TournamentPlanner

Generated by Doxygen 1.9.1

Chapter 1

TournamentPlanner

This repository contains both the practical and the theoretical (writing) part of my master's thesis.

To be able to run the project, you need to add a reference to the `Gurobi90.NET.dll` (or a newer version) at `TournamentPlanner.csproj` file (or using VS's add reference option).

The `latex` folder in the root directory contains the writing part of the thesis in the `latex/dp.pdf` file.

Chapter 2

The MIT License (MIT)

Chapter 3

Namespace Index

3.1 Packages

Here are the packages with brief descriptions (if available):

Chapter 4

Hierarchical Index

4.1 Class Hierarchy

This inheritance list is sorted roughly, but not completely, alphabetically:

        - TournamentPlanner.Algorithms.ConstructionAlgorithms.LatinSquares.Orthogonal.GeneralMOLSPairConstruction 96
        - TournamentPlanner.Algorithms.ConstructionAlgorithms.LatinSquares.Orthogonal.OddMOLSPairConstruction 127
- TournamentPlanner.Algorithms.ConstructionAlgorithms.IDesignConstruction< OrthogonalArray > 101
    - TournamentPlanner.Algorithms.ConstructionAlgorithms.LatinSquares.OA.OrthogonalArrayConstruction 134
- TournamentPlanner.Algorithms.ConstructionAlgorithms.IDesignConstruction< PBD > 101
    - TournamentPlanner.Algorithms.ConstructionAlgorithms.BlockDesigns.PBDConstruction.GeneralPBD3NPlus1Construction 98
    - TournamentPlanner.Algorithms.ConstructionAlgorithms.BlockDesigns.PBDConstruction.LargerOrderPBDConstruction 110
    - TournamentPlanner.Algorithms.ConstructionAlgorithms.BlockDesigns.PBDConstruction.PBD34Construction 166
    - TournamentPlanner.Algorithms.ConstructionAlgorithms.BlockDesigns.PBDConstruction.SmallOrderPBDConstruction 205
        - TournamentPlanner.Algorithms.ConstructionAlgorithms.BlockDesigns.PBDConstruction.QuasigroupPBDConstruction 176
- TournamentPlanner.Algorithms.ConstructionAlgorithms.IDesignConstruction< SOLS > 101
    - TournamentPlanner.Algorithms.ConstructionAlgorithms.LatinSquares.SelfOrthogonal.SOLSWithNoGCD6Construction 212
- TournamentPlanner.Algorithms.ConstructionAlgorithms.IDesignConstruction< TD > 101
    - TournamentPlanner.Algorithms.ConstructionAlgorithms.BlockDesigns.TDConstruction.TDConstruction 218
        - TournamentPlanner.Algorithms.ConstructionAlgorithms.BlockDesigns.TDConstruction.RTDConstruction 187
- TournamentPlannerTests.Algorithms.ConstructionAlgorithms.BlockDesigns.BIBDConstruction.IncrementalBIBDConstructionTests 103
- TournamentPlanner.Algorithms.TournamentPlanners.ITournamentPlanner< BaseTournament > 105
- TournamentPlanner.Algorithms.TournamentPlanners.ITournamentPlanner< BipartiteTournament > 105
    - TournamentPlanner.Algorithms.TournamentPlanners.BipartiteTournamentPlanner 62
- TournamentPlanner.Algorithms.TournamentPlanners.ITournamentPlanner< SAMDRR > 105
    - TournamentPlanner.Algorithms.TournamentPlanners.Spouses.SAMDRR38Planner 191
    - TournamentPlanner.Algorithms.TournamentPlanners.Spouses.SAMDRR8Planner 193
    - TournamentPlanner.Algorithms.TournamentPlanners.Spouses.SimpleSAMDRRPlanner 194
- TournamentPlanner.Algorithms.TournamentPlanners.ITournamentPlanner< ThreePlayerTournament > 105
    - TournamentPlanner.Algorithms.TournamentPlanners.Balanced3PlayerTournamentPlanner 39
    - TournamentPlanner.Algorithms.TournamentPlanners.PartiallyBalanced3PlayerTournamentPlanner 162
- TournamentPlanner.Algorithms.TournamentPlanners.ITournamentPlanner< TwoPlayerRoundRobin > 105
    - TournamentPlanner.Algorithms.TournamentPlanners.TwoPlayerRoundRobinPlanner 227
- TournamentPlanner.Algorithms.TournamentPlanners.ITournamentPlanner< UniformGameSizeTournament > 105
    - TournamentPlanner.Algorithms.TournamentPlanners.UniformGameSizeTournamentPlanner 233
- TournamentPlanner.Algorithms.TournamentPlanners.ITournamentPlanner< WhistTournament > 105
    - TournamentPlanner.Algorithms.TournamentPlanners.Whist.BaseWhistPlanner 52
        - TournamentPlanner.Algorithms.TournamentPlanners.Whist.Even.CyclicWhistPlanner 72
        - TournamentPlanner.Algorithms.TournamentPlanners.Whist.Even.EvenEvenProductWhistPlanner 78
        - TournamentPlanner.Algorithms.TournamentPlanners.Whist.Even.EvenOddProductWhistPlanner 80
        - TournamentPlanner.Algorithms.TournamentPlanners.Whist.Even.GeneralEvenWhistPlanner 93
        - TournamentPlanner.Algorithms.TournamentPlanners.Whist.Even.LargerWhistPlanner 112
        - TournamentPlanner.Algorithms.TournamentPlanners.Whist.Even.MooreWhistPlanner 125
        - TournamentPlanner.Algorithms.TournamentPlanners.Whist.Even.OddSAMDRRWhistPlanner 129
        - TournamentPlanner.Algorithms.TournamentPlanners.Whist.Even.TDWhistPlanner 220
        - TournamentPlanner.Algorithms.TournamentPlanners.Whist.Even.Whist152Planner 237
        - TournamentPlanner.Algorithms.TournamentPlanners.Whist.Odd.CyclicOddWhistPlanner 70
        - TournamentPlanner.Algorithms.TournamentPlanners.Whist.Odd.GeneralOddWhistPlanner 97
        - TournamentPlanner.Algorithms.TournamentPlanners.Whist.Odd.PBDWhistPlanner 169
        - TournamentPlanner.Algorithms.TournamentPlanners.Whist.Odd.PrimePowerWhistPlanner 172

    - TournamentPlanner.Pages.PageModels.Testing.PairwiseTestingModel 161
    - TournamentPlanner.Pages.PageModels.Tournaments.BipartiteTournamentModel 60
    - TournamentPlanner.Pages.PageModels.Tournaments.ThreePlayerTournamentsModel 224
    - TournamentPlanner.Pages.PageModels.Tournaments.TwoPlayerTournamentModel 230
    - TournamentPlanner.Pages.PageModels.Tournaments.UniformGamesModel 235
    - TournamentPlanner.Pages.PageModels.Tournaments.WhistTournamentsModel 241
  - TournamentPlanner.Pages.PrivacyModel 174
- TournamentPlanner.Model.Testing.PairwiseTestCases 157
- TournamentPlanner.Algorithms.Testing.PairwiseTestCasesConstruction 159
- TournamentPlannerTests.Algorithms.Testing.PairwiseTestCasesConstructionTests 160
- TournamentPlannerTests.Algorithms.ConstructionAlgorithms.BlockDesigns.PBDConstruction.PBD34ConstructionTest 167
- TournamentPlannerTests.Algorithms.TournamentPlanners.Whist.Odd.PBDWhistPlannerTests 170
- TournamentPlannerTests.Utils.PrimeFactorizationTests 171
- TournamentPlannerTests.Algorithms.TournamentPlanners.Whist.Odd.PrimePowerWhistPlannerTests 173
- TournamentPlanner.Program 174
- TournamentPlannerTests.Algorithms.ConstructionAlgorithms.BlockDesigns.BIBDConstruction.QuadraticProgrammingBIBDConstructionTests 176
- TournamentPlannerTests.Algorithms.ConstructionAlgorithms.BlockDesigns.PBDConstruction.QuasigroupPBDConstructionTests 178
- global.MicrosoftAspNetCore.Mvc.Razor.RazorPage
  - TournamentPlanner.Pages.Pages__ViewImports 136
  - TournamentPlanner.Pages.Pages__ViewImports 136
  - TournamentPlanner.Pages.Pages__ViewStart 137
  - TournamentPlanner.Pages.Pages__ViewStart 137
  - TournamentPlanner.Pages.Shared.Pages_Shared__Layout 153
  - TournamentPlanner.Pages.Shared.Pages_Shared__Layout 153
  - TournamentPlanner.Pages.Shared.Pages_Shared__ValidationScriptsPartial 154
  - TournamentPlanner.Pages.Shared.Pages_Shared__ValidationScriptsPartial 154
- TournamentPlannerTests.Algorithms.ConstructionAlgorithms.BlockDesigns.BIBDConstruction.RecursiveKTSConstructionTests 183
- TournamentPlannerTests.Algorithms.ConstructionAlgorithms.BlockDesigns.GDDConstruction.Resolvable3GDDConstructionTests 185
- TournamentPlannerTests.Algorithms.ConstructionAlgorithms.BlockDesigns.TDConstruction.RTDConstructionTests 189
- TournamentPlannerTests.Algorithms.TournamentPlanners.Spouses.SAMDRR38PlannerTests 192
- TournamentPlannerTests.Algorithms.TournamentPlanners.Spouses.SAMDRR8PlannerTests 194
- TournamentPlannerTests.Algorithms.TournamentPlanners.Spouses.SimpleSAMDRRPlannerTests 195
- TournamentPlannerTests.Algorithms.ConstructionAlgorithms.LatinSquares.SingleLatinSquareConstructionTests 198
- TournamentPlannerTests.Algorithms.ConstructionAlgorithms.BlockDesigns.BIBDConstruction.SkolemSTSConstructionTests 200
- TournamentPlannerTests.Algorithms.ConstructionAlgorithms.BlockDesigns.BIBDConstruction.SmallBIBDForWhistConstructionTests 202
- TournamentPlannerTests.Algorithms.ConstructionAlgorithms.BlockDesigns.GDDConstruction.SmallGDDConstructionTests 204
- TournamentPlannerTests.Algorithms.ConstructionAlgorithms.BlockDesigns.PBDConstruction.SmallOrderPBDConstructionTests 207
- TournamentPlannerTests.Algorithms.ConstructionAlgorithms.BlockDesigns.BIBDConstruction.SmoothKTSConstructionTests 209
- TournamentPlannerTests.Algorithms.ConstructionAlgorithms.LatinSquares.SelfOrthogonal.SOLSWithNoGCD6ConstructionTests 213
- TournamentPlanner.Startup 214
- TournamentPlannerTests.Algorithms.ConstructionAlgorithms.BlockDesigns.TDConstruction.TD17ConstructionTests 218
- TournamentPlannerTests.Algorithms.TournamentPlanners.Whist.Even.TDWhistPlannerTests 221
- TournamentPlannerTests.Algorithms.TournamentPlanners.UniformGameSizeTournamentPlannerTests 234
- TournamentPlannerTests.Algorithms.TournamentPlanners.Whist.Even.Whist152PlannerTests 238

Chapter 5

Class Index

5.1 Class List

Here are the classes, structs, unions and interfaces with brief descriptions:

TournamentPlanner.Algorithms.ConstructionAlgorithms.BlockDesigns.PBDConstruction.PBD34Construction
The construction of PBD 34 was supposed to be in the SmallOrderPBDConstruction, but was a bit too complicated so i extracted it to its custom class. For the construction of PBD34 a PBD27 needs to be constructed first. Then it is modified and new blocks are added to get a PBD34 . . 166

TournamentPlannerTests.Algorithms.ConstructionAlgorithms.BlockDesigns.PBDConstruction.PBD34ConstructionTest
167

TournamentPlanner.Pages.PageModels.Designs.PBDModel
Page model for the pairwise balanced design. Construction of PBD of orders 3M + 1 is implemented 168

TournamentPlanner.Algorithms.TournamentPlanners.Whist.Odd.PBDWhistPlanner
Class facilitating the construction of Whist(21) and Whist(69) using PBDs 169

TournamentPlannerTests.Algorithms.TournamentPlanners.Whist.Odd.PBDWhistPlannerTests
Tests for the PBDWhistPlannerTests 170

TournamentPlannerTests.Utils.PrimeFactorizationTests
Tests for the PrimeFactorization static class 171

TournamentPlanner.Algorithms.TournamentPlanners.Whist.Odd.PrimePowerWhistPlanner
Whist tournament planner for prime power number of players in the form of v = 4n + 1. It uses finite fields for the tournament construction 172

TournamentPlannerTests.Algorithms.TournamentPlanners.Whist.Odd.PrimePowerWhistPlannerTests
Tests for the PrimeNWhistPlanner 173

TournamentPlanner.Pages.PrivacyModel 174

TournamentPlanner.Program 174

TournamentPlanner.Algorithms.ConstructionAlgorithms.BlockDesigns.BIBDConstruction.QuadraticProgrammingBIBDConstructio
A quadratic programming approach to the construction of BIBDs as described in the thesis. Uses CP-SAT as solver 175

TournamentPlannerTests.Algorithms.ConstructionAlgorithms.BlockDesigns.BIBDConstruction.QuadraticProgrammingBIBDConst
Tests for the QuadraticProgrammingBIBDConstruction 176

TournamentPlanner.Algorithms.ConstructionAlgorithms.BlockDesigns.PBDConstruction.QuasigroupPBDConstruction
Method for construction of PBDs of orders 13, 16 and 28. This method uses pairs of MOLS as quasigroups 176

TournamentPlannerTests.Algorithms.ConstructionAlgorithms.BlockDesigns.PBDConstruction.QuasigroupPBDConstructionTests
Tests for the QuasigroupPBDConstruction 178

TournamentPlanner.Model.Designs.BlockDesigns.RBIBD
Class representing a resolvable balanced incomplete blocks design 179

TournamentPlanner.Algorithms.ConstructionAlgorithms.BlockDesigns.BIBDConstruction.RecursiveKTSConstruction
Algorithm for the recursive construction of KTS of order 6N + 3. Method described in https↩ ://books.google.tg/books?id=OoRFisqn-aIC. KTSs of orders 9, 15, 21 and 39 are constructed statically, all the others are constructed in a recursive manner 180

TournamentPlannerTests.Algorithms.ConstructionAlgorithms.BlockDesigns.BIBDConstruction.RecursiveKTSConstructionTests
Tests for the RecursiveKTSConstruction 183

TournamentPlanner.Algorithms.ConstructionAlgorithms.BlockDesigns.GDDConstruction.Resolvable3GDDConstruction
Construct resolvable group divisible designs with block size 3 using MOLS. Should be able to construct GDDs where the order is a multiple of 3 except 6 and 18 183

TournamentPlannerTests.Algorithms.ConstructionAlgorithms.BlockDesigns.GDDConstruction.Resolvable3GDDConstructionTests
Tests the creation of resolvable GDDs by Resolvable3GDDConstruction 185

TournamentPlanner.Model.Designs.BlockDesigns.RTD
Resolvable transversal design 186

TournamentPlanner.Algorithms.ConstructionAlgorithms.BlockDesigns.TDConstruction.RTDConstruction
Construction method for resolvable transversal designs 187

TournamentPlannerTests.Algorithms.ConstructionAlgorithms.BlockDesigns.TDConstruction.RTDConstructionTests
Tests for the RTDConstruction 189

TournamentPlanner.Model.Tournaments.SAMDRR
Spouse-avoiding mixed doubles round robin tournament. Games are represented as 4 integers, (i, l, j, k) - H_i W_l v H_j W_k 190

TournamentPlanner.Algorithms.TournamentPlanners.Spouses.SAMDRR38Planner
SAMDRR 38 planner. GCD(38, 6) != 1 so a special construction is required. SAMDRR 38 is used to construct Whist(152) 191

Chapter 6

Namespace Documentation

6.1 TournamentPlanner Namespace Reference

Classes

- class Program
- class Startup

6.2 TournamentPlanner.Algorithms Namespace Reference

6.3 TournamentPlanner.Algorithms.ConstructionAlgorithms Namespace Reference

Classes

- interface IDesignConstruction

  Interface for the design construction algorithms.

6.4 TournamentPlanner.Algorithms.ConstructionAlgorithms.Block↩ Designs Namespace Reference

6.5 TournamentPlanner.Algorithms.ConstructionAlgorithms.Block↩ Designs.BIBDConstruction Namespace Reference

Classes

- class BaseBIBDConstruction

  Base class for the BIBD construction algorithms.
- class BoseSTSConstruction

Bose's algorithm for the construction of STS of size V = 6N + 3. http://math.ucdenver.edu/∼wcherowi/courses/m6406/sts.pdf

- class IncrementalBIBDConstruction

  Incremental method that constructs BIBDs block by block. Implementation of the algorithm described in https://www.researchgate.net/publication/220660995_A_mathematical_programming_approach_to_the_construction_of_BIBDs. Uses CP-SAT as solver.
- class MooreRBIBDConstruction

  Construction of RBIBD(3q+1, 4, 1) where q is a prime power and can be written as q = 4m + 1. These RBIBD are used to construct Whist tournaments.
- class QuadraticProgrammingBIBDConstruction

  A quadratic programming approach to the construction of BIBDs as described in the thesis. Uses CP-SAT as solver.
- class RecursiveKTSConstruction

  Algorithm for the recursive construction of KTS of order 6N + 3. Method described in https://books.google.tg/books?id=OoRFisqn-aIC. KTSs of orders 9, 15, 21 and 39 are constructed statically, all the others are constructed in a recursive manner.
- class SkolemSTSConstruction

  Skolem construction algorithm for STS of size 6N + 1. http://math.ucdenver.edu/∼wcherowi/courses/m6406/sts.pdf
- class SmallBIBDForWhistConstruction

  Static construction of BIBD(21, 5, 1) and BIBD(25, 5, 1). These designs have been constructed by the incremental / quadratic methods. Those methods are, however, unreliable (license and memory issues) so the constructed designs are stored here for faster and more reliable access.
- class SmoothKTSConstruction

  Algorithm for construction of smooth Kirkman triple systems. The size of smooth KTS is a power of 3. https://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.501.8928&rep=rep1&type=pdf

6.6 TournamentPlanner.Algorithms.ConstructionAlgorithms.BlockDesigns.GDDConstruction Namespace Reference

Classes

- class GDDForWhistConstruction

  Construction of GDD(17g, + 4u + 4v, {5, 17}, {g, g + 4u, g + 4v}). These are used to construct whist tournaments.
- class Resolvable3GDDConstruction

  Construct resolvable group divisible designs with block size 3 using MOLS. Should be able to construct GDDs where the order is a multiple of 3 except 6 and 18.
- class SmallGDDConstruction

  Construct a few small order GDDs which are then used i Wilsons fundamental construction.
- class WilsonGDDConstruction

  Wilsons construction methods using smaller GDDs to construct bigger ones. https://www.semanticscholar.org/paper/An-Existence-Theory-for-Pairwise-Balanced-Designs-Wilson/b9c1c6bb1d7828b4f47f2ee3330dab26adb946dc

6.7 TournamentPlanner.Algorithms.ConstructionAlgorithms.BlockDesigns.PBDConstruction Namespace Reference

Classes

- class GeneralPBD3NPlus1Construction

  Class that is used for construction of any PBD(3n+1). It uses the required depending on the order. It is also used in the recursive step of the LargerOrderPBDConstruction.

- class LargerOrderPBDConstruction

  Construction of larger order PBDs, where the order is V = 3N + 1.
- class PBD34Construction

  The construction of PBD 34 was supposed to be in the SmallOrderPBDConstruction, but was a bit too complicated so i extracted it to its custom class. For the construction of PBD34 a PBD27 needs to be constructed first. Then it is modified and new blocks are added to get a PBD34.
- class QuasigroupPBDConstruction

  Method for construction of PBDs of orders 13, 16 and 28. This method uses pairs of MOLS as quasigroups.
- class SmallOrderPBDConstruction

  A class facilitating the construction of PBDs of smaller orders in form 3N + 1. All the constructed design have block sizes in {4, 7, 10, 19}. There are several construction methods, not a single unified one.

6.8 TournamentPlanner.Algorithms.ConstructionAlgorithms.Block↩ Designs.TDConstruction Namespace Reference

Classes

- class RTDConstruction

  Construction method for resolvable transversal designs.
- class TDConstruction

  Construction of TD(k, g) that are then used to construct Whist tournaments.

6.9 TournamentPlanner.Algorithms.ConstructionAlgorithms.Latin↩ Squares Namespace Reference

Classes

- class CommutativeIdempotentLatinSquareConstruction

  Algorithm for the creation od commutative idempotent Latin squares of odd order
- class SingleLatinSquareConstruction

  Recursive construction of a single Latin square of any order.

6.10 TournamentPlanner.Algorithms.ConstructionAlgorithms.Latin↩ Squares.OA Namespace Reference

Classes

- class OrthogonalArrayConstruction

  Construction method for OAs using MOLS.

6.11 TournamentPlanner.Algorithms.ConstructionAlgorithms.Latin↩ Squares.Orthogonal Namespace Reference

Classes

- class BaseMOLSConstruction

  Base class for the mutually orthogonal latin square construction algorithms.
- class ConstraintProgrammingMOLSPairConstruction

  Constraint a pair of MOLS using constraint programming approach with CP-SAT solver. This method only works for MOLS of small orders.
- class DatastoreMOLSConstruction

  This class creates MOLS objects based on the ones in the data store. This is not a construction algorithm per se, but acts like one. It implements the same interface as the other MOLS construction methods so it is easier to use in other algorithms.
- class DifferenceMOLSPairConstruction

  Construct a pair of MOLS of order 3M + 1. This way, MOLS for all orders 12t + 10 (which is a subset of 4k + 2) can be constructed. This method was introduced in https://www.math.uci.edu/∼brusso/[14]BosShrParCJM1960.pdf.
- class FourKMOLSPairConstruction

  Construction of MOLS of order V = 4 ∗ K.
- class GeneralMOLSPairConstruction

  Class facilitating the construction of MOLS of all the supported orders. Uses other construction algorithms.
- class OddMOLSPairConstruction

  Algorithm to construct pairs of mutually orthogonal Latin squares of odd order. https://www.whitman.↩ edu/mathematics/cgt_online/book/section04.03.html

6.12 TournamentPlanner.Algorithms.ConstructionAlgorithms.Latin↩ Squares.SelfOrthogonal Namespace Reference

Classes

- class SOLSWithNoGCD6Construction

  Construct a cyclic SOLS such that GCD(n,6) = 1.

6.13 TournamentPlanner.Algorithms.Testing Namespace Reference

Classes

- class PairwiseTestCasesConstruction

  Class implementing the construction of the pairwise test cases using orthogonal arrays.

6.14 TournamentPlanner.Algorithms.TournamentPlanners Namespace Reference

Classes

- class Balanced3PlayerTournamentPlanner

  Plane a balanced three player tournament. Perfectly balanced three player tournament can only be planed, if the number of players V = 6N + 3.
- class BipartiteTournamentPlanner

  Bipartite tournament planner using Latin squares.
- interface ITournamentPlanner

  Interface for the tournament planners.
- class PartiallyBalanced3PlayerTournamentPlanner

  This class facilitates the construction of partially balanced three player tournaments. The number of players in the tournament has to be divisible by 3. For the construction of the tournament, GDD are used.
- class TwoPlayerRoundRobinPlanner

  Tournament planner for two player round robins.
- class UniformGameSizeTournamentPlanner

  CP approach to the planning of tournaments, where the game sizes are uniform.

6.15 TournamentPlanner.Algorithms.TournamentPlanners.Spouses Namespace Reference

Classes

- class SAMDRR38Planner

  SAMDRR 38 planner. GCD(38, 6) != 1 so a special construction is required. SAMDRR 38 is used to construct Whist(152).
- class SAMDRR8Planner

  SAMDRR 8 construction class.
- class SimpleSAMDRRPlanner

  SAMDRR planner using cyclic SOLS such that for the SOLS order n, GCD(n, 6) = 1.

6.16 TournamentPlanner.Algorithms.TournamentPlanners.Whist Namespace Reference

Classes

- class BaseWhistPlanner

  The base class for whist tournament planners.

6.17 TournamentPlanner.Algorithms.TournamentPlanners.Whist.Even Namespace Reference

Classes

- class CyclicWhistPlanner

  Cyclic construction method for several orders of Whist(V = 4N).
- class EvenEvenProductWhistPlanner

  Construction of a Whist tournament as a product of 2 existing Wh(4n) tournaments.
- class EvenOddProductWhistPlanner

  Construction of a Whist tournament as a product of a Wh(4n) and Wh(4n + 1).
- class GeneralEvenWhistPlanner

  Plan a whist tournament of order 4n or 4n + 1 using some of the implemented approaches. This class basically just selects the method to use for the construction.
- class LargerWhistPlanner

  Construct Wh(4n), where n $>$ 80.
- class MooreWhistPlanner

  This class implements the construction of Wh(v = 4n), where v = 3q + 1 and q = 1 (mod 4) = 4m + 1 and is a prime power. v = 28, 40, 52, 76, 88 ... https://www.jstor.org/stable/2369860?seq=1::metadata_info_tab_contents.
- class OddSAMDRRWhistPlanner

  Use odd order SAMMDR to construct several orders od Whist(4n).
- class TDWhistPlanner

  Same method as in the larger whist construction but with TD (which is also a GDD basically). TODO unite.
- class Whist152Planner

  Class for the construction of Whist 152. Uses even SAMDRR.

6.18 TournamentPlanner.Algorithms.TournamentPlanners.Whist.Odd Namespace Reference

Classes

- class CyclicOddWhistPlanner

  Cyclic odd whist planner for v = 33, 57.
- class GeneralOddWhistPlanner

  Construct some Whist tournaments of order 4n + 1. Not all orders are supported unfortunately.
- class PBDWhistPlanner

  Class facilitating the construction of Whist(21) and Whist(69) using PBDs.
- class PrimePowerWhistPlanner

  Whist tournament planner for prime power number of players in the form of v = 4n + 1. It uses finite fields for the tournament construction.

6.19 TournamentPlanner.Algorithms.Utils Namespace Reference

Classes

- class **OrthogonalArrayMOLSConverter**

  Static class facilitating the conversion between MOLS and orthogonal arrays. MOLS and OAs are equivalent structures for some parameter values.

6.20 TournamentPlanner.Datastore Namespace Reference

Classes

- class **DataReader**

  Static class that facilitates the access to the data stored in file or MongoDB.
- class MongoMOLS

  Class representing a MOLS in mongo DB.

6.21 TournamentPlanner.Model Namespace Reference

6.22 TournamentPlanner.Model.Designs Namespace Reference

Classes

- interface IDesign

  Common interface for all the design structures.
- interface IResolvableDesign

  Interface for resolvable designs.
- class OrthogonalArray

  Class representing the Orthogonal array structure. In some literature, this representation is actually the transpose of an orthogonal array.

6.23 TournamentPlanner.Model.Designs.BlockDesigns Namespace Reference

Classes

- class BaseBlockDesign

  Base class for the block designs.
- class BIBD

  Class representing a balanced incomplete block design.
- class GDD

  Class representing a group divisible design.
- class KTS

  A class representing a Kirkman triple system. A KTS is a resolvable BIBD(V, 3, 1), where V = 6N + 3. Basically, a resolvable STS i a KTS.
- class PBD

  Class representing a Pairwise balanced design. PBD is similar to BIBD, but the block sizes are from K - they don't have to be uniform. We only consider PBDs with lambda = 1.
- class RBIBD

  Class representing a resolvable balanced incomplete blocks design.
- class RTD

  Resolvable transversal design.
- class STS

  Class representing a Steiner triple system. A Steiner triple system is a BIBD with K = 3 and Lambda = 1.
- class TD

  Class representing the transversal design.

6.24 TournamentPlanner.Model.Designs.LatinSquares Namespace Reference

Classes

- class LatinSquare

  The Latin square.
- class MOLS

  Set of mutually orthogonal Latin squares. For simplicity, it is allowed for the set of MOLS to contain only a single Latin square.
- class SOLS

  Self orthogonal Latin square.

6.25 TournamentPlanner.Model.Testing Namespace Reference

Classes

- class PairwiseTestCases

  A class representing the pairwise test cases created using orthogonal arrays a some post processing methods.

6.26 TournamentPlanner.Model.Tournaments Namespace Reference

Classes

- class BaseTournament

  Base abstract class for the tournaments.
- class BipartiteTournament

  Class representing a bipartite tournament. In a bipartite tournament, there are 2 teams with the same number of players. The goal is, that every player from the first team should play a game with every player from the second team.
- class SAMDRR

  Spouse-avoiding mixed doubles round robin tournament. Games are represented as 4 integers, (i, l, j, k) - H_i W_l v H_j W_k
- class ThreePlayerTournament

  Class representing a three player tournament.
- class TwoPlayerRoundRobin

  Represents a 2 player round robin tournament schedule.
- class UniformGameSizeTournament

  Model for the tournament, where the sizes of the games are uniform - all the games have k-players.
- class WhistTournament

  The model of the whist tournament.

6.27 TournamentPlanner.Pages Namespace Reference

Classes

- class Pages__ViewImports
- class Pages__ViewStart
- class Pages_Error
- class Pages_Index
- class Pages_Privacy
- class ErrorModel
- class IndexModel
- class PrivacyModel

6.28 TournamentPlanner.Pages.PageModels Namespace Reference

Classes

- class BasePageModel

  The base class for the page models of designs and tournaments.

6.29 TournamentPlanner.Pages.PageModels.Designs Namespace Reference

Classes

- class Pages_PageModels_Designs_BIBD
- class Pages_PageModels_Designs_GDD
- class Pages_PageModels_Designs_KTS
- class Pages_PageModels_Designs_MOLS
- class Pages_PageModels_Designs_PBD
- class BIBDModel

  The page model for BIBDs.
- class GDDModel

  Page model for resolvable group divisible designs with block size 3.
- class KTSModel

  Page model for the Kirkman triple system creation.
- class MOLSModel

  Page model for mutually orthogonal Latin squares.
- class PBDModel

  Page model for the pairwise balanced design. Construction of PBD of orders 3M + 1 is implemented.

Enumerations

- enum class Method { QuadraticProgramming , BoseBlockBuilding , SkolemBlockBuilding , IncrementalBlockBuilding }

  Enum of the possible construction methods.
- enum class MOLSMethod {
  Automatic , OddOrderMOLS , FourKOrder , CPApproach ,
  DifferenceApproach }

  Enum of the possible construction methods.

6.29.1 Enumeration Type Documentation

6.29.1.1 Method

```
enum TournamentPlanner.Pages.PageModels.Designs.Method  [strong]
```

Enum of the possible construction methods.

Enumerator

QuadraticProgramming	Quadratic programming construction method.
BoseBlockBuilding	Bose STS construction method.
SkolemBlockBuilding	Skolem STS construction method.
IncrementalBlockBuilding	Incremental BIBD construction method.

6.29.1.2 MOLSMethod

```
enum TournamentPlanner.Pages.PageModels.Designs.MOLSMethod  [strong]
```

Enum of the possible construction methods.

Enumerator

Automatic	Select the method according to the order automatically.
OddOrderMOLS	Construction method for MOLS of odd order.
FourKOrder	Construction method for MOLS of order 4K.
CPApproach	Construction method using constraint programming.
DifferenceApproach	Construction method using the difference set approach.

6.30 TournamentPlanner.Pages.PageModels.Testing Namespace Reference

Classes

- class Pages_PageModels_Testing_PairwiseTesting
- class PairwiseTestingModel

  Page model for the construction of pairwise tests.

6.31 TournamentPlanner.Pages.PageModels.Tournaments Namespace Reference

Classes

- class Pages_PageModels_Tournaments_BipartiteTournament
- class Pages_PageModels_Tournaments_ThreePlayerTournaments
- class Pages_PageModels_Tournaments_TwoPlayerTournament
- class Pages_PageModels_Tournaments_UniformGames
- class Pages_PageModels_Tournaments_WhistTournaments
- class BipartiteTournamentModel

  Page model for the construction of bipartite tournaments.
- class ThreePlayerTournamentsModel

  Page model for three player tournaments.
- class TwoPlayerTournamentModel

  Page model for the construction of two player round robin tournaments.
- class UniformGamesModel

  Page model for the tournament, in which the games are uniformly sized.
- class WhistTournamentsModel

  Page model for the Whist tournaments.

6.32 TournamentPlanner.Pages.Shared Namespace Reference

Classes

- class Pages_Shared__Layout
- class Pages_Shared__ValidationScriptsPartial

6.33 TournamentPlanner.Pages.Tournament Namespace Reference

Classes

- class Pages_Tournament_BipartiteTournament
- class Pages_Tournament_TwoPlayerTournament

6.34 TournamentPlanner.Utils Namespace Reference

Classes

- class **MultidimensionalArrayExtension**

  Extension methods of 2D multidimensional arrays to get rows and columns as single 1D arrays..
- class **PrimeFactorization**

  Static class providing factorization methods.

6.35 TournamentPlannerTests Namespace Reference

6.36 TournamentPlannerTests.Algorithms Namespace Reference

6.37 TournamentPlannerTests.Algorithms.ConstructionAlgorithms Namespace Reference

6.38 TournamentPlannerTests.Algorithms.ConstructionAlgorithms.↩ BlockDesigns Namespace Reference

6.39 TournamentPlannerTests.Algorithms.ConstructionAlgorithms.↩ BlockDesigns.BIBDConstruction Namespace Reference

Classes

- class BoseSTSConstructionTests

  Tests for the BoseSTSConstruction.
- class GurobiIncrementalBIBDConstructionTests

  Tests for the GurobiIncrementalBIBDConstruction.
- class GurobiQuadraticProgrammingBIBDConstructionTests

  Tests fot the QuadraticProgrammingBIBDConstruction.
- class IncrementalBIBDConstructionTests

  Test for the IncrementalBIBDConstruction.
- class MooreRBIBDConstructionTests

  Tests for the MooreRBIBDConstruction.
- class QuadraticProgrammingBIBDConstructionTests

  Tests for the QuadraticProgrammingBIBDConstruction.
- class RecursiveKTSConstructionTests

  Tests for the RecursiveKTSConstruction.
- class SkolemSTSConstructionTests

  Tests for the SkolemSTSConstruction.
- class SmallBIBDForWhistConstructionTests

  Tests for SmallBIBDForWhistConstruction.
- class SmoothKTSConstructionTests

  Tests for the SmoothKTSConstruction.

6.40 TournamentPlannerTests.Algorithms.ConstructionAlgorithms.↩ BlockDesigns.GDDConstruction Namespace Reference

Classes

- class GDDForWhistConstructionTests

  Tests for the GDDForWhistConstruction.
- class Resolvable3GDDConstructionTests

  Tests the creation of resolvable GDDs by Resolvable3GDDConstruction.
- class SmallGDDConstructionTests

  Tests for the SmallGDDConstruction.
- class WilsonGDDConstructionTests

  Tests for the WilsonGDDConstruction.

6.41 TournamentPlannerTests.Algorithms.ConstructionAlgorithms.↩ BlockDesigns.PBDConstruction Namespace Reference

Classes

- class LargerOrderPBDConstructionTests

  Tests for the LargerOrderPBDConstruction.
- class PBD34ConstructionTest
- class QuasigroupPBDConstructionTests

  Tests for the QuasigroupPBDConstruction.
- class SmallOrderPBDConstructionTests

  Tests for the SmallOrderPBDConstruction.

6.42 TournamentPlannerTests.Algorithms.ConstructionAlgorithms.↩ BlockDesigns.TDConstruction Namespace Reference

Classes

- class RTDConstructionTests

  Tests for the RTDConstruction.
- class TD17ConstructionTests

  Tests for the TD17Construction.

6.43 TournamentPlannerTests.Algorithms.ConstructionAlgorithms.↩ LatinSquares Namespace Reference

Classes

- class SingleLatinSquareConstructionTests

  Tests for the SingleLatinSquareConstruction.

6.44 TournamentPlannerTests.Algorithms.ConstructionAlgorithms.↩ LatinSquares.Orthogonal Namespace Reference

Classes

- class ConstraintProgrammingMOLSPairConstructionTests

  Tests for the ConstraintProgrammingMOLSPairConstruction.
- class DatastoreMOLSConstructionTests

  Tests for the DatastoreMOLSConstruction.
- class DifferenceMOLSPairConstructionTests

  Tests for the DifferenceMOLSPairConstruction.
- class FourKMOLSPairConstructionTests

  Tests for the FourKMOLSPairConstruction.
- class OddMOLSPairConstructionTests

  Tests for the OddMOLSPairConstruction.

6.45 TournamentPlannerTests.Algorithms.ConstructionAlgorithms.↩ LatinSquares.SelfOrthogonal Namespace Reference

Classes

- class SOLSWithNoGCD6ConstructionTests

  Tests for SOLSWithNoGCD6Construction.

6.46 TournamentPlannerTests.Algorithms.Testing Namespace Reference

Classes

- class PairwiseTestCasesConstructionTests

6.47 TournamentPlannerTests.Algorithms.TournamentPlanners Namespace Reference

Classes

- class UniformGameSizeTournamentPlannerTests

  Tests for the UniformGameSizeTournamentPlanner.

6.48 TournamentPlannerTests.Algorithms.TournamentPlanners.Spouses Namespace Reference

Classes

- class SAMDRR38PlannerTests

  Tests for the SAMDRR38Planner.
- class SAMDRR8PlannerTests

  Tests for the SAMDRR8Planner.
- class SimpleSAMDRRPlannerTests

  Tests for the SimpleSAMDRRPlanner.

6.49 TournamentPlannerTests.Algorithms.TournamentPlanners.Whist Namespace Reference

6.50 TournamentPlannerTests.Algorithms.TournamentPlanners.Whist.↩ Even Namespace Reference

Classes

- class CyclicWhistPlannerTests

  Tests for the CyclicWhistPlanner.
- class EvenEvenProductWhistPlannerTests

  Tests for the ProductWhistPlanner.
- class EvenOddProductWhistPlannerTests

  Tests for the EvenOddProductWhistPlanner.
- class GeneralEvenWhistPlannerTests

  Tests for the General4NWhistPlanner.
- class LargerWhistPlannerTests

  Tests for the LargerWhistPlanner.
- class MooreWhistPlannerTests

  Tests for the MooreWhistPlanner.
- class OddSAMDRRWhistPlannerTests

  Tests for the SAMDRRWhistPlanner.
- class TDWhistPlannerTests

  Tests for the TDWhistPlanner.
- class Whist152PlannerTests

  Tests for the Whist152Planner.

6.51 TournamentPlannerTests.Algorithms.TournamentPlanners.Whist.↩ Odd Namespace Reference

Classes

- class CyclicOddWhistPlannerTests

  Tests for the CyclicOddWhistPlanner.
- class PBDWhistPlannerTests

  Tests for the PBDWhistPlannerTests.
- class PrimePowerWhistPlannerTests

  Tests for the PrimeNWhistPlanner.

6.52 TournamentPlannerTests.Datastore Namespace Reference

Classes

- class DataReaderTests

  Tests for the DataReader.

6.53 TournamentPlannerTests.Designs Namespace Reference

Classes

- class MOLSSetValidatorTests

  Tests for the MOLSSetValidator.

6.54 TournamentPlannerTests.Model Namespace Reference

6.55 TournamentPlannerTests.Model.Designs Namespace Reference

Classes

- class **ResolvableDesignValidator**

  Validator for resolvable designs.

6.56 TournamentPlannerTests.Model.Designs.BlockDesigns Namespace Reference

Classes

- class **BIBDValidator**

  A static class providing methods for BIBD validation.
- class **GDDValidator**

  A static class for validating GDDs.
- class **PBDValidator**

  A static class for validating PBDs.
- class **TDValidator**

  Validator for transversal designs.

6.57 TournamentPlannerTests.Model.Designs.LatinSquares Namespace Reference

Classes

- class **LatinSquareValidator**

  A static class to validate the properties of a Latin square.
- class **MOLSValidator**

  Validator for MOLS. Validates the mutually orthogonality of all the Latin squares in the set.

6.58 TournamentPlannerTests.Model.Tournaments Namespace Reference

Classes

- class **SAMDRRValidator**

  SAMDRR validator.
- class WhistTournamentValidator

  A validator for whist tournaments.

6.59 TournamentPlannerTests.Utils Namespace Reference

Classes

- class OrthogonalArrayMOLSConverterTests

  Tests for the OrthogonalArrayMOLSConverter.
- class PrimeFactorizationTests

  Tests for the PrimeFactorization static class.

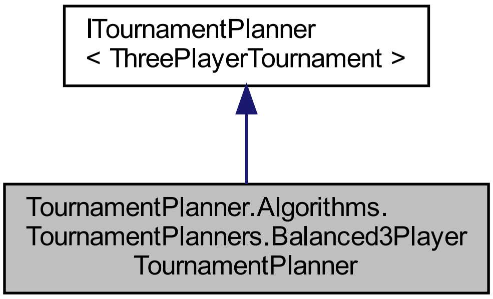
Chapter 7

Class Documentation

7.1 TournamentPlanner.Algorithms.TournamentPlanners.Balanced3↩ PlayerTournamentPlanner Class Reference

Plane a balanced three player tournament. Perfectly balanced three player tournament can only be planed, if the number of players V = 6N + 3.

Inheritance diagram for TournamentPlanner.Algorithms.TournamentPlanners.Balanced3PlayerTournament↩ Planner:


ITournamentPlanner
< ThreePlayerTournament >
TournamentPlanner.Algorithms.
TournamentPlanners.Balanced3Player
TournamentPlanner

Public Member Functions

- Balanced3PlayerTournamentPlanner (int v)

  Initializes a new instance of the Balanced3PlayerTournamentPlanner class.
- ThreePlayerTournament PlanTournament ()

7.1.1 Detailed Description

Plane a balanced three player tournament. Perfectly balanced three player tournament can only be planed, if the number of players V = 6N + 3.

7.1.2 Constructor & Destructor Documentation

7.1.2.1 Balanced3PlayerTournamentPlanner()

```
TournamentPlanner.Algorithms.TournamentPlanners.Balanced3PlayerTournamentPlanner.Balanced3↩
PlayerTournamentPlanner (
            int v )
```

Initializes a new instance of the Balanced3PlayerTournamentPlanner class.

Parameters

v	Number of players.

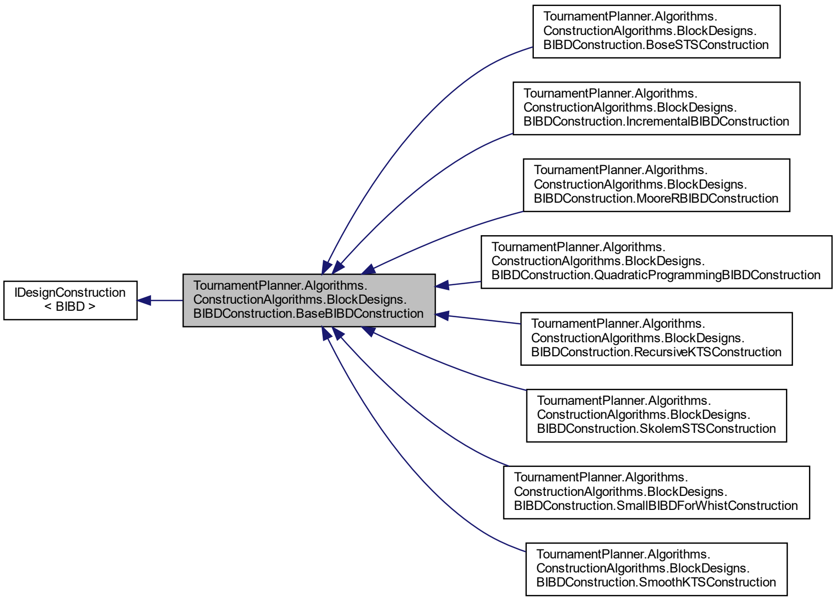
The documentation for this class was generated from the following file:

- Balanced3PlayerTournamentPlanner.cs

7.2 TournamentPlanner.Algorithms.ConstructionAlgorithms.BlockDesigns.BIBDConstruction.BaseBIBDConstruction Class Reference

Base class for the BIBD construction algorithms.

Inheritance diagram for TournamentPlanner.Algorithms.ConstructionAlgorithms.BlockDesigns.BIBDConstruction.↩ BaseBIBDConstruction:


TournamentPlanner.Algorithms. ConstructionAlgorithms.BlockDesigns. BIBDConstruction.BoseSTSConstruction
TournamentPlanner.Algorithms. ConstructionAlgorithms.BlockDesigns. BIBDConstruction.IncrementalBIBDConstruction
TournamentPlanner.Algorithms. ConstructionAlgorithms.BlockDesigns. BIBDConstruction.MooreRBIBDConstruction
TournamentPlanner.Algorithms. ConstructionAlgorithms.BlockDesigns. BIBDConstruction.QuadraticProgrammingBIBDConstruction
IDesignConstruction < BIBD >
TournamentPlanner.Algorithms. ConstructionAlgorithms.BlockDesigns. BIBDConstruction.BaseBIBDConstruction
TournamentPlanner.Algorithms. ConstructionAlgorithms.BlockDesigns. BIBDConstruction.RecursiveKTSConstruction
TournamentPlanner.Algorithms. ConstructionAlgorithms.BlockDesigns. BIBDConstruction.SkolemSTSConstruction
TournamentPlanner.Algorithms. ConstructionAlgorithms.BlockDesigns. BIBDConstruction.SmallBIBDForWhistConstruction
TournamentPlanner.Algorithms. ConstructionAlgorithms.BlockDesigns. BIBDConstruction.SmoothKTSConstruction

Public Member Functions

- abstract BIBD ConstructDesign ()

  Construct the BIBD.

Protected Member Functions

- BaseBIBDConstruction (int v, int k, int lambda)

  Initializes a new instance of the BaseBIBDConstruction class.

Properties

- int V `[get]`

  Gets the number of elements in the BIBD to construct.
- int K `[get]`

  Gets the size of a block in the BIBD to construct.
- int Lambda `[get]`

  Gets the number of times each pair of elements should occur together.
- int R `[get]`

  Gets the number of occurrences of an element in the blocks.
- int B `[get]`

  Gets the supposed number of block the design should contain.

7.2.1 Detailed Description

Base class for the BIBD construction algorithms.

7.2.2 Constructor & Destructor Documentation

7.2.2.1 BaseBIBDConstruction()

```
TournamentPlanner.Algorithms.ConstructionAlgorithms.BlockDesigns.BIBDConstruction.BaseBIBDConstruction.↩
BaseBIBDConstruction (
             int v,
             int k,
             int lambda )  [protected]
```

Initializes a new instance of the BaseBIBDConstruction class.

Parameters

v	Set size.
k	Block size.
lambda	Pair occurrences.

7.2.3 Member Function Documentation

7.2.3.1 ConstructDesign()

```
abstract BIBD TournamentPlanner.Algorithms.ConstructionAlgorithms.BlockDesigns.BIBDConstruction.↩
BaseBIBDConstruction.ConstructDesign ( )  [pure virtual]
```

Construct the BIBD.

Returns

Constructed BIBD.

Implemented in TournamentPlanner.Algorithms.ConstructionAlgorithms.BlockDesigns.BIBDConstruction.SmoothKTSConstruction, TournamentPlanner.Algorithms.ConstructionAlgorithms.BlockDesigns.BIBDConstruction.SmallBIBDForWhistConstruction, TournamentPlanner.Algorithms.ConstructionAlgorithms.BlockDesigns.BIBDConstruction.SkolemSTSConstruction, TournamentPlanner.Algorithms.ConstructionAlgorithms.BlockDesigns.BIBDConstruction.RecursiveKTSConstruction, TournamentPlanner.Algorithms.ConstructionAlgorithms.BlockDesigns.BIBDConstruction.QuadraticProgrammingBIBDConstruction, TournamentPlanner.Algorithms.ConstructionAlgorithms.BlockDesigns.BIBDConstruction.MooreRBIBDConstruction, TournamentPlanner.Algorithms.ConstructionAlgorithms.BlockDesigns.BIBDConstruction.IncrementalBIBDConstruction, and TournamentPlanner.Algorithms.ConstructionAlgorithms.BlockDesigns.BIBDConstruction.BoseSTSConstruction.

7.2.4 Property Documentation

7.2.4.1 B

```
int TournamentPlanner.Algorithms.ConstructionAlgorithms.BlockDesigns.BIBDConstruction.Base↩
BIBDConstruction.B  [get], [protected]
```

Gets the supposed number of block the design should contain.

7.2.4.2 K

```
int TournamentPlanner.Algorithms.ConstructionAlgorithms.BlockDesigns.BIBDConstruction.Base↩
BIBDConstruction.K  [get], [protected]
```

Gets the size of a block in the BIBD to construct.

7.2.4.3 Lambda

```
int TournamentPlanner.Algorithms.ConstructionAlgorithms.BlockDesigns.BIBDConstruction.Base←
BIBDConstruction.Lambda  [get], [protected]
```

Gets the number of times each pair of elements should occur together.

7.2.4.4 R

```
int TournamentPlanner.Algorithms.ConstructionAlgorithms.BlockDesigns.BIBDConstruction.Base←
BIBDConstruction.R  [get], [protected]
```

Gets the number of occurrences of an element in the blocks.

7.2.4.5 V

```
int TournamentPlanner.Algorithms.ConstructionAlgorithms.BlockDesigns.BIBDConstruction.Base←
BIBDConstruction.V  [get], [protected]
```

Gets the number of elements in the BIBD to construct.

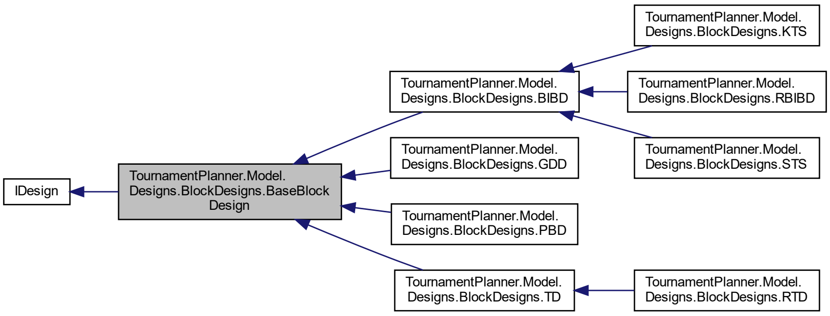
The documentation for this class was generated from the following file:

- BaseBIBDConstruction.cs

7.3 TournamentPlanner.Model.Designs.BlockDesigns.BaseBlockDesign Class Reference

Base class for the block designs.

Inheritance diagram for TournamentPlanner.Model.Designs.BlockDesigns.BaseBlockDesign:



Protected Member Functions

- BaseBlockDesign (int v)

  Initializes a new instance of the BaseBlockDesign class.

Properties

- int V `[get, protected set]`

  Gets or sets the number of elements (treatments) in the block design.
- List< List< int > > Blocks = new List<List<int>>() `[get, protected set]`

  Gets or sets the blocks of the block design.

7.3.1 Detailed Description

Base class for the block designs.

7.3.2 Constructor & Destructor Documentation

7.3.2.1 BaseBlockDesign()

```
TournamentPlanner.Model.Designs.BlockDesigns.BaseBlockDesign.BaseBlockDesign (
            int v ) [protected]
```

Initializes a new instance of the BaseBlockDesign class.

Parameters

v	Number of elements.

7.3.3 Property Documentation

7.3.3.1 Blocks

```
List<List<int> > TournamentPlanner.Model.Designs.BlockDesigns.BaseBlockDesign.Blocks = new
List<List<int>>() [get], [protected set]
```

Gets or sets the blocks of the block design.

7.3.3.2 V

```
int TournamentPlanner.Model.Designs.BlockDesigns.BaseBlockDesign.V  [get], [protected set]
```

Gets or sets the number of elements (treatments) in the block design.

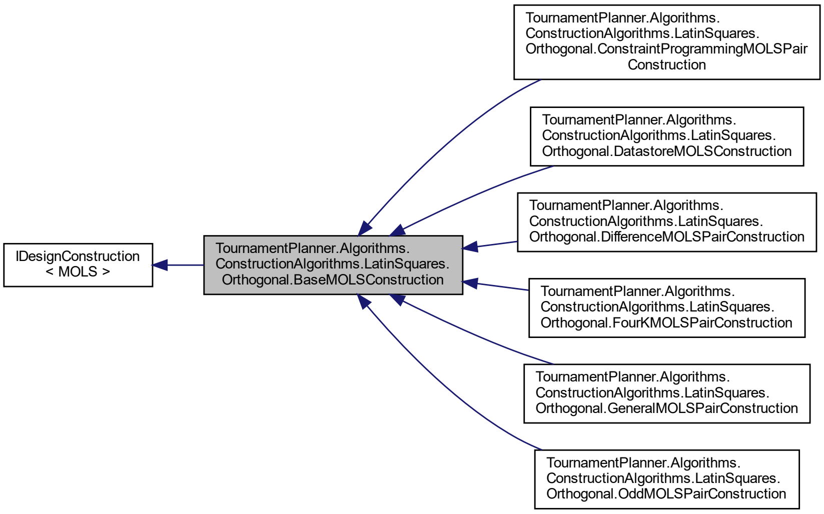
The documentation for this class was generated from the following file:

- BaseBlockDesign.cs

7.4 TournamentPlanner.Algorithms.ConstructionAlgorithms.Latin↩ Squares.Orthogonal.BaseMOLSConstruction Class Reference

Base class for the mutually orthogonal latin square construction algorithms.

Inheritance diagram for TournamentPlanner.Algorithms.ConstructionAlgorithms.LatinSquares.Orthogonal.Base↩
MOLSConstruction:


TournamentPlanner.Algorithms.ConstructionAlgorithms.LatinSquares.Orthogonal.ConstraintProgrammingMOLSPairConstruction
TournamentPlanner.Algorithms.ConstructionAlgorithms.LatinSquares.Orthogonal.DatastoreMOLSConstruction
TournamentPlanner.Algorithms.ConstructionAlgorithms.LatinSquares.Orthogonal.DifferenceMOLSPairConstruction
IDesignConstruction < MOLS >
TournamentPlanner.Algorithms.ConstructionAlgorithms.LatinSquares.Orthogonal.BaseMOLSConstruction
TournamentPlanner.Algorithms.ConstructionAlgorithms.LatinSquares.Orthogonal.FourKMOLSPairConstruction
TournamentPlanner.Algorithms.ConstructionAlgorithms.LatinSquares.Orthogonal.GeneralMOLSPairConstruction
TournamentPlanner.Algorithms.ConstructionAlgorithms.LatinSquares.Orthogonal.OddMOLSPairConstruction

Public Member Functions

- abstract MOLS ConstructDesign ()

  Construct MOLS.

Protected Member Functions

- BaseMOLSConstruction (int order)

  Initializes a new instance of the BaseMOLSConstruction class.
- abstract bool ValidateOrder (int order)

  Method to validate the order in the constructor.

Properties

- int Order `[get]`

  Gets the order of the MOLS to construct.
- abstract string ArgumentExceptionMessage `[get]`

  Gets the argument exception message of the MOLS construction algorithm.

7.4.1 Detailed Description

Base class for the mutually orthogonal latin square construction algorithms.

7.4.2 Constructor & Destructor Documentation

7.4.2.1 BaseMOLSConstruction()

```
TournamentPlanner.Algorithms.ConstructionAlgorithms.LatinSquares.Orthogonal.BaseMOLSConstruction.↩
BaseMOLSConstruction (
            int order )  [protected]
```

Initializes a new instance of the BaseMOLSConstruction class.

Parameters

order	The order of the MOLS.

7.4.3 Member Function Documentation

7.4.3.1 ConstructDesign()

```
abstract MOLS TournamentPlanner.Algorithms.ConstructionAlgorithms.LatinSquares.Orthogonal.↩
BaseMOLSConstruction.ConstructDesign ( )  [pure virtual]
```

Construct MOLS.

Returns

Constructed MOLS.

Implemented in TournamentPlanner.Algorithms.ConstructionAlgorithms.LatinSquares.Orthogonal.OddMOLSPairConstruction, TournamentPlanner.Algorithms.ConstructionAlgorithms.LatinSquares.Orthogonal.GeneralMOLSPairConstruction, TournamentPlanner.Algorithms.ConstructionAlgorithms.LatinSquares.Orthogonal.FourKMOLSPairConstruction, TournamentPlanner.Algorithms.ConstructionAlgorithms.LatinSquares.Orthogonal.DifferenceMOLSPairConstruction, TournamentPlanner.Algorithms.ConstructionAlgorithms.LatinSquares.Orthogonal.DatastoreMOLSConstruction, and TournamentPlanner.Algorithms.ConstructionAlgorithms.LatinSquares.Orthogonal.ConstraintProgrammingMOLSPairConstructio

7.4.3.2 ValidateOrder()

```
abstract bool TournamentPlanner.Algorithms.ConstructionAlgorithms.LatinSquares.Orthogonal.↩
BaseMOLSConstruction.ValidateOrder (
            int order )  [protected], [pure virtual]
```

Method to validate the order in the constructor.

Parameters

order	The order to validate.

Returns

True if the algorithm can construct MOLS of given order; otherwise false.

Implemented in TournamentPlanner.Algorithms.ConstructionAlgorithms.LatinSquares.Orthogonal.OddMOLSPairConstruction, TournamentPlanner.Algorithms.ConstructionAlgorithms.LatinSquares.Orthogonal.GeneralMOLSPairConstruction, TournamentPlanner.Algorithms.ConstructionAlgorithms.LatinSquares.Orthogonal.FourKMOLSPairConstruction, TournamentPlanner.Algorithms.ConstructionAlgorithms.LatinSquares.Orthogonal.DifferenceMOLSPairConstruction, TournamentPlanner.Algorithms.ConstructionAlgorithms.LatinSquares.Orthogonal.DatastoreMOLSConstruction, and TournamentPlanner.Algorithms.ConstructionAlgorithms.LatinSquares.Orthogonal.ConstraintProgrammingMOLSPairConstruction

7.4.4 Property Documentation

7.4.4.1 ArgumentExceptionMessage

```
abstract string TournamentPlanner.Algorithms.ConstructionAlgorithms.LatinSquares.Orthogonal.↩
BaseMOLSConstruction.ArgumentExceptionMessage  [get], [protected]
```

Gets the argument exception message of the MOLS construction algorithm.

7.4.4.2 Order

```
int TournamentPlanner.Algorithms.ConstructionAlgorithms.LatinSquares.Orthogonal.BaseMOLSConstruction.↩
Order  [get], [protected]
```

Gets the order of the MOLS to construct.

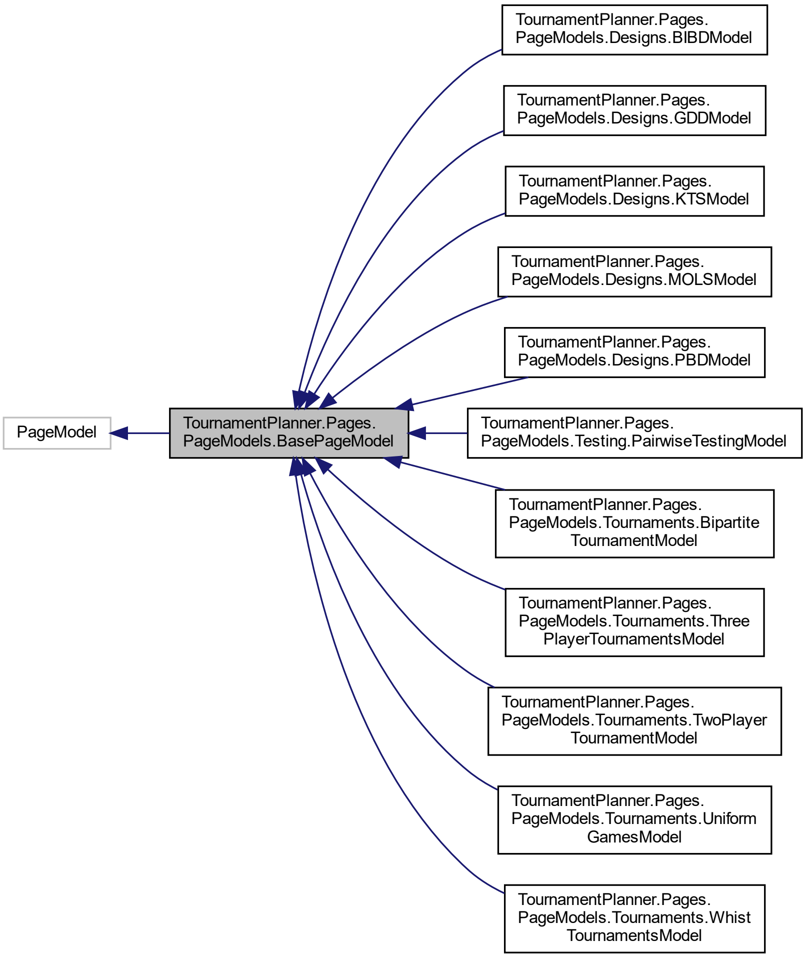
The documentation for this class was generated from the following file:

- BaseMOLSConstruction.cs

7.5 TournamentPlanner.Pages.PageModels.BasePageModel Class Reference

The base class for the page models of designs and tournaments.

Inheritance diagram for TournamentPlanner.Pages.PageModels.BasePageModel:


TournamentPlanner.Pages.PageModels.Designs.BIBDModel
TournamentPlanner.Pages.PageModels.Designs.GDDModel
TournamentPlanner.Pages.PageModels.Designs.KTSModel
TournamentPlanner.Pages.PageModels.Designs.MOLSModel
TournamentPlanner.Pages.PageModels.Designs.PBDModel
PageModel
TournamentPlanner.Pages.PageModels.BasePageModel
TournamentPlanner.Pages.PageModels.Testing.PairwiseTestingModel
TournamentPlanner.Pages.PageModels.Tournaments.BipartiteTournamentModel
TournamentPlanner.Pages.PageModels.Tournaments.ThreePlayerTournamentsModel
TournamentPlanner.Pages.PageModels.Tournaments.TwoPlayerTournamentModel
TournamentPlanner.Pages.PageModels.Tournaments.UniformGamesModel
TournamentPlanner.Pages.PageModels.Tournaments.WhistTournamentsModel

Properties

- bool DisplayResults = false `[get, protected set]`

  Gets or sets a value indicating whether to display the results.
- List< string > ErrorMessages = new List<string>() `[get]`

  Gets the error messages. If empty, the there was no error. Otherwise, the error messages shall be displayed.

7.5.1 Detailed Description

The base class for the page models of designs and tournaments.

7.5.2 Property Documentation

7.5.2.1 DisplayResults

```
bool TournamentPlanner.Pages.PageModels.BasePageModel.DisplayResults = false  [get], [protected
set]
```

Gets or sets a value indicating whether to display the results.

7.5.2.2 ErrorMessages

```
List<string> TournamentPlanner.Pages.PageModels.BasePageModel.ErrorMessages = new List<string>()
[get]
```

Gets the error messages. If empty, the there was no error. Otherwise, the error messages shall be displayed.

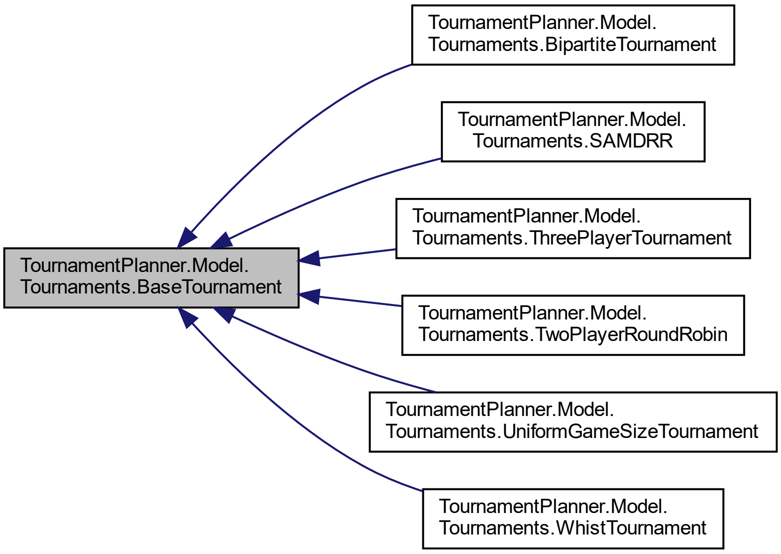
The documentation for this class was generated from the following file:

- BasePageModel.cs

7.6 TournamentPlanner.Model.Tournaments.BaseTournament Class Reference

Base abstract class for the tournaments.

Inheritance diagram for TournamentPlanner.Model.Tournaments.BaseTournament:



Public Member Functions

- void AddRound (List< List< int >> round)

  Add round to the tournament.
- string GameToString (List< int > game)

  Gets the string representation of the game.

Protected Member Functions

- BaseTournament (int order)

  Initializes a new instance of the BaseTournament class.

Properties

- int Order `[get, protected set]`

  Gets or sets the order of the tournament. The order is the number of players or 'teams'.
- List< List< List< int > > > Rounds = new List<List<List<int>>>() `[get, protected set]`

  Gets or sets the round of the tournament.

7.6.1 Detailed Description

Base abstract class for the tournaments.

7.6.2 Constructor & Destructor Documentation

7.6.2.1 BaseTournament()

```
TournamentPlanner.Model.Tournaments.BaseTournament.BaseTournament (
            int order )  [protected]
```

Initializes a new instance of the BaseTournament class.

Parameters

order	The order.

7.6.3 Member Function Documentation

7.6.3.1 AddRound()

```
void TournamentPlanner.Model.Tournaments.BaseTournament.AddRound (
            List< List< int >> round )
```

Add round to the tournament.

Parameters

round	The round.

7.6.3.2 GameToString()

```
string TournamentPlanner.Model.Tournaments.BaseTournament.GameToString (
            List< int > game )
```

Gets the string representation of the game.

Parameters

game	The game.

Returns

The string representation of the game.

7.6.4 Property Documentation

7.6.4.1 Order

```
int TournamentPlanner.Model.Tournaments.BaseTournament.Order  [get], [protected set]
```

Gets or sets the order of the tournament. The order is the number of players or 'teams'.

7.6.4.2 Rounds

```
List<List<List<int> > > TournamentPlanner.Model.Tournaments.BaseTournament.Rounds = new
List<List<List<int>>>()  [get], [protected set]
```

Gets or sets the round of the tournament.

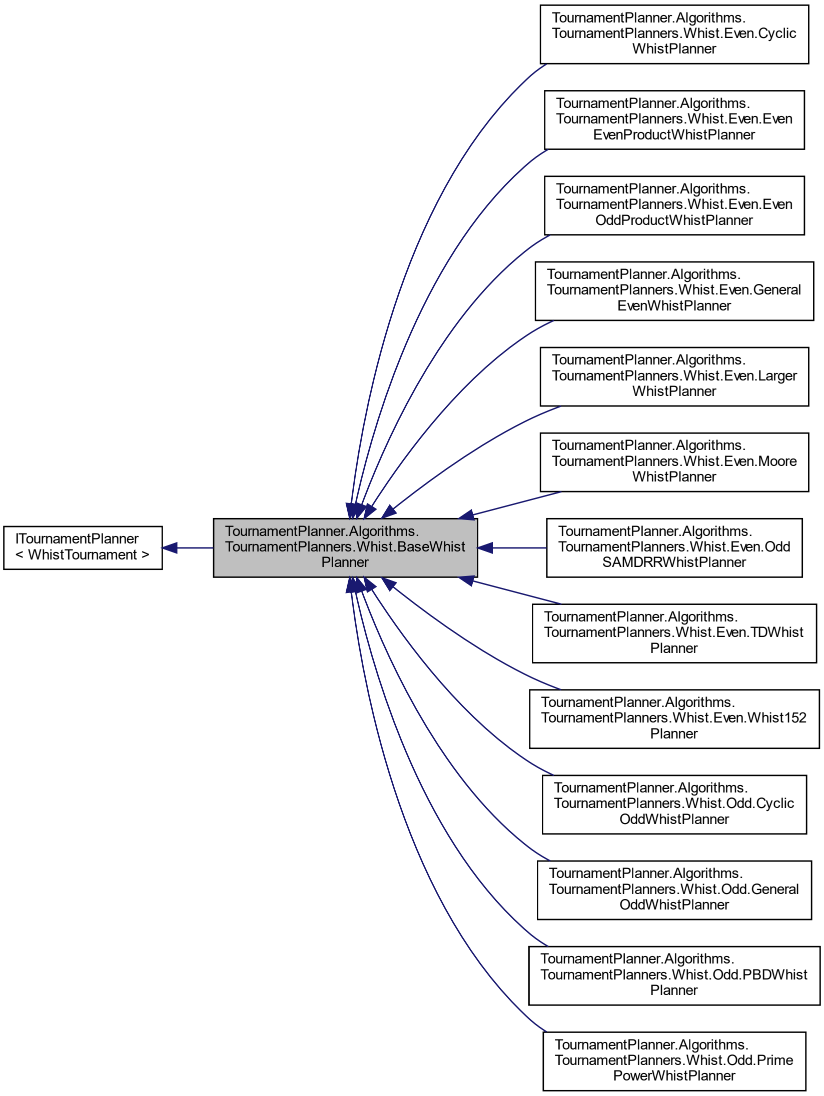
The documentation for this class was generated from the following file:

- BaseTournament.cs

7.7 TournamentPlanner.Algorithms.TournamentPlanners.Whist.Base↩ WhistPlanner Class Reference

The base class for whist tournament planners.

Inheritance diagram for TournamentPlanner.Algorithms.TournamentPlanners.Whist.BaseWhistPlanner:


TournamentPlanner.Algorithms.
TournamentPlanners.Whist.Even.Cyclic
WhistPlanner
TournamentPlanner.Algorithms.
TournamentPlanners.Whist.Even.Even
EvenProductWhistPlanner
TournamentPlanner.Algorithms.
TournamentPlanners.Whist.Even.Even
OddProductWhistPlanner
TournamentPlanner.Algorithms.
TournamentPlanners.Whist.Even.General
EvenWhistPlanner
TournamentPlanner.Algorithms.
TournamentPlanners.Whist.Even.Larger
WhistPlanner
TournamentPlanner.Algorithms.
TournamentPlanners.Whist.Even.Moore
WhistPlanner
ITournamentPlanner
< WhistTournament >
TournamentPlanner.Algorithms.
TournamentPlanners.Whist.BaseWhist
Planner
TournamentPlanner.Algorithms.
TournamentPlanners.Whist.Even.Odd
SAMDRRWhistPlanner
TournamentPlanner.Algorithms.
TournamentPlanners.Whist.Even.TDWhist
Planner
TournamentPlanner.Algorithms.
TournamentPlanners.Whist.Even.Whist152
Planner
TournamentPlanner.Algorithms.
TournamentPlanners.Whist.Odd.Cyclic
OddWhistPlanner
TournamentPlanner.Algorithms.
TournamentPlanners.Whist.Odd.General
OddWhistPlanner
TournamentPlanner.Algorithms.
TournamentPlanners.Whist.Odd.PBDWhist
Planner
TournamentPlanner.Algorithms.
TournamentPlanners.Whist.Odd.Prime
PowerWhistPlanner

Public Member Functions

- abstract WhistTournament PlanTournament ()

  Plan the Whist tournament.

Protected Member Functions

- BaseWhistPlanner (int v)

  Initializes a new instance of the BaseWhistPlanner class.

Properties

- int V `[get, set]`

  Gets or sets the order of the tournament to be constructed.

7.7.1 Detailed Description

The base class for whist tournament planners.

7.7.2 Constructor & Destructor Documentation

7.7.2.1 BaseWhistPlanner()

```
TournamentPlanner.Algorithms.TournamentPlanners.Whist.BaseWhistPlanner.BaseWhistPlanner (
            int v ) [protected]
```

Initializes a new instance of the BaseWhistPlanner class.

Parameters

v	The order of the whist tournament.

7.7.3 Member Function Documentation

7.7.3.1 PlanTournament()

```
abstract WhistTournament TournamentPlanner.Algorithms.TournamentPlanners.Whist.BaseWhist↩
Planner.PlanTournament ( ) [pure virtual]
```

Plan the Whist tournament.

Returns

The constructed whist tournament.

Implemented in TournamentPlanner.Algorithms.TournamentPlanners.Whist.Odd.PrimePowerWhistPlanner, TournamentPlanner.Algorithms.TournamentPlanners.Whist.Odd.PBDWhistPlanner, TournamentPlanner.Algorithms.TournamentPlanr
TournamentPlanner.Algorithms.TournamentPlanners.Whist.Odd.CyclicOddWhistPlanner, TournamentPlanner.Algorithms.Tournamen
TournamentPlanner.Algorithms.TournamentPlanners.Whist.Even.TDWhistPlanner, TournamentPlanner.Algorithms.TournamentPlann
TournamentPlanner.Algorithms.TournamentPlanners.Whist.Even.MooreWhistPlanner, TournamentPlanner.Algorithms.TournamentPla
TournamentPlanner.Algorithms.TournamentPlanners.Whist.Even.GeneralEvenWhistPlanner, TournamentPlanner.Algorithms.Tournan
TournamentPlanner.Algorithms.TournamentPlanners.Whist.Even.EvenEvenProductWhistPlanner, and TournamentPlanner.Algorithms

7.7.4 Property Documentation

7.7.4.1 V

```
int TournamentPlanner.Algorithms.TournamentPlanners.Whist.BaseWhistPlanner.V  [get], [set],
[protected]
```

Gets or sets the order of the tournament to be constructed.

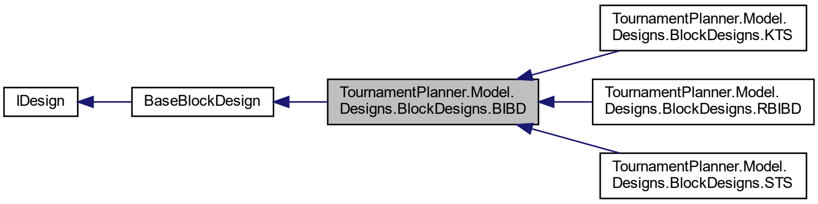
The documentation for this class was generated from the following file:

- BaseWhistPlanner.cs

7.8 TournamentPlanner.Model.Designs.BlockDesigns.BIBD Class Reference

Class representing a balanced incomplete block design.

Inheritance diagram for TournamentPlanner.Model.Designs.BlockDesigns.BIBD:


IDesign
BaseBlockDesign
TournamentPlanner.Model.Designs.BlockDesigns.BIBD
TournamentPlanner.Model.Designs.BlockDesigns.KTS
TournamentPlanner.Model.Designs.BlockDesigns.RBIBD
TournamentPlanner.Model.Designs.BlockDesigns.STS

Public Member Functions

- BIBD (int v, int k, int lambda)

  Initializes a new instance of the BIBD class.
- BIBD (int v, int k, int lambda, List< List< int >> blocks)

  Initializes a new instance of the BIBD class.
- void AddBlock (List< int > block)

  Add a single block to the list of blocks.
- void AddBlocks (List< List< int >> blocks)

  Add multiple block at once.

Properties

- int K `[get]`

  Gets the size of a block.
- int Lambda `[get]`

  Gets the number of times every pair of elements occurs together in a block.
- int R `[get]`

  Gets the number of occurrence of an element in a block. In a valid BIBD, it has to be an integer. The check is done in the construction algorithm
- int B `[get]`

  Gets the number of blocks. In a valid BIBD, it has to be an integer. The check is done in the construction algorithm

Additional Inherited Members

7.8.1 Detailed Description

Class representing a balanced incomplete block design.

7.8.2 Constructor & Destructor Documentation

7.8.2.1 BIBD() [1/2]

```
TournamentPlanner.Model.Designs.BlockDesigns.BIBD.BIBD (
            int v,
            int k,
            int lambda )
```

Initializes a new instance of the BIBD class.

Parameters

v	Set size.
k	Block size.
lambda	Pair occurrences.

7.8.2.2 BIBD() [2/2]

```
TournamentPlanner.Model.Designs.BlockDesigns.BIBD.BIBD (
            int v,
            int k,
            int lambda,
            List< List< int >> blocks )
```

Initializes a new instance of the BIBD class.

Parameters

v	Set size.
k	Block size.
lambda	Pair occurrences.
blocks	Blocks of the design.

7.8.3 Member Function Documentation

7.8.3.1 AddBlock()

```
void TournamentPlanner.Model.Designs.BlockDesigns.BIBD.AddBlock (
            List< int > block )
```

Add a single block to the list of blocks.

Parameters

block	The block to add.

7.8.3.2 AddBlocks()

```
void TournamentPlanner.Model.Designs.BlockDesigns.BIBD.AddBlocks (
            List< List< int >> blocks )
```

Add multiple block at once.

Parameters

blocks	The blocks to add.

7.8.4 Property Documentation

7.8.4.1 B

```
int TournamentPlanner.Model.Designs.BlockDesigns.BIBD.B  [get]
```

Gets the number of blocks. In a valid BIBD, it has to be an integer. The check is done in the construction algorithm

7.8.4.2 K

```
int TournamentPlanner.Model.Designs.BlockDesigns.BIBD.K  [get]
```

Gets the size of a block.

7.8.4.3 Lambda

```
int TournamentPlanner.Model.Designs.BlockDesigns.BIBD.Lambda  [get]
```

Gets the number of times every pair of elements occurs together in a block.

7.8.4.4 R

```
int TournamentPlanner.Model.Designs.BlockDesigns.BIBD.R  [get]
```

Gets the number of occurrence of an element in a block. In a valid BIBD, it has to be an integer. The check is done in the construction algorithm

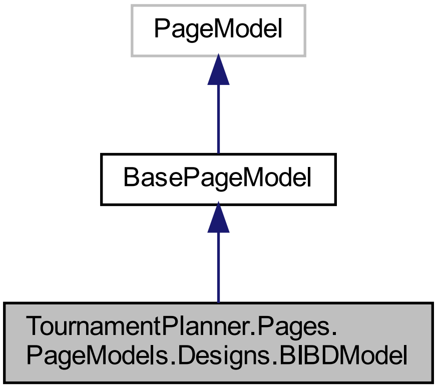
The documentation for this class was generated from the following file:

- BIBD.cs

7.9 TournamentPlanner.Pages.PageModels.Designs.BIBDModel Class Reference

The page model for BIBDs.

Inheritance diagram for TournamentPlanner.Pages.PageModels.Designs.BIBDModel:



Public Member Functions

- void OnPost ()

Properties

- int V = 7 [get, set]

  Gets or sets the size of the constructed BIBD.
- int K = 3 [get, set]

  Gets or sets the size of a single block.
- int L = 1 [get, set]

  Gets or sets the lambda of the BIBD.
- Method BlockBuildingMethod = Method.IncrementalBlockBuilding [get, set]

  Gets or sets the method to use for the construction of BIBD.
- BIBD BIBD [get]

  Gets the constructed BIBD.

7.9.1 Detailed Description

The page model for BIBDs.

7.9.2 Property Documentation

7.9.2.1 BIBD

```
BIBD TournamentPlanner.Pages.PageModels.Designs.BIBDModel.BIBD [get]
```

Gets the constructed BIBD.

7.9.2.2 BlockBuildingMethod

```
Method TournamentPlanner.Pages.PageModels.Designs.BIBDModel.BlockBuildingMethod = Method.IncrementalBlockBuil
[get], [set]
```

Gets or sets the method to use for the construction of BIBD.

7.9.2.3 K

```
int TournamentPlanner.Pages.PageModels.Designs.BIBDModel.K = 3 [get], [set]
```

Gets or sets the size of a single block.

7.9.2.4 L

```
int TournamentPlanner.Pages.PageModels.Designs.BIBDModel.L = 1  [get], [set]
```

Gets or sets the lambda of the BIBD.

7.9.2.5 V

```
int TournamentPlanner.Pages.PageModels.Designs.BIBDModel.V = 7  [get], [set]
```

Gets or sets the size of the constructed BIBD.

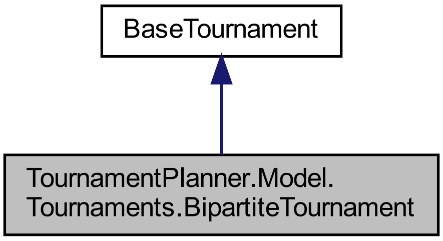
The documentation for this class was generated from the following file:

- BIBD.cshtml.cs

7.10 TournamentPlanner.Model.Tournaments.BipartiteTournament Class Reference

Class representing a bipartite tournament. In a bipartite tournament, there are 2 teams with the same number of players. The goal is, that every player from the first team should play a game with every player from the second team.

Inheritance diagram for TournamentPlanner.Model.Tournaments.BipartiteTournament:



Public Member Functions

- BipartiteTournament (int teamSize)

  Initializes a new instance of the BipartiteTournament class.
- new string GameToString (List< int > game)

Additional Inherited Members

7.10.1 Detailed Description

Class representing a bipartite tournament. In a bipartite tournament, there are 2 teams with the same number of players. The goal is, that every player from the first team should play a game with every player from the second team.

7.10.2 Constructor & Destructor Documentation

7.10.2.1 BipartiteTournament()

```
TournamentPlanner.Model.Tournaments.BipartiteTournament.BipartiteTournament (
            int teamSize )
```

Initializes a new instance of the BipartiteTournament class.

Parameters

teamSize	The size of a team.

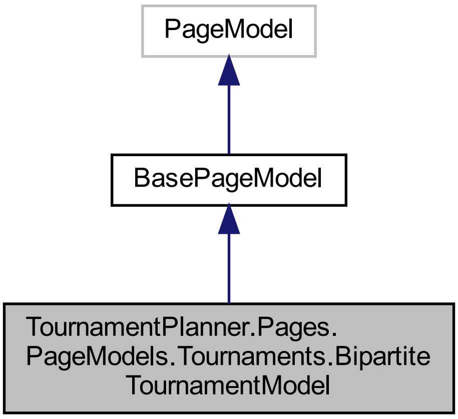
The documentation for this class was generated from the following file:

- BipartiteTournament.cs

7.11 TournamentPlanner.Pages.PageModels.Tournaments.Bipartite↩ TournamentModel Class Reference

Page model for the construction of bipartite tournaments.

Inheritance diagram for TournamentPlanner.Pages.PageModels.Tournaments.BipartiteTournamentModel:



Public Member Functions

- void OnPost ()

Properties

- int TeamSize `[get, set]`

  Gets or sets the size of teams.
- BipartiteTournament Tournament `[get]`

  Gets the constructed tournament.

7.11.1 Detailed Description

Page model for the construction of bipartite tournaments.

7.11.2 Property Documentation

7.11.2.1 TeamSize

```
int TournamentPlanner.Pages.PageModels.Tournaments.BipartiteTournamentModel.TeamSize  [get],
[set]
```

Gets or sets the size of teams.

7.11.2.2 Tournament

```
BipartiteTournament TournamentPlanner.Pages.PageModels.Tournaments.BipartiteTournamentModel.↩
Tournament  [get]
```

Gets the constructed tournament.

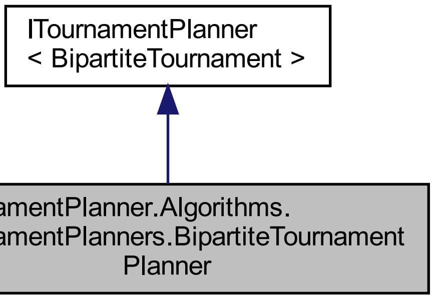
The documentation for this class was generated from the following file:

- BipartiteTournament.cshtml.cs

7.12 TournamentPlanner.Algorithms.TournamentPlanners.Bipartite↩ TournamentPlanner Class Reference

Bipartite tournament planner using Latin squares.

Inheritance diagram for TournamentPlanner.Algorithms.TournamentPlanners.BipartiteTournamentPlanner:


ITournamentPlanner
< BipartiteTournament >
TournamentPlanner.Algorithms.
TournamentPlanners.BipartiteTournament
Planner

Public Member Functions

- BipartiteTournamentPlanner (int teamSize)

  Initializes a new instance of the BipartiteTournamentPlanner class.
- BipartiteTournament PlanTournament ()

  Plane a bipartite tournament.

7.12.1 Detailed Description

Bipartite tournament planner using Latin squares.

7.12.2 Constructor & Destructor Documentation

7.12.2.1 BipartiteTournamentPlanner()

```
TournamentPlanner.Algorithms.TournamentPlanners.BipartiteTournamentPlanner.BipartiteTournament↩
Planner (
            int teamSize )
```

Initializes a new instance of the BipartiteTournamentPlanner class.

Parameters

teamSize	Size of teams.

7.12.3 Member Function Documentation

7.12.3.1 PlanTournament()

```
BipartiteTournament TournamentPlanner.Algorithms.TournamentPlanners.BipartiteTournament↩
Planner.PlanTournament ( )
```

Plane a bipartite tournament.

Returns

A bipartite tournament.

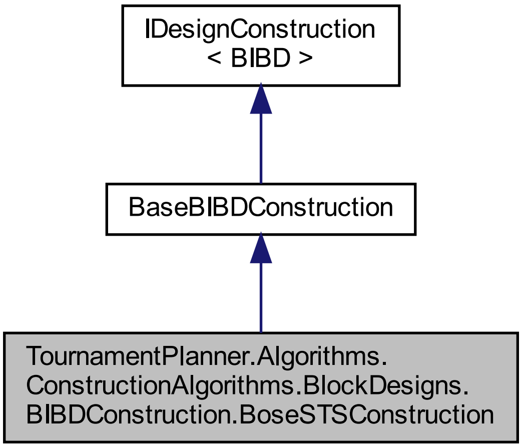
The documentation for this class was generated from the following file:

- BipartiteTournamentPlanner.cs

7.13 TournamentPlanner.Algorithms.ConstructionAlgorithms.Block↩ Designs.BIBDConstruction.BoseSTSConstruction Class Reference

Bose's algorithm for the construction of STS of size V = 6N + 3. http://math.ucdenver.edu/∼wcherowi/courses/m6406/sts.↩
pdf

Inheritance diagram for TournamentPlanner.Algorithms.ConstructionAlgorithms.BlockDesigns.BIBDConstruction.↩ BoseSTSConstruction:


IDesignConstruction
< BIBD >
BaseBIBDConstruction
TournamentPlanner.Algorithms.
ConstructionAlgorithms.BlockDesigns.
BIBDConstruction.BoseSTSConstruction

Public Member Functions

- BoseSTSConstruction (int v)

  Initializes a new instance of the BoseSTSConstruction class.
- override STS ConstructDesign ()

  Construct STS of size 6N + 3.

Additional Inherited Members

7.13.1 Detailed Description

Bose's algorithm for the construction of STS of size V = 6N + 3. http://math.ucdenver.edu/∼wcherowi/courses/m6406/sts.↩ pdf

7.13.2 Constructor & Destructor Documentation

7.13.2.1 BoseSTSConstruction()

```
TournamentPlanner.Algorithms.ConstructionAlgorithms.BlockDesigns.BIBDConstruction.BoseSTSConstruction.↩
BoseSTSConstruction (
              int v )
```

Initializes a new instance of the BoseSTSConstruction class.

Parameters

v	Set size.

7.13.3 Member Function Documentation

7.13.3.1 ConstructDesign()

```
override STS TournamentPlanner.Algorithms.ConstructionAlgorithms.BlockDesigns.BIBDConstruction.↩
BoseSTSConstruction.ConstructDesign ( ) [virtual]
```

Construct STS of size 6N + 3.

Returns

The constructed STS.

Implements TournamentPlanner.Algorithms.ConstructionAlgorithms.BlockDesigns.BIBDConstruction.BaseBIBDConstruction.

The documentation for this class was generated from the following file:

- BoseSTSConstruction.cs

7.14 TournamentPlannerTests.Algorithms.ConstructionAlgorithms.↩ BlockDesigns.BIBDConstruction.BoseSTSConstructionTests Class Reference

Tests for the BoseSTSConstruction.

Public Member Functions

- void BoseSTSConstructionTest ()

  Test for a few values of set size 6N + 3.

7.14.1 Detailed Description

Tests for the BoseSTSConstruction.

7.14.2 Member Function Documentation

7.14.2.1 BoseSTSConstructionTest()

```
void TournamentPlannerTests.Algorithms.ConstructionAlgorithms.BlockDesigns.BIBDConstruction.↩
BoseSTSConstructionTests.BoseSTSConstructionTest ( )
```

Test for a few values of set size 6N + 3.

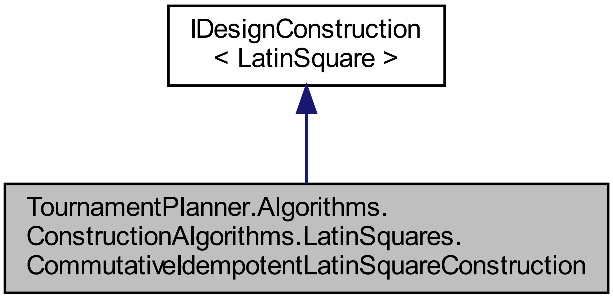
The documentation for this class was generated from the following file:

- BoseSTSConstructionTests.cs

7.15 TournamentPlanner.Algorithms.ConstructionAlgorithms.Latin↩Squares.CommutativeIdempotentLatinSquareConstruction Class Reference

Algorithm for the creation od commutative idempotent Latin squares of odd order

Inheritance diagram for TournamentPlanner.Algorithms.ConstructionAlgorithms.LatinSquares.Commutative↩IdempotentLatinSquareConstruction:


IDesignConstruction
< LatinSquare >
TournamentPlanner.Algorithms.
ConstructionAlgorithms.LatinSquares.
CommutativeIdempotentLatinSquareConstruction

Public Member Functions

- CommutativeIdempotentLatinSquareConstruction (int order)

    Initializes a new instance of the CommutativeIdempotentLatinSquareConstruction class.
- LatinSquare ConstructDesign ()

    Construct a commutative idempotent Latin square of odd order.

7.15.1 Detailed Description

Algorithm for the creation od commutative idempotent Latin squares of odd order

7.15.2 Constructor & Destructor Documentation

7.15.2.1 CommutativeIdempotentLatinSquareConstruction()

```
TournamentPlanner.Algorithms.ConstructionAlgorithms.LatinSquares.CommutativeIdempotentLatin↩
SquareConstruction.CommutativeIdempotentLatinSquareConstruction (
            int order )
```

Initializes a new instance of the CommutativeIdempotentLatinSquareConstruction class.

Parameters

order	The LS order.

7.15.3 Member Function Documentation

7.15.3.1 ConstructDesign()

```
LatinSquare TournamentPlanner.Algorithms.ConstructionAlgorithms.LatinSquares.Commutative↩
IdempotentLatinSquareConstruction.ConstructDesign ( )
```

Construct a commutative idempotent Latin square of odd order.

Returns

Latin square.

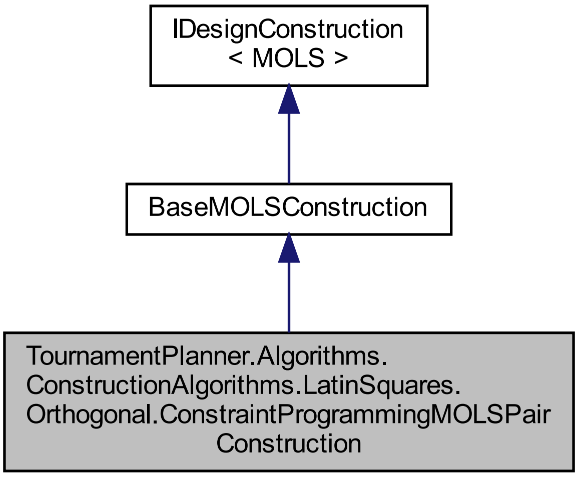
The documentation for this class was generated from the following file:

- CommutativeIdempotentLatinSquareConstruction.cs

7.16 TournamentPlanner.Algorithms.ConstructionAlgorithms.Latin↩ Squares.Orthogonal.ConstraintProgrammingMOLSPair↩ Construction Class Reference

Constraint a pair of MOLS using constraint programming approach with CP-SAT solver. This method only works for MOLS of small orders.

Inheritance diagram for TournamentPlanner.Algorithms.ConstructionAlgorithms.LatinSquares.Orthogonal.↩ ConstraintProgrammingMOLSPairConstruction:


IDesignConstruction < MOLS >
BaseMOLSConstruction
TournamentPlanner.Algorithms. ConstructionAlgorithms.LatinSquares. Orthogonal.ConstraintProgrammingMOLSPair Construction

Public Member Functions

- ConstraintProgrammingMOLSPairConstruction (int order)
- override MOLS ConstructDesign ()

  Construct MOLS.

Protected Member Functions

- override bool ValidateOrder (int order)

  Method to validate the order in the constructor.

Properties

- override string ArgumentExceptionMessage `[get]`

7.16.1 Detailed Description

Constraint a pair of MOLS using constraint programming approach with CP-SAT solver. This method only works for MOLS of small orders.

The documentation for this class was generated from the following file:

- ConstraintProgrammingMOLSPairConstruction.cs

7.17 TournamentPlannerTests.Algorithms.ConstructionAlgorithms.↩ LatinSquares.Orthogonal.ConstraintProgrammingMOLSPair↩ ConstructionTests Class Reference

Tests for the ConstraintProgrammingMOLSPairConstruction.

Public Member Functions

- void InvalidOrderShouldThrowException ()

  Test that an exception is thrown when the order is 2 or 6.
- void OrthogonalityTest ()

  Test the orthogonality of a few pairs of MOLS of low order.

7.17.1 Detailed Description

Tests for the ConstraintProgrammingMOLSPairConstruction.

7.17.2 Member Function Documentation

7.17.2.1 InvalidOrderShouldThrowException()

```
void TournamentPlannerTests.Algorithms.ConstructionAlgorithms.LatinSquares.Orthogonal.Constraint↩
ProgrammingMOLSPairConstructionTests.InvalidOrderShouldThrowException ( )
```

Test that an exception is thrown when the order is 2 or 6.

7.17.2.2 OrthogonalityTest()

```
void TournamentPlannerTests.Algorithms.ConstructionAlgorithms.LatinSquares.Orthogonal.Constraint↩
ProgrammingMOLSPairConstructionTests.OrthogonalityTest ( )
```

Test the orthogonality of a few pairs of MOLS of low order.

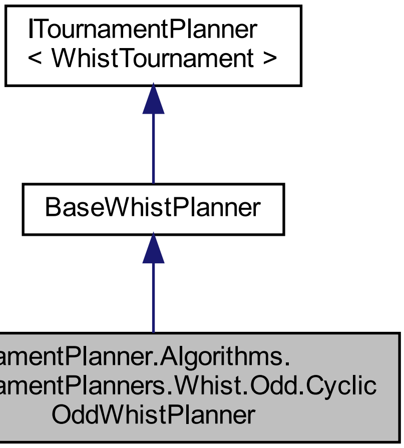
The documentation for this class was generated from the following file:

- ConstraintProgrammingMOLSPairConstructionTests.cs

7.18 TournamentPlanner.Algorithms.TournamentPlanners.Whist.Odd.↩ CyclicOddWhistPlanner Class Reference

Cyclic odd whist planner for v = 33, 57.

Inheritance diagram for TournamentPlanner.Algorithms.TournamentPlanners.Whist.Odd.CyclicOddWhistPlanner:



Public Member Functions

- CyclicOddWhistPlanner (int v)

    Initializes a new instance of the CyclicOddWhistPlanner class.
- override WhistTournament PlanTournament ()

    Plan the Whist tournament.

Additional Inherited Members

7.18.1 Detailed Description

Cyclic odd whist planner for v = 33, 57.

7.18.2 Constructor & Destructor Documentation

7.18.2.1 CyclicOddWhistPlanner()

```
TournamentPlanner.Algorithms.TournamentPlanners.Whist.Odd.CyclicOddWhistPlanner.CyclicOdd↩
WhistPlanner (
              int v )
```

Initializes a new instance of the CyclicOddWhistPlanner class.

Parameters

v	The order.

The documentation for this class was generated from the following file:

- CyclicOddWhistPlanner.cs

7.19 TournamentPlannerTests.Algorithms.TournamentPlanners.Whist.↩ Odd.CyclicOddWhistPlannerTests Class Reference

Tests for the CyclicOddWhistPlanner.

Public Member Functions

- void TestWhist33Construction ()

  Tests the construction of Whist(33).
- void TestWhist57Construction ()

  Test the construction of Whist 57.

7.19.1 Detailed Description

Tests for the CyclicOddWhistPlanner.

7.19.2 Member Function Documentation

7.19.2.1 TestWhist33Construction()

```
void TournamentPlannerTests.Algorithms.TournamentPlanners.Whist.Odd.CyclicOddWhistPlanner↩
Tests.TestWhist33Construction ( )
```

Tests the construction of Whist(33).

7.19.2.2 TestWhist57Construction()

```
void TournamentPlannerTests.Algorithms.TournamentPlanners.Whist.Odd.CyclicOddWhistPlanner↩
Tests.TestWhist57Construction ( )
```

Test the construction of Whist 57.

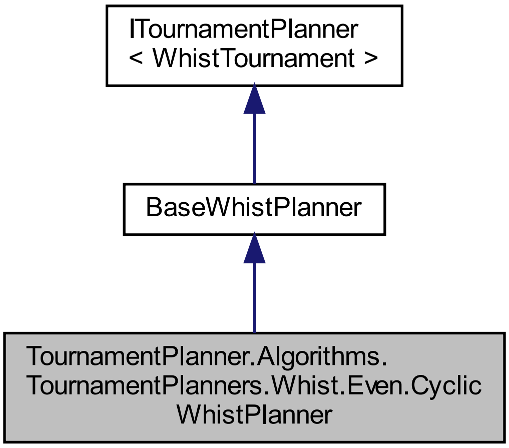
The documentation for this class was generated from the following file:

- CyclicOddWhistPlannerTests.cs

7.20 TournamentPlanner.Algorithms.TournamentPlanners.Whist.Even.↩ CyclicWhistPlanner Class Reference

Cyclic construction method for several orders of Whist(V = 4N).

Inheritance diagram for TournamentPlanner.Algorithms.TournamentPlanners.Whist.Even.CyclicWhistPlanner:


ITournamentPlanner
< WhistTournament >
BaseWhistPlanner
TournamentPlanner.Algorithms.
TournamentPlanners.Whist.Even.Cyclic
WhistPlanner

Public Member Functions

- CyclicWhistPlanner (int v)

  Initializes a new instance of the CyclicWhistPlanner class.
- override WhistTournament PlanTournament ()

  Plan the Whist tournament.

Additional Inherited Members

7.20.1 Detailed Description

Cyclic construction method for several orders of Whist(V = 4N).

7.20.2 Constructor & Destructor Documentation

7.20.2.1 CyclicWhistPlanner()

```
TournamentPlanner.Algorithms.TournamentPlanners.Whist.Even.CyclicWhistPlanner.CyclicWhist↩
Planner (
            int v )
```

Initializes a new instance of the CyclicWhistPlanner class.

Parameters

v	The order of the whist tournament.

The documentation for this class was generated from the following file:

- CyclicWhistPlanner.cs

7.21 TournamentPlannerTests.Algorithms.TournamentPlanners.Whist.↩ Even.CyclicWhistPlannerTests Class Reference

Tests for the CyclicWhistPlanner.

Public Member Functions

- void CyclicWhistConstructionsTest ()

  Tests the construction of the supported cyclic whist tournaments..

7.21.1 Detailed Description

Tests for the CyclicWhistPlanner.

7.21.2 Member Function Documentation

7.21.2.1 CyclicWhistConstructionsTest()

```
void TournamentPlannerTests.Algorithms.TournamentPlanners.Whist.Even.CyclicWhistPlanner↩
Tests.CyclicWhistConstructionsTest ( )
```

Tests the construction of the supported cyclic whist tournaments..

The documentation for this class was generated from the following file:

- CyclicWhistPlannerTests.cs

7.22 TournamentPlannerTests.Datastore.DataReaderTests Class Reference

Tests for the DataReader.

Public Member Functions

- void **TestExistingNbMOLS** ()

7.22.1 Detailed Description

Tests for the DataReader.

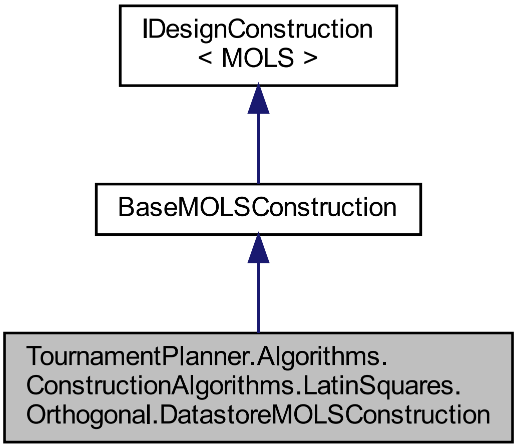
The documentation for this class was generated from the following file:

- DataReaderTests.cs

7.23 TournamentPlanner.Algorithms.ConstructionAlgorithms.Latin↩ Squares.Orthogonal.DatastoreMOLSConstruction Class Reference

This class creates MOLS objects based on the ones in the data store. This is not a construction algorithm per se, but acts like one. It implements the same interface as the other MOLS construction methods so it is easier to use in other algorithms.

Inheritance diagram for TournamentPlanner.Algorithms.ConstructionAlgorithms.LatinSquares.Orthogonal.↩ DatastoreMOLSConstruction:


IDesignConstruction < MOLS >
BaseMOLSConstruction
TournamentPlanner.Algorithms. ConstructionAlgorithms.LatinSquares. Orthogonal.DatastoreMOLSConstruction

Public Member Functions

- DatastoreMOLSConstruction (int order, int nbMols=2)
- override MOLS ConstructDesign ()

    Construct MOLS.

Protected Member Functions

- override bool ValidateOrder (int order)

  Method to validate the order in the constructor.

Properties

- override string ArgumentExceptionMessage `[get]`

7.23.1 Detailed Description

This class creates MOLS objects based on the ones in the data store. This is not a construction algorithm per se, but acts like one. It implements the same interface as the other MOLS construction methods so it is easier to use in other algorithms.

The documentation for this class was generated from the following file:

- DatastoreMOLSConstruction.cs

7.24 TournamentPlannerTests.Algorithms.ConstructionAlgorithms.↩ LatinSquares.Orthogonal.DatastoreMOLSConstructionTests Class Reference

Tests for the DatastoreMOLSConstruction.

Public Member Functions

- void SeveralOrdersTest ()

  Test the retrieval of MOLS of several orders and their validity.

7.24.1 Detailed Description

Tests for the DatastoreMOLSConstruction.

7.24.2 Member Function Documentation

7.24.2.1 SeveralOrdersTest()

```
void TournamentPlannerTests.Algorithms.ConstructionAlgorithms.LatinSquares.Orthogonal.Datastore↩
MOLSConstructionTests.SeveralOrdersTest ( )
```

Test the retrieval of MOLS of several orders and their validity.

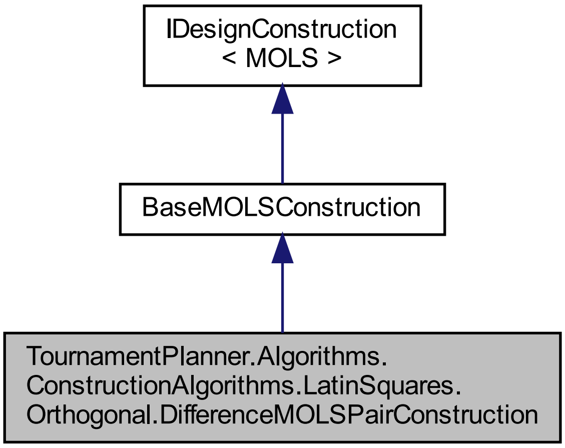
The documentation for this class was generated from the following file:

- DatastoreMOLSConstructionTests.cs

7.25 TournamentPlanner.Algorithms.ConstructionAlgorithms.Latin↩ Squares.Orthogonal.DifferenceMOLSPairConstruction Class Reference

Construct a pair of MOLS of order 3M + 1. This way, MOLS for all orders 12t + 10 (which is a subset of 4k + 2) can be constructed. This method was introduced in https://www.math.uci.edu/∼brusso/[14]BosShrParCJM1960.pdf.

Inheritance diagram for TournamentPlanner.Algorithms.ConstructionAlgorithms.LatinSquares.Orthogonal.↩ DifferenceMOLSPairConstruction:


IDesignConstruction < MOLS >
BaseMOLSConstruction
TournamentPlanner.Algorithms. ConstructionAlgorithms.LatinSquares. Orthogonal.DifferenceMOLSPairConstruction

Public Member Functions

- DifferenceMOLSPairConstruction (int order)
- override MOLS ConstructDesign ()

  Construct MOLS.

Protected Member Functions

- override bool ValidateOrder (int order)

  Method to validate the order in the constructor.

Properties

- override string ArgumentExceptionMessage `[get]`

7.25.1 Detailed Description

Construct a pair of MOLS of order 3M + 1. This way, MOLS for all orders 12t + 10 (which is a subset of 4k + 2) can be constructed. This method was introduced in https://www.math.uci.edu/∼brusso/[14]BosShrParCJM1960.pdf.

The documentation for this class was generated from the following file:

- DifferenceMOLSPairConstruction.cs

7.26 TournamentPlannerTests.Algorithms.ConstructionAlgorithms.↩ LatinSquares.Orthogonal.DifferenceMOLSPairConstructionTests Class Reference

Tests for the DifferenceMOLSPairConstruction.

Public Member Functions

- void InvalidOrderTest ()

  Test that an exception is thrown when order is not 3m + 1.
- void OrthogonalityTest ()

  Test the orthogonality of a few pairs of MOLS of order 3M + 1.

7.26.1 Detailed Description

Tests for the DifferenceMOLSPairConstruction.

7.26.2 Member Function Documentation

7.26.2.1 InvalidOrderTest()

```
void TournamentPlannerTests.Algorithms.ConstructionAlgorithms.LatinSquares.Orthogonal.Difference↩
MOLSPairConstructionTests.InvalidOrderTest ( )
```

Test that an exception is thrown when order is not 3m + 1.

7.26.2.2 OrthogonalityTest()

```
void TournamentPlannerTests.Algorithms.ConstructionAlgorithms.LatinSquares.Orthogonal.Difference↩
MOLSPairConstructionTests.OrthogonalityTest ( )
```

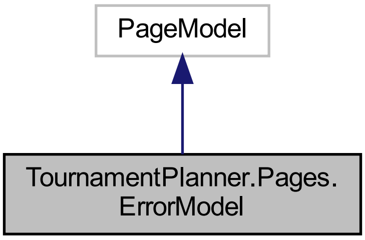
Test the orthogonality of a few pairs of MOLS of order 3M + 1.

The documentation for this class was generated from the following file:

- DifferenceMOLSPairConstructionTests.cs

7.27 TournamentPlanner.Pages.ErrorModel Class Reference

Inheritance diagram for TournamentPlanner.Pages.ErrorModel:



Public Member Functions

- **ErrorModel** (ILogger< ErrorModel > logger)
- void **OnGet** ()

Properties

- string **RequestId** `[get, set]`
- bool **ShowRequestId** `[get]`

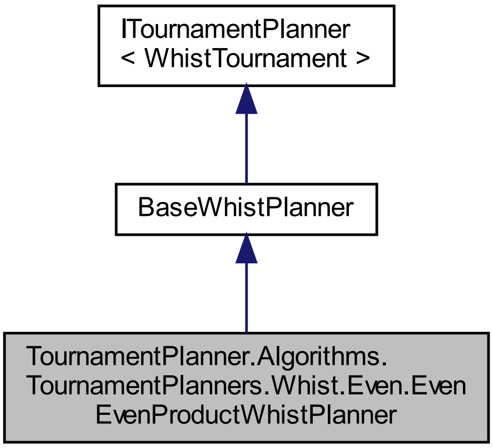
The documentation for this class was generated from the following file:

- Error.cshtml.cs

7.28 TournamentPlanner.Algorithms.TournamentPlanners.Whist.Even.↩ EvenEvenProductWhistPlanner Class Reference

Construction of a Whist tournament as a product of 2 existing Wh(4n) tournaments.

Inheritance diagram for TournamentPlanner.Algorithms.TournamentPlanners.Whist.Even.EvenEvenProductWhist↩ Planner:



Public Member Functions

- EvenEvenProductWhistPlanner (int v, WhistTournament whistV, WhistTournament whistW)

  Initializes a new instance of the EvenEvenProductWhistPlanner class.
- override WhistTournament PlanTournament ()

  Plan the Whist tournament.

Additional Inherited Members

7.28.1 Detailed Description

Construction of a Whist tournament as a product of 2 existing Wh(4n) tournaments.

7.28.2 Constructor & Destructor Documentation

7.28.2.1 EvenEvenProductWhistPlanner()

```
TournamentPlanner.Algorithms.TournamentPlanners.Whist.Even.EvenEvenProductWhistPlanner.Even↩
EvenProductWhistPlanner (
             int v,
             WhistTournament whistV,
             WhistTournament whistW )
```

Initializes a new instance of the EvenEvenProductWhistPlanner class.

Parameters

v	Order of new whist.
whistV	First Wh(v).
whistW	Second Wh(w).

The documentation for this class was generated from the following file:

- EvenEvenProductWhistPlanner.cs

7.29 TournamentPlannerTests.Algorithms.TournamentPlanners.Whist.↩ Even.EvenEvenProductWhistPlannerTests Class Reference

Tests for the ProductWhistPlanner.

Public Member Functions

- void **TwoEvenProductTest** ()
- void **Whist32Test** ()
- void **Whit64Test** ()
- void **Whit80Test** ()
- void **Whit96Test** ()
- void **Whit144Test** ()

7.29.1 Detailed Description

Tests for the ProductWhistPlanner.

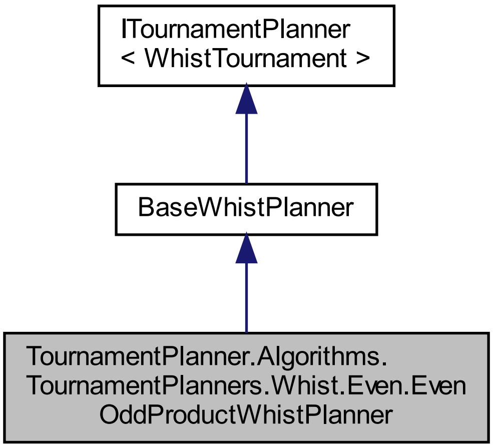
The documentation for this class was generated from the following file:

- EvenEvenProductWhistPlannerTests.cs

7.30 TournamentPlanner.Algorithms.TournamentPlanners.Whist.Even.↩ EvenOddProductWhistPlanner Class Reference

Construction of a Whist tournament as a product of a Wh(4n) and Wh(4n + 1).

Inheritance diagram for TournamentPlanner.Algorithms.TournamentPlanners.Whist.Even.EvenOddProductWhistPlanner:



Public Member Functions

- EvenOddProductWhistPlanner (int v, WhistTournament whistV, WhistTournament whistW)

  Initializes a new instance of the EvenOddProductWhistPlanner class.
- override WhistTournament PlanTournament ()

  Plan the Whist tournament.

Additional Inherited Members

7.30.1 Detailed Description

Construction of a Whist tournament as a product of a Wh(4n) and Wh(4n + 1).

7.30.2 Constructor & Destructor Documentation

7.30.2.1 EvenOddProductWhistPlanner()

```
TournamentPlanner.Algorithms.TournamentPlanners.Whist.Even.EvenOddProductWhistPlanner.EvenOddProductWhistPlanner (
             int v,
             WhistTournament whistV,
             WhistTournament whistW )
```

Initializes a new instance of the EvenOddProductWhistPlanner class.

Parameters

v	Order of new whist.
whistV	First Wh(4n).
whistW	Second Wh(4n+1).

The documentation for this class was generated from the following file:

- EvenOddProductWhistPlanner.cs

7.31 TournamentPlannerTests.Algorithms.TournamentPlanners.Whist.←Even.EvenOddProductWhistPlannerTests Class Reference

Tests for the EvenOddProductWhistPlanner.

Public Member Functions

- void **Whist36ConstructionTest** ()
- void **Whist60ConstructionTest** ()

7.31.1 Detailed Description

Tests for the EvenOddProductWhistPlanner.

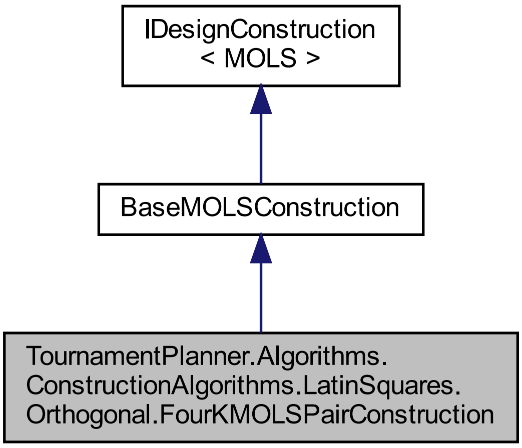
The documentation for this class was generated from the following file:

- EvenOddProductWhistPlannerTests.cs

7.32 TournamentPlanner.Algorithms.ConstructionAlgorithms.Latin←Squares.Orthogonal.FourKMOLSPairConstruction Class Reference

Construction of MOLS of order V = 4 $*$ K.

Inheritance diagram for TournamentPlanner.Algorithms.ConstructionAlgorithms.LatinSquares.Orthogonal.FourKMOLSPairConstruction:


IDesignConstruction
< MOLS >
BaseMOLSConstruction
TournamentPlanner.Algorithms.
ConstructionAlgorithms.LatinSquares.
Orthogonal.FourKMOLSPairConstruction

Public Member Functions

- FourKMOLSPairConstruction (int order)
- override MOLS ConstructDesign ()

  Construct MOLS.

Protected Member Functions

- override bool ValidateOrder (int order)

  Method to validate the order in the constructor.

Properties

- override string ArgumentExceptionMessage `[get]`

7.32.1 Detailed Description

Construction of MOLS of order V = 4 ∗ K.

The documentation for this class was generated from the following file:

- FourKMOLSPairConstruction.cs

7.33 TournamentPlannerTests.Algorithms.ConstructionAlgorithms.↩ LatinSquares.Orthogonal.FourKMOLSPairConstructionTests Class Reference

Tests for the FourKMOLSPairConstruction.

Public Member Functions

- void InvalidOrderTest ()

  Test that an exception is thrown when the order is not a multiple of 4.
- void OrthogonalityTest ()

  Test the orthogonality of a few pairs of MOLS of order 4 K.

7.33.1 Detailed Description

Tests for the FourKMOLSPairConstruction.

7.33.2 Member Function Documentation

7.33.2.1 InvalidOrderTest()

```
void TournamentPlannerTests.Algorithms.ConstructionAlgorithms.LatinSquares.Orthogonal.Four↩
KMOLSPairConstructionTests.InvalidOrderTest ( )
```

Test that an exception is thrown when the order is not a multiple of 4.

7.33.2.2 OrthogonalityTest()

```
void TournamentPlannerTests.Algorithms.ConstructionAlgorithms.LatinSquares.Orthogonal.Four↩
KMOLSPairConstructionTests.OrthogonalityTest ( )
```

Test the orthogonality of a few pairs of MOLS of order 4 K.

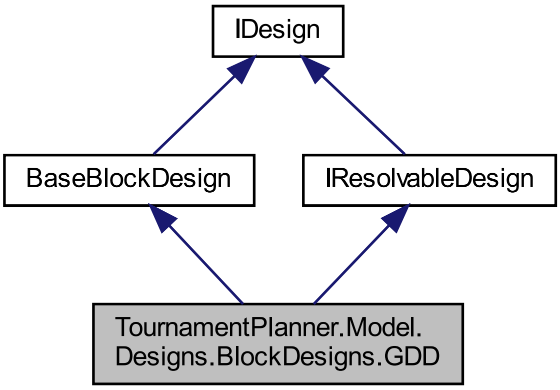
The documentation for this class was generated from the following file:

- FourKMOLSPairConstructionTests.cs

7.34 TournamentPlanner.Model.Designs.BlockDesigns.GDD Class Reference

Class representing a group divisible design.

Inheritance diagram for TournamentPlanner.Model.Designs.BlockDesigns.GDD:



Public Member Functions

- GDD (int v)

  Initializes a new instance of the GDD class.
- GDD (int v, List< List< int >> groups)

  Initializes a new instance of the GDD class.
- GDD (int v, int k, List< List< int >> groups)

  Initializes a new instance of the GDD class.
- GDD (int v, int k, List< List< int >> groups, List< List< int >> blocks)

  Initializes a new instance of the GDD class.
- GDD (int v, List< int > blockSizes, List< List< int >> groups, List< List< int >> blocks)

  Initializes a new instance of the GDD class.
- void AddParallelClassAndBlocks (List< List< int >> parallelClass)

  Add a parallel class and its blocks at the same time.
- void AddBlock (List< int > block)

  Ass a single block to the blocks of the design.
- List< int > GetGroupSizes ()

  Get the size(s) of the groups in the GDD.

Properties

- int K = 3 `[get]`

  Gets the block size.
- List< int > BlockSizes = new List<int> { 3 } `[get]`

  Gets the allowed block sizes.
- List< List< int > > Groups = new () `[get]`

  Gets the groups of the GDD.
- List< List< List< int > > > ParallelClasses = new () `[get]`

  Gets the parallel classes of the GDD.

Additional Inherited Members

7.34.1 Detailed Description

Class representing a group divisible design.

7.34.2 Constructor & Destructor Documentation

7.34.2.1 GDD() [1/5]

```
TournamentPlanner.Model.Designs.BlockDesigns.GDD.GDD (
             int v )
```

Initializes a new instance of the GDD class.

Parameters

v	Size of the GDD.

7.34.2.2 GDD() [2/5]

```
TournamentPlanner.Model.Designs.BlockDesigns.GDD.GDD (
             int v,
             List< List< int >> groups )
```

Initializes a new instance of the GDD class.

Parameters

v	Size of the GDD.
groups	The groups of the GDD.

7.34.2.3 GDD() [3/5]

```
TournamentPlanner.Model.Designs.BlockDesigns.GDD.GDD (
            int v,
            int k,
            List< List< int >> groups )
```

Initializes a new instance of the GDD class.

Parameters

v	Size of the GDD.
k	Size of the blocks.
groups	The groups of the GDD.

7.34.2.4 GDD() [4/5]

```
TournamentPlanner.Model.Designs.BlockDesigns.GDD.GDD (
            int v,
            int k,
            List< List< int >> groups,
            List< List< int >> blocks )
```

Initializes a new instance of the GDD class.

Parameters

v	Size of the GDD.
k	Size of the blocks.
groups	The groups of the GDD.
blocks	The blocks of the GDD.

7.34.2.5 GDD() [5/5]

```
TournamentPlanner.Model.Designs.BlockDesigns.GDD.GDD (
            int v,
            List< int > blockSizes,
            List< List< int >> groups,
            List< List< int >> blocks )
```

Initializes a new instance of the GDD class.

Parameters

v	Size of the GDD.
blockSizes	Sizes of the blocks.
groups	The groups of the GDD.
blocks	The blocks of the GDD.

7.34.3 Member Function Documentation

7.34.3.1 AddBlock()

```
void TournamentPlanner.Model.Designs.BlockDesigns.GDD.AddBlock (
            List< int > block )
```

Ass a single block to the blocks of the design.

Parameters

block	The block to add.

7.34.3.2 AddParallelClassAndBlocks()

```
void TournamentPlanner.Model.Designs.BlockDesigns.GDD.AddParallelClassAndBlocks (
            List< List< int >> parallelClass )
```

Add a parallel class and its blocks at the same time.

Parameters

parallelClass	The parallel class to add.

7.34.3.3 GetGroupSizes()

```
List<int> TournamentPlanner.Model.Designs.BlockDesigns.GDD.GetGroupSizes ( )
```

Get the size(s) of the groups in the GDD.

Returns

List of sizes.

7.34.4 Property Documentation

7.34.4.1 BlockSizes

```
List<int> TournamentPlanner.Model.Designs.BlockDesigns.GDD.BlockSizes = new List<int> { 3 }
[get]
```

Gets the allowed block sizes.

7.34.4.2 Groups

```
List<List<int> > TournamentPlanner.Model.Designs.BlockDesigns.GDD.Groups = new ()  [get]
```

Gets the groups of the GDD.

7.34.4.3 K

```
int TournamentPlanner.Model.Designs.BlockDesigns.GDD.K = 3  [get]
```

Gets the block size.

7.34.4.4 ParallelClasses

```
List<List<List<int> > > TournamentPlanner.Model.Designs.BlockDesigns.GDD.ParallelClasses =
new ()  [get]
```

Gets the parallel classes of the GDD.

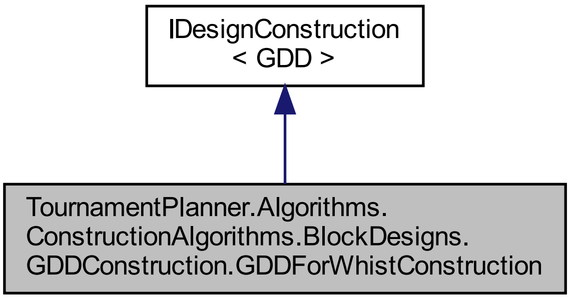
The documentation for this class was generated from the following file:

- GDD.cs

7.35 TournamentPlanner.Algorithms.ConstructionAlgorithms.Block↩ Designs.GDDConstruction.GDDForWhistConstruction Class Reference

Construction of GDD(17g, + 4u + 4v, {5, 17}, {g, g + 4u, g + 4v}). These are used to construct whist tournaments.

Inheritance diagram for TournamentPlanner.Algorithms.ConstructionAlgorithms.BlockDesigns.GDDConstruction.↩ GDDForWhistConstruction:


IDesignConstruction
< GDD >
TournamentPlanner.Algorithms.
ConstructionAlgorithms.BlockDesigns.
GDDConstruction.GDDForWhistConstruction

Public Member Functions

- GDDForWhistConstruction (int g, int u, int v)

  Initializes a new instance of the GDDForWhistConstruction class.
- GDD ConstructDesign ()

7.35.1 Detailed Description

Construction of GDD(17g, + 4u + 4v, {5, 17}, {g, g + 4u, g + 4v}). These are used to construct whist tournaments.

7.35.2 Constructor & Destructor Documentation

7.35.2.1 GDDForWhistConstruction()

```
TournamentPlanner.Algorithms.ConstructionAlgorithms.BlockDesigns.GDDConstruction.GDDForWhist↩
Construction.GDDForWhistConstruction (
            int g,
            int u,
            int v )
```

Initializes a new instance of the GDDForWhistConstruction class.

Parameters

g	Initial size of groups.
u	Number of elements to replace in the first group.
v	Number of elements to replace in the second group.

The documentation for this class was generated from the following file:

- GDDForWhistConstruction.cs

7.36 TournamentPlannerTests.Algorithms.ConstructionAlgorithms.↩ BlockDesigns.GDDConstruction.GDDForWhistConstructionTests Class Reference

Tests for the GDDForWhistConstruction.

Public Member Functions

- void **GDDConstructionTest** ()
- void **GDDConstructionWithReplacementsIn2GroupsTest** ()

7.36.1 Detailed Description

Tests for the GDDForWhistConstruction.

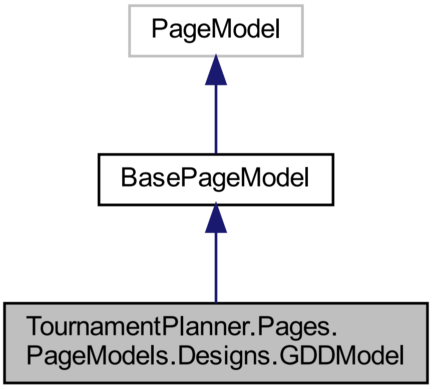
The documentation for this class was generated from the following file:

- GDDForWhistConstructionTests.cs

7.37 TournamentPlanner.Pages.PageModels.Designs.GDDModel Class Reference

Page model for resolvable group divisible designs with block size 3.

Inheritance diagram for TournamentPlanner.Pages.PageModels.Designs.GDDModel:



Public Member Functions

- void OnPost ()

Properties

- int M = 4 [get, set]

  Gets or sets the M which is the number of groups and the order of MOLS used for the construction.
- int V = 12 [get, set]

  Gets or sets the size of the GDD. V = 3 ∗ M.
- GDD GDD [get]

  Gets the constructed GDD.

7.37.1 Detailed Description

Page model for resolvable group divisible designs with block size 3.

7.37.2 Property Documentation

7.37.2.1 GDD

```
GDD TournamentPlanner.Pages.PageModels.Designs.GDDModel.GDD  [get]
```

Gets the constructed GDD.

7.37.2.2 M

```
int TournamentPlanner.Pages.PageModels.Designs.GDDModel.M = 4  [get], [set]
```

Gets or sets the M which is the number of groups and the order of MOLS used for the construction.

7.37.2.3 V

```
int TournamentPlanner.Pages.PageModels.Designs.GDDModel.V = 12  [get], [set]
```

Gets or sets the size of the GDD. V = 3 ∗ M.

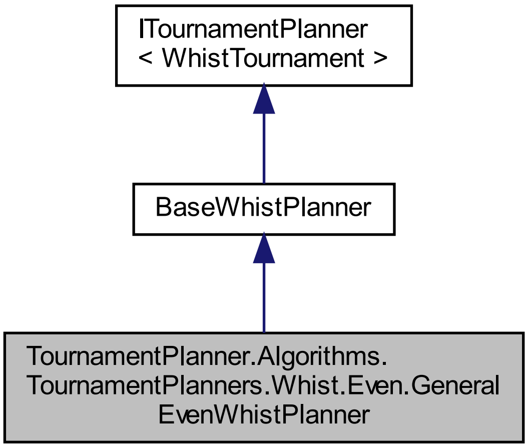
The documentation for this class was generated from the following file:

- GDD.cshtml.cs

7.38 TournamentPlanner.Algorithms.TournamentPlanners.Whist.Even.GeneralEvenWhistPlanner Class Reference

Plan a whist tournament of order 4n or 4n + 1 using some of the implemented approaches. This class basically just selects the method to use for the construction.

Inheritance diagram for TournamentPlanner.Algorithms.TournamentPlanners.Whist.Even.GeneralEvenWhistPlanner:



Public Member Functions

- GeneralEvenWhistPlanner (int v)

  Initializes a new instance of the GeneralEvenWhistPlanner class.
- override WhistTournament PlanTournament ()

  Plan the Whist tournament.

Additional Inherited Members

7.38.1 Detailed Description

Plan a whist tournament of order 4n or 4n + 1 using some of the implemented approaches. This class basically just selects the method to use for the construction.

7.38.2 Constructor & Destructor Documentation

7.38.2.1 GeneralEvenWhistPlanner()

```
TournamentPlanner.Algorithms.TournamentPlanners.Whist.Even.GeneralEvenWhistPlanner.General↩
EvenWhistPlanner (
             int v )
```

Initializes a new instance of the GeneralEvenWhistPlanner class.

Parameters

v	The order of the tournament.

The documentation for this class was generated from the following file:

- GeneralEvenWhistPlanner.cs

7.39 TournamentPlannerTests.Algorithms.TournamentPlanners.Whist.↩ Even.GeneralEvenWhistPlannerTests Class Reference

Tests for the General4NWhistPlanner.

Public Member Functions

- void SmallerWhistConstructionTest ()

  Validate constructed whist tournaments up to 81.
- void LargerWhistConstructionTest ()

  Test the construction of larger whist tournaments. Commented because it takes a long time but passes.
- void LargerConstructionWithoutValidationTest ()

  Construction without validation. Takes a long ass time too.

7.39.1 Detailed Description

Tests for the General4NWhistPlanner.

7.39.2 Member Function Documentation

7.39.2.1 LargerConstructionWithoutValidationTest()

```
void TournamentPlannerTests.Algorithms.TournamentPlanners.Whist.Even.GeneralEvenWhistPlanner←↩
Tests.LargerConstructionWithoutValidationTest ( )
```

Construction without validation. Takes a long ass time too.

[Test]

7.39.2.2 LargerWhistConstructionTest()

```
void TournamentPlannerTests.Algorithms.TournamentPlanners.Whist.Even.GeneralEvenWhistPlanner←↩
Tests.LargerWhistConstructionTest ( )
```

Test the construction of larger whist tournaments. Commented because it takes a long time but passes.

[Test]

7.39.2.3 SmallerWhistConstructionTest()

```
void TournamentPlannerTests.Algorithms.TournamentPlanners.Whist.Even.GeneralEvenWhistPlanner←↩
Tests.SmallerWhistConstructionTest ( )
```

Validate constructed whist tournaments up to 81.

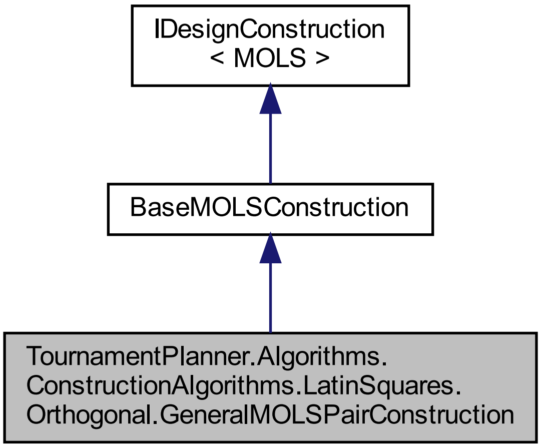
The documentation for this class was generated from the following file:

- GeneralEvenWhistPlannerTests.cs

7.40 TournamentPlanner.Algorithms.ConstructionAlgorithms.Latin↩ Squares.Orthogonal.GeneralMOLSPairConstruction Class Reference

Class facilitating the construction of MOLS of all the supported orders. Uses other construction algorithms.

Inheritance diagram for TournamentPlanner.Algorithms.ConstructionAlgorithms.LatinSquares.Orthogonal.↩ GeneralMOLSPairConstruction:


IDesignConstruction < MOLS >

TournamentPlanner.Algorithms. ConstructionAlgorithms.LatinSquares. Orthogonal.GeneralMOLSPairConstruction

Public Member Functions

- GeneralMOLSPairConstruction (int order)
- override MOLS ConstructDesign ()

  Construct MOLS.

Protected Member Functions

- override bool ValidateOrder (int order)

  Method to validate the order in the constructor.

Properties

- override string ArgumentExceptionMessage `[get]`

7.40.1 Detailed Description

Class facilitating the construction of MOLS of all the supported orders. Uses other construction algorithms.

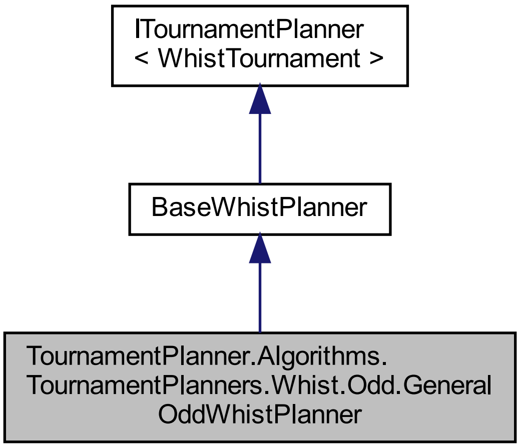
The documentation for this class was generated from the following file:

- GeneralMOLSPairConstruction.cs

7.41 TournamentPlanner.Algorithms.TournamentPlanners.Whist.Odd.↩ GeneralOddWhistPlanner Class Reference

Construct some Whist tournaments of order 4n + 1. Not all orders are supported unfortunately.

Inheritance diagram for TournamentPlanner.Algorithms.TournamentPlanners.Whist.Odd.GeneralOddWhist↩ Planner:



Public Member Functions

- GeneralOddWhistPlanner (int v)

  Initializes a new instance of the GeneralOddWhistPlanner class.
- override WhistTournament PlanTournament ()

  Plan the Whist tournament.

Additional Inherited Members

7.41.1 Detailed Description

Construct some Whist tournaments of order 4n + 1. Not all orders are supported unfortunately.

7.41.2 Constructor & Destructor Documentation

7.41.2.1 GeneralOddWhistPlanner()

```
TournamentPlanner.Algorithms.TournamentPlanners.Whist.Odd.GeneralOddWhistPlanner.GeneralOdd↩
WhistPlanner (
             int v )
```

Initializes a new instance of the GeneralOddWhistPlanner class.

Parameters

v	Number of players.

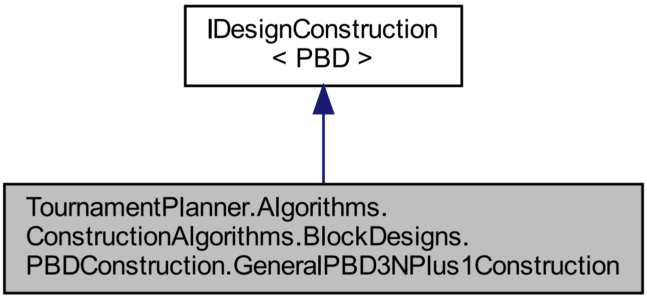
The documentation for this class was generated from the following file:

- GeneralOddWhistPlanner.cs

7.42 TournamentPlanner.Algorithms.ConstructionAlgorithms.Block↩ Designs.PBDConstruction.GeneralPBD3NPlus1Construction Class Reference

Class that is used for construction of any PBD(3n+1). It uses the required depending on the order. It is also used in the recursive step of the LargerOrderPBDConstruction.

Inheritance diagram for TournamentPlanner.Algorithms.ConstructionAlgorithms.BlockDesigns.PBDConstruction.↩ GeneralPBD3NPlus1Construction:


IDesignConstruction
< PBD >
TournamentPlanner.Algorithms.
ConstructionAlgorithms.BlockDesigns.
PBDConstruction.GeneralPBD3NPlus1Construction

Public Member Functions

- GeneralPBD3NPlus1Construction (int v)

    Initializes a new instance of the GeneralPBD3NPlus1Construction class.
- PBD ConstructDesign ()

7.42.1 Detailed Description

Class that is used for construction of any PBD(3n+1). It uses the required depending on the order. It is also used in the recursive step of the LargerOrderPBDConstruction.

7.42.2 Constructor & Destructor Documentation

7.42.2.1 GeneralPBD3NPlus1Construction()

```
TournamentPlanner.Algorithms.ConstructionAlgorithms.BlockDesigns.PBDConstruction.GeneralPBD3↩
NPlus1Construction.GeneralPBD3NPlus1Construction (
            int v )
```

Initializes a new instance of the GeneralPBD3NPlus1Construction class.

Parameters

v	The PBD order.

The documentation for this class was generated from the following file:

- GeneralPBD3NPlus1Construction.cs

7.43 TournamentPlannerTests.Algorithms.ConstructionAlgorithms.↩ BlockDesigns.BIBDConstruction.GurobiIncremental↩ BIBDConstructionTests Class Reference

Tests for the GurobiIncrementalBIBDConstruction.

Public Member Functions

- void IncrementalBIBDConstructionTest ()

  Test the algorithm on a few BIBDs.

7.43.1 Detailed Description

Tests for the GurobiIncrementalBIBDConstruction.

7.43.2 Member Function Documentation

7.43.2.1 IncrementalBIBDConstructionTest()

```
void TournamentPlannerTests.Algorithms.ConstructionAlgorithms.BlockDesigns.BIBDConstruction.↩
GurobiIncrementalBIBDConstructionTests.IncrementalBIBDConstructionTest ( )
```

Test the algorithm on a few BIBDs.

The documentation for this class was generated from the following file:

- GurobiIncrementalBIBDConstructionTests.cs

7.44 TournamentPlannerTests.Algorithms.ConstructionAlgorithms.↩ BlockDesigns.BIBDConstruction.GurobiQuadraticProgramming↩ BIBDConstructionTests Class Reference

Tests fot the QuadraticProgrammingBIBDConstruction.

Public Member Functions

- void **QudraticConstructionTest** ()

7.44.1 Detailed Description

Tests fot the QuadraticProgrammingBIBDConstruction.

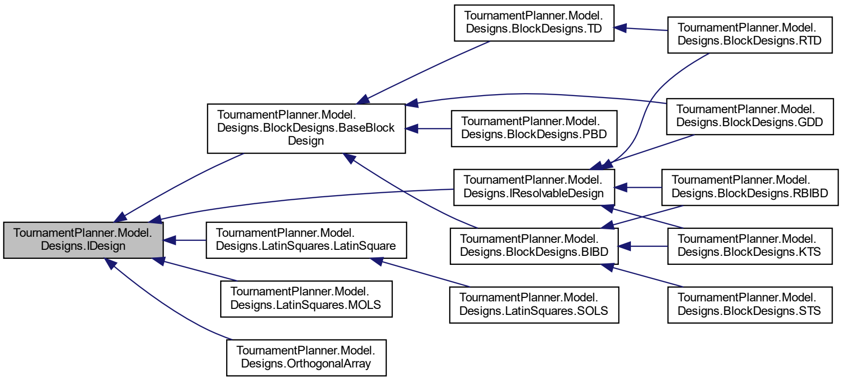
The documentation for this class was generated from the following file:

- GurobiQuadraticProgrammingBIBDConstructionTests.cs

7.45 TournamentPlanner.Model.Designs.IDesign Interface Reference

Common interface for all the design structures.

Inheritance diagram for TournamentPlanner.Model.Designs.IDesign:


TournamentPlanner.Model. Designs.BlockDesigns.TD
TournamentPlanner.Model. Designs.BlockDesigns.RTD
TournamentPlanner.Model. Designs.BlockDesigns.BaseBlock Design
TournamentPlanner.Model. Designs.BlockDesigns.PBD
TournamentPlanner.Model. Designs.BlockDesigns.GDD
TournamentPlanner.Model. Designs.IResolvableDesign
TournamentPlanner.Model. Designs.BlockDesigns.RBIBD
TournamentPlanner.Model. Designs.IDesign
TournamentPlanner.Model. Designs.LatinSquares.LatinSquare
TournamentPlanner.Model. Designs.BlockDesigns.BIBD
TournamentPlanner.Model. Designs.BlockDesigns.KTS
TournamentPlanner.Model. Designs.LatinSquares.MOLS
TournamentPlanner.Model. Designs.LatinSquares.SOLS
TournamentPlanner.Model. Designs.BlockDesigns.STS
TournamentPlanner.Model. Designs.OrthogonalArray

Properties

- int V `[get]`

  Gets the size of the design. In literature, V denotes the size of the design - the number of distinct elements in the design. It is probably the only common parameter of all the designs. From the tournament POV, it denotes the number of players. In Latin squares, V is the order of the square. In the context of orthogonal arrays, V denotes the level of the OA.

7.45.1 Detailed Description

Common interface for all the design structures.

7.45.2 Property Documentation

7.45.2.1 V

```
int TournamentPlanner.Model.Designs.IDesign.V  [get]
```

Gets the size of the design. In literature, V denotes the size of the design - the number of distinct elements in the design. It is probably the only common parameter of all the designs. From the tournament POV, it denotes the number of players. In Latin squares, V is the order of the square. In the context of orthogonal arrays, V denotes the level of the OA.

The documentation for this interface was generated from the following file:

- IDesign.cs

7.46 TournamentPlanner.Algorithms.ConstructionAlgorithms.IDesign↩ Construction< IDesign > Interface Template Reference

Interface for the design construction algorithms.

Public Member Functions

- IDesign ConstructDesign ()

  Construct the design.

7.46.1 Detailed Description

Interface for the design construction algorithms.

Template Parameters

IDesign	The type of IDesign.

7.46.2 Member Function Documentation

7.46.2.1 ConstructDesign()

```
IDesign TournamentPlanner.Algorithms.ConstructionAlgorithms.IDesignConstruction< IDesign >.Construct↩
Design ( )
```

Construct the design.

Returns

Concrete design.

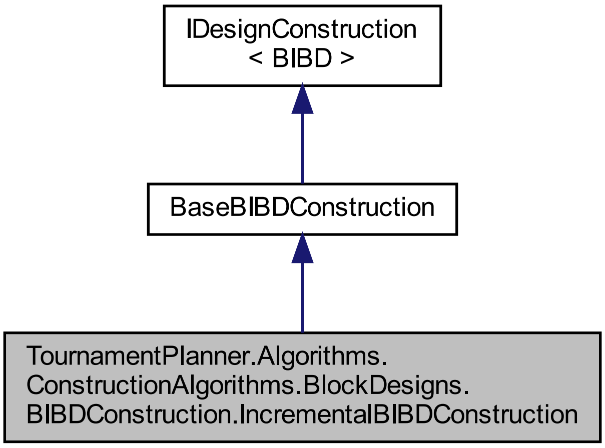
The documentation for this interface was generated from the following file:

- IDesignConstruction.cs

7.47 TournamentPlanner.Algorithms.ConstructionAlgorithms.Block↩ Designs.BIBDConstruction.IncrementalBIBDConstruction Class Reference

Incremental method that constructs BIBDs block by block. Implementation of the algorithm described in https↩ ://www.researchgate.net/publication/220660995_A_mathematical_programming_approach_to_the_construction↩ _of_BIBDs. Uses CP-SAT as solver.

Inheritance diagram for TournamentPlanner.Algorithms.ConstructionAlgorithms.BlockDesigns.BIBDConstruction.↩ IncrementalBIBDConstruction:


IDesignConstruction
< BIBD >
BaseBIBDConstruction
TournamentPlanner.Algorithms.
ConstructionAlgorithms.BlockDesigns.
BIBDConstruction.IncrementalBIBDConstruction

Public Member Functions

- IncrementalBIBDConstruction (int v, int k, int lambda, int timeLimit=10)
- override BIBD ConstructDesign ()

  Construct the BIBD.

Additional Inherited Members

7.47.1 Detailed Description

Incremental method that constructs BIBDs block by block. Implementation of the algorithm described in https://www.researchgate.net/publication/220660995_A_mathematical_programming_approach_to_the_construction_of_BIBDs. Uses CP-SAT as solver.

The documentation for this class was generated from the following file:

- IncrementalBIBDConstruction.cs

7.48 TournamentPlannerTests.Algorithms.ConstructionAlgorithms.↩ BlockDesigns.BIBDConstruction.IncrementalBIBDConstruction↩ Tests Class Reference

Test for the IncrementalBIBDConstruction.

Public Member Functions

- void IncrementalBIBDConstructionTest ()

  Test the algorithm on a few BIBDs.

7.48.1 Detailed Description

Test for the IncrementalBIBDConstruction.

7.48.2 Member Function Documentation

7.48.2.1 IncrementalBIBDConstructionTest()

```
void TournamentPlannerTests.Algorithms.ConstructionAlgorithms.BlockDesigns.BIBDConstruction.↩
IncrementalBIBDConstructionTests.IncrementalBIBDConstructionTest ( )
```

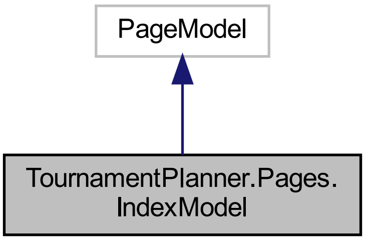
Test the algorithm on a few BIBDs.

The documentation for this class was generated from the following file:

- IncrementalBIBDConstructionTests.cs

7.49 TournamentPlanner.Pages.IndexModel Class Reference

Inheritance diagram for TournamentPlanner.Pages.IndexModel:



Public Member Functions

- **IndexModel** (ILogger< IndexModel > logger)
- void **OnGet** ()

Properties

- string **Message** = "PageModel in C#" `[get]`

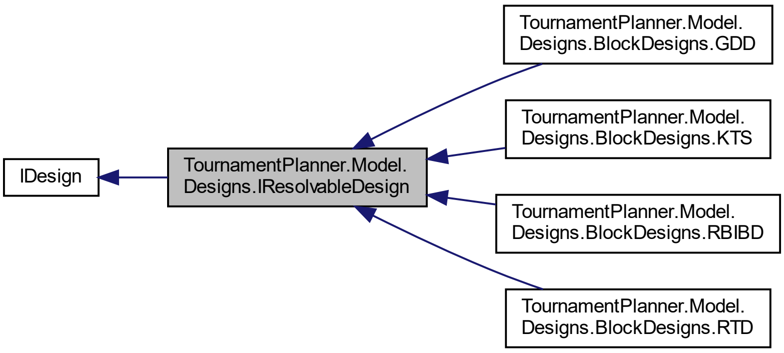
The documentation for this class was generated from the following file:

- Index.cshtml.cs

7.50 TournamentPlanner.Model.Designs.IResolvableDesign Interface Reference

Interface for resolvable designs.

Inheritance diagram for TournamentPlanner.Model.Designs.IResolvableDesign:



Properties

- List< List< List< int > > > ParallelClasses `[get]`

  Gets the parallel classes of the resolvable design.

7.50.1 Detailed Description

Interface for resolvable designs.

7.50.2 Property Documentation

7.50.2.1 ParallelClasses

```
List<List<List<int> > > TournamentPlanner.Model.Designs.IResolvableDesign.ParallelClasses
[get]
```

Gets the parallel classes of the resolvable design.

The documentation for this interface was generated from the following file:

- IResolvableDesign.cs

7.51 TournamentPlanner.Algorithms.TournamentPlanners.ITournamentPlanner< BaseTournament > Interface Template Reference

Interface for the tournament planners.

Public Member Functions

- BaseTournament PlanTournament ()

  Plan the given tournament.

7.51.1 Detailed Description

Interface for the tournament planners.

Template Parameters

BaseTournament	Type of the tournament to plan.

7.51.2 Member Function Documentation

7.51.2.1 PlanTournament()

```
BaseTournament TournamentPlanner.Algorithms.TournamentPlanners.ITournamentPlanner< BaseTournament
>.PlanTournament ( )
```

Plan the given tournament.

Returns

The planned tournament.

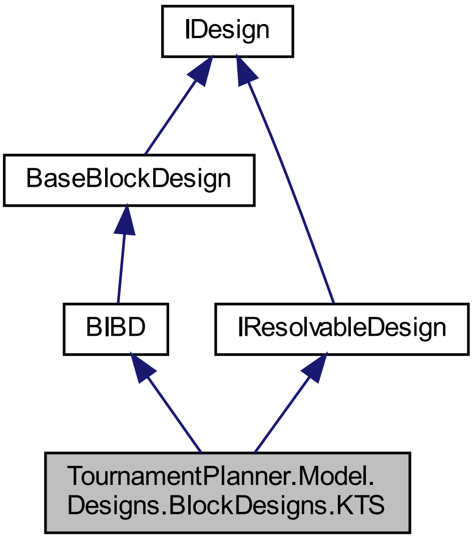
The documentation for this interface was generated from the following file:

- ITournamentPlanner.cs

7.52 TournamentPlanner.Model.Designs.BlockDesigns.KTS Class Reference

A class representing a Kirkman triple system. A KTS is a resolvable BIBD(V, 3, 1), where V = 6N + 3. Basically, a resolvable STS i a KTS.

Inheritance diagram for TournamentPlanner.Model.Designs.BlockDesigns.KTS:



Public Member Functions

- KTS (int v)

  Initializes a new instance of the KTS class.
- KTS (int v, List< List< List< int >>> parallelClasses)

  Initializes a new instance of the KTS class.
- void AddParallelClass (List< List< int >> parallelClass)

  Add a parallel class to the list of parallel classes.
- void AddParallelClassAndBlocks (List< List< int >> parallelClass)

  Add a parallel class to the list of parallel classes. At the same time, it adds the blocks of the parallel class to the list of blocks.

Properties

- List< List< List< int > > > ParallelClasses = new List<List<List<int>>>() `[get]`

Additional Inherited Members

7.52.1 Detailed Description

A class representing a Kirkman triple system. A KTS is a resolvable BIBD(V, 3, 1), where V = 6N + 3. Basically, a resolvable STS i a KTS.

7.52.2 Constructor & Destructor Documentation

7.52.2.1 KTS() [1/2]

```
TournamentPlanner.Model.Designs.BlockDesigns.KTS.KTS (
            int v )
```

Initializes a new instance of the KTS class.

Parameters

v	Set size.

7.52.2.2 KTS() [2/2]

```
TournamentPlanner.Model.Designs.BlockDesigns.KTS.KTS (
            int v,
            List< List< List< int >>> parallelClasses )
```

Initializes a new instance of the KTS class.

Parameters

v	Set size.
parallelClasses	The parallel classes.

7.52.3 Member Function Documentation

7.52.3.1 AddParallelClass()

```
void TournamentPlanner.Model.Designs.BlockDesigns.KTS.AddParallelClass (
            List< List< int >> parallelClass )
```

Add a parallel class to the list of parallel classes.

Parameters

parallelClass	Parallel class to add.

7.52.3.2 AddParallelClassAndBlocks()

```
void TournamentPlanner.Model.Designs.BlockDesigns.KTS.AddParallelClassAndBlocks (
            List< List< int >> parallelClass )
```

Add a parallel class to the list of parallel classes. At the same time, it adds the blocks of the parallel class to the list of blocks.

Parameters

parallelClass	Parallel class to add.

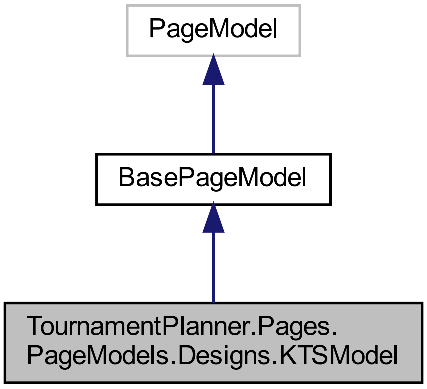
The documentation for this class was generated from the following file:

- KTS.cs

7.53 TournamentPlanner.Pages.PageModels.Designs.KTSModel Class Reference

Page model for the Kirkman triple system creation.

Inheritance diagram for TournamentPlanner.Pages.PageModels.Designs.KTSModel:



Public Member Functions

- void OnPost ()

Properties

- int Multiplier = 4 `[get, set]`

  Gets or sets the multiplier.
- int V = 27 `[get, set]`

  Gets or sets the V - the order of the KTS to construct. Should be a multiple equal to 6 $*$ N + 3.
- KTS KTS `[get]`

  Gets the constructed KTS.

7.53.1 Detailed Description

Page model for the Kirkman triple system creation.

7.53.2 Property Documentation

7.53.2.1 KTS

```
KTS TournamentPlanner.Pages.PageModels.Designs.KTSModel.KTS  [get]
```

Gets the constructed KTS.

7.53.2.2 Multiplier

```
int TournamentPlanner.Pages.PageModels.Designs.KTSModel.Multiplier = 4  [get], [set]
```

Gets or sets the multiplier.

7.53.2.3 V

```
int TournamentPlanner.Pages.PageModels.Designs.KTSModel.V = 27  [get], [set]
```

Gets or sets the V - the order of the KTS to construct. Should be a multiple equal to 6 ∗ N + 3.

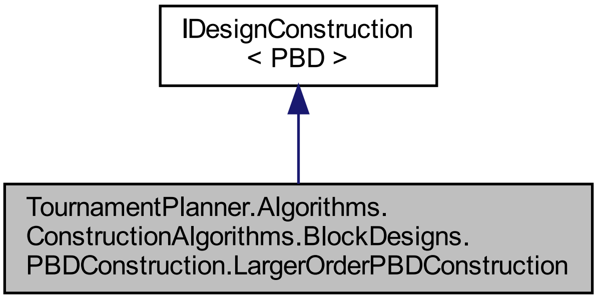
The documentation for this class was generated from the following file:

- KTS.cshtml.cs

7.54 TournamentPlanner.Algorithms.ConstructionAlgorithms.Block↩ Designs.PBDConstruction.LargerOrderPBDConstruction Class Reference

Construction of larger order PBDs, where the order is V = 3N + 1.

Inheritance diagram for TournamentPlanner.Algorithms.ConstructionAlgorithms.BlockDesigns.PBDConstruction.↩ LargerOrderPBDConstruction:


IDesignConstruction
< PBD >
TournamentPlanner.Algorithms.
ConstructionAlgorithms.BlockDesigns.
PBDConstruction.LargerOrderPBDConstruction

Public Member Functions

- LargerOrderPBDConstruction (int v)

  Initializes a new instance of the LargerOrderPBDConstruction class.
- PBD ConstructDesign ()

7.54.1 Detailed Description

Construction of larger order PBDs, where the order is V = 3N + 1.

7.54.2 Constructor & Destructor Documentation

7.54.2.1 LargerOrderPBDConstruction()

```
TournamentPlanner.Algorithms.ConstructionAlgorithms.BlockDesigns.PBDConstruction.LargerOrder↩
PBDConstruction.LargerOrderPBDConstruction (
            int v )
```

Initializes a new instance of the LargerOrderPBDConstruction class.

Parameters

v	The order of the PBD to construct.

The documentation for this class was generated from the following file:

- LargerOrderPBDConstruction.cs

7.55 TournamentPlannerTests.Algorithms.ConstructionAlgorithms.↩ BlockDesigns.PBDConstruction.LargerOrderPBDConstructionTests Class Reference

Tests for the LargerOrderPBDConstruction.

Public Member Functions

- void TestPBD49To76Construction ()

  Test the construction of PBD(3n+1) from order 49 up to order 76.
- void PBD178ConstructionTest ()

  Tests the construction of PBD(178).

7.55.1 Detailed Description

Tests for the LargerOrderPBDConstruction.

7.55.2 Member Function Documentation

7.55.2.1 PBD178ConstructionTest()

```
void TournamentPlannerTests.Algorithms.ConstructionAlgorithms.BlockDesigns.PBDConstruction.↩
LargerOrderPBDConstructionTests.PBD178ConstructionTest ( )
```

Tests the construction of PBD(178).

7.55.2.2 TestPBD49To76Construction()

```
void TournamentPlannerTests.Algorithms.ConstructionAlgorithms.BlockDesigns.PBDConstruction.↩
LargerOrderPBDConstructionTests.TestPBD49To76Construction ( )
```

Test the construction of PBD(3n+1) from order 49 up to order 76.

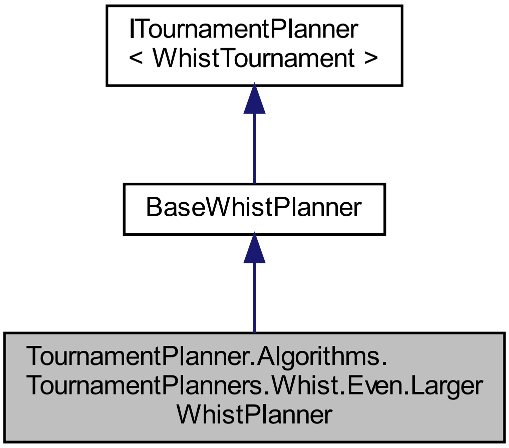
The documentation for this class was generated from the following file:

- LargerOrderPBDConstructionTests.cs

7.56 TournamentPlanner.Algorithms.TournamentPlanners.Whist.Even.↩ LargerWhistPlanner Class Reference

Construct Wh(4n), where n $>$ 80.

Inheritance diagram for TournamentPlanner.Algorithms.TournamentPlanners.Whist.Even.LargerWhistPlanner:


ITournamentPlanner
< WhistTournament >
BaseWhistPlanner
TournamentPlanner.Algorithms.
TournamentPlanners.Whist.Even.Larger
WhistPlanner

Public Member Functions

- LargerWhistPlanner (int v)
- override WhistTournament PlanTournament ()

  Plan the Whist tournament.

Additional Inherited Members

7.56.1 Detailed Description

Construct Wh(4n), where n $>$ 80.

The documentation for this class was generated from the following file:

- LargerWhistPlanner.cs

7.57 TournamentPlannerTests.Algorithms.TournamentPlanners.Whist.↩ Even.LargerWhistPlannerTests Class Reference

Tests for the LargerWhistPlanner.

Public Member Functions

- void **Whist324ConstructionTest** ()

7.57.1 Detailed Description

Tests for the LargerWhistPlanner.

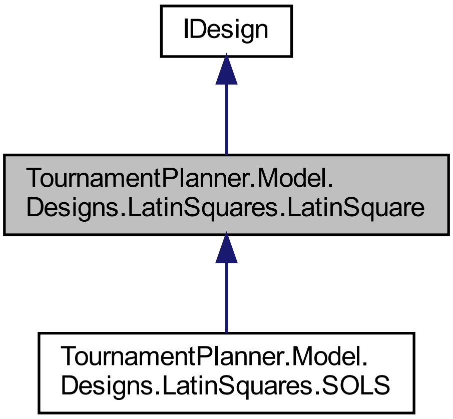
The documentation for this class was generated from the following file:

- LargerWhistPlannerTests.cs

7.58 TournamentPlanner.Model.Designs.LatinSquares.LatinSquare Class Reference

The Latin square.

Inheritance diagram for TournamentPlanner.Model.Designs.LatinSquares.LatinSquare:



Public Member Functions

- LatinSquare (int v, int[,] ls)

  Initializes a new instance of the LatinSquare class.

Properties

- int V `[get, protected set]`

  Gets or sets the order of the Latin square.
- int[,] Matrix `[get, protected set]`

  Gets or sets the Latin square matrix. It should be a V times V matrix. Only V distinct elements should appear in it.

7.58.1 Detailed Description

The Latin square.

7.58.2 Constructor & Destructor Documentation

7.58.2.1 LatinSquare()

```
TournamentPlanner.Model.Designs.LatinSquares.LatinSquare.LatinSquare (
            int v,
            int ls[,] )
```

Initializes a new instance of the LatinSquare class.

Parameters

v	The order od the LS.
ls	The LS matrix.

7.58.3 Property Documentation

7.58.3.1 Matrix

```
int [,] TournamentPlanner.Model.Designs.LatinSquares.LatinSquare.Matrix  [get], [protected set]
```

Gets or sets the Latin square matrix. It should be a V times V matrix. Only V distinct elements should appear in it.

7.58.3.2 V

```
int TournamentPlanner.Model.Designs.LatinSquares.LatinSquare.V  [get], [protected set]
```

Gets or sets the order of the Latin square.

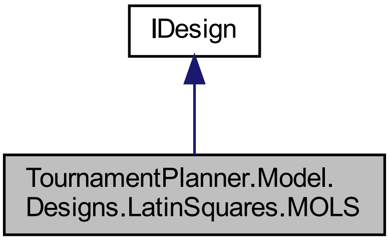
The documentation for this class was generated from the following file:

- LatinSquare.cs

7.59 TournamentPlanner.Model.Designs.LatinSquares.MOLS Class Reference

Set of mutually orthogonal Latin squares. For simplicity, it is allowed for the set of MOLS to contain only a single Latin square.

Inheritance diagram for TournamentPlanner.Model.Designs.LatinSquares.MOLS:



Public Member Functions

- MOLS (int order)

  Initializes a new instance of the MOLS class.
- MOLS (int order, LatinSquare lsA, LatinSquare lsB)

  Initializes a new instance of the MOLS class. Often, only 2 MOLS are given, not the whole set.
- MOLS (int order, List< LatinSquare > latinSquares)

  Initializes a new instance of the MOLS class.
- MOLS (int order, LatinSquare latinSquare)

  Initializes a new instance of the MOLS class.
- List< int[,]> GetMOLSMatrices ()

  Retrieve the matrices of the mutually orthogonal Latin squares.
- int GetNumberOfMOLS ()

  Retrieve the number of mutually orthogonal Latin squares.
- void AddOrthogonalLatinSquare (LatinSquare ls)

  Add a MOLS.

Properties

- int V `[get]`

  Gets the order of the mutually orthogonal Latin squares.
- List< LatinSquare > LatinSquares `[get]`

  Gets the list of mutually orthogonal Latin squares

7.59.1 Detailed Description

Set of mutually orthogonal Latin squares. For simplicity, it is allowed for the set of MOLS to contain only a single Latin square.

7.59.2 Constructor & Destructor Documentation

7.59.2.1 MOLS() [1/4]

```
TournamentPlanner.Model.Designs.LatinSquares.MOLS.MOLS (
              int order )
```

Initializes a new instance of the MOLS class.

Parameters

order	The order of MOLS.

7.59.2.2 MOLS() [2/4]

```
TournamentPlanner.Model.Designs.LatinSquares.MOLS.MOLS (
            int order,
            LatinSquare lsA,
            LatinSquare lsB )
```

Initializes a new instance of the MOLS class. Often, only 2 MOLS are given, not the whole set.

Parameters

order	The order of MOLS.
lsA	First Latin square.
lsB	Second Latin square.

7.59.2.3 MOLS() [3/4]

```
TournamentPlanner.Model.Designs.LatinSquares.MOLS.MOLS (
            int order,
            List< LatinSquare > latinSquares )
```

Initializes a new instance of the MOLS class.

Parameters

order	The order of MOLS.
latinSquares	A list of mutually orthogonal Latin squares.

7.59.2.4 MOLS() [4/4]

```
TournamentPlanner.Model.Designs.LatinSquares.MOLS.MOLS (
            int order,
            LatinSquare latinSquare )
```

Initializes a new instance of the MOLS class.

Parameters

order	The order of MOLS.
latinSquare	A single Latin square.

7.59.3 Member Function Documentation

7.59.3.1 AddOrthogonalLatinSquare()

```
void TournamentPlanner.Model.Designs.LatinSquares.MOLS.AddOrthogonalLatinSquare (
             LatinSquare ls )
```

Add a MOLS.

Parameters

ls	Latin square that should be orthogonal to the others in the list.

7.59.3.2 GetMOLSMatrices()

```
List<int[,]> TournamentPlanner.Model.Designs.LatinSquares.MOLS.GetMOLSMatrices ( )
```

Retrieve the matrices of the mutually orthogonal Latin squares.

Returns

List of MOLS matrices.

7.59.3.3 GetNumberOfMOLS()

```
int TournamentPlanner.Model.Designs.LatinSquares.MOLS.GetNumberOfMOLS ( )
```

Retrieve the number of mutually orthogonal Latin squares.

Returns

Number of MOLS.

7.59.4 Property Documentation

7.59.4.1 LatinSquares

```
List<LatinSquare> TournamentPlanner.Model.Designs.LatinSquares.MOLS.LatinSquares  [get]
```

Gets the list of mutually orthogonal Latin squares

7.59.4.2 V

```
int TournamentPlanner.Model.Designs.LatinSquares.MOLS.V  [get]
```

Gets the order of the mutually orthogonal Latin squares.

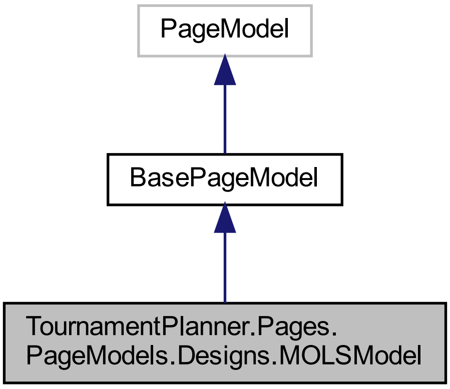
The documentation for this class was generated from the following file:

- MOLS.cs

7.60 TournamentPlanner.Pages.PageModels.Designs.MOLSModel Class Reference

Page model for mutually orthogonal Latin squares.

Inheritance diagram for TournamentPlanner.Pages.PageModels.Designs.MOLSModel:



Public Member Functions

- void OnPost ()

Properties

- MOLS MOLS [get]

  Gets the constructed MOLS.
- OrthogonalArray OA [get]

  Gets the orthogonal array equivalent to the constructed MOLS.
- int Order = 4 [get, set]

  Gets or sets the order of the Latin square.
- MOLSMethod MOLSConstructionMethod = MOLSMethod.Automatic [get, set]

  Gets or sets the construction method for the Latin square.

7.60.1 Detailed Description

Page model for mutually orthogonal Latin squares.

7.60.2 Property Documentation

7.60.2.1 MOLS

```
MOLS TournamentPlanner.Pages.PageModels.Designs.MOLSModel.MOLS  [get]
```

Gets the constructed MOLS.

7.60.2.2 MOLSConstructionMethod

```
MOLSMethod TournamentPlanner.Pages.PageModels.Designs.MOLSModel.MOLSConstructionMethod = MOLSMethod.Automatic
[get], [set]
```

Gets or sets the construction method for the Latin square.

7.60.2.3 OA

```
OrthogonalArray TournamentPlanner.Pages.PageModels.Designs.MOLSModel.OA  [get]
```

Gets the orthogonal array equivalent to the constructed MOLS.

7.60.2.4 Order

```
int TournamentPlanner.Pages.PageModels.Designs.MOLSModel.Order = 4  [get], [set]
```

Gets or sets the order of the Latin square.

The documentation for this class was generated from the following file:

- MOLS.cshtml.cs

7.61 TournamentPlannerTests.Designs.MOLSSetValidatorTests Class Reference

Tests for the MOLSSetValidator.

Public Member Functions

- void ValidMOLSTest ()

  Test where the MOLS are valid.
- void InvalidMOLSTest ()

  Test where some of the MOLS are not valid.

7.61.1 Detailed Description

Tests for the MOLSSetValidator.

7.61.2 Member Function Documentation

7.61.2.1 InvalidMOLSTest()

```
void TournamentPlannerTests.Designs.MOLSSetValidatorTests.InvalidMOLSTest ( )
```

Test where some of the MOLS are not valid.

7.61.2.2 ValidMOLSTest()

```
void TournamentPlannerTests.Designs.MOLSSetValidatorTests.ValidMOLSTest ( )
```

Test where the MOLS are valid.

The documentation for this class was generated from the following file:

- MOLSSetValidatorTests.cs

7.62 TournamentPlanner.Datastore.MongoMOLS Class Reference

Class representing a MOLS in mongo DB.

Properties

- int Order `[get, set]`

  Gets or sets the order of MOLS.
- int NbMOLS `[get, set]`

  Gets or sets the number of MOLS.
- List< int[,]> Matrices `[get, set]`

  Gets or sets the Latin square matrices of the MOLS.

7.62.1 Detailed Description

Class representing a MOLS in mongo DB.

7.62.2 Property Documentation

7.62.2.1 Matrices

```
List<int[,]> TournamentPlanner.Datastore.MongoMOLS.Matrices  [get], [set]
```

Gets or sets the Latin square matrices of the MOLS.

7.62.2.2 NbMOLS

```
int TournamentPlanner.Datastore.MongoMOLS.NbMOLS  [get], [set]
```

Gets or sets the number of MOLS.

7.62.2.3 Order

```
int TournamentPlanner.Datastore.MongoMOLS.Order  [get], [set]
```

Gets or sets the order of MOLS.

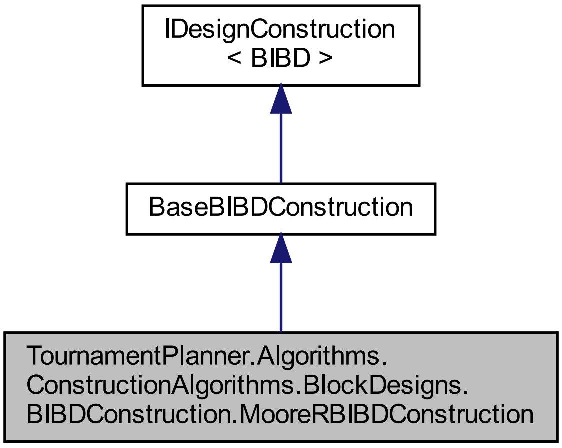
The documentation for this class was generated from the following file:

- MongoMOLS.cs

7.63 TournamentPlanner.Algorithms.ConstructionAlgorithms.Block↩ Designs.BIBDConstruction.MooreRBIBDConstruction Class Reference

Construction of RBIBD(3q+1, 4, 1) where q is a prime power and can be written as q = 4m + 1. These RBIBD are used to construct Whist tournaments.

Inheritance diagram for TournamentPlanner.Algorithms.ConstructionAlgorithms.BlockDesigns.BIBDConstruction.↩ MooreRBIBDConstruction:


IDesignConstruction
< BIBD >
BaseBIBDConstruction
TournamentPlanner.Algorithms.
ConstructionAlgorithms.BlockDesigns.
BIBDConstruction.MooreRBIBDConstruction

Public Member Functions

- MooreRBIBDConstruction (int v)

  Initializes a new instance of the MooreRBIBDConstruction class.
- override RBIBD ConstructDesign ()

  Construct a resolvable BIBD.

Additional Inherited Members

7.63.1 Detailed Description

Construction of RBIBD(3q+1, 4, 1) where q is a prime power and can be written as q = 4m + 1. These RBIBD are used to construct Whist tournaments.

7.63.2 Constructor & Destructor Documentation

7.63.2.1 MooreRBIBDConstruction()

```
TournamentPlanner.Algorithms.ConstructionAlgorithms.BlockDesigns.BIBDConstruction.Moore←↩
RBIBDConstruction.MooreRBIBDConstruction (
            int v )
```

Initializes a new instance of the MooreRBIBDConstruction class.

Parameters

v	Order of the RBIBD to construct.

7.63.3 Member Function Documentation

7.63.3.1 ConstructDesign()

```
override RBIBD TournamentPlanner.Algorithms.ConstructionAlgorithms.BlockDesigns.BIBDConstruction.←↩
MooreRBIBDConstruction.ConstructDesign ( )  [virtual]
```

Construct a resolvable BIBD.

Returns

Constructed RBIBD.

Implements TournamentPlanner.Algorithms.ConstructionAlgorithms.BlockDesigns.BIBDConstruction.BaseBIBDConstruction.

The documentation for this class was generated from the following file:

- MooreRBIBDConstruction.cs

7.64 TournamentPlannerTests.Algorithms.ConstructionAlgorithms.←↩ BlockDesigns.BIBDConstruction.MooreRBIBDConstructionTests Class Reference

Tests for the MooreRBIBDConstruction.

Public Member Functions

- void ConstructionTestForSeveralOrdersOfRBIBD ()
  Test the PBIBD construction for all the important orders.

7.64.1 Detailed Description

Tests for the MooreRBIBDConstruction.

7.64.2 Member Function Documentation

7.64.2.1 ConstructionTestForSeveralOrdersOfRBIBD()

```
void TournamentPlannerTests.Algorithms.ConstructionAlgorithms.BlockDesigns.BIBDConstruction.↩
MooreRBIBDConstructionTests.ConstructionTestForSeveralOrdersOfRBIBD ( )
```

Test the PBIBD construction for all the important orders.

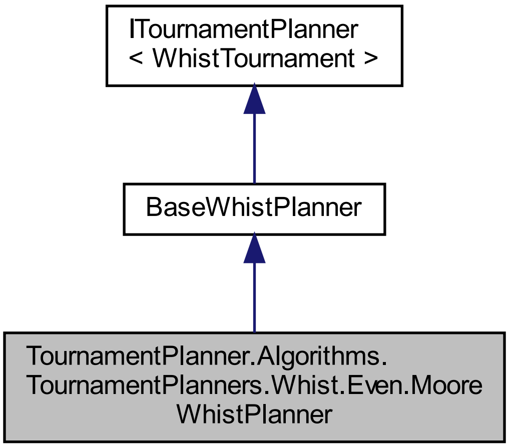
The documentation for this class was generated from the following file:

- MooreRBIBDConstructionTests.cs

7.65 TournamentPlanner.Algorithms.TournamentPlanners.Whist.Even.↩ MooreWhistPlanner Class Reference

This class implements the construction of Wh(v = 4n), where v = 3q + 1 and q = 1 (mod 4) = 4m + 1 and is a prime power. v = 28, 40, 52, 76, 88 ... https://www.jstor.org/stable/2369860?seq=1::metadata_info_tab_contents.

Inheritance diagram for TournamentPlanner.Algorithms.TournamentPlanners.Whist.Even.MooreWhistPlanner:



Public Member Functions

- MooreWhistPlanner (int v)

  Initializes a new instance of the MooreWhistPlanner class.
- override WhistTournament PlanTournament ()

  Plan the Whist tournament.

Additional Inherited Members

7.65.1 Detailed Description

This class implements the construction of Wh(v = 4n), where v = 3q + 1 and q = 1 (mod 4) = 4m + 1 and is a prime power. v = 28, 40, 52, 76, 88 ... https://www.jstor.org/stable/2369860?seq=1::metadata_info_tab_contents.

7.65.2 Constructor & Destructor Documentation

7.65.2.1 MooreWhistPlanner()

```
TournamentPlanner.Algorithms.TournamentPlanners.Whist.Even.MooreWhistPlanner.MooreWhistPlanner
(
             int v )
```

Initializes a new instance of the MooreWhistPlanner class.

Parameters

v	Number of players.

The documentation for this class was generated from the following file:

- MooreWhistPlanner.cs

7.66 TournamentPlannerTests.Algorithms.TournamentPlanners.Whist.↩ Even.MooreWhistPlannerTests Class Reference

Tests for the MooreWhistPlanner.

Public Member Functions

- void TestValidOrdersOfMooreConstruction ()

  Test with the valid orders of whist tournaments for the Moore construction.

7.66.1 Detailed Description

Tests for the MooreWhistPlanner.

7.66.2 Member Function Documentation

7.66.2.1 TestValidOrdersOfMooreConstruction()

```
void TournamentPlannerTests.Algorithms.TournamentPlanners.Whist.Even.MooreWhistPlannerTests.↩
TestValidOrdersOfMooreConstruction ( )
```

Test with the valid orders of whist tournaments for the Moore construction.

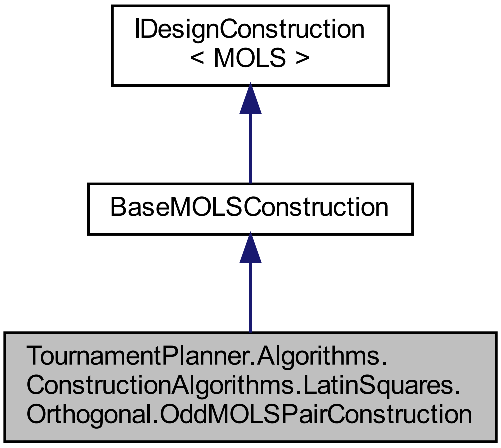
The documentation for this class was generated from the following file:

- MooreWhistPlannerTests.cs

7.67 TournamentPlanner.Algorithms.ConstructionAlgorithms.Latin↩ Squares.Orthogonal.OddMOLSPairConstruction Class Reference

Algorithm to construct pairs of mutually orthogonal Latin squares of odd order. https://www.whitman.↩
edu/mathematics/cgt_online/book/section04.03.html

Inheritance diagram for TournamentPlanner.Algorithms.ConstructionAlgorithms.LatinSquares.Orthogonal.Odd↩
MOLSPairConstruction:


IDesignConstruction
< MOLS >
BaseMOLSConstruction
TournamentPlanner.Algorithms.
ConstructionAlgorithms.LatinSquares.
Orthogonal.OddMOLSPairConstruction

Public Member Functions

- OddMOLSPairConstruction (int order)
- override MOLS ConstructDesign ()

  Construct MOLS.

Protected Member Functions

- override bool ValidateOrder (int order)

  Method to validate the order in the constructor.

Properties

- override string ArgumentExceptionMessage `[get]`

7.67.1 Detailed Description

Algorithm to construct pairs of mutually orthogonal Latin squares of odd order. https://www.whitman.↩ edu/mathematics/cgt_online/book/section04.03.html

The documentation for this class was generated from the following file:

- OddMOLSPairConstruction.cs

7.68 TournamentPlannerTests.Algorithms.ConstructionAlgorithms.↩ LatinSquares.Orthogonal.OddMOLSPairConstructionTests Class Reference

Tests for the OddMOLSPairConstruction.

Public Member Functions

- void EvenOrderShouldThrowExceptionTest ()

  Test that an exception is thrown when the order is even.
- void OrthogonalityTest ()

  Test the orthogonality of a few pairs of MOLS of odd orders.

7.68.1 Detailed Description

Tests for the OddMOLSPairConstruction.

7.68.2 Member Function Documentation

7.68.2.1 EvenOrderShouldThrowExceptionTest()

```
void TournamentPlannerTests.Algorithms.ConstructionAlgorithms.LatinSquares.Orthogonal.Odd↵
MOLSPairConstructionTests.EvenOrderShouldThrowExceptionTest ( )
```

Test that an exception is thrown when the order is even.

7.68.2.2 OrthogonalityTest()

```
void TournamentPlannerTests.Algorithms.ConstructionAlgorithms.LatinSquares.Orthogonal.Odd↵
MOLSPairConstructionTests.OrthogonalityTest ( )
```

Test the orthogonality of a few pairs of MOLS of odd orders.

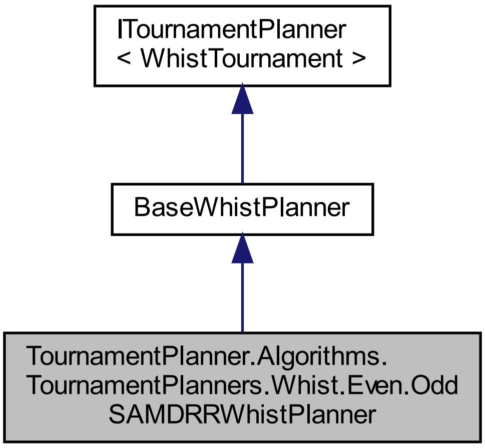
The documentation for this class was generated from the following file:

- OddMOLSPairConstructionTests.cs

7.69 TournamentPlanner.Algorithms.TournamentPlanners.Whist.Even.↵ OddSAMDRRWhistPlanner Class Reference

Use odd order SAMMDR to construct several orders od Whist(4n).

Inheritance diagram for TournamentPlanner.Algorithms.TournamentPlanners.Whist.Even.OddSAMDRRWhist↵ Planner:



Public Member Functions

- OddSAMDRRWhistPlanner (int v)

  Initializes a new instance of the OddSAMDRRWhistPlanner class.
- override WhistTournament PlanTournament ()

  Plan the Whist tournament.

Additional Inherited Members

7.69.1 Detailed Description

Use odd order SAMMDR to construct several orders od Whist(4n).

7.69.2 Constructor & Destructor Documentation

7.69.2.1 OddSAMDRRWhistPlanner()

```
TournamentPlanner.Algorithms.TournamentPlanners.Whist.Even.OddSAMDRRWhistPlanner.OddSAMDRRWhist↩
Planner (
             int v )
```

Initializes a new instance of the OddSAMDRRWhistPlanner class.

Parameters

v	The order.

The documentation for this class was generated from the following file:

- OddSAMDRRWhistPlanner.cs

7.70 TournamentPlannerTests.Algorithms.TournamentPlanners.Whist.↩ Even.OddSAMDRRWhistPlannerTests Class Reference

Tests for the SAMDRRWhistPlanner.

Public Member Functions

- void TestWhistConstructionUsingSAMDRR ()

  Test the construction of Whist(44).

7.70.1 Detailed Description

Tests for the SAMDRRWhistPlanner.

7.70.2 Member Function Documentation

7.70.2.1 TestWhistConstructionUsingSAMDRR()

```
void TournamentPlannerTests.Algorithms.TournamentPlanners.Whist.Even.OddSAMDRRWhistPlanner↩
Tests.TestWhistConstructionUsingSAMDRR ( )
```

Test the construction of Whist(44).

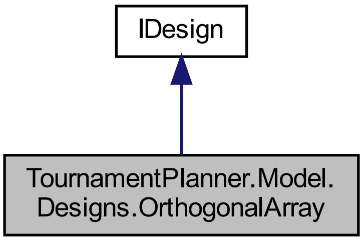
The documentation for this class was generated from the following file:

- OddSAMDRRWhistPlannerTests.cs

7.71 TournamentPlanner.Model.Designs.OrthogonalArray Class Reference

Class representing the Orthogonal array structure. In some literature, this representation is actually the transpose of an orthogonal array.

Inheritance diagram for TournamentPlanner.Model.Designs.OrthogonalArray:



Public Member Functions

- OrthogonalArray (int v, int[,] matrix)

    Initializes a new instance of the OrthogonalArray class.
- int Width ()

    Gets the width of the orthogonal array.
- int Height ()

    Gets the height of the orthogonal array.
- int RepresentedMOLSCount ()

    Get the number of MOLS this orthogonal array might represent.

Properties

- int V [get]

  Gets the level of the orthogonal array.
- int[,] Matrix [get]

  Gets the matrix representation of the OA.

7.71.1 Detailed Description

Class representing the Orthogonal array structure. In some literature, this representation is actually the transpose of an orthogonal array.

7.71.2 Constructor & Destructor Documentation

7.71.2.1 OrthogonalArray()

```
TournamentPlanner.Model.Designs.OrthogonalArray.OrthogonalArray (
             int v,
             int matrix[,] )
```

Initializes a new instance of the OrthogonalArray class.

Parameters

v	The level of the orthogonal array.
matrix	The matrix representing the orthogonal array.

7.71.3 Member Function Documentation

7.71.3.1 Height()

```
int TournamentPlanner.Model.Designs.OrthogonalArray.Height ( )
```

Gets the height of the orthogonal array.

Returns

Height of the OA matrix.

7.71.3.2 RepresentedMOLSCount()

```
int TournamentPlanner.Model.Designs.OrthogonalArray.RepresentedMOLSCount ( )
```

Get the number of MOLS this orthogonal array might represent.

Returns

Number of MOLS.

7.71.3.3 Width()

```
int TournamentPlanner.Model.Designs.OrthogonalArray.Width ( )
```

Gets the width of the orthogonal array.

Returns

Width of the OA matrix.

7.71.4 Property Documentation

7.71.4.1 Matrix

```
int [,] TournamentPlanner.Model.Designs.OrthogonalArray.Matrix  [get]
```

Gets the matrix representation of the OA.

7.71.4.2 V

```
int TournamentPlanner.Model.Designs.OrthogonalArray.V  [get]
```

Gets the level of the orthogonal array.

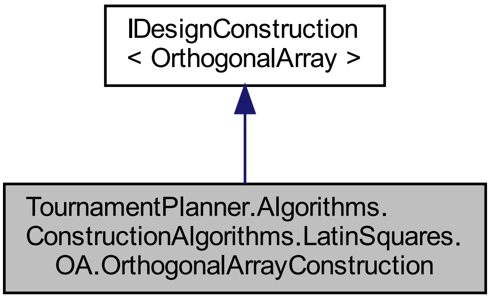
The documentation for this class was generated from the following file:

- OrthogonalArray.cs

7.72 TournamentPlanner.Algorithms.ConstructionAlgorithms.Latin↩ Squares.OA.OrthogonalArrayConstruction Class Reference

Construction method for OAs using MOLS.

Inheritance diagram for TournamentPlanner.Algorithms.ConstructionAlgorithms.LatinSquares.OA.Orthogonal↩ ArrayConstruction:


IDesignConstruction < OrthogonalArray >
TournamentPlanner.Algorithms. ConstructionAlgorithms.LatinSquares. OA.OrthogonalArrayConstruction

Public Member Functions

- OrthogonalArrayConstruction (int k, int m)

    Initializes a new instance of the OrthogonalArrayConstruction class.
- OrthogonalArray ConstructDesign ()

    Construct the orthogonal array.

7.72.1 Detailed Description

Construction method for OAs using MOLS.

7.72.2 Constructor & Destructor Documentation

7.72.2.1 OrthogonalArrayConstruction()

```
TournamentPlanner.Algorithms.ConstructionAlgorithms.LatinSquares.OA.OrthogonalArrayConstruction.↩
OrthogonalArrayConstruction (
             int k,
             int m )
```

Initializes a new instance of the OrthogonalArrayConstruction class.

Parameters

k	The factor.
m	The level.

7.72.3 Member Function Documentation

7.72.3.1 ConstructDesign()

```
OrthogonalArray TournamentPlanner.Algorithms.ConstructionAlgorithms.LatinSquares.OA.Orthogonal←
ArrayConstruction.ConstructDesign ( )
```

Construct the orthogonal array.

Returns

Constructed orthogonal array.

The documentation for this class was generated from the following file:

- OrthogonalArrayConstruction.cs

7.73 TournamentPlannerTests.Utils.OrthogonalArrayMOLSConverter← Tests Class Reference

Tests for the OrthogonalArrayMOLSConverter.

Public Member Functions

- void MOLS2OATest ()

  Test the conversion of MOLS to OA.
- void OA2MOLSTest ()

  Test the conversion of an OA into MOLS.

7.73.1 Detailed Description

Tests for the OrthogonalArrayMOLSConverter.

7.73.2 Member Function Documentation

7.73.2.1 MOLS2OATest()

```
void TournamentPlannerTests.Utils.OrthogonalArrayMOLSConverterTests.MOLS2OATest ( )
```

Test the conversion of MOLS to OA.

7.73.2.2 OA2MOLSTest()

```
void TournamentPlannerTests.Utils.OrthogonalArrayMOLSConverterTests.OA2MOLSTest ( )
```

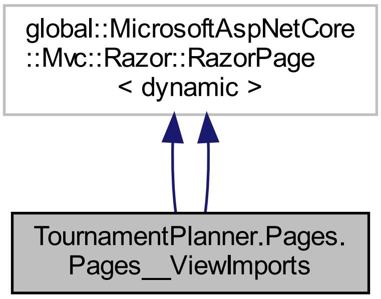
Test the conversion of an OA into MOLS.

The documentation for this class was generated from the following file:

- OrthogonalArrayMOLSConverterTests.cs

7.74 TournamentPlanner.Pages.Pages__ViewImports Class Reference

Inheritance diagram for TournamentPlanner.Pages.Pages__ViewImports:



Public Member Functions

- async override global::System.Threading.Tasks.Task **ExecuteAsync** ()
- async override global::System.Threading.Tasks.Task **ExecuteAsync** ()

Properties

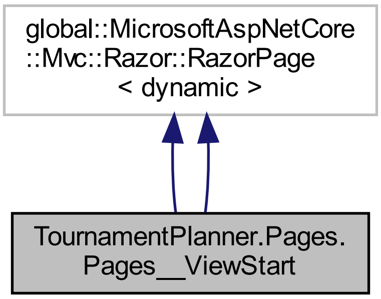
- global::Microsoft.AspNetCore.Mvc.ViewFeatures.IModelExpressionProvider **ModelExpressionProvider** `[get]`
- global::Microsoft.AspNetCore.Mvc.IUrlHelper **Url** `[get]`
- global::Microsoft.AspNetCore.Mvc.IViewComponentHelper **Component** `[get]`
- global::Microsoft.AspNetCore.Mvc.Rendering.IJsonHelper **Json** `[get]`
- global::Microsoft.AspNetCore.Mvc.Rendering.IHtmlHelper< dynamic > **Html** `[get]`

The documentation for this class was generated from the following file:

- Debug/net5.0/Razor/Pages/_ViewImports.cshtml.g.cs

7.75 TournamentPlanner.Pages.Pages__ViewStart Class Reference

Inheritance diagram for TournamentPlanner.Pages.Pages__ViewStart:



Public Member Functions

- async override global::System.Threading.Tasks.Task **ExecuteAsync** ()
- async override global::System.Threading.Tasks.Task **ExecuteAsync** ()

Properties

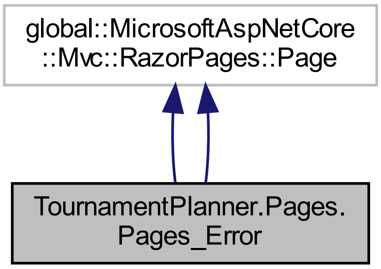
- global::Microsoft.AspNetCore.Mvc.ViewFeatures.IModelExpressionProvider **ModelExpressionProvider** `[get]`
- global::Microsoft.AspNetCore.Mvc.IUrlHelper **Url** `[get]`
- global::Microsoft.AspNetCore.Mvc.IViewComponentHelper **Component** `[get]`
- global::Microsoft.AspNetCore.Mvc.Rendering.IJsonHelper **Json** `[get]`
- global::Microsoft.AspNetCore.Mvc.Rendering.IHtmlHelper< dynamic > **Html** `[get]`

The documentation for this class was generated from the following file:

- Debug/net5.0/Razor/Pages/_ViewStart.cshtml.g.cs

7.76 TournamentPlanner.Pages.Pages_Error Class Reference

Inheritance diagram for TournamentPlanner.Pages.Pages_Error:



Public Member Functions

- async override global::System.Threading.Tasks.Task **ExecuteAsync** ()
- async override global::System.Threading.Tasks.Task **ExecuteAsync** ()

Properties

- global::Microsoft.AspNetCore.Mvc.ViewFeatures.IModelExpressionProvider **ModelExpressionProvider** [get]
- global::Microsoft.AspNetCore.Mvc.IUrlHelper **Url** [get]
- global::Microsoft.AspNetCore.Mvc.IViewComponentHelper **Component** [get]
- global::Microsoft.AspNetCore.Mvc.Rendering.IJsonHelper **Json** [get]
- global::Microsoft.AspNetCore.Mvc.Rendering.IHtmlHelper< ErrorModel > **Html** [get]
- global::Microsoft.AspNetCore.Mvc.ViewFeatures.ViewDataDictionary< ErrorModel > **ViewData** [get]
- ErrorModel **Model** [get]

7.76.1 Property Documentation

7.76.1.1 ModelExpressionProvider

```
Microsoft AspNetCore Mvc ViewFeatures global.IModelExpressionProvider TournamentPlanner.↩
Pages.Pages_Error.ModelExpressionProvider  [get]
```

Initial value:

```
<h3>Development Mode</h3>
<p>
    Swapping to the <strong>Development</strong> environment displays detailed information about the error
       that occurred.
</p>
<p>
```

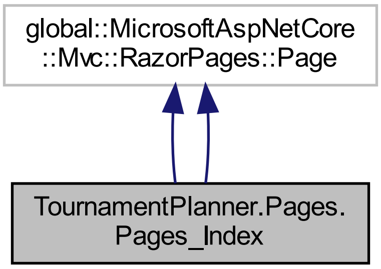
```
    <strong>The Development environment shouldn’t be enabled for deployed applications.</strong>
    It can result in displaying sensitive information from exceptions to end users.
    For local debugging, enable the <strong>Development</strong> environment by setting the
       <strong>ASPNETCORE_ENVIRONMENT</strong> environment variable to <strong>Development</strong>
    and restarting the app.
</p>
"
```

The documentation for this class was generated from the following file:

- Debug/net5.0/Razor/Pages/Error.cshtml.g.cs

7.77 TournamentPlanner.Pages.Pages_Index Class Reference

Inheritance diagram for TournamentPlanner.Pages.Pages_Index:



Public Member Functions

- async override global::System.Threading.Tasks.Task **ExecuteAsync** ()
- async override global::System.Threading.Tasks.Task **ExecuteAsync** ()

Properties

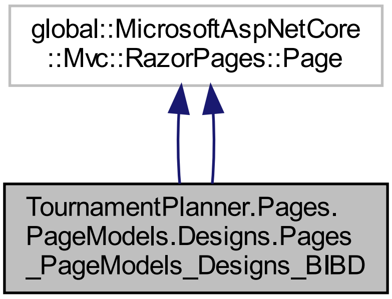
- global::Microsoft.AspNetCore.Mvc.ViewFeatures.IModelExpressionProvider **ModelExpressionProvider** [get]
- global::Microsoft.AspNetCore.Mvc.IUrlHelper **Url** [get]
- global::Microsoft.AspNetCore.Mvc.IViewComponentHelper **Component** [get]
- global::Microsoft.AspNetCore.Mvc.Rendering.IJsonHelper **Json** [get]
- global::Microsoft.AspNetCore.Mvc.Rendering.IHtmlHelper< IndexModel > **Html** [get]
- global::Microsoft.AspNetCore.Mvc.ViewFeatures.ViewDataDictionary< IndexModel > **ViewData** [get]
- IndexModel **Model** [get]

The documentation for this class was generated from the following file:

- Debug/net5.0/Razor/Pages/Index.cshtml.g.cs

7.78 TournamentPlanner.Pages.PageModels.Designs.Pages_Page↩ Models_Designs_BIBD Class Reference

Inheritance diagram for TournamentPlanner.Pages.PageModels.Designs.Pages_PageModels_Designs_BIBD:


global::MicrosoftAspNetCore ::Mvc::RazorPages::Page
TournamentPlanner.Pages. PageModels.Designs.Pages _PageModels_Designs_BIBD

Public Member Functions

- async override global::System.Threading.Tasks.Task **ExecuteAsync** ()
- async override global::System.Threading.Tasks.Task **ExecuteAsync** ()

Properties

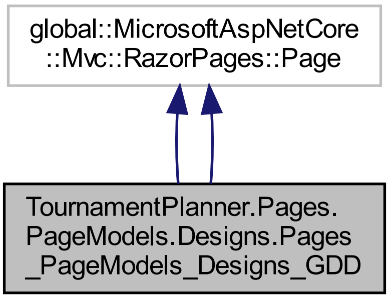
- global::Microsoft.AspNetCore.Mvc.ViewFeatures.IModelExpressionProvider **ModelExpressionProvider** `[get]`
- global::Microsoft.AspNetCore.Mvc.IUrlHelper **Url** `[get]`
- global::Microsoft.AspNetCore.Mvc.IViewComponentHelper **Component** `[get]`
- global::Microsoft.AspNetCore.Mvc.Rendering.IJsonHelper **Json** `[get]`
- global::Microsoft.AspNetCore.Mvc.Rendering.IHtmlHelper< TournamentPlanner.Pages.PageModels.Designs.BIBDModel > **Html** `[get]`
- global::Microsoft.AspNetCore.Mvc.ViewFeatures.ViewDataDictionary< TournamentPlanner.Pages.PageModels.Designs.BIBD > **ViewData** `[get]`
- TournamentPlanner.Pages.PageModels.Designs.BIBDModel **Model** `[get]`

The documentation for this class was generated from the following file:

- Debug/net5.0/Razor/Pages/PageModels/Designs/BIBD.cshtml.g.cs

7.79 TournamentPlanner.Pages.PageModels.Designs.Pages_Page↩ Models_Designs_GDD Class Reference

Inheritance diagram for TournamentPlanner.Pages.PageModels.Designs.Pages_PageModels_Designs_GDD:


global::MicrosoftAspNetCore ::Mvc::RazorPages::Page
TournamentPlanner.Pages. PageModels.Designs.Pages _PageModels_Designs_GDD

Public Member Functions

- async override global::System.Threading.Tasks.Task **ExecuteAsync** ()
- async override global::System.Threading.Tasks.Task **ExecuteAsync** ()

Properties

- global::Microsoft.AspNetCore.Mvc.ViewFeatures.IModelExpressionProvider **ModelExpressionProvider** [get]
- global::Microsoft.AspNetCore.Mvc.IUrlHelper **Url** [get]
- global::Microsoft.AspNetCore.Mvc.IViewComponentHelper **Component** [get]
- global::Microsoft.AspNetCore.Mvc.Rendering.IJsonHelper **Json** [get]
- global::Microsoft.AspNetCore.Mvc.Rendering.IHtmlHelper< TournamentPlanner.Pages.PageModels.Designs.GDDModel > **Html** [get]
- global::Microsoft.AspNetCore.Mvc.ViewFeatures.ViewDataDictionary< TournamentPlanner.Pages.PageModels.Designs.GDD > **ViewData** [get]
- TournamentPlanner.Pages.PageModels.Designs.GDDModel **Model** [get]

7.79.1 Property Documentation

7.79.1.1 ModelExpressionProvider

```
Microsoft AspNetCore Mvc ViewFeatures global.IModelExpressionProvider TournamentPlanner.↩
Pages.PageModels.Designs.Pages_PageModels_Designs_GDD.ModelExpressionProvider  [get]
```

Initial value:

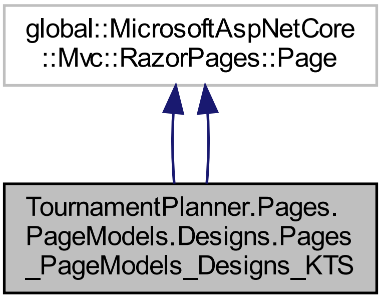
```
<div class=""content-section"">
    <div class=""content-section-header"">
        <h4>Group Divisible Designs</h4>
    </div>
    <div class=""content-section-body"">
        A <i>Group divisible design</i> or <b>GDD</b> of order \(v\) consists of a \(v\)-sized set \(S\)
   of elements,
        a collection of subsets \(G\) called <b>groups</b> and a collection of blocks \(B\) such that
        <ul>
            <li>the groups form a partition of \(S\);</li>
            <li>each pair of elements from the same group do not occur together in any of the
   blocks;</li>
            <li>each pair of elements from different groups occur together in exactly one of the
   blocks.</li>
        </ul>
        Resolvabe GDDs can be constructed using MOLS. This construction is possible for any \(v = 3m\)
   except \(m = 2, 6\). They are
        then used to schedule patrially balanced tournaments with three player games.
    </div>
</div>
<div class=""c"    <div class=""content-section"">
    <div class=""content-section-header"">
        <h4>Constructed resolvable GDD with block size 3</h4>
    </div>
    <div class=""content-section-body"">
        <table class=""result-table"">
            <tr>
                <td>V</td>
                <td>"
```

The documentation for this class was generated from the following file:

- Debug/net5.0/Razor/Pages/PageModels/Designs/GDD.cshtml.g.cs

7.80 TournamentPlanner.Pages.PageModels.Designs.Pages_Page↩ Models_Designs_KTS Class Reference

Inheritance diagram for TournamentPlanner.Pages.PageModels.Designs.Pages_PageModels_Designs_KTS:



Public Member Functions

- async override global::System.Threading.Tasks.Task **ExecuteAsync** ()
- async override global::System.Threading.Tasks.Task **ExecuteAsync** ()

Properties

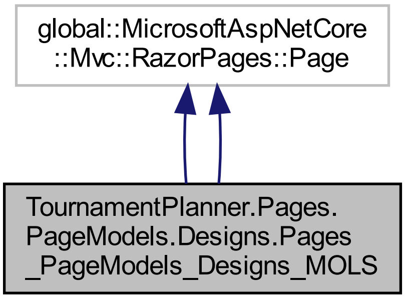
- global::Microsoft.AspNetCore.Mvc.ViewFeatures.IModelExpressionProvider **ModelExpressionProvider** [get]
- global::Microsoft.AspNetCore.Mvc.IUrlHelper **Url** [get]
- global::Microsoft.AspNetCore.Mvc.IViewComponentHelper **Component** [get]
- global::Microsoft.AspNetCore.Mvc.Rendering.IJsonHelper **Json** [get]
- global::Microsoft.AspNetCore.Mvc.Rendering.IHtmlHelper< TournamentPlanner.Pages.PageModels.Designs.KTSModel > **Html** [get]
- global::Microsoft.AspNetCore.Mvc.ViewFeatures.ViewDataDictionary< TournamentPlanner.Pages.PageModels.Designs.KTSM > **ViewData** [get]
- TournamentPlanner.Pages.PageModels.Designs.KTSModel **Model** [get]

The documentation for this class was generated from the following file:

- Debug/net5.0/Razor/Pages/PageModels/Designs/KTS.cshtml.g.cs

7.81 TournamentPlanner.Pages.PageModels.Designs.Pages_PageModels_Designs_MOLS Class Reference

Inheritance diagram for TournamentPlanner.Pages.PageModels.Designs.Pages_PageModels_Designs_MOLS:



Public Member Functions

- async override global::System.Threading.Tasks.Task **ExecuteAsync** ()
- async override global::System.Threading.Tasks.Task **ExecuteAsync** ()

Properties

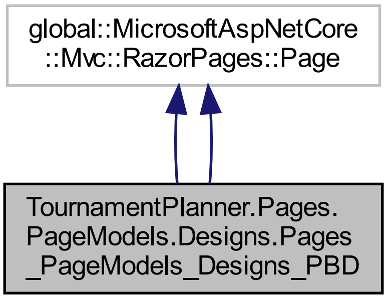
- global::Microsoft.AspNetCore.Mvc.ViewFeatures.IModelExpressionProvider **ModelExpressionProvider** [get]
- global::Microsoft.AspNetCore.Mvc.IUrlHelper **Url** [get]
- global::Microsoft.AspNetCore.Mvc.IViewComponentHelper **Component** [get]
- global::Microsoft.AspNetCore.Mvc.Rendering.IJsonHelper **Json** [get]
- global::Microsoft.AspNetCore.Mvc.Rendering.IHtmlHelper< TournamentPlanner.Pages.PageModels.Designs.MOLSModel > **Html** [get]
- global::Microsoft.AspNetCore.Mvc.ViewFeatures.ViewDataDictionary< TournamentPlanner.Pages.PageModels.Designs.MOLS > **ViewData** [get]
- TournamentPlanner.Pages.PageModels.Designs.MOLSModel **Model** [get]

The documentation for this class was generated from the following file:

- Debug/net5.0/Razor/Pages/PageModels/Designs/MOLS.cshtml.g.cs

7.82 TournamentPlanner.Pages.PageModels.Designs.Pages_PageModels_Designs_PBD Class Reference

Inheritance diagram for TournamentPlanner.Pages.PageModels.Designs.Pages_PageModels_Designs_PBD:


global::MicrosoftAspNetCore ::Mvc::RazorPages::Page
TournamentPlanner.Pages. PageModels.Designs.Pages _PageModels_Designs_PBD

Public Member Functions

- async override global::System.Threading.Tasks.Task **ExecuteAsync** ()
- async override global::System.Threading.Tasks.Task **ExecuteAsync** ()

Properties

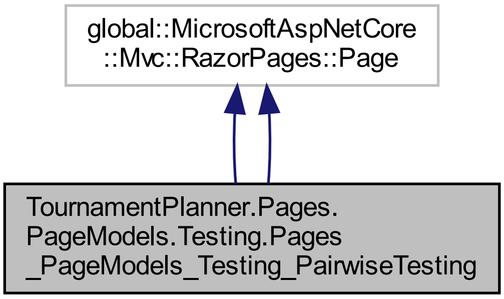
- global::Microsoft.AspNetCore.Mvc.ViewFeatures.IModelExpressionProvider **ModelExpressionProvider** [get]
- global::Microsoft.AspNetCore.Mvc.IUrlHelper **Url** [get]
- global::Microsoft.AspNetCore.Mvc.IViewComponentHelper **Component** [get]
- global::Microsoft.AspNetCore.Mvc.Rendering.IJsonHelper **Json** [get]
- global::Microsoft.AspNetCore.Mvc.Rendering.IHtmlHelper< TournamentPlanner.Pages.PageModels.Designs.PBDModel > **Html** [get]
- global::Microsoft.AspNetCore.Mvc.ViewFeatures.ViewDataDictionary< TournamentPlanner.Pages.PageModels.Designs.PBDM > **ViewData** [get]
- TournamentPlanner.Pages.PageModels.Designs.PBDModel **Model** [get]

The documentation for this class was generated from the following file:

- Debug/net5.0/Razor/Pages/PageModels/Designs/PBD.cshtml.g.cs

7.83 TournamentPlanner.Pages.PageModels.Testing.Pages_PageModels_Testing_PairwiseTesting Class Reference

Inheritance diagram for TournamentPlanner.Pages.PageModels.Testing.Pages_PageModels_Testing_PairwiseTesting:


global::MicrosoftAspNetCore ::Mvc::RazorPages::Page
TournamentPlanner.Pages. PageModels.Testing.Pages _PageModels_Testing_PairwiseTesting

Public Member Functions

- async override global::System.Threading.Tasks.Task **ExecuteAsync** ()
- async override global::System.Threading.Tasks.Task **ExecuteAsync** ()

Properties

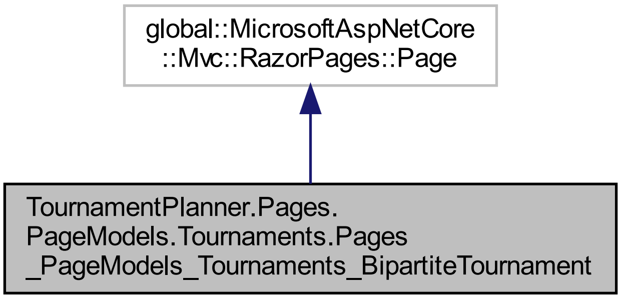
- global::Microsoft.AspNetCore.Mvc.ViewFeatures.IModelExpressionProvider **ModelExpressionProvider** [get]
- global::Microsoft.AspNetCore.Mvc.IUrlHelper **Url** [get]
- global::Microsoft.AspNetCore.Mvc.IViewComponentHelper **Component** [get]
- global::Microsoft.AspNetCore.Mvc.Rendering.IJsonHelper **Json** [get]
- global::Microsoft.AspNetCore.Mvc.Rendering.IHtmlHelper< TournamentPlanner.Pages.PageModels.Testing.PairwiseTestingM
  > **Html** [get]
- global::Microsoft.AspNetCore.Mvc.ViewFeatures.ViewDataDictionary< TournamentPlanner.Pages.PageModels.Testing.Pairwi
  > **ViewData** [get]
- TournamentPlanner.Pages.PageModels.Testing.PairwiseTestingModel **Model** [get]
- global::Microsoft.AspNetCore.Mvc.Rendering.IHtmlHelper< TournamentPlanner.Pages.PageModels.Testing.PairwiseTesting2Model > **Html** [get]
- global::Microsoft.AspNetCore.Mvc.ViewFeatures.ViewDataDictionary< TournamentPlanner.Pages.PageModels.Testing.PairwiseTesting2Model > **ViewData** [get]
- TournamentPlanner.Pages.PageModels.Testing.PairwiseTesting2Model **Model** [get]

The documentation for this class was generated from the following file:

- Debug/net5.0/Razor/Pages/PageModels/Testing/PairwiseTesting.cshtml.g.cs

7.84 TournamentPlanner.Pages.PageModels.Tournaments.Pages_PageModels_Tournaments_BipartiteTournament Class Reference

Inheritance diagram for TournamentPlanner.Pages.PageModels.Tournaments.Pages_PageModels_Tournaments_BipartiteTournament:


global::MicrosoftAspNetCore
::Mvc::RazorPages::Page
TournamentPlanner.Pages.
PageModels.Tournaments.Pages
_PageModels_Tournaments_BipartiteTournament

Public Member Functions

- async override global::System.Threading.Tasks.Task **ExecuteAsync** ()

Properties

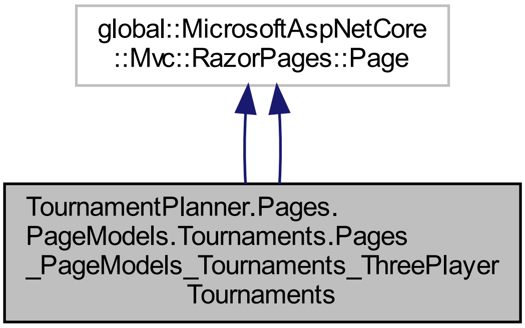
- global::Microsoft.AspNetCore.Mvc.ViewFeatures.IModelExpressionProvider **ModelExpressionProvider** [get]
- global::Microsoft.AspNetCore.Mvc.IUrlHelper **Url** [get]
- global::Microsoft.AspNetCore.Mvc.IViewComponentHelper **Component** [get]
- global::Microsoft.AspNetCore.Mvc.Rendering.IJsonHelper **Json** [get]
- global::Microsoft.AspNetCore.Mvc.Rendering.IHtmlHelper< TournamentPlanner.Pages.PageModels.Tournaments.BipartiteTou
  > **Html** [get]
- global::Microsoft.AspNetCore.Mvc.ViewFeatures.ViewDataDictionary< TournamentPlanner.Pages.PageModels.Tournaments.
  > **ViewData** [get]
- TournamentPlanner.Pages.PageModels.Tournaments.BipartiteTournamentModel **Model** [get]

The documentation for this class was generated from the following file:

- Debug/net5.0/Razor/Pages/PageModels/Tournaments/BipartiteTournament.cshtml.g.cs

7.85 TournamentPlanner.Pages.PageModels.Tournaments.Pages_PageModels_Tournaments_ThreePlayerTournaments Class Reference

Inheritance diagram for TournamentPlanner.Pages.PageModels.Tournaments.Pages_PageModels_Tournaments←_ThreePlayerTournaments:


global::MicrosoftAspNetCore
::Mvc::RazorPages::Page
TournamentPlanner.Pages.
PageModels.Tournaments.Pages
_PageModels_Tournaments_ThreePlayer
Tournaments

Public Member Functions

- async override global::System.Threading.Tasks.Task **ExecuteAsync** ()
- async override global::System.Threading.Tasks.Task **ExecuteAsync** ()

Properties

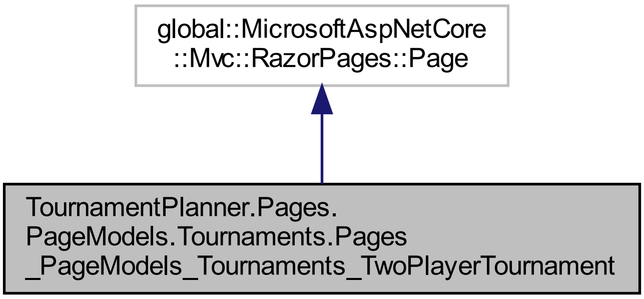
- global::Microsoft.AspNetCore.Mvc.ViewFeatures.IModelExpressionProvider **ModelExpressionProvider** [get]
- global::Microsoft.AspNetCore.Mvc.IUrlHelper **Url** [get]
- global::Microsoft.AspNetCore.Mvc.IViewComponentHelper **Component** [get]
- global::Microsoft.AspNetCore.Mvc.Rendering.IJsonHelper **Json** [get]
- global::Microsoft.AspNetCore.Mvc.Rendering.IHtmlHelper< TournamentPlanner.Pages.PageModels.Tournaments.ThreePlaye
  > **Html** [get]
- global::Microsoft.AspNetCore.Mvc.ViewFeatures.ViewDataDictionary< TournamentPlanner.Pages.PageModels.Tournaments.T
  > **ViewData** [get]
- TournamentPlanner.Pages.PageModels.Tournaments.ThreePlayerTournamentsModel **Model** [get]

The documentation for this class was generated from the following file:

- Debug/net5.0/Razor/Pages/PageModels/Tournaments/ThreePlayerTournaments.cshtml.g.cs

7.86 TournamentPlanner.Pages.PageModels.Tournaments.Pages_PageModels_Tournaments_TwoPlayerTournament Class Reference

Inheritance diagram for TournamentPlanner.Pages.PageModels.Tournaments.Pages_PageModels_Tournaments_TwoPlayerTournament:


global::MicrosoftAspNetCore ::Mvc::RazorPages::Page
TournamentPlanner.Pages. PageModels.Tournaments.Pages _PageModels_Tournaments_TwoPlayerTournament

Public Member Functions

- async override global::System.Threading.Tasks.Task **ExecuteAsync** ()

Properties

- global::Microsoft.AspNetCore.Mvc.ViewFeatures.IModelExpressionProvider **ModelExpressionProvider** [get]
- global::Microsoft.AspNetCore.Mvc.IUrlHelper **Url** [get]
- global::Microsoft.AspNetCore.Mvc.IViewComponentHelper **Component** [get]
- global::Microsoft.AspNetCore.Mvc.Rendering.IJsonHelper **Json** [get]
- global::Microsoft.AspNetCore.Mvc.Rendering.IHtmlHelper< PageModels.Tournaments.TwoPlayerTournamentModel
  > **Html** [get]
- global::Microsoft.AspNetCore.Mvc.ViewFeatures.ViewDataDictionary< PageModels.Tournaments.TwoPlayerTournamentMode
  > **ViewData** [get]
- PageModels.Tournaments.TwoPlayerTournamentModel **Model** [get]

7.86.1 Property Documentation

7.86.1.1 ModelExpressionProvider

```
global.Microsoft.AspNetCore.Mvc.ViewFeatures.IModelExpressionProvider TournamentPlanner.↩
Pages.PageModels.Tournaments.Pages_PageModels_Tournaments_TwoPlayerTournament.ModelExpression↩
Provider  [get]
```

Initial value:

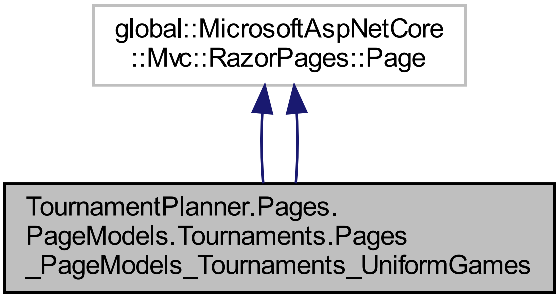
```
<div class=""content"">
    <div class=""content-section"">
        <div class=""content-section-header"">
            <h4>Round Robin Tournament</h4>
        </div>
        <div class=""content-section-body"">
            <p>
                A round-robin tournament (or all-play-all tournament) is a competition in which each
       contestant meets every other participant,
                usually in turn. A round-robin contrasts with an elimination tournament, in which
       participants are eliminated after
                a certain number of losses. <a
       href=""https://en.wikipedia.org/wiki/Round-robin_tournament"">wiki</a>
            </p>
        </div>
    </div>
    <div class=""content-section"">
        <div class=""content-section-header"">
            <h4>Construct Tournament</h4>
        </div>
        <div class=""content-section-body"">
            <div class=""row"">
                <div class=""col-md-4"">
                    "             <div class=""content-section"">
                <div class=""content-section-header"">
                    <h4>Resulting Tournament</h4>
                </div>
                <div class=""content-section-body"">
                    <table class=""result-table"">
"
```

The documentation for this class was generated from the following file:

- Debug/net5.0/Razor/Pages/PageModels/Tournaments/TwoPlayerTournament.cshtml.g.cs

7.87 TournamentPlanner.Pages.PageModels.Tournaments.Pages_Page↩ Models_Tournaments_UniformGames Class Reference

Inheritance diagram for TournamentPlanner.Pages.PageModels.Tournaments.Pages_PageModels_Tournaments↩ _UniformGames:


global::MicrosoftAspNetCore ::Mvc::RazorPages::Page
TournamentPlanner.Pages. PageModels.Tournaments.Pages _PageModels_Tournaments_UniformGames

Public Member Functions

- async override global::System.Threading.Tasks.Task **ExecuteAsync** ()
- async override global::System.Threading.Tasks.Task **ExecuteAsync** ()

Properties

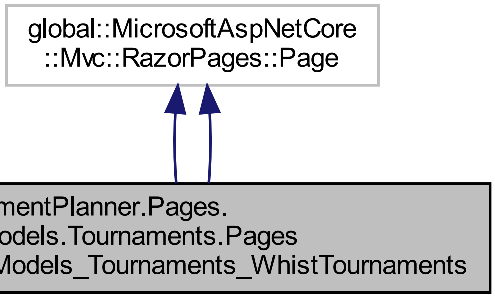
- global::Microsoft.AspNetCore.Mvc.ViewFeatures.IModelExpressionProvider **ModelExpressionProvider** [get]
- global::Microsoft.AspNetCore.Mvc.IUrlHelper **Url** [get]
- global::Microsoft.AspNetCore.Mvc.IViewComponentHelper **Component** [get]
- global::Microsoft.AspNetCore.Mvc.Rendering.IJsonHelper **Json** [get]
- global::Microsoft.AspNetCore.Mvc.Rendering.IHtmlHelper< TournamentPlanner.Pages.PageModels.Tournaments.UniformGa > **Html** [get]
- global::Microsoft.AspNetCore.Mvc.ViewFeatures.ViewDataDictionary< TournamentPlanner.Pages.PageModels.Tournaments. > **ViewData** [get]
- TournamentPlanner.Pages.PageModels.Tournaments.UniformGamesModel **Model** [get]

The documentation for this class was generated from the following file:

- Debug/net5.0/Razor/Pages/PageModels/Tournaments/UniformGames.cshtml.g.cs

7.88 TournamentPlanner.Pages.PageModels.Tournaments.Pages_Page← Models_Tournaments_WhistTournaments Class Reference

Inheritance diagram for TournamentPlanner.Pages.PageModels.Tournaments.Pages_PageModels_Tournaments← _WhistTournaments:


global::MicrosoftAspNetCore ::Mvc::RazorPages::Page
TournamentPlanner.Pages. PageModels.Tournaments.Pages _PageModels_Tournaments_WhistTournaments

Public Member Functions

- async override global::System.Threading.Tasks.Task **ExecuteAsync** ()
- async override global::System.Threading.Tasks.Task **ExecuteAsync** ()

Properties

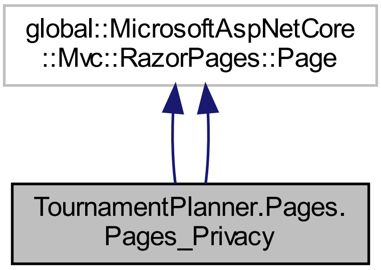
- global::Microsoft.AspNetCore.Mvc.ViewFeatures.IModelExpressionProvider **ModelExpressionProvider** [get]
- global::Microsoft.AspNetCore.Mvc.IUrlHelper **Url** [get]
- global::Microsoft.AspNetCore.Mvc.IViewComponentHelper **Component** [get]
- global::Microsoft.AspNetCore.Mvc.Rendering.IJsonHelper **Json** [get]
- global::Microsoft.AspNetCore.Mvc.Rendering.IHtmlHelper< TournamentPlanner.Pages.PageModels.Tournaments.WhistTourn > **Html** [get]
- global::Microsoft.AspNetCore.Mvc.ViewFeatures.ViewDataDictionary< TournamentPlanner.Pages.PageModels.Tournaments.\ > **ViewData** [get]
- TournamentPlanner.Pages.PageModels.Tournaments.WhistTournamentsModel **Model** [get]

The documentation for this class was generated from the following file:

- Debug/net5.0/Razor/Pages/PageModels/Tournaments/WhistTournaments.cshtml.g.cs

7.89 TournamentPlanner.Pages.Pages_Privacy Class Reference

Inheritance diagram for TournamentPlanner.Pages.Pages_Privacy:



Public Member Functions

- async override global::System.Threading.Tasks.Task **ExecuteAsync** ()
- async override global::System.Threading.Tasks.Task **ExecuteAsync** ()

Properties

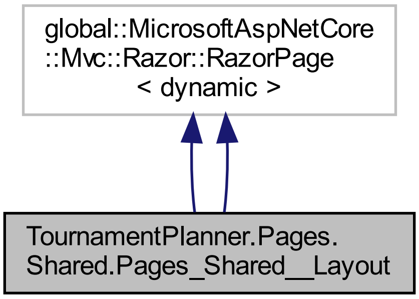
- global::Microsoft.AspNetCore.Mvc.ViewFeatures.IModelExpressionProvider **ModelExpressionProvider** [get]
- global::Microsoft.AspNetCore.Mvc.IUrlHelper **Url** [get]
- global::Microsoft.AspNetCore.Mvc.IViewComponentHelper **Component** [get]
- global::Microsoft.AspNetCore.Mvc.Rendering.IJsonHelper **Json** [get]
- global::Microsoft.AspNetCore.Mvc.Rendering.IHtmlHelper< PrivacyModel > **Html** [get]
- global::Microsoft.AspNetCore.Mvc.ViewFeatures.ViewDataDictionary< PrivacyModel > **ViewData** [get]
- PrivacyModel **Model** [get]

The documentation for this class was generated from the following file:

- Debug/net5.0/Razor/Pages/Privacy.cshtml.g.cs

7.90 TournamentPlanner.Pages.Shared.Pages_Shared__Layout Class Reference

Inheritance diagram for TournamentPlanner.Pages.Shared.Pages_Shared__Layout:



Public Member Functions

- async override global::System.Threading.Tasks.Task **ExecuteAsync** ()
- async override global::System.Threading.Tasks.Task **ExecuteAsync** ()

Properties

- global::Microsoft.AspNetCore.Mvc.ViewFeatures.IModelExpressionProvider **ModelExpressionProvider** `[get]`
- global::Microsoft.AspNetCore.Mvc.IUrlHelper **Url** `[get]`
- global::Microsoft.AspNetCore.Mvc.IViewComponentHelper **Component** `[get]`
- global::Microsoft.AspNetCore.Mvc.Rendering.IJsonHelper **Json** `[get]`
- global::Microsoft.AspNetCore.Mvc.Rendering.IHtmlHelper< dynamic > **Html** `[get]`

7.90.1 Property Documentation

7.90.1.1 ModelExpressionProvider

```
Microsoft AspNetCore Mvc ViewFeatures global.IModelExpressionProvider TournamentPlanner.↩
Pages.Shared.Pages_Shared__Layout.ModelExpressionProvider  [get]
```

Initial value:

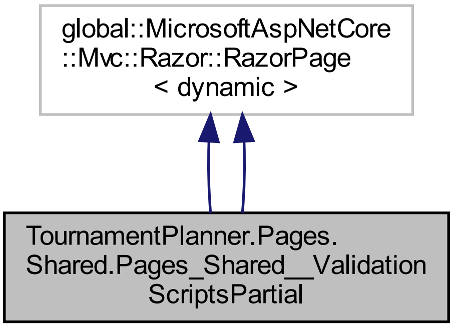
```
    <button class=""navbar-toggler"" type=""button"" data-toggle=""collapse""
   data-target="".navbar-collapse"" aria-controls=""navbarSupportedContent""
        aria-expanded=""false"" aria-label=""Toggle navigation"">
      <span class=""navbar-toggler-icon""></span>
    </button>
    </nav>
    <div class=""all"">
        <nav class=""left"">
            <ul class=""content"">
                <li class=""nav-head""><b>Designs</b></li>
                <li class=""nav-item"">
                    "

    <script>
        var sandwichToggler = document.querySelector(""#sandwich"");
        sandwichToggler.addEventListener('click', toggleContentMenu, false);
        function toggleContentMenu() {
            let leftMenu = document.querySelector('.left');
            leftMenu.classList.toggle('toggled');
        }
    </script>

    "
```

The documentation for this class was generated from the following file:

- Debug/net5.0/Razor/Pages/Shared/_Layout.cshtml.g.cs

7.91 TournamentPlanner.Pages.Shared.Pages_Shared__Validation↩ ScriptsPartial Class Reference

Inheritance diagram for TournamentPlanner.Pages.Shared.Pages_Shared__ValidationScriptsPartial:



Public Member Functions

- async override global::System.Threading.Tasks.Task **ExecuteAsync** ()
- async override global::System.Threading.Tasks.Task **ExecuteAsync** ()

Properties

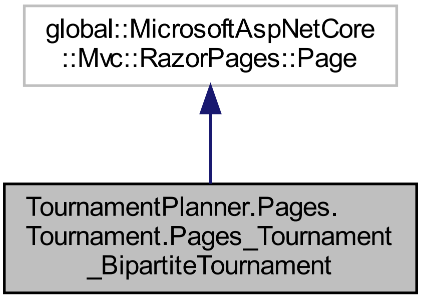
- global::Microsoft.AspNetCore.Mvc.ViewFeatures.IModelExpressionProvider **ModelExpressionProvider** [get]
- global::Microsoft.AspNetCore.Mvc.IUrlHelper **Url** [get]
- global::Microsoft.AspNetCore.Mvc.IViewComponentHelper **Component** [get]
- global::Microsoft.AspNetCore.Mvc.Rendering.IJsonHelper **Json** [get]
- global::Microsoft.AspNetCore.Mvc.Rendering.IHtmlHelper< dynamic > **Html** [get]

The documentation for this class was generated from the following file:

- Debug/net5.0/Razor/Pages/Shared/_ValidationScriptsPartial.cshtml.g.cs

7.92 TournamentPlanner.Pages.Tournament.Pages_Tournament_↩ BipartiteTournament Class Reference

Inheritance diagram for TournamentPlanner.Pages.Tournament.Pages_Tournament_BipartiteTournament:


global::MicrosoftAspNetCore ::Mvc::RazorPages::Page
TournamentPlanner.Pages. Tournament.Pages_Tournament _BipartiteTournament

Public Member Functions

- async override global::System.Threading.Tasks.Task **ExecuteAsync** ()

Properties

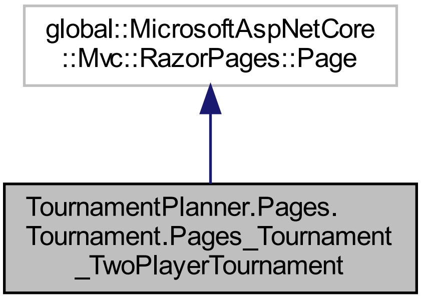
- global::Microsoft.AspNetCore.Mvc.ViewFeatures.IModelExpressionProvider **ModelExpressionProvider** `[get]`
- global::Microsoft.AspNetCore.Mvc.IUrlHelper **Url** `[get]`
- global::Microsoft.AspNetCore.Mvc.IViewComponentHelper **Component** `[get]`
- global::Microsoft.AspNetCore.Mvc.Rendering.IJsonHelper **Json** `[get]`
- global::Microsoft.AspNetCore.Mvc.Rendering.IHtmlHelper< TournamentPlanner.Pages.Tournament.↩ BipartiteTournamentModel > **Html** `[get]`
- global::Microsoft.AspNetCore.Mvc.ViewFeatures.ViewDataDictionary< TournamentPlanner.Pages.↩ Tournament.BipartiteTournamentModel > **ViewData** `[get]`
- TournamentPlanner.Pages.Tournament.BipartiteTournamentModel **Model** `[get]`

The documentation for this class was generated from the following file:

- Release/net5.0/Razor/Pages/Tournament/BipartiteTournament.cshtml.g.cs

7.93 TournamentPlanner.Pages.Tournament.Pages_Tournament_TwoPlayerTournament Class Reference

Inheritance diagram for TournamentPlanner.Pages.Tournament.Pages_Tournament_TwoPlayerTournament:



Public Member Functions

- async override global::System.Threading.Tasks.Task **ExecuteAsync** ()

Properties

- global::Microsoft.AspNetCore.Mvc.ViewFeatures.IModelExpressionProvider **ModelExpressionProvider** `[get]`
- global::Microsoft.AspNetCore.Mvc.IUrlHelper **Url** `[get]`
- global::Microsoft.AspNetCore.Mvc.IViewComponentHelper **Component** `[get]`
- global::Microsoft.AspNetCore.Mvc.Rendering.IJsonHelper **Json** `[get]`
- global::Microsoft.AspNetCore.Mvc.Rendering.IHtmlHelper< TournamentPlanner.Pages.Tournament.TwoPlayerTournamentModel > **Html** `[get]`
- global::Microsoft.AspNetCore.Mvc.ViewFeatures.ViewDataDictionary< TournamentPlanner.Pages.Tournament.TwoPlayerTournamentModel > **ViewData** `[get]`
- TournamentPlanner.Pages.Tournament.TwoPlayerTournamentModel **Model** `[get]`

7.93.1 Property Documentation

7.93.1.1 ModelExpressionProvider

```
global.Microsoft.AspNetCore.Mvc.ViewFeatures.IModelExpressionProvider TournamentPlanner.↩
Pages.Tournament.Pages_Tournament_TwoPlayerTournament.ModelExpressionProvider  [get]
```

Initial value:

```
<div class=""content"">
    <div class=""content-section"">
        <div class=""content-section-header"">
            <h4>Round Robin Tournament</h4>
        </div>
        <div class=""content-section-body"">
            Etiam commodo dui eget wisi. Vivamus luctus egestas leo. Nullam sit amet magna in magna gravida
       vehicula.
            Sed vel lectus. Donec odio tempus molestie, porttitor ut, iaculis quis, sem.
            Aliquam ante. In laoreet, magna id viverra tincidunt, sem odio bibendum justo, vel imperdiet
       sapien wisi sed libero.
            Vivamus ac leo pretium faucibus. Fusce nibh. Praesent in mauris eu tortor porttitor accumsan.
       Integer malesuada.
        </div>
    </div>
    <div class=""content-section"">
        <div class=""content-section-header"">
            <h4>Construct Tournament</h4>
        </div>
        <div class=""content-section-body"">
            <div class=""row"">
                <div class=""col-md-4"">
                    "
```

The documentation for this class was generated from the following file:

- Release/net5.0/Razor/Pages/Tournament/TwoPlayerTournament.cshtml.g.cs

7.94 TournamentPlanner.Model.Testing.PairwiseTestCases Class Reference

A class representing the pairwise test cases created using orthogonal arrays a some post processing methods.

Public Member Functions

- PairwiseTestCases (List< string > variableNames, Dictionary< int, List< string >> variableValues, List< List< int >> testCases)

  Initializes a new instance of the PairwiseTestCases class.
- List< List< string > > TranslatedTestCases ()

  Test cases as list of strings.

Properties

- List< string > VariableNames [get]

  Gets the list of variable names.
- Dictionary< int, List< string > > VariableValues [get]

  Gets the dictionary of variable values. The key is the index of the variable name in its list.
- List< List< int > > TestCases [get]

  Gets the test cases.

7.94.1 Detailed Description

A class representing the pairwise test cases created using orthogonal arrays a some post processing methods.

7.94.2 Constructor & Destructor Documentation

7.94.2.1 PairwiseTestCases()

```
TournamentPlanner.Model.Testing.PairwiseTestCases.PairwiseTestCases (
            List< string > variableNames,
            Dictionary< int, List< string >> variableValues,
            List< List< int >> testCases )
```

Initializes a new instance of the PairwiseTestCases class.

Parameters

variableNames	Variable names.
variableValues	Variable values.
testCases	Test cases.

7.94.3 Member Function Documentation

7.94.3.1 TranslatedTestCases()

```
List<List<string> > TournamentPlanner.Model.Testing.PairwiseTestCases.TranslatedTestCases ( )
```

Test cases as list of strings.

Returns

Translated test cases.

7.94.4 Property Documentation

7.94.4.1 TestCases

```
List<List<int> > TournamentPlanner.Model.Testing.PairwiseTestCases.TestCases  [get]
```

Gets the test cases.

7.94.4.2 VariableNames

```
List<string> TournamentPlanner.Model.Testing.PairwiseTestCases.VariableNames  [get]
```

Gets the list of variable names.

7.94.4.3 VariableValues

```
Dictionary<int, List<string> > TournamentPlanner.Model.Testing.PairwiseTestCases.Variable←
Values  [get]
```

Gets the dictionary of variable values. The key is the index of the variable name in its list.

The documentation for this class was generated from the following file:

- PairwiseTestCases.cs

7.95 TournamentPlanner.Algorithms.Testing.PairwiseTestCases← Construction Class Reference

Class implementing the construction of the pairwise test cases using orthogonal arrays.

Public Member Functions

- PairwiseTestCasesConstruction (List< string > variableNames, Dictionary< int, List< string >> variable← Values)

  Initializes a new instance of the PairwiseTestCasesConstruction class.
- PairwiseTestCases ConstructPairwiseTestCases ()

  Construct the pairwise test cases.

7.95.1 Detailed Description

Class implementing the construction of the pairwise test cases using orthogonal arrays.

7.95.2 Constructor & Destructor Documentation

7.95.2.1 PairwiseTestCasesConstruction()

```
TournamentPlanner.Algorithms.Testing.PairwiseTestCasesConstruction.PairwiseTestCasesConstruction
(
            List< string > variableNames,
            Dictionary< int, List< string >> variableValues )
```

Initializes a new instance of the PairwiseTestCasesConstruction class.

Parameters

variableNames	Variable names.
variableValues	Variable values.

7.95.3 Member Function Documentation

7.95.3.1 ConstructPairwiseTestCases()

```
PairwiseTestCases TournamentPlanner.Algorithms.Testing.PairwiseTestCasesConstruction.Construct↩
PairwiseTestCases ( )
```

Construct the pairwise test cases.

Returns

Constructed test cases.

The documentation for this class was generated from the following file:

- PairwiseTestCasesConstruction.cs

7.96 TournamentPlannerTests.Algorithms.Testing.PairwiseTestCases↩ ConstructionTests Class Reference

Public Member Functions

- void **TariffFairsPairwiseTestsConstructionTest** ()

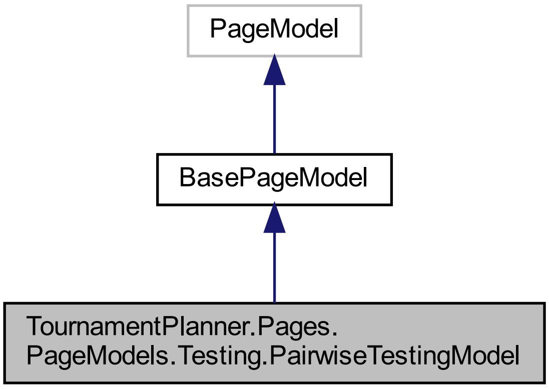
The documentation for this class was generated from the following file:

- PairwiseTestCasesConstructionTests.cs

7.97 TournamentPlanner.Pages.PageModels.Testing.PairwiseTesting↩ Model Class Reference

Page model for the construction of pairwise tests.

Inheritance diagram for TournamentPlanner.Pages.PageModels.Testing.PairwiseTestingModel:



Public Member Functions

- PairwiseTestingModel ()

  Initializes a new instance of the PairwiseTestingModel class.
- void OnPost (IFormCollection data)

Properties

- Dictionary< string, List< string > > VariableValues [get]

  Gets the variable values.
- PairwiseTestCases TestCases [get]

  Gets the constructed test cases.

7.97.1 Detailed Description

Page model for the construction of pairwise tests.

7.97.2 Constructor & Destructor Documentation

7.97.2.1 PairwiseTestingModel()

```
TournamentPlanner.Pages.PageModels.Testing.PairwiseTestingModel.PairwiseTestingModel ( )
```

Initializes a new instance of the PairwiseTestingModel class.

7.97.3 Property Documentation

7.97.3.1 TestCases

```
PairwiseTestCases TournamentPlanner.Pages.PageModels.Testing.PairwiseTestingModel.TestCases
[get]
```

Gets the constructed test cases.

7.97.3.2 VariableValues

```
Dictionary<string, List<string> > TournamentPlanner.Pages.PageModels.Testing.PairwiseTesting←
Model.VariableValues [get]
```

Gets the variable values.

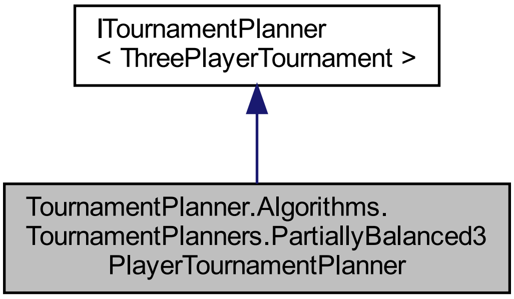
The documentation for this class was generated from the following file:

- PairwiseTesting.cshtml.cs

7.98 TournamentPlanner.Algorithms.TournamentPlanners.PartiallyBalanced3PlayerTournamentPlanner Class Reference

This class facilitates the construction of partially balanced three player tournaments. The number of players in the tournament has to be divisible by 3. For the construction of the tournament, GDD are used.

Inheritance diagram for TournamentPlanner.Algorithms.TournamentPlanners.PartiallyBalanced3PlayerTournamentPlanner:


ITournamentPlanner
< ThreePlayerTournament >
TournamentPlanner.Algorithms.
TournamentPlanners.PartiallyBalanced3
PlayerTournamentPlanner

Public Member Functions

- PartiallyBalanced3PlayerTournamentPlanner (int v)

  Initializes a new instance of the PartiallyBalanced3PlayerTournamentPlanner class.
- ThreePlayerTournament PlanTournament ()

7.98.1 Detailed Description

This class facilitates the construction of partially balanced three player tournaments. The number of players in the tournament has to be divisible by 3. For the construction of the tournament, GDD are used.

7.98.2 Constructor & Destructor Documentation

7.98.2.1 PartiallyBalanced3PlayerTournamentPlanner()

```
TournamentPlanner.Algorithms.TournamentPlanners.PartiallyBalanced3PlayerTournamentPlanner.↩
PartiallyBalanced3PlayerTournamentPlanner (
            int v )
```

Initializes a new instance of the PartiallyBalanced3PlayerTournamentPlanner class.

Parameters

v	Number of players.

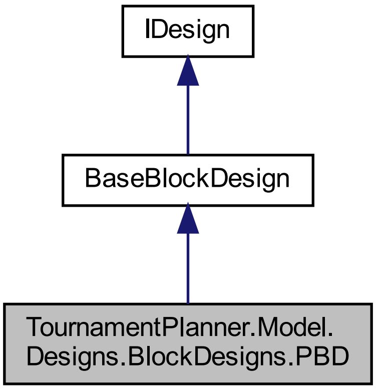
The documentation for this class was generated from the following file:

- PartiallyBalanced3PlayerTournamentPlanner.cs

7.99 TournamentPlanner.Model.Designs.BlockDesigns.PBD Class Reference

Class representing a Pairwise balanced design. PBD is similar to BIBD, but the block sizes are from K - they don't have to be uniform. We only consider PBDs with lambda = 1.

Inheritance diagram for TournamentPlanner.Model.Designs.BlockDesigns.PBD:



Public Member Functions

- PBD (int v, List< int > k)

  Initializes a new instance of the PBD class.
- PBD (int v, List< int > k, List< List< int >> blocks)

  Initializes a new instance of the PBD class.
- PBD (int v, List< List< int >> blocks)

  Initializes a new instance of the PBD class.
- void AddBlock (List< int > block)

  Add a block to the list of blocks.

Properties

- List< int > K `[get]`

  Gets the block sizes of the PBD.

Additional Inherited Members

7.99.1 Detailed Description

Class representing a Pairwise balanced design. PBD is similar to BIBD, but the block sizes are from K - they don't have to be uniform. We only consider PBDs with lambda = 1.

7.99.2 Constructor & Destructor Documentation

7.99.2.1 PBD() [1/3]

```
TournamentPlanner.Model.Designs.BlockDesigns.PBD.PBD (
            int v,
            List< int > k )
```

Initializes a new instance of the PBD class.

Parameters

v	The order of the PBD.
k	The block sizes of the PBD.

7.99.2.2 PBD() [2/3]

```
TournamentPlanner.Model.Designs.BlockDesigns.PBD.PBD (
            int v,
            List< int > k,
            List< List< int >> blocks )
```

Initializes a new instance of the PBD class.

Parameters

v	The order of the PBD.
k	The block sizes of the PBD.
blocks	The blocks of the PBD.

7.99.2.3 PBD() [3/3]

```
TournamentPlanner.Model.Designs.BlockDesigns.PBD.PBD (
            int v,
            List< List< int >> blocks )
```

Initializes a new instance of the PBD class.

Parameters

v	The order of the PBD.
blocks	The blocks of the PBD.

7.99.3 Member Function Documentation

7.99.3.1 AddBlock()

```
void TournamentPlanner.Model.Designs.BlockDesigns.PBD.AddBlock (
            List< int > block )
```

Add a block to the list of blocks.

Parameters

block	The block to add.

7.99.4 Property Documentation

7.99.4.1 K

```
List<int> TournamentPlanner.Model.Designs.BlockDesigns.PBD.K  [get]
```

Gets the block sizes of the PBD.

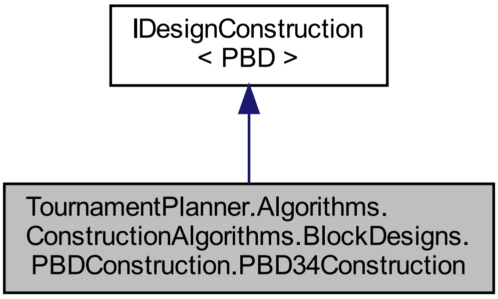
The documentation for this class was generated from the following file:

- PBD.cs

7.100 TournamentPlanner.Algorithms.ConstructionAlgorithms.Block↩ Designs.PBDConstruction.PBD34Construction Class Reference

The construction of PBD 34 was supposed to be in the SmallOrderPBDConstruction, but was a bit too complicated so i extracted it to its custom class. For the construction of PBD34 a PBD27 needs to be constructed first. Then it is modified and new blocks are added to get a PBD34.

Inheritance diagram for TournamentPlanner.Algorithms.ConstructionAlgorithms.BlockDesigns.PBDConstruction.↩ PBD34Construction:


IDesignConstruction < PBD >
TournamentPlanner.Algorithms. ConstructionAlgorithms.BlockDesigns. PBDConstruction.PBD34Construction

Public Member Functions

- PBD ConstructDesign ()

  Construct PBD of order 34.

7.100.1 Detailed Description

The construction of PBD 34 was supposed to be in the SmallOrderPBDConstruction, but was a bit too complicated so i extracted it to its custom class. For the construction of PBD34 a PBD27 needs to be constructed first. Then it is modified and new blocks are added to get a PBD34.

7.100.2 Member Function Documentation

7.100.2.1 ConstructDesign()

```
PBD TournamentPlanner.Algorithms.ConstructionAlgorithms.BlockDesigns.PBDConstruction.PBD34↩
Construction.ConstructDesign ( )
```

Construct PBD of order 34.

Returns

Constructed PBD.

The documentation for this class was generated from the following file:

- PBD34Construction.cs

7.101 TournamentPlannerTests.Algorithms.ConstructionAlgorithms.↩ BlockDesigns.PBDConstruction.PBD34ConstructionTest Class Reference

Public Member Functions

- void **TestPBD34Construction** ()

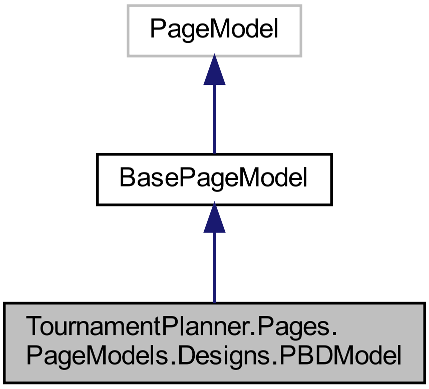
The documentation for this class was generated from the following file:

- PBD34ConstructionTest.cs

7.102 TournamentPlanner.Pages.PageModels.Designs.PBDModel Class Reference

Page model for the pairwise balanced design. Construction of PBD of orders 3M + 1 is implemented.

Inheritance diagram for TournamentPlanner.Pages.PageModels.Designs.PBDModel:



Public Member Functions

- void OnPost ()

Properties

- int Multiplier = 4 `[get, set]`

  Gets or sets the multiplier.
- int V = 13 `[get, set]`

  Gets or sets the order of the PBD to construct.
- PBD PBD `[get]`

  Gets the constructed PBD.

7.102.1 Detailed Description

Page model for the pairwise balanced design. Construction of PBD of orders 3M + 1 is implemented.

7.102.2 Property Documentation

7.102.2.1 Multiplier

```
int TournamentPlanner.Pages.PageModels.Designs.PBDModel.Multiplier = 4  [get], [set]
```

Gets or sets the multiplier.

7.102.2.2 PBD

```
PBD TournamentPlanner.Pages.PageModels.Designs.PBDModel.PBD  [get]
```

Gets the constructed PBD.

7.102.2.3 V

```
int TournamentPlanner.Pages.PageModels.Designs.PBDModel.V = 13  [get], [set]
```

Gets or sets the order of the PBD to construct.

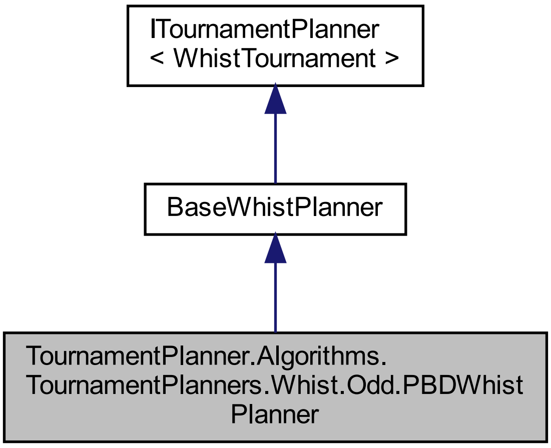
The documentation for this class was generated from the following file:

- PBD.cshtml.cs

7.103 TournamentPlanner.Algorithms.TournamentPlanners.Whist.Odd.↩ PBDWhistPlanner Class Reference

Class facilitating the construction of Whist(21) and Whist(69) using PBDs.

Inheritance diagram for TournamentPlanner.Algorithms.TournamentPlanners.Whist.Odd.PBDWhistPlanner:


ITournamentPlanner
< WhistTournament >
BaseWhistPlanner
TournamentPlanner.Algorithms.
TournamentPlanners.Whist.Odd.PBDWhist
Planner

Public Member Functions

- PBDWhistPlanner (int v)
- override WhistTournament PlanTournament ()

    Plan the Whist tournament.

Additional Inherited Members

7.103.1 Detailed Description

Class facilitating the construction of Whist(21) and Whist(69) using PBDs.

The documentation for this class was generated from the following file:

- PBDWhistPlanner.cs

7.104 TournamentPlannerTests.Algorithms.TournamentPlanners.↩ Whist.Odd.PBDWhistPlannerTests Class Reference

Tests for the PBDWhistPlannerTests.

Public Member Functions

- void Whist21ConstructionTest ()

    Test the construction of whist 21.
- void **Whist69ConstructionTest** ()

7.104.1 Detailed Description

Tests for the PBDWhistPlannerTests.

7.104.2 Member Function Documentation

7.104.2.1 Whist21ConstructionTest()

```
void TournamentPlannerTests.Algorithms.TournamentPlanners.Whist.Odd.PBDWhistPlannerTests.↩
Whist21ConstructionTest ( )
```

Test the construction of whist 21.

The documentation for this class was generated from the following file:

- PBDWhistPlannerTests.cs

7.105 TournamentPlannerTests.Utils.PrimeFactorizationTests Class Reference

Tests for the PrimeFactorization static class.

Public Member Functions

- void IsPrimeTest ()

  Test the recognition of primes.
- void IsPrimeOrPrimePowerTest ()

  Test the recognition of prime powers.

7.105.1 Detailed Description

Tests for the PrimeFactorization static class.

7.105.2 Member Function Documentation

7.105.2.1 IsPrimeOrPrimePowerTest()

```
void TournamentPlannerTests.Utils.PrimeFactorizationTests.IsPrimeOrPrimePowerTest ( )
```

Test the recognition of prime powers.

7.105.2.2 IsPrimeTest()

```
void TournamentPlannerTests.Utils.PrimeFactorizationTests.IsPrimeTest ( )
```

Test the recognition of primes.

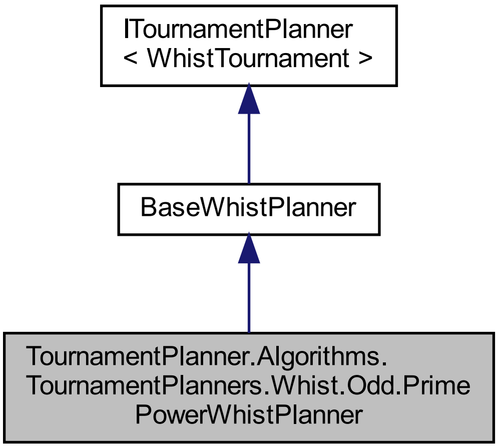
The documentation for this class was generated from the following file:

- PrimeFactorizationTests.cs

7.106 TournamentPlanner.Algorithms.TournamentPlanners.Whist.Odd.PrimePowerWhistPlanner Class Reference

Whist tournament planner for prime power number of players in the form of v = 4n + 1. It uses finite fields for the tournament construction.

Inheritance diagram for TournamentPlanner.Algorithms.TournamentPlanners.Whist.Odd.PrimePowerWhistPlanner:



Public Member Functions

- PrimePowerWhistPlanner (int v)

  Initializes a new instance of the PrimePowerWhistPlanner class.
- override WhistTournament PlanTournament ()

  Plan the Whist tournament.

Additional Inherited Members

7.106.1 Detailed Description

Whist tournament planner for prime power number of players in the form of v = 4n + 1. It uses finite fields for the tournament construction.

7.106.2 Constructor & Destructor Documentation

7.106.2.1 PrimePowerWhistPlanner()

```
TournamentPlanner.Algorithms.TournamentPlanners.Whist.Odd.PrimePowerWhistPlanner.PrimePowerWhistPlanner (
                int v )
```

Initializes a new instance of the PrimePowerWhistPlanner class.

Parameters

v	The order od the whist tournament.

The documentation for this class was generated from the following file:

- PrimePowerWhistPlanner.cs

7.107 TournamentPlannerTests.Algorithms.TournamentPlanners.Whist.Odd.PrimePowerWhistPlannerTests Class Reference

Tests for the PrimeNWhistPlanner.

Public Member Functions

- void **TestGFWhist5** ()
- void TestSeveralValidInputs ()

  Test for several V = 4N + 1 where V is prime.

7.107.1 Detailed Description

Tests for the PrimeNWhistPlanner.

7.107.2 Member Function Documentation

7.107.2.1 TestSeveralValidInputs()

```
void TournamentPlannerTests.Algorithms.TournamentPlanners.Whist.Odd.PrimePowerWhistPlanner↩
Tests.TestSeveralValidInputs ( )
```

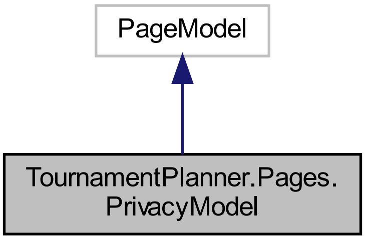
Test for several V = 4N + 1 where V is prime.

The documentation for this class was generated from the following file:

- PrimePowerWhistPlannerTests.cs

7.108 TournamentPlanner.Pages.PrivacyModel Class Reference

Inheritance diagram for TournamentPlanner.Pages.PrivacyModel:



Public Member Functions

- **PrivacyModel** (ILogger< PrivacyModel > logger)
- void **OnGet** ()

The documentation for this class was generated from the following file:

- Privacy.cshtml.cs

7.109 TournamentPlanner.Program Class Reference

Static Public Member Functions

- static void **Main** (string[ ] args)
- static IHostBuilder **CreateHostBuilder** (string[ ] args)

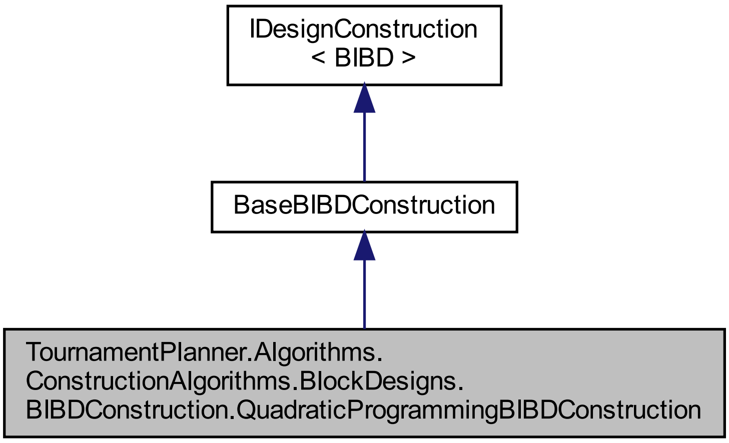
The documentation for this class was generated from the following file:

- Program.cs

7.110 TournamentPlanner.Algorithms.ConstructionAlgorithms.Block↩ Designs.BIBDConstruction.QuadraticProgramming↩ BIBDConstruction Class Reference

A quadratic programming approach to the construction of BIBDs as described in the thesis. Uses CP-SAT as solver.

Inheritance diagram for TournamentPlanner.Algorithms.ConstructionAlgorithms.BlockDesigns.BIBDConstruction.↩ QuadraticProgrammingBIBDConstruction:


IDesignConstruction
< BIBD >
BaseBIBDConstruction
TournamentPlanner.Algorithms.
ConstructionAlgorithms.BlockDesigns.
BIBDConstruction.QuadraticProgrammingBIBDConstruction

Public Member Functions

- QuadraticProgrammingBIBDConstruction (int v, int k, int lambda, int timeLimit=30)
- override BIBD ConstructDesign ()

  Construct the BIBD.

Additional Inherited Members

7.110.1 Detailed Description

A quadratic programming approach to the construction of BIBDs as described in the thesis. Uses CP-SAT as solver.

The documentation for this class was generated from the following file:

- QuadraticProgrammingBIBDConstruction.cs

7.111 TournamentPlannerTests.Algorithms.ConstructionAlgorithms.↩ BlockDesigns.BIBDConstruction.QuadraticProgramming↩ BIBDConstructionTests Class Reference

Tests for the QuadraticProgrammingBIBDConstruction.

Public Member Functions

- void QudraticConstructionTest ()

  Simple BIBD construction test.

7.111.1 Detailed Description

Tests for the QuadraticProgrammingBIBDConstruction.

7.111.2 Member Function Documentation

7.111.2.1 QudraticConstructionTest()

```
void TournamentPlannerTests.Algorithms.ConstructionAlgorithms.BlockDesigns.BIBDConstruction.↩
QuadraticProgrammingBIBDConstructionTests.QudraticConstructionTest ( )
```

Simple BIBD construction test.

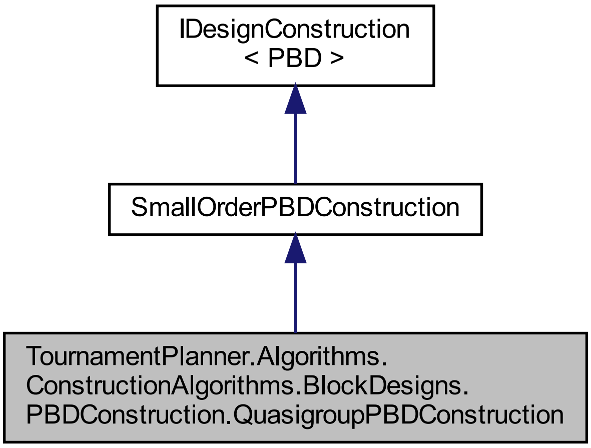
The documentation for this class was generated from the following file:

- QuadraticProgrammingBIBDConstructionTests.cs

7.112 TournamentPlanner.Algorithms.ConstructionAlgorithms.Block↩ Designs.PBDConstruction.QuasigroupPBDConstruction Class Reference

Method for construction of PBDs of orders 13, 16 and 28. This method uses pairs of MOLS as quasigroups.

Inheritance diagram for TournamentPlanner.Algorithms.ConstructionAlgorithms.BlockDesigns.PBDConstruction.↩
QuasigroupPBDConstruction:


IDesignConstruction
< PBD >
SmallOrderPBDConstruction
TournamentPlanner.Algorithms.
ConstructionAlgorithms.BlockDesigns.
PBDConstruction.QuasigroupPBDConstruction

Public Member Functions

- QuasigroupPBDConstruction (int v)

  Initializes a new instance of the QuasigroupPBDConstruction class.
- new PBD ConstructDesign ()

Additional Inherited Members

7.112.1 Detailed Description

Method for construction of PBDs of orders 13, 16 and 28. This method uses pairs of MOLS as quasigroups.

7.112.2 Constructor & Destructor Documentation

7.112.2.1 QuasigroupPBDConstruction()

```
TournamentPlanner.Algorithms.ConstructionAlgorithms.BlockDesigns.PBDConstruction.Quasigroup↩
PBDConstruction.QuasigroupPBDConstruction (
            int v )
```

Initializes a new instance of the QuasigroupPBDConstruction class.

Parameters

v	The order of PBD.

The documentation for this class was generated from the following file:

- QuasigroupPBDConstruction.cs

7.113 TournamentPlannerTests.Algorithms.ConstructionAlgorithms.↩ BlockDesigns.PBDConstruction.QuasigroupPBDConstruction↩ Tests Class Reference

Tests for the QuasigroupPBDConstruction.

Public Member Functions

- void QuasigroupConstructionTest ()
  Tests the construction of PBDs of orders 13, 16, 28.

7.113.1 Detailed Description

Tests for the QuasigroupPBDConstruction.

7.113.2 Member Function Documentation

7.113.2.1 QuasigroupConstructionTest()

```
void TournamentPlannerTests.Algorithms.ConstructionAlgorithms.BlockDesigns.PBDConstruction.↩
QuasigroupPBDConstructionTests.QuasigroupConstructionTest ( )
```

Tests the construction of PBDs of orders 13, 16, 28.

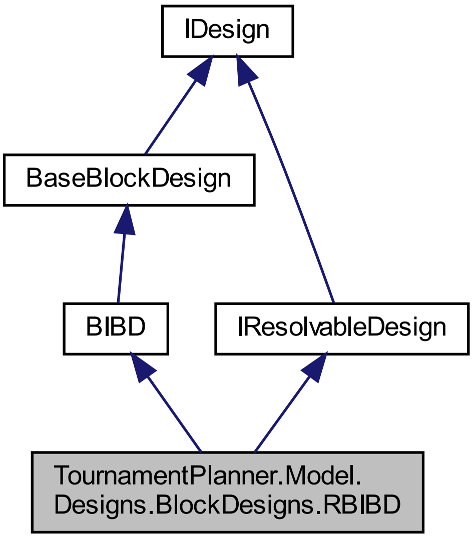
The documentation for this class was generated from the following file:

- QuasigroupPBDConstructionTests.cs

7.114 TournamentPlanner.Model.Designs.BlockDesigns.RBIBD Class Reference

Class representing a resolvable balanced incomplete blocks design.

Inheritance diagram for TournamentPlanner.Model.Designs.BlockDesigns.RBIBD:



Public Member Functions

- RBIBD (int v, int k, int lambda)

  Initializes a new instance of the RBIBD class.
- void AddParallelClass (List< List< int >> parallelClass)

  Add a parallel class. Also add its block to the list of blocks.

Properties

- List< List< List< int > > > ParallelClasses = new () `[get, protected set]`

Additional Inherited Members

7.114.1 Detailed Description

Class representing a resolvable balanced incomplete blocks design.

7.114.2 Constructor & Destructor Documentation

7.114.2.1 RBIBD()

```
TournamentPlanner.Model.Designs.BlockDesigns.RBIBD.RBIBD (
            int v,
            int k,
            int lambda )
```

Initializes a new instance of the RBIBD class.

Parameters

v	Order of the RBIBD.
k	Size of blocks.
lambda	Pair occurrences.

7.114.3 Member Function Documentation

7.114.3.1 AddParallelClass()

```
void TournamentPlanner.Model.Designs.BlockDesigns.RBIBD.AddParallelClass (
            List< List< int >> parallelClass )
```

Add a parallel class. Also add its block to the list of blocks.

Parameters

parallelClass	The parallel class to add.

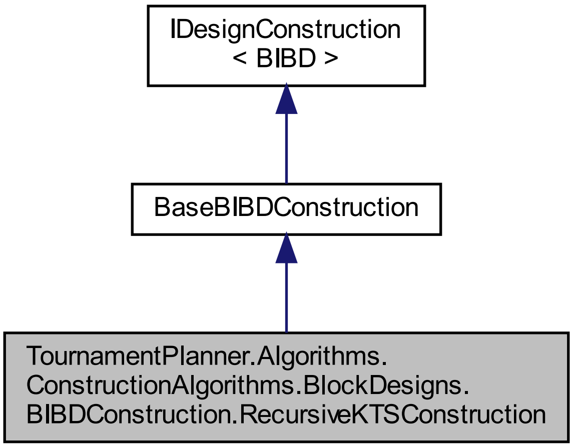
The documentation for this class was generated from the following file:

- RBIBD.cs

7.115 TournamentPlanner.Algorithms.ConstructionAlgorithms.Block↩ Designs.BIBDConstruction.RecursiveKTSConstruction Class Reference

Algorithm for the recursive construction of KTS of order 6N + 3. Method described in https://books.google.↩ tg/books?id=OoRFisqn-aIC. KTSs of orders 9, 15, 21 and 39 are constructed statically, all the others are constructed in a recursive manner.

Inheritance diagram for TournamentPlanner.Algorithms.ConstructionAlgorithms.BlockDesigns.BIBDConstruction.↩ RecursiveKTSConstruction:


IDesignConstruction
< BIBD >
BaseBIBDConstruction
TournamentPlanner.Algorithms.
ConstructionAlgorithms.BlockDesigns.
BIBDConstruction.RecursiveKTSConstruction

Public Member Functions

- RecursiveKTSConstruction (int v)

  Initializes a new instance of the RecursiveKTSConstruction class.
- override KTS ConstructDesign ()

  Construct a KTS.
- List< List< List< Tuple< int, int > > > > ConstructBlockInducedParallelClasses (List< int > block, KTS kts)

  Construct block induced parallel classes of tuples (before translation).

Additional Inherited Members

7.115.1 Detailed Description

Algorithm for the recursive construction of KTS of order 6N + 3. Method described in https://books.google.↩ tg/books?id=OoRFisqn-aIC. KTSs of orders 9, 15, 21 and 39 are constructed statically, all the others are constructed in a recursive manner.

7.115.2 Constructor & Destructor Documentation

7.115.2.1 RecursiveKTSConstruction()

```
TournamentPlanner.Algorithms.ConstructionAlgorithms.BlockDesigns.BIBDConstruction.Recursive↩
KTSConstruction.RecursiveKTSConstruction (
            int v )
```

Initializes a new instance of the RecursiveKTSConstruction class.

Parameters

v	Set size.

7.115.3 Member Function Documentation

7.115.3.1 ConstructBlockInducedParallelClasses()

```
List<List<List<Tuple<int, int> > > > TournamentPlanner.Algorithms.ConstructionAlgorithms.↩
BlockDesigns.BIBDConstruction.RecursiveKTSConstruction.ConstructBlockInducedParallelClasses (
            List< int > block,
            KTS kts )
```

Construct block induced parallel classes of tuples (before translation).

Parameters

block	The block of PBD.
kts	The initial KTS.

Returns

Parallel classes of tuples.

7.115.3.2 ConstructDesign()

```
override KTS TournamentPlanner.Algorithms.ConstructionAlgorithms.BlockDesigns.BIBDConstruction.↩
RecursiveKTSConstruction.ConstructDesign ( ) [virtual]
```

Construct a KTS.

Returns

Constructed KTS.

Implements TournamentPlanner.Algorithms.ConstructionAlgorithms.BlockDesigns.BIBDConstruction.BaseBIBDConstruction.

The documentation for this class was generated from the following file:

- RecursiveKTSConstruction.cs

7.116 TournamentPlannerTests.Algorithms.ConstructionAlgorithms.↩ BlockDesigns.BIBDConstruction.RecursiveKTSConstructionTests Class Reference

Tests for the RecursiveKTSConstruction.

Public Member Functions

- void **KTS357ConstructionTest** ()
- void RecursiveKTSConstructionTest ()

  Test the construction of KTS of orders 6N + 3.

7.116.1 Detailed Description

Tests for the RecursiveKTSConstruction.

7.116.2 Member Function Documentation

7.116.2.1 RecursiveKTSConstructionTest()

```
void TournamentPlannerTests.Algorithms.ConstructionAlgorithms.BlockDesigns.BIBDConstruction.↩
RecursiveKTSConstructionTests.RecursiveKTSConstructionTest ( )
```

Test the construction of KTS of orders 6N + 3.

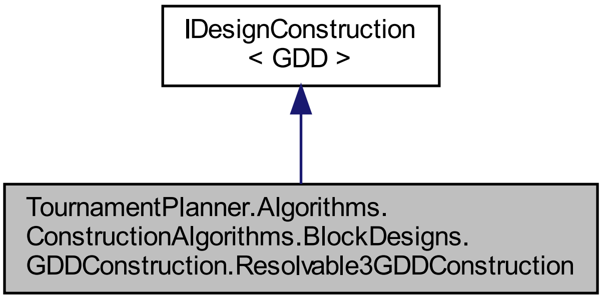
The documentation for this class was generated from the following file:

- RecursiveKTSConstructionTests.cs

7.117 TournamentPlanner.Algorithms.ConstructionAlgorithms.Block↩ Designs.GDDConstruction.Resolvable3GDDConstruction Class Reference

Construct resolvable group divisible designs with block size 3 using MOLS. Should be able to construct GDDs where the order is a multiple of 3 except 6 and 18.

Inheritance diagram for TournamentPlanner.Algorithms.ConstructionAlgorithms.BlockDesigns.GDDConstruction.↩ Resolvable3GDDConstruction:


IDesignConstruction < GDD >
TournamentPlanner.Algorithms. ConstructionAlgorithms.BlockDesigns. GDDConstruction.Resolvable3GDDConstruction

Public Member Functions

- Resolvable3GDDConstruction (int v)

    Initializes a new instance of the Resolvable3GDDConstruction class.
- GDD ConstructDesign ()

    Construct a GDD with block size 3.

7.117.1 Detailed Description

Construct resolvable group divisible designs with block size 3 using MOLS. Should be able to construct GDDs where the order is a multiple of 3 except 6 and 18.

7.117.2 Constructor & Destructor Documentation

7.117.2.1 Resolvable3GDDConstruction()

```
TournamentPlanner.Algorithms.ConstructionAlgorithms.BlockDesigns.GDDConstruction.Resolvable3↩
GDDConstruction.Resolvable3GDDConstruction (
              int v )
```

Initializes a new instance of the Resolvable3GDDConstruction class.

Parameters

v	Number of elements in GDD.

7.117.3 Member Function Documentation

7.117.3.1 ConstructDesign()

```
GDD TournamentPlanner.Algorithms.ConstructionAlgorithms.BlockDesigns.GDDConstruction.Resolvable3↩
GDDConstruction.ConstructDesign ( )
```

Construct a GDD with block size 3.

Returns

Group divisible design.

The documentation for this class was generated from the following file:

- Resolvable3GDDConstruction.cs

7.118 TournamentPlannerTests.Algorithms.ConstructionAlgorithms.↩ BlockDesigns.GDDConstruction.Resolvable3GDDConstruction↩ Tests Class Reference

Tests the creation of resolvable GDDs by Resolvable3GDDConstruction.

Public Member Functions

- void GDDConstructionTest ()

  Test the construction, the subsequent validity and the parallel classes of GDDs of several orders. Valid orders are

7.118.1 Detailed Description

Tests the creation of resolvable GDDs by Resolvable3GDDConstruction.

7.118.2 Member Function Documentation

7.118.2.1 GDDConstructionTest()

```
void TournamentPlannerTests.Algorithms.ConstructionAlgorithms.BlockDesigns.GDDConstruction.↩
Resolvable3GDDConstructionTests.GDDConstructionTest ( )
```

Test the construction, the subsequent validity and the parallel classes of GDDs of several orders. Valid orders are

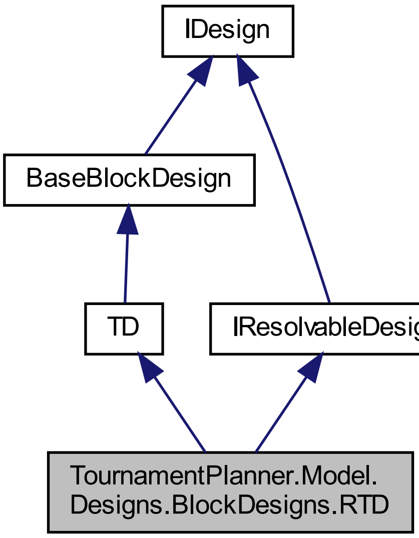
The documentation for this class was generated from the following file:

- Resolvable3GDDConstructionTests.cs

7.119 TournamentPlanner.Model.Designs.BlockDesigns.RTD Class Reference

Resolvable transversal design.

Inheritance diagram for TournamentPlanner.Model.Designs.BlockDesigns.RTD:



Public Member Functions

- RTD (int k, int m, List< List< int >> groups)

    Initializes a new instance of the RTD class.
- void AddParallelClass (List< List< int >> parallelClass)

    Add parallel class and its blocks.

Properties

- List< List< List< int > > > ParallelClasses = new () `[get]`

Additional Inherited Members

7.119.1 Detailed Description

Resolvable transversal design.

7.119.2 Constructor & Destructor Documentation

7.119.2.1 RTD()

```
TournamentPlanner.Model.Designs.BlockDesigns.RTD.RTD (
            int k,
            int m,
            List< List< int >> groups )
```

Initializes a new instance of the RTD class.

Parameters

k	Block sizes.
m	Group sizes.
groups	Groups of the design.

7.119.3 Member Function Documentation

7.119.3.1 AddParallelClass()

```
void TournamentPlanner.Model.Designs.BlockDesigns.RTD.AddParallelClass (
            List< List< int >> parallelClass )
```

Add parallel class and its blocks.

Parameters

parallelClass	The parallel class to add.

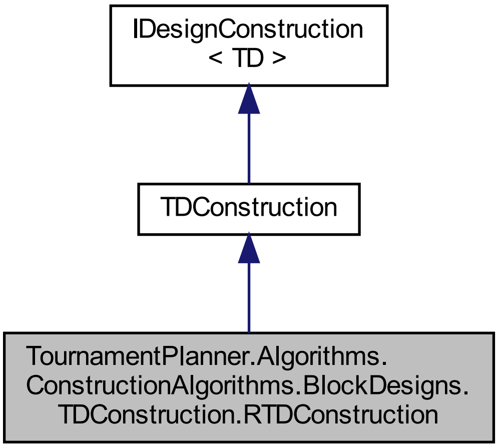
The documentation for this class was generated from the following file:

- RTD.cs

7.120 TournamentPlanner.Algorithms.ConstructionAlgorithms.BlockDesigns.TDConstruction.RTDConstruction Class Reference

Construction method for resolvable transversal designs.

Inheritance diagram for TournamentPlanner.Algorithms.ConstructionAlgorithms.BlockDesigns.TDConstruction.↩
RTDConstruction:


IDesignConstruction
< TD >
TDConstruction
TournamentPlanner.Algorithms.
ConstructionAlgorithms.BlockDesigns.
TDConstruction.RTDConstruction

Public Member Functions

- RTDConstruction (int k, int m)

  Initializes a new instance of the RTDConstruction class.
- new RTD ConstructDesign ()

  Construct the resolvable transversal design.

Additional Inherited Members

7.120.1 Detailed Description

Construction method for resolvable transversal designs.

7.120.2 Constructor & Destructor Documentation

7.120.2.1 RTDConstruction()

```
TournamentPlanner.Algorithms.ConstructionAlgorithms.BlockDesigns.TDConstruction.RTDConstruction.↩
RTDConstruction (
            int k,
            int m )
```

Initializes a new instance of the RTDConstruction class.

Parameters

k	Block sizes.
m	Group sizes.

7.120.3 Member Function Documentation

7.120.3.1 ConstructDesign()

```
new RTD TournamentPlanner.Algorithms.ConstructionAlgorithms.BlockDesigns.TDConstruction.↩
RTDConstruction.ConstructDesign ( )
```

Construct the resolvable transversal design.

Returns

Resolvable transversal design.

The documentation for this class was generated from the following file:

- RTDConstruction.cs

7.121 TournamentPlannerTests.Algorithms.ConstructionAlgorithms.↩ BlockDesigns.TDConstruction.RTDConstructionTests Class Reference

Tests for the RTDConstruction.

Public Member Functions

- void RTDConstructionTest ()

  Test the construction of resolvable TD(18, 19).

7.121.1 Detailed Description

Tests for the RTDConstruction.

7.121.2 Member Function Documentation

7.121.2.1 RTDConstructionTest()

```
void TournamentPlannerTests.Algorithms.ConstructionAlgorithms.BlockDesigns.TDConstruction.↩
RTDConstructionTests.RTDConstructionTest ( )
```

Test the construction of resolvable TD(18, 19).

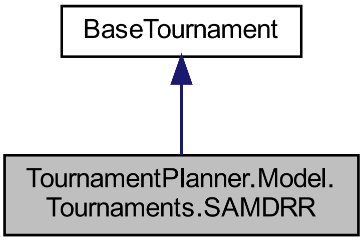
The documentation for this class was generated from the following file:

- RTDConstructionTests.cs

7.122 TournamentPlanner.Model.Tournaments.SAMDRR Class Reference

Spouse-avoiding mixed doubles round robin tournament. Games are represented as 4 integers, (i, l, j, k) - H_i W_l v H_j W_k

Inheritance diagram for TournamentPlanner.Model.Tournaments.SAMDRR:



Public Member Functions

- SAMDRR (int order, List< List< List< int >>> rounds)

  Initializes a new instance of the SAMDRR class.

Additional Inherited Members

7.122.1 Detailed Description

Spouse-avoiding mixed doubles round robin tournament. Games are represented as 4 integers, (i, l, j, k) - H_i W_l v H_j W_k

7.122.2 Constructor & Destructor Documentation

7.122.2.1 SAMDRR()

```
TournamentPlanner.Model.Tournaments.SAMDRR.SAMDRR (
             int order,
             List< List< List< int >>> rounds )
```

Initializes a new instance of the SAMDRR class.

Parameters

order	The order.
rounds	The rounds.

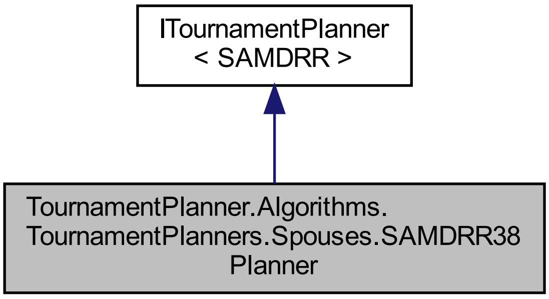
The documentation for this class was generated from the following file:

- SAMDRR.cs

7.123 TournamentPlanner.Algorithms.TournamentPlanners.Spouses.↩ SAMDRR38Planner Class Reference

SAMDRR 38 planner. GCD(38, 6) != 1 so a special construction is required. SAMDRR 38 is used to construct Whist(152).

Inheritance diagram for TournamentPlanner.Algorithms.TournamentPlanners.Spouses.SAMDRR38Planner:


ITournamentPlanner
< SAMDRR >
TournamentPlanner.Algorithms.
TournamentPlanners.Spouses.SAMDRR38
Planner

Public Member Functions

- SAMDRR PlanTournament ()

  Plan a SAMDRR 38.

7.123.1 Detailed Description

SAMDRR 38 planner. GCD(38, 6) != 1 so a special construction is required. SAMDRR 38 is used to construct Whist(152).

7.123.2 Member Function Documentation

7.123.2.1 PlanTournament()

```
SAMDRR TournamentPlanner.Algorithms.TournamentPlanners.Spouses.SAMDRR38Planner.PlanTournament
( )
```

Plan a SAMDRR 38.

Returns

SAMDRR 38.

The documentation for this class was generated from the following file:

- SAMDRR38Planner.cs

7.124 TournamentPlannerTests.Algorithms.TournamentPlanners.↩ Spouses.SAMDRR38PlannerTests Class Reference

Tests for the SAMDRR38Planner.

Public Member Functions

- void **SAMDRR38ConstructionTest** ()

7.124.1 Detailed Description

Tests for the SAMDRR38Planner.

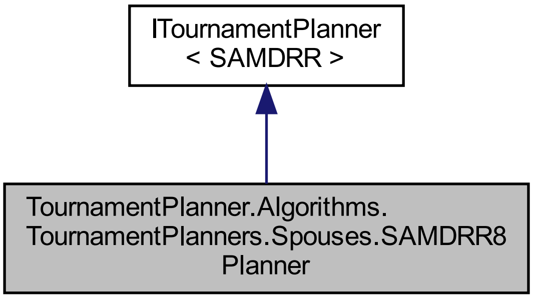
The documentation for this class was generated from the following file:

- SAMDRR38PlannerTests.cs

7.125 TournamentPlanner.Algorithms.TournamentPlanners.Spouses.↩ SAMDRR8Planner Class Reference

SAMDRR 8 construction class.

Inheritance diagram for TournamentPlanner.Algorithms.TournamentPlanners.Spouses.SAMDRR8Planner:


ITournamentPlanner
< SAMDRR >
TournamentPlanner.Algorithms.
TournamentPlanners.Spouses.SAMDRR8
Planner

Public Member Functions

- SAMDRR PlanTournament ()

  Plan SAMDRR 8.

7.125.1 Detailed Description

SAMDRR 8 construction class.

7.125.2 Member Function Documentation

7.125.2.1 PlanTournament()

```
SAMDRR TournamentPlanner.Algorithms.TournamentPlanners.Spouses.SAMDRR8Planner.PlanTournament (
)
```

Plan SAMDRR 8.

Returns

SAMDRR 8.

The documentation for this class was generated from the following file:

- SAMDRR8Planner.cs

7.126 TournamentPlannerTests.Algorithms.TournamentPlanners.↩ Spouses.SAMDRR8PlannerTests Class Reference

Tests for the SAMDRR8Planner.

Public Member Functions

- void **SAMDRR8ConstructionTest** ()

7.126.1 Detailed Description

Tests for the SAMDRR8Planner.

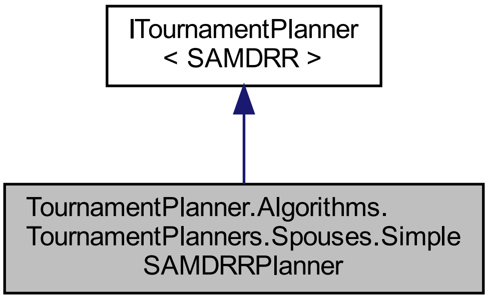
The documentation for this class was generated from the following file:

- SAMDRR8PlannerTests.cs

7.127 TournamentPlanner.Algorithms.TournamentPlanners.Spouses.↩ SimpleSAMDRRPlanner Class Reference

SAMDRR planner using cyclic SOLS such that for the SOLS order n, GCD(n, 6) = 1.

Inheritance diagram for TournamentPlanner.Algorithms.TournamentPlanners.Spouses.SimpleSAMDRRPlanner:


ITournamentPlanner
< SAMDRR >
TournamentPlanner.Algorithms.
TournamentPlanners.Spouses.Simple
SAMDRRPlanner

Public Member Functions

- SimpleSAMDRRPlanner (int order)

  Initializes a new instance of the SimpleSAMDRRPlanner class.
- SAMDRR PlanTournament ()

  Plan a SAMDRR.

7.127.1 Detailed Description

SAMDRR planner using cyclic SOLS such that for the SOLS order n, GCD(n, 6) = 1.

7.127.2 Constructor & Destructor Documentation

7.127.2.1 SimpleSAMDRRPlanner()

```
TournamentPlanner.Algorithms.TournamentPlanners.Spouses.SimpleSAMDRRPlanner.SimpleSAMDRRPlanner
(
             int order )
```

Initializes a new instance of the SimpleSAMDRRPlanner class.

Parameters

order	The order.

7.127.3 Member Function Documentation

7.127.3.1 PlanTournament()

```
SAMDRR TournamentPlanner.Algorithms.TournamentPlanners.Spouses.SimpleSAMDRRPlanner.Plan↩
Tournament ( )
```

Plan a SAMDRR.

Returns

The scheduled SAMDRR.

The documentation for this class was generated from the following file:

- SimpleSAMDRRPlanner.cs

7.128 TournamentPlannerTests.Algorithms.TournamentPlanners.↩ Spouses.SimpleSAMDRRPlannerTests Class Reference

Tests for the SimpleSAMDRRPlanner.

Public Member Functions

- void TestSAMDRR11Construction ()

  Test the construction of SAMDRR 11.

7.128.1 Detailed Description

Tests for the SimpleSAMDRRPlanner.

7.128.2 Member Function Documentation

7.128.2.1 TestSAMDRR11Construction()

```
void TournamentPlannerTests.Algorithms.TournamentPlanners.Spouses.SimpleSAMDRRPlannerTests.↩
TestSAMDRR11Construction ( )
```

Test the construction of SAMDRR 11.

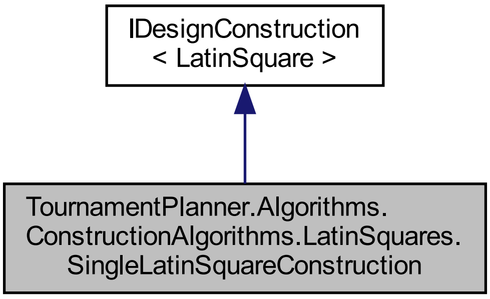
The documentation for this class was generated from the following file:

- SimpleSAMDRRPlannerTests.cs

7.129 TournamentPlanner.Algorithms.ConstructionAlgorithms.Latin↩ Squares.SingleLatinSquareConstruction Class Reference

Recursive construction of a single Latin square of any order.

Inheritance diagram for TournamentPlanner.Algorithms.ConstructionAlgorithms.LatinSquares.SingleLatinSquare↩ Construction:


IDesignConstruction
< LatinSquare >
TournamentPlanner.Algorithms.
ConstructionAlgorithms.LatinSquares.
SingleLatinSquareConstruction

Public Member Functions

- SingleLatinSquareConstruction (int order)

  Initializes a new instance of the SingleLatinSquareConstruction class.
- LatinSquare ConstructDesign ()

  Construct the Latin square.

7.129.1 Detailed Description

Recursive construction of a single Latin square of any order.

7.129.2 Constructor & Destructor Documentation

7.129.2.1 SingleLatinSquareConstruction()

```
TournamentPlanner.Algorithms.ConstructionAlgorithms.LatinSquares.SingleLatinSquareConstruction.↩
SingleLatinSquareConstruction (
            int order )
```

Initializes a new instance of the SingleLatinSquareConstruction class.

Parameters

order	The LS order.

7.129.3 Member Function Documentation

7.129.3.1 ConstructDesign()

```
LatinSquare TournamentPlanner.Algorithms.ConstructionAlgorithms.LatinSquares.SingleLatin↩
SquareConstruction.ConstructDesign ( )
```

Construct the Latin square.

Returns

Constructed Latin square.

The documentation for this class was generated from the following file:

- SingleLatinSquareConstruction.cs

7.130 TournamentPlannerTests.Algorithms.ConstructionAlgorithms.↩ LatinSquares.SingleLatinSquareConstructionTests Class Reference

Tests for the SingleLatinSquareConstruction.

Public Member Functions

- void SingleLatinSquareOfOrder10Test ()

  Test the creation and validate a Latin square of order 10.
- void SeveralOrdersTest ()

  Test Latin squares or orders 2 to 10.

7.130.1 Detailed Description

Tests for the SingleLatinSquareConstruction.

7.130.2 Member Function Documentation

7.130.2.1 SeveralOrdersTest()

```
void TournamentPlannerTests.Algorithms.ConstructionAlgorithms.LatinSquares.SingleLatinSquare↩
ConstructionTests.SeveralOrdersTest ( )
```

Test Latin squares or orders 2 to 10.

7.130.2.2 SingleLatinSquareOfOrder10Test()

```
void TournamentPlannerTests.Algorithms.ConstructionAlgorithms.LatinSquares.SingleLatinSquare↩
ConstructionTests.SingleLatinSquareOfOrder10Test ( )
```

Test the creation and validate a Latin square of order 10.

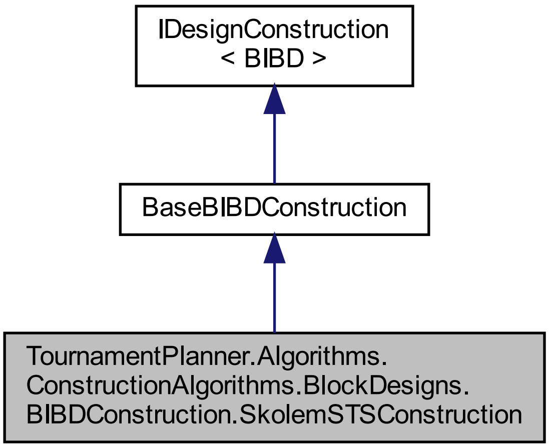
The documentation for this class was generated from the following file:

- SingleLatinSquareConstructionTests.cs

7.131 TournamentPlanner.Algorithms.ConstructionAlgorithms.Block↩ Designs.BIBDConstruction.SkolemSTSConstruction Class Reference

Skolem construction algorithm for STS of size 6N + 1. http://math.ucdenver.edu/∼wcherowi/courses/m6406/sts.pdf

Inheritance diagram for TournamentPlanner.Algorithms.ConstructionAlgorithms.BlockDesigns.BIBDConstruction.↩
SkolemSTSConstruction:


IDesignConstruction
< BIBD >
BaseBIBDConstruction
TournamentPlanner.Algorithms.
ConstructionAlgorithms.BlockDesigns.
BIBDConstruction.SkolemSTSConstruction

Public Member Functions

- SkolemSTSConstruction (int v)

  Initializes a new instance of the SkolemSTSConstruction class.
- override STS ConstructDesign ()

  Construct STS of size 6N + 1.

Additional Inherited Members

7.131.1 Detailed Description

Skolem construction algorithm for STS of size 6N + 1. http://math.ucdenver.edu/∼wcherowi/courses/m6406/sts.pdf

7.131.2 Constructor & Destructor Documentation

7.131.2.1 SkolemSTSConstruction()

```
TournamentPlanner.Algorithms.ConstructionAlgorithms.BlockDesigns.BIBDConstruction.Skolem↩
STSConstruction.SkolemSTSConstruction (
            int v )
```

Initializes a new instance of the SkolemSTSConstruction class.

Parameters

v	Set size.

7.131.3 Member Function Documentation

7.131.3.1 ConstructDesign()

```
override STS TournamentPlanner.Algorithms.ConstructionAlgorithms.BlockDesigns.BIBDConstruction.↩
SkolemSTSConstruction.ConstructDesign ( ) [virtual]
```

Construct STS of size 6N + 1.

Returns

The constructed STS.

Implements TournamentPlanner.Algorithms.ConstructionAlgorithms.BlockDesigns.BIBDConstruction.BaseBIBDConstruction.

The documentation for this class was generated from the following file:

- SkolemSTSConstruction.cs

7.132 TournamentPlannerTests.Algorithms.ConstructionAlgorithms.↩ BlockDesigns.BIBDConstruction.SkolemSTSConstructionTests Class Reference

Tests for the SkolemSTSConstruction.

Public Member Functions

- void SkolemSTSConstructionTest ()

  Test the creation of a few STS of order 6N + 1.
- void SkolemSTSExceptionTest ()

  Test the exception if the order is not valid.

7.132.1 Detailed Description

Tests for the SkolemSTSConstruction.

7.132.2 Member Function Documentation

7.132.2.1 SkolemSTSConstructionTest()

```
void TournamentPlannerTests.Algorithms.ConstructionAlgorithms.BlockDesigns.BIBDConstruction.↩
SkolemSTSConstructionTests.SkolemSTSConstructionTest ( )
```

Test the creation of a few STS of order 6N + 1.

7.132.2.2 SkolemSTSExceptionTest()

```
void TournamentPlannerTests.Algorithms.ConstructionAlgorithms.BlockDesigns.BIBDConstruction.↩
SkolemSTSConstructionTests.SkolemSTSExceptionTest ( )
```

Test the exception if the order is not valid.

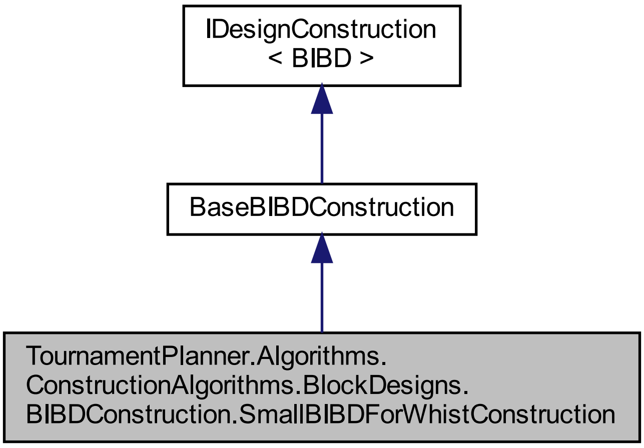
The documentation for this class was generated from the following file:

- SkolemSTSConstructionTests.cs

7.133 TournamentPlanner.Algorithms.ConstructionAlgorithms.Block↩ Designs.BIBDConstruction.SmallBIBDForWhistConstruction Class Reference

Static construction of BIBD(21, 5, 1) and BIBD(25, 5, 1). These designs have been constructed by the incremental / quadratic methods. Those methods are, however, unreliable (license and memory issues) so the constructed designs are stored here for faster and more reliable access.

Inheritance diagram for TournamentPlanner.Algorithms.ConstructionAlgorithms.BlockDesigns.BIBDConstruction.↩ SmallBIBDForWhistConstruction:


IDesignConstruction < BIBD >
BaseBIBDConstruction
TournamentPlanner.Algorithms. ConstructionAlgorithms.BlockDesigns. BIBDConstruction.SmallBIBDForWhistConstruction

Public Member Functions

- SmallBIBDForWhistConstruction (int v)

  Initializes a new instance of the SmallBIBDForWhistConstruction class.
- override BIBD ConstructDesign ()

  Construct the BIBD.

Additional Inherited Members

7.133.1 Detailed Description

Static construction of BIBD(21, 5, 1) and BIBD(25, 5, 1). These designs have been constructed by the incremental / quadratic methods. Those methods are, however, unreliable (license and memory issues) so the constructed designs are stored here for faster and more reliable access.

7.133.2 Constructor & Destructor Documentation

7.133.2.1 SmallBIBDForWhistConstruction()

```
TournamentPlanner.Algorithms.ConstructionAlgorithms.BlockDesigns.BIBDConstruction.Small↩
BIBDForWhistConstruction.SmallBIBDForWhistConstruction (
            int v )
```

Initializes a new instance of the SmallBIBDForWhistConstruction class.

Parameters

v	BIBD order.

The documentation for this class was generated from the following file:

- SmallBIBDForWhistConstruction.cs

7.134 TournamentPlannerTests.Algorithms.ConstructionAlgorithms.↩ BlockDesigns.BIBDConstruction.SmallBIBDForWhist↩ ConstructionTests Class Reference

Tests for SmallBIBDForWhistConstruction.

Public Member Functions

- void **BIBDConstructionTest** ()

7.134.1 Detailed Description

Tests for SmallBIBDForWhistConstruction.

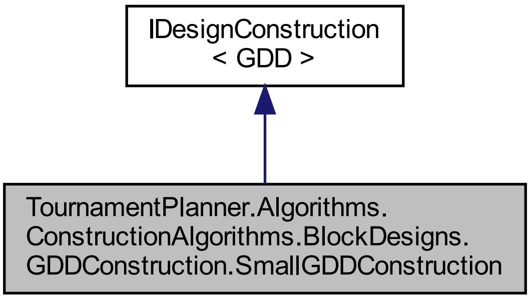
The documentation for this class was generated from the following file:

- SmallBIBDForWhistConstructionTests.cs

7.135 TournamentPlanner.Algorithms.ConstructionAlgorithms.Block↩ Designs.GDDConstruction.SmallGDDConstruction Class Reference

Construct a few small order GDDs which are then used i Wilsons fundamental construction.

Inheritance diagram for TournamentPlanner.Algorithms.ConstructionAlgorithms.BlockDesigns.GDDConstruction.↩ SmallGDDConstruction:


IDesignConstruction
< GDD >
TournamentPlanner.Algorithms.
ConstructionAlgorithms.BlockDesigns.
GDDConstruction.SmallGDDConstruction

Public Member Functions

- SmallGDDConstruction (int v)

  Initializes a new instance of the SmallGDDConstruction class.
- GDD ConstructDesign ()

  Construct a GDD of small order.

7.135.1 Detailed Description

Construct a few small order GDDs which are then used i Wilsons fundamental construction.

7.135.2 Constructor & Destructor Documentation

7.135.2.1 SmallGDDConstruction()

```
TournamentPlanner.Algorithms.ConstructionAlgorithms.BlockDesigns.GDDConstruction.SmallGDDConstruction.↩
SmallGDDConstruction (
            int v )
```

Initializes a new instance of the SmallGDDConstruction class.

Parameters

v	Size of GDD.

7.135.3 Member Function Documentation

7.135.3.1 ConstructDesign()

```
GDD TournamentPlanner.Algorithms.ConstructionAlgorithms.BlockDesigns.GDDConstruction.Small↩
GDDConstruction.ConstructDesign ( )
```

Construct a GDD of small order.

Returns

Constructed GDD.

The documentation for this class was generated from the following file:

- SmallGDDConstruction.cs

7.136 TournamentPlannerTests.Algorithms.ConstructionAlgorithms.↩ BlockDesigns.GDDConstruction.SmallGDDConstructionTests Class Reference

Tests for the SmallGDDConstruction.

Public Member Functions

- void TestSmallGDDConstructionForAvailableOrders ()
  Test the small GDD construction for all the available orders.

7.136.1 Detailed Description

Tests for the SmallGDDConstruction.

7.136.2 Member Function Documentation

7.136.2.1 TestSmallGDDConstructionForAvailableOrders()

```
void TournamentPlannerTests.Algorithms.ConstructionAlgorithms.BlockDesigns.GDDConstruction.↩
SmallGDDConstructionTests.TestSmallGDDConstructionForAvailableOrders ( )
```

Test the small GDD construction for all the available orders.

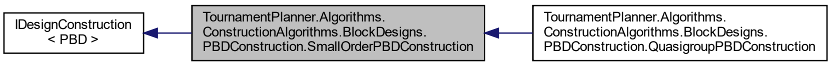
The documentation for this class was generated from the following file:

- SmallGDDConstructionTests.cs

7.137 TournamentPlanner.Algorithms.ConstructionAlgorithms.Block↩ Designs.PBDConstruction.SmallOrderPBDConstruction Class Reference

A class facilitating the construction of PBDs of smaller orders in form 3N + 1. All the constructed design have block sizes in {4, 7, 10, 19}. There are several construction methods, not a single unified one.

Inheritance diagram for TournamentPlanner.Algorithms.ConstructionAlgorithms.BlockDesigns.PBDConstruction.↩
SmallOrderPBDConstruction:


IDesignConstruction
< PBD >
TournamentPlanner.Algorithms.
ConstructionAlgorithms.BlockDesigns.
PBDConstruction.SmallOrderPBDConstruction
TournamentPlanner.Algorithms.
ConstructionAlgorithms.BlockDesigns.
PBDConstruction.QuasigroupPBDConstruction

Public Member Functions

- SmallOrderPBDConstruction (int v)

  Initializes a new instance of the SmallOrderPBDConstruction class.
- PBD ConstructDesign ()

  Construct a PBD.

Properties

- int V `[get]`

  Gets the order of the PBD to construct V = 3N + 1.

7.137.1 Detailed Description

A class facilitating the construction of PBDs of smaller orders in form 3N + 1. All the constructed design have block sizes in {4, 7, 10, 19}. There are several construction methods, not a single unified one.

7.137.2 Constructor & Destructor Documentation

7.137.2.1 SmallOrderPBDConstruction()

```
TournamentPlanner.Algorithms.ConstructionAlgorithms.BlockDesigns.PBDConstruction.SmallOrder↩
PBDConstruction.SmallOrderPBDConstruction (
             int v )
```

Initializes a new instance of the SmallOrderPBDConstruction class.

Parameters

v	The order of the PBD.

7.137.3 Member Function Documentation

7.137.3.1 ConstructDesign()

```
PBD TournamentPlanner.Algorithms.ConstructionAlgorithms.BlockDesigns.PBDConstruction.Small↩
OrderPBDConstruction.ConstructDesign ( )
```

Construct a PBD.

Returns

Constructed PBD.

7.137.4 Property Documentation

7.137.4.1 V

```
int TournamentPlanner.Algorithms.ConstructionAlgorithms.BlockDesigns.PBDConstruction.Small↩
OrderPBDConstruction.V  [get], [protected]
```

Gets the order of the PBD to construct V = 3N + 1.

The documentation for this class was generated from the following file:

- SmallOrderPBDConstruction.cs

7.138 TournamentPlannerTests.Algorithms.ConstructionAlgorithms.↩ BlockDesigns.PBDConstruction.SmallOrderPBDConstruction↩ Tests Class Reference

Tests for the SmallOrderPBDConstruction.

Public Member Functions

- void SmallPBDUpToOrder46ConstructionTest ()

  Test the construction of PBDs up to the order 46.
- void PBD79And82ConstructionTest ()

  Tests the PBD(79) and PBD(82) construction.

7.138.1 Detailed Description

Tests for the SmallOrderPBDConstruction.

7.138.2 Member Function Documentation

7.138.2.1 PBD79And82ConstructionTest()

```
void TournamentPlannerTests.Algorithms.ConstructionAlgorithms.BlockDesigns.PBDConstruction.↩
SmallOrderPBDConstructionTests.PBD79And82ConstructionTest ( )
```

Tests the PBD(79) and PBD(82) construction.

7.138.2.2 SmallPBDUpToOrder46ConstructionTest()

```
void TournamentPlannerTests.Algorithms.ConstructionAlgorithms.BlockDesigns.PBDConstruction.↩
SmallOrderPBDConstructionTests.SmallPBDUpToOrder46ConstructionTest ( )
```

Test the construction of PBDs up to the order 46.

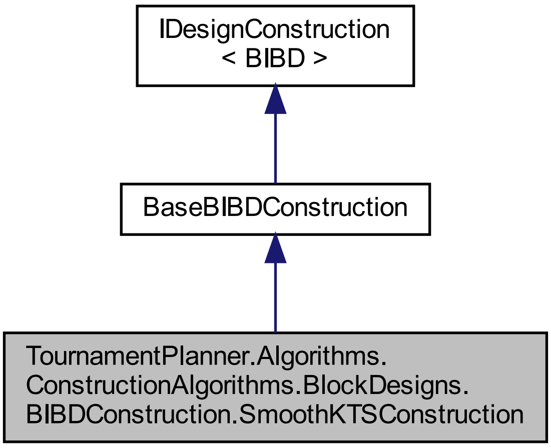
The documentation for this class was generated from the following file:

- SmallOrderPBDConstructionTests.cs

7.139 TournamentPlanner.Algorithms.ConstructionAlgorithms.BlockDesigns.BIBDConstruction.SmoothKTSConstruction Class Reference

Algorithm for construction of smooth Kirkman triple systems. The size of smooth KTS is a power of 3. https://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.501.8928&rep=rep1&type=pdf

Inheritance diagram for TournamentPlanner.Algorithms.ConstructionAlgorithms.BlockDesigns.BIBDConstruction.SmoothKTSConstruction:


IDesignConstruction
< BIBD >
BaseBIBDConstruction
TournamentPlanner.Algorithms.
ConstructionAlgorithms.BlockDesigns.
BIBDConstruction.SmoothKTSConstruction

Public Member Functions

- SmoothKTSConstruction (int v)

  Initializes a new instance of the SmoothKTSConstruction class.
- override STS ConstructDesign ()

  Construct a smooth KTS where the size is a power of 3.

Additional Inherited Members

7.139.1 Detailed Description

Algorithm for construction of smooth Kirkman triple systems. The size of smooth KTS is a power of 3. https://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.501.8928&rep=rep1&type=pdf

7.139.2 Constructor & Destructor Documentation

7.139.2.1 SmoothKTSConstruction()

```
TournamentPlanner.Algorithms.ConstructionAlgorithms.BlockDesigns.BIBDConstruction.Smooth↩
KTSConstruction.SmoothKTSConstruction (
            int v )
```

Initializes a new instance of the SmoothKTSConstruction class.

Parameters

v	Set size.

7.139.3 Member Function Documentation

7.139.3.1 ConstructDesign()

```
override STS TournamentPlanner.Algorithms.ConstructionAlgorithms.BlockDesigns.BIBDConstruction.↩
SmoothKTSConstruction.ConstructDesign ( )  [virtual]
```

Construct a smooth KTS where the size is a power of 3.

Returns

Smooth KTS.

Implements TournamentPlanner.Algorithms.ConstructionAlgorithms.BlockDesigns.BIBDConstruction.BaseBIBDConstruction.

The documentation for this class was generated from the following file:

- SmoothKTSConstruction.cs

7.140 TournamentPlannerTests.Algorithms.ConstructionAlgorithms.↩ BlockDesigns.BIBDConstruction.SmoothKTSConstructionTests Class Reference

Tests for the SmoothKTSConstruction.

Public Member Functions

- void SmoothKTSConstructionTest ()

  Tests the construction of KTS where the order is a power of 3.

7.140.1 Detailed Description

Tests for the SmoothKTSConstruction.

7.140.2 Member Function Documentation

7.140.2.1 SmoothKTSConstructionTest()

```
void TournamentPlannerTests.Algorithms.ConstructionAlgorithms.BlockDesigns.BIBDConstruction.↩
SmoothKTSConstructionTests.SmoothKTSConstructionTest ( )
```

Tests the construction of KTS where the order is a power of 3.

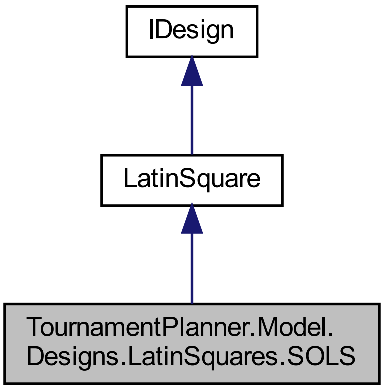
The documentation for this class was generated from the following file:

- SmoothKTSConstructionTests.cs

7.141 TournamentPlanner.Model.Designs.LatinSquares.SOLS Class Reference

Self orthogonal Latin square.

Inheritance diagram for TournamentPlanner.Model.Designs.LatinSquares.SOLS:



Public Member Functions

- SOLS (int v, int[,] ls)

  Initializes a new instance of the SOLS class.
- int[,] Transpose ()

  Get the transpose of the matrix.

Additional Inherited Members

7.141.1 Detailed Description

Self orthogonal Latin square.

7.141.2 Constructor & Destructor Documentation

7.141.2.1 SOLS()

```
TournamentPlanner.Model.Designs.LatinSquares.SOLS.SOLS (
            int v,
            int ls[,] )
```

Initializes a new instance of the SOLS class.

Parameters

v	The order.
ls	The Latin square matrix.

7.141.3 Member Function Documentation

7.141.3.1 Transpose()

```
int [,] TournamentPlanner.Model.Designs.LatinSquares.SOLS.Transpose ( )
```

Get the transpose of the matrix.

Returns

The transpose.

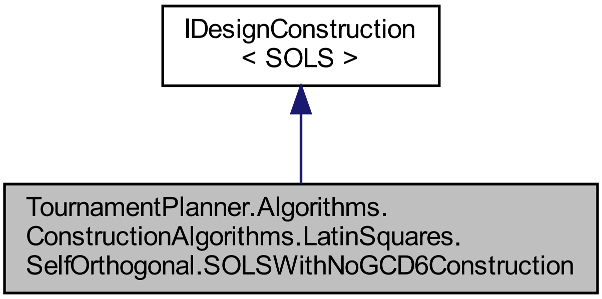
The documentation for this class was generated from the following file:

- SOLS.cs

7.142 TournamentPlanner.Algorithms.ConstructionAlgorithms.Latin↩ Squares.SelfOrthogonal.SOLSWithNoGCD6Construction Class Reference

Construct a cyclic SOLS such that GCD(n,6) = 1.

Inheritance diagram for TournamentPlanner.Algorithms.ConstructionAlgorithms.LatinSquares.SelfOrthogonal.↩ SOLSWithNoGCD6Construction:


IDesignConstruction
< SOLS >
TournamentPlanner.Algorithms.
ConstructionAlgorithms.LatinSquares.
SelfOrthogonal.SOLSWithNoGCD6Construction

Public Member Functions

- SOLSWithNoGCD6Construction (int order)

  Initializes a new instance of the SOLSWithNoGCD6Construction class.
- SOLS ConstructDesign ()

  Construct a SOLS.

7.142.1 Detailed Description

Construct a cyclic SOLS such that GCD(n,6) = 1.

7.142.2 Constructor & Destructor Documentation

7.142.2.1 SOLSWithNoGCD6Construction()

```
TournamentPlanner.Algorithms.ConstructionAlgorithms.LatinSquares.SelfOrthogonal.SOLSWithNo↩
GCD6Construction.SOLSWithNoGCD6Construction (
                int order )
```

Initializes a new instance of the SOLSWithNoGCD6Construction class.

Parameters

order	Order of the SOLS.

7.142.3 Member Function Documentation

7.142.3.1 ConstructDesign()

```
SOLS TournamentPlanner.Algorithms.ConstructionAlgorithms.LatinSquares.SelfOrthogonal.SOLSWith↩
NoGCD6Construction.ConstructDesign ( )
```

Construct a SOLS.

Returns

Constructed SOLS.

The documentation for this class was generated from the following file:

- SOLSWithNoGCD6Construction.cs

7.143 TournamentPlannerTests.Algorithms.ConstructionAlgorithms.↩ LatinSquares.SelfOrthogonal.SOLSWithNoGCD6Construction↩ Tests Class Reference

Tests for SOLSWithNoGCD6Construction.

Public Member Functions

- void SOLSConstructionTest ()
  Test on the orders required for

7.143.1 Detailed Description

Tests for SOLSWithNoGCD6Construction.

7.143.2 Member Function Documentation

7.143.2.1 SOLSConstructionTest()

```
void TournamentPlannerTests.Algorithms.ConstructionAlgorithms.LatinSquares.SelfOrthogonal.↩
SOLSWithNoGCD6ConstructionTests.SOLSConstructionTest ( )
```

Test on the orders required for

The documentation for this class was generated from the following file:

- SOLSWithNoGCD6ConstructionTests.cs

7.144 TournamentPlanner.Startup Class Reference

Public Member Functions

- **Startup** (IConfiguration configuration)
- void **ConfigureServices** (IServiceCollection services)
- void **Configure** (IApplicationBuilder app, IWebHostEnvironment env)

Properties

- IConfiguration **Configuration** `[get]`

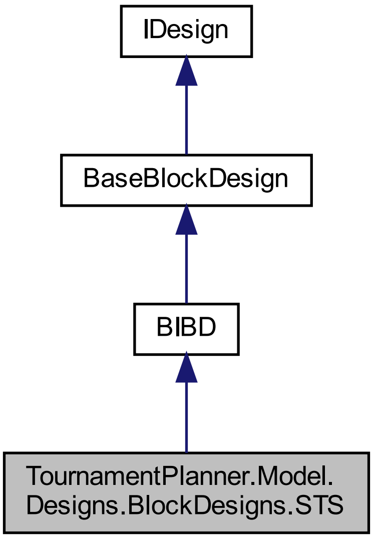
The documentation for this class was generated from the following file:

- Startup.cs

7.145 TournamentPlanner.Model.Designs.BlockDesigns.STS Class Reference

Class representing a Steiner triple system. A Steiner triple system is a BIBD with K = 3 and Lambda = 1.

Inheritance diagram for TournamentPlanner.Model.Designs.BlockDesigns.STS:



Public Member Functions

- STS (int v)

  Initializes a new instance of the STS class.
- STS (int v, List< List< int >> blocks)

  Initializes a new instance of the STS class.

Additional Inherited Members

7.145.1 Detailed Description

Class representing a Steiner triple system. A Steiner triple system is a BIBD with K = 3 and Lambda = 1.

7.145.2 Constructor & Destructor Documentation

7.145.2.1 STS() [1/2]

```
TournamentPlanner.Model.Designs.BlockDesigns.STS.STS (
            int v )
```

Initializes a new instance of the STS class.

Parameters

v	Set size.

7.145.2.2 STS() [2/2]

```
TournamentPlanner.Model.Designs.BlockDesigns.STS.STS (
            int v,
            List< List< int >> blocks )
```

Initializes a new instance of the STS class.

Parameters

v	Set size.
blocks	The blocks.

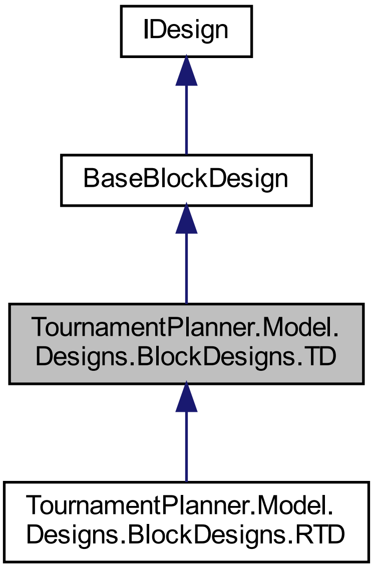
The documentation for this class was generated from the following file:

- STS.cs

7.146 TournamentPlanner.Model.Designs.BlockDesigns.TD Class Reference

Class representing the transversal design.

Inheritance diagram for TournamentPlanner.Model.Designs.BlockDesigns.TD:


IDesign
BaseBlockDesign
TournamentPlanner.Model.Designs.BlockDesigns.TD
TournamentPlanner.Model.Designs.BlockDesigns.RTD

Public Member Functions

- TD (int k, int m)

  Initializes a new instance of the TD class.
- TD (int k, int m, List< List< int >> groups, List< List< int >> blocks)

  Initializes a new instance of the TD class.

Properties

- int K `[get]`

  Gets the block size.
- int M `[get]`

  Gets the group size.
- List< List< int > > Groups `[get, protected set]`

  Gets or sets the groups partitioning the elements. Should be of size M.

Additional Inherited Members

7.146.1 Detailed Description

Class representing the transversal design.

7.146.2 Constructor & Destructor Documentation

7.146.2.1 TD() [1/2]

```
TournamentPlanner.Model.Designs.BlockDesigns.TD.TD (
             int k,
             int m )
```

Initializes a new instance of the TD class.

Parameters

k	Block sizes.
m	Group sizes.

7.146.2.2 TD() [2/2]

```
TournamentPlanner.Model.Designs.BlockDesigns.TD.TD (
             int k,
             int m,
             List< List< int >> groups,
             List< List< int >> blocks )
```

Initializes a new instance of the TD class.

Parameters

k	Block sizes.
m	Group sizes.
groups	The groups.
blocks	The blocks.

7.146.3 Property Documentation

7.146.3.1 Groups

```
List<List<int> > TournamentPlanner.Model.Designs.BlockDesigns.TD.Groups  [get], [protected
set]
```

Gets or sets the groups partitioning the elements. Should be of size M.

7.146.3.2 K

```
int TournamentPlanner.Model.Designs.BlockDesigns.TD.K  [get]
```

Gets the block size.

7.146.3.3 M

```
int TournamentPlanner.Model.Designs.BlockDesigns.TD.M  [get]
```

Gets the group size.

The documentation for this class was generated from the following file:

- TD.cs

7.147 TournamentPlannerTests.Algorithms.ConstructionAlgorithms.↩ BlockDesigns.TDConstruction.TD17ConstructionTests Class Reference

Tests for the TD17Construction.

Public Member Functions

- void **TestTD17_19Construction** ()

7.147.1 Detailed Description

Tests for the TD17Construction.

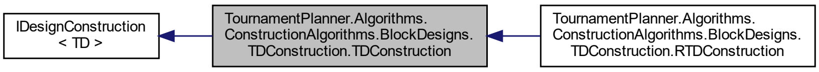
The documentation for this class was generated from the following file:

- TD17ConstructionTests.cs

7.148 TournamentPlanner.Algorithms.ConstructionAlgorithms.Block↩ Designs.TDConstruction.TDConstruction Class Reference

Construction of TD(k, g) that are then used to construct Whist tournaments.

Inheritance diagram for TournamentPlanner.Algorithms.ConstructionAlgorithms.BlockDesigns.TDConstruction.↩ TDConstruction:


IDesignConstruction
< TD >
TournamentPlanner.Algorithms.
ConstructionAlgorithms.BlockDesigns.
TDConstruction.TDConstruction
TournamentPlanner.Algorithms.
ConstructionAlgorithms.BlockDesigns.
TDConstruction.RTDConstruction

Public Member Functions

- TDConstruction (int k, int m)

  Initializes a new instance of the TDConstruction class.
- TD ConstructDesign ()

  Construct a TD(17, g).

Protected Attributes

- int k

  Block sizes of the TD.
- int m

  Group sizes of the TD.

7.148.1 Detailed Description

Construction of TD(k, g) that are then used to construct Whist tournaments.

7.148.2 Constructor & Destructor Documentation

7.148.2.1 TDConstruction()

```
TournamentPlanner.Algorithms.ConstructionAlgorithms.BlockDesigns.TDConstruction.TDConstruction.↩
TDConstruction (
             int k,
             int m )
```

Initializes a new instance of the TDConstruction class.

Parameters

k	Size of the blocks.
m	Size of the groups.

7.148.3 Member Function Documentation

7.148.3.1 ConstructDesign()

```
TD TournamentPlanner.Algorithms.ConstructionAlgorithms.BlockDesigns.TDConstruction.TDConstruction.↩
ConstructDesign ( )
```

Construct a TD(17, g).

Returns

Constructed TD.

7.148.4 Member Data Documentation

7.148.4.1 k

```
int TournamentPlanner.Algorithms.ConstructionAlgorithms.BlockDesigns.TDConstruction.TDConstruction.↩
k  [protected]
```

Block sizes of the TD.

7.148.4.2 m

```
int TournamentPlanner.Algorithms.ConstructionAlgorithms.BlockDesigns.TDConstruction.TDConstruction.↩
m  [protected]
```

Group sizes of the TD.

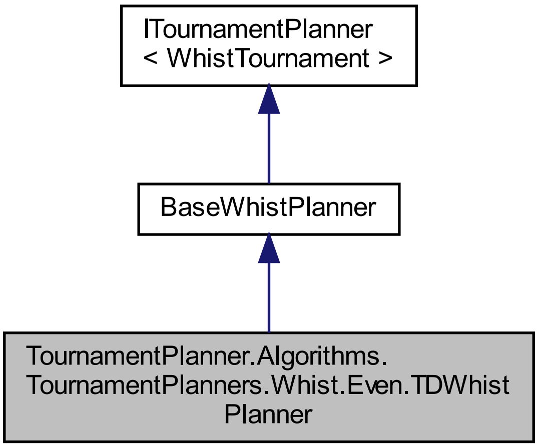
The documentation for this class was generated from the following file:

- TDConstruction.cs

7.149 TournamentPlanner.Algorithms.TournamentPlanners.Whist.↩ Even.TDWhistPlanner Class Reference

Same method as in the larger whist construction but with TD (which is also a GDD basically). TODO unite.

Inheritance diagram for TournamentPlanner.Algorithms.TournamentPlanners.Whist.Even.TDWhistPlanner:


ITournamentPlanner
< WhistTournament >
BaseWhistPlanner
TournamentPlanner.Algorithms.
TournamentPlanners.Whist.Even.TDWhist
Planner

Public Member Functions

- TDWhistPlanner (int v)

  Initializes a new instance of the TDWhistPlanner class.
- override WhistTournament PlanTournament ()

  Plan the Whist tournament.

Additional Inherited Members

7.149.1 Detailed Description

Same method as in the larger whist construction but with TD (which is also a GDD basically). TODO unite.

7.149.2 Constructor & Destructor Documentation

7.149.2.1 TDWhistPlanner()

```
TournamentPlanner.Algorithms.TournamentPlanners.Whist.Even.TDWhistPlanner.TDWhistPlanner (
             int v )
```

Initializes a new instance of the TDWhistPlanner class.

Parameters

v	The order.

The documentation for this class was generated from the following file:

- TDWhistPlanner.cs

7.150 TournamentPlannerTests.Algorithms.TournamentPlanners.↩ Whist.Even.TDWhistPlannerTests Class Reference

Tests for the TDWhistPlanner.

Public Member Functions

- void **TestWhist56Construction** ()

7.150.1 Detailed Description

Tests for the TDWhistPlanner.

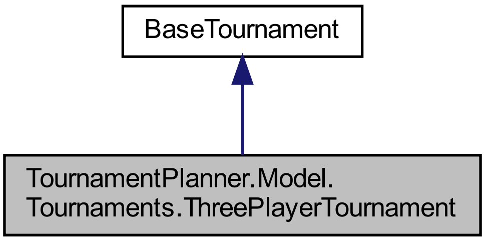
The documentation for this class was generated from the following file:

- TDWhistPlannerTests.cs

7.151 TournamentPlanner.Model.Tournaments.ThreePlayerTournament Class Reference

Class representing a three player tournament.

Inheritance diagram for TournamentPlanner.Model.Tournaments.ThreePlayerTournament:



Public Member Functions

- ThreePlayerTournament (int nbPlayers, List< List< List< int >>> gamesByRounds, List< List< int >> groups=null)

  Initializes a new instance of the ThreePlayerTournament class.
- int TotalNumberOfGames ()

  Get the total number of games in the tournament.

Properties

- List< List< int > > PlayerGroups `[get]`

  Gets the player groups. Player that are together in a group won't play with each others.

Additional Inherited Members

7.151.1 Detailed Description

Class representing a three player tournament.

7.151.2 Constructor & Destructor Documentation

7.151.2.1 ThreePlayerTournament()

```
TournamentPlanner.Model.Tournaments.ThreePlayerTournament.ThreePlayerTournament (
            int nbPlayers,
            List< List< List< int >>> gamesByRounds,
            List< List< int >> groups = null )
```

Initializes a new instance of the ThreePlayerTournament class.

Parameters

nbPlayers	Number of players.
gamesByRounds	Games by round.
groups	The groups. Optional.

7.151.3 Member Function Documentation

7.151.3.1 TotalNumberOfGames()

```
int TournamentPlanner.Model.Tournaments.ThreePlayerTournament.TotalNumberOfGames ( )
```

Get the total number of games in the tournament.

Returns

The total number of games.

7.151.4 Property Documentation

7.151.4.1 PlayerGroups

```
List<List<int> > TournamentPlanner.Model.Tournaments.ThreePlayerTournament.PlayerGroups  [get]
```

Gets the player groups. Player that are together in a group won't play with each others.

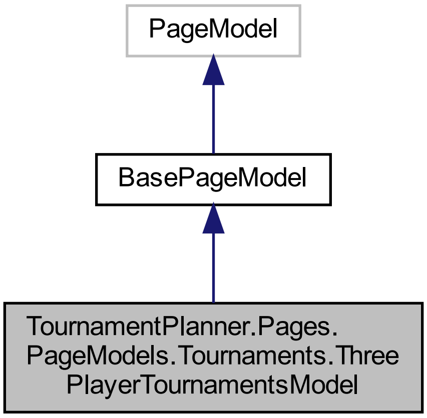
The documentation for this class was generated from the following file:

- ThreePlayerTournament.cs

7.152 TournamentPlanner.Pages.PageModels.Tournaments.Three↩ PlayerTournamentsModel Class Reference

Page model for three player tournaments.

Inheritance diagram for TournamentPlanner.Pages.PageModels.Tournaments.ThreePlayerTournamentsModel:



Public Member Functions

- void OnPost ()

Properties

- int NbPlayers = 15 `[get, set]`

  Gets or sets the number of players in the tournament.
- ThreePlayerTournament KTSTournament `[get]`

  Gets the tournament constructed using KTS.
- ThreePlayerTournament GDDTournament `[get]`

  Gets the tournament constructed using GDD.

7.152.1 Detailed Description

Page model for three player tournaments.

7.152.2 Property Documentation

7.152.2.1 GDDTournament

```
ThreePlayerTournament TournamentPlanner.Pages.PageModels.Tournaments.ThreePlayerTournaments↩
Model.GDDTournament  [get]
```

Gets the tournament constructed using GDD.

7.152.2.2 KTSTournament

```
ThreePlayerTournament TournamentPlanner.Pages.PageModels.Tournaments.ThreePlayerTournaments↩
Model.KTSTournament  [get]
```

Gets the tournament constructed using KTS.

7.152.2.3 NbPlayers

```
int TournamentPlanner.Pages.PageModels.Tournaments.ThreePlayerTournamentsModel.NbPlayers = 15
[get], [set]
```

Gets or sets the number of players in the tournament.

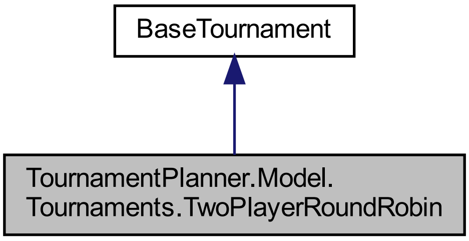
The documentation for this class was generated from the following file:

- ThreePlayerTournaments.cshtml.cs

7.153 TournamentPlanner.Model.Tournaments.TwoPlayerRoundRobin Class Reference

Represents a 2 player round robin tournament schedule.

Inheritance diagram for TournamentPlanner.Model.Tournaments.TwoPlayerRoundRobin:



Public Member Functions

- TwoPlayerRoundRobin (int order)

  Initializes a new instance of the TwoPlayerRoundRobin class.
- void AddRoundSchedule (List< List< int >> round)

  Add round.
- void AddRoundSchedule (List< List< int >> round, int inactive)

  Add round and inactive player.
- new string GameToString (List< int > game)

Public Attributes

- List< int > InactiveByRounds = new List<int>()

  Gets or sets the inactive players by round.

Additional Inherited Members

7.153.1 Detailed Description

Represents a 2 player round robin tournament schedule.

7.153.2 Constructor & Destructor Documentation

7.153.2.1 TwoPlayerRoundRobin()

```
TournamentPlanner.Model.Tournaments.TwoPlayerRoundRobin.TwoPlayerRoundRobin (
            int order )
```

Initializes a new instance of the TwoPlayerRoundRobin class.

Parameters

order	The order.

7.153.3 Member Function Documentation

7.153.3.1 AddRoundSchedule() [1/2]

```
void TournamentPlanner.Model.Tournaments.TwoPlayerRoundRobin.AddRoundSchedule (
            List< List< int >> round )
```

Add round.

Parameters

round	The round.

7.153.3.2 AddRoundSchedule() [2/2]

```
void TournamentPlanner.Model.Tournaments.TwoPlayerRoundRobin.AddRoundSchedule (
            List< List< int >> round,
            int inactive )
```

Add round and inactive player.

Parameters

round	The round.
inactive	The inactive player.

7.153.4 Member Data Documentation

7.153.4.1 InactiveByRounds

```
List<int> TournamentPlanner.Model.Tournaments.TwoPlayerRoundRobin.InactiveByRounds = new
List<int>()
```

Gets or sets the inactive players by round.

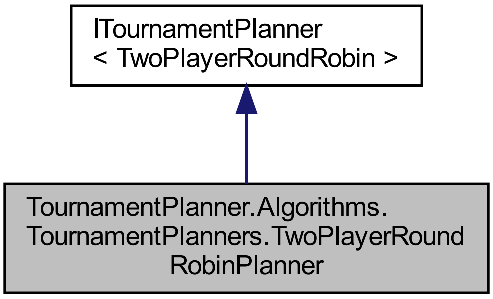
The documentation for this class was generated from the following file:

- TwoPlayerRoundRobin.cs

7.154 TournamentPlanner.Algorithms.TournamentPlanners.TwoPlayer↩ RoundRobinPlanner Class Reference

Tournament planner for two player round robins.

Inheritance diagram for TournamentPlanner.Algorithms.TournamentPlanners.TwoPlayerRoundRobinPlanner:


ITournamentPlanner
< TwoPlayerRoundRobin >
TournamentPlanner.Algorithms.
TournamentPlanners.TwoPlayerRound
RobinPlanner

Public Member Functions

- TwoPlayerRoundRobinPlanner (int nbPlayers)

  Initializes a new instance of the TwoPlayerRoundRobinPlanner class.
- TwoPlayerRoundRobin PlanTournament ()

  Plan a two player round robin tournament.

Properties

- int NbPlayers `[get]`

  Gets the number of players.
- int N `[get]`

  Gets N as the number of players divided by 2 and floored.
- bool OddNbPlayers `[get]`

  Gets a value indicating whether the number of players is odd.

7.154.1 Detailed Description

Tournament planner for two player round robins.

7.154.2 Constructor & Destructor Documentation

7.154.2.1 TwoPlayerRoundRobinPlanner()

```
TournamentPlanner.Algorithms.TournamentPlanners.TwoPlayerRoundRobinPlanner.TwoPlayerRound←↩
RobinPlanner (
            int nbPlayers )
```

Initializes a new instance of the TwoPlayerRoundRobinPlanner class.

Parameters

nbPlayers	The number of players.

7.154.3 Member Function Documentation

7.154.3.1 PlanTournament()

```
TwoPlayerRoundRobin TournamentPlanner.Algorithms.TournamentPlanners.TwoPlayerRoundRobin↩
Planner.PlanTournament ( )
```

Plan a two player round robin tournament.

Returns

The tournament.

7.154.4 Property Documentation

7.154.4.1 N

```
int TournamentPlanner.Algorithms.TournamentPlanners.TwoPlayerRoundRobinPlanner.N  [get]
```

Gets N as the number of players divided by 2 and floored.

7.154.4.2 NbPlayers

```
int TournamentPlanner.Algorithms.TournamentPlanners.TwoPlayerRoundRobinPlanner.NbPlayers  [get]
```

Gets the number of players.

7.154.4.3 OddNbPlayers

```
bool TournamentPlanner.Algorithms.TournamentPlanners.TwoPlayerRoundRobinPlanner.OddNbPlayers
[get]
```

Gets a value indicating whether the number of players is odd.

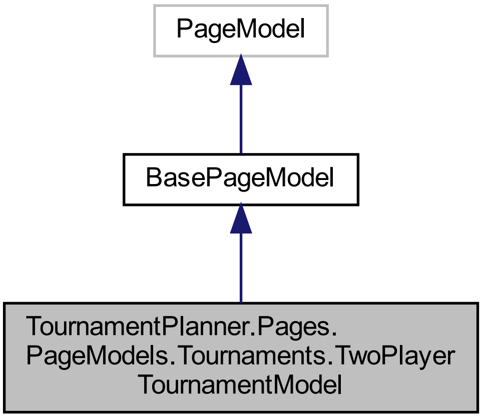
The documentation for this class was generated from the following file:

- TwoPlayerRoundRobinPlanner.cs

7.155 TournamentPlanner.Pages.PageModels.Tournaments.TwoPlayer↩ TournamentModel Class Reference

Page model for the construction of two player round robin tournaments.

Inheritance diagram for TournamentPlanner.Pages.PageModels.Tournaments.TwoPlayerTournamentModel:



Public Member Functions

- void OnPost ()

Properties

- int NbPlayers = 8 `[get, set]`

    Gets or sets the number of players.
- TwoPlayerRoundRobin Tournament `[get]`

    Gets the constructed tournament.

7.155.1 Detailed Description

Page model for the construction of two player round robin tournaments.

7.155.2 Property Documentation

7.155.2.1 NbPlayers

```
int TournamentPlanner.Pages.PageModels.Tournaments.TwoPlayerTournamentModel.NbPlayers = 8
[get], [set]
```

Gets or sets the number of players.

7.155.2.2 Tournament

```
TwoPlayerRoundRobin TournamentPlanner.Pages.PageModels.Tournaments.TwoPlayerTournamentModel.↩
Tournament  [get]
```

Gets the constructed tournament.

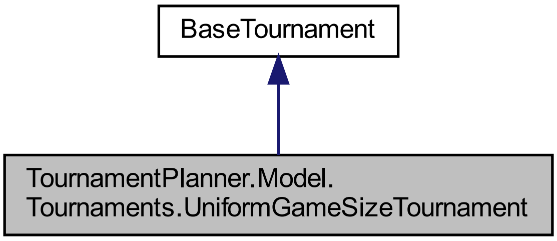
The documentation for this class was generated from the following file:

- TwoPlayerTournament.cshtml.cs

7.156 TournamentPlanner.Model.Tournaments.UniformGameSizeTournament Class Reference

Model for the tournament, where the sizes of the games are uniform - all the games have k-players.

Inheritance diagram for TournamentPlanner.Model.Tournaments.UniformGameSizeTournament:



Public Member Functions

- UniformGameSizeTournament (int nbPlayers, List< List< List< int >>> rounds, int objectiveValue, List< List< int >> leftOverPlayersByRounds=null)

  Initializes a new instance of the UniformGameSizeTournament class.
- int TotalNumberOfGames ()

  Gets the total number of games.

Properties

- List< List< int > > LeftOverPlayersByRounds `[get]`

  Gets the left over players.
- int ObjectiveValue `[get]`

  Gets the criterion value of the solution.

Additional Inherited Members

7.156.1 Detailed Description

Model for the tournament, where the sizes of the games are uniform - all the games have k-players.

7.156.2 Constructor & Destructor Documentation

7.156.2.1 UniformGameSizeTournament()

```
TournamentPlanner.Model.Tournaments.UniformGameSizeTournament.UniformGameSizeTournament (
            int nbPlayers,
            List< List< List< int >>> rounds,
            int objectiveValue,
            List< List< int >> leftOverPlayersByRounds = null )
```

Initializes a new instance of the UniformGameSizeTournament class.

Parameters

nbPlayers	Number of players.
rounds	The rounds.
objectiveValue	The objective value reached by the solver.
leftOverPlayersByRounds	The left over players by rounds.

7.156.3 Member Function Documentation

7.156.3.1 TotalNumberOfGames()

```
int TournamentPlanner.Model.Tournaments.UniformGameSizeTournament.TotalNumberOfGames ( )
```

Gets the total number of games.

Returns

The total number of games.

7.156.4 Property Documentation

7.156.4.1 LeftOverPlayersByRounds

```
List<List<int> > TournamentPlanner.Model.Tournaments.UniformGameSizeTournament.LeftOver←
PlayersByRounds  [get]
```

Gets the left over players.

7.156.4.2 ObjectiveValue

```
int TournamentPlanner.Model.Tournaments.UniformGameSizeTournament.ObjectiveValue  [get]
```

Gets the criterion value of the solution.

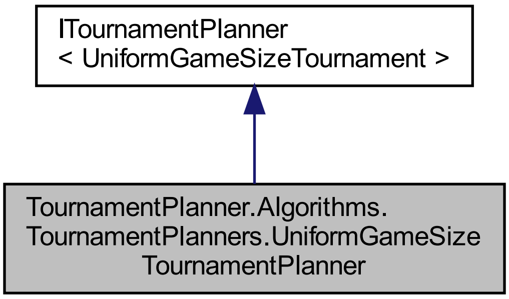
The documentation for this class was generated from the following file:

- UniformGameSizeTournament.cs

7.157 TournamentPlanner.Algorithms.TournamentPlanners.UniformGameSizeTournamentPlanner Class Reference

CP approach to the planning of tournaments, where the game sizes are uniform.

Inheritance diagram for TournamentPlanner.Algorithms.TournamentPlanners.UniformGameSizeTournamentPlanner:


ITournamentPlanner
< UniformGameSizeTournament >
TournamentPlanner.Algorithms.
TournamentPlanners.UniformGameSize
TournamentPlanner

Public Member Functions

- UniformGameSizeTournamentPlanner (int nbPlayers, int gameSize, int nbRounds, int timeLimit, CpSolver↵ SolutionCallback cb=null)

  Initializes a new instance of the UniformGameSizeTournamentPlanner class.
- UniformGameSizeTournament PlanTournament ()

7.157.1 Detailed Description

CP approach to the planning of tournaments, where the game sizes are uniform.

7.157.2 Constructor & Destructor Documentation

7.157.2.1 UniformGameSizeTournamentPlanner()

```
TournamentPlanner.Algorithms.TournamentPlanners.UniformGameSizeTournamentPlanner.UniformGame↵
SizeTournamentPlanner (
                int nbPlayers,
                int gameSize,
                int nbRounds,
                int timeLimit,
                CpSolverSolutionCallback cb = null )
```

Initializes a new instance of the UniformGameSizeTournamentPlanner class.

Parameters

nbPlayers	Then number of players.
gameSize	The size of a game.
nbRounds	The number of rounds.
timeLimit	The time limit for the solver.
cb	The optional solution callback.

The documentation for this class was generated from the following file:

- UniformGameSizeTournamentPlanner.cs

7.158 TournamentPlannerTests.Algorithms.TournamentPlanners.↵ UniformGameSizeTournamentPlannerTests Class Reference

Tests for the UniformGameSizeTournamentPlanner.

Public Member Functions

- void **KTS9EquivalentTest** ()

7.158.1 Detailed Description

Tests for the UniformGameSizeTournamentPlanner.

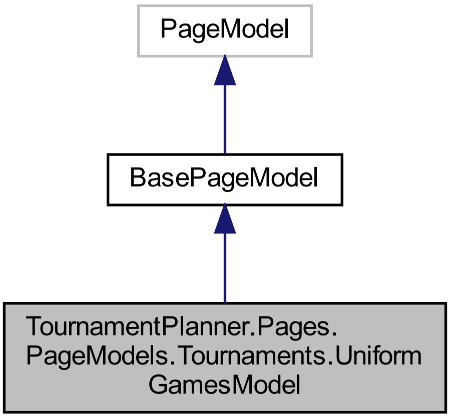
The documentation for this class was generated from the following file:

- UniformGameSizeTournamentPlannerTests.cs

7.159 TournamentPlanner.Pages.PageModels.Tournaments.Uniform↩ GamesModel Class Reference

Page model for the tournament, in which the games are uniformly sized.

Inheritance diagram for TournamentPlanner.Pages.PageModels.Tournaments.UniformGamesModel:



Public Member Functions

- void OnPost ()

Properties

- int NbPlayers = 13 `[get, set]`

  Gets or sets the number of players.
- int GameSize = 4 `[get, set]`

  Gets or sets the game size.
- int NbRounds = 3 `[get, set]`

  Gets or sets the number of rounds.
- int TimeLimit = 10 `[get, set]`

  Gets or sets the time limit.
- UniformGameSizeTournament Tournament `[get]`

  Gets the constructed tournament.

7.159.1 Detailed Description

Page model for the tournament, in which the games are uniformly sized.

7.159.2 Property Documentation

7.159.2.1 GameSize

```
int TournamentPlanner.Pages.PageModels.Tournaments.UniformGamesModel.GameSize = 4  [get], [set]
```

Gets or sets the game size.

7.159.2.2 NbPlayers

```
int TournamentPlanner.Pages.PageModels.Tournaments.UniformGamesModel.NbPlayers = 13  [get],
[set]
```

Gets or sets the number of players.

7.159.2.3 NbRounds

```
int TournamentPlanner.Pages.PageModels.Tournaments.UniformGamesModel.NbRounds = 3  [get], [set]
```

Gets or sets the number of rounds.

7.159.2.4 TimeLimit

```
int TournamentPlanner.Pages.PageModels.Tournaments.UniformGamesModel.TimeLimit = 10  [get],
[set]
```

Gets or sets the time limit.

7.159.2.5 Tournament

```
UniformGameSizeTournament TournamentPlanner.Pages.PageModels.Tournaments.UniformGamesModel.↩
Tournament [get]
```

Gets the constructed tournament.

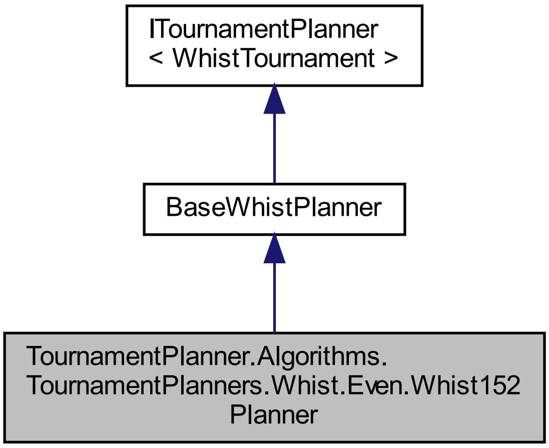
The documentation for this class was generated from the following file:

- UniformGames.cshtml.cs

7.160 TournamentPlanner.Algorithms.TournamentPlanners.Whist.↩ Even.Whist152Planner Class Reference

Class for the construction of Whist 152. Uses even SAMDRR.

Inheritance diagram for TournamentPlanner.Algorithms.TournamentPlanners.Whist.Even.Whist152Planner:


ITournamentPlanner
< WhistTournament >
BaseWhistPlanner
TournamentPlanner.Algorithms.
TournamentPlanners.Whist.Even.Whist152
Planner

Public Member Functions

- Whist152Planner ()

  Initializes a new instance of the Whist152Planner class.
- override WhistTournament PlanTournament ()

  Plan the Whist tournament.

Additional Inherited Members

7.160.1 Detailed Description

Class for the construction of Whist 152. Uses even SAMDRR.

7.160.2 Constructor & Destructor Documentation

7.160.2.1 Whist152Planner()

```
TournamentPlanner.Algorithms.TournamentPlanners.Whist.Even.Whist152Planner.Whist152Planner ( )
```

Initializes a new instance of the Whist152Planner class.

The documentation for this class was generated from the following file:

- Whist152Planner.cs

7.161 TournamentPlannerTests.Algorithms.TournamentPlanners.↩ Whist.Even.Whist152PlannerTests Class Reference

Tests for the Whist152Planner.

Public Member Functions

- void Whist152ConstructionTest ()

  Test the construction of Whist 152.

7.161.1 Detailed Description

Tests for the Whist152Planner.

7.161.2 Member Function Documentation

7.161.2.1 Whist152ConstructionTest()

```
void TournamentPlannerTests.Algorithms.TournamentPlanners.Whist.Even.Whist152PlannerTests.↩
Whist152ConstructionTest ( )
```

Test the construction of Whist 152.

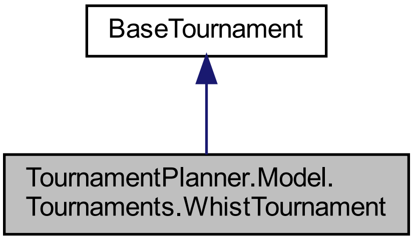
The documentation for this class was generated from the following file:

- Whist152PlannerTests.cs

7.162 TournamentPlanner.Model.Tournaments.WhistTournament Class Reference

The model of the whist tournament.

Inheritance diagram for TournamentPlanner.Model.Tournaments.WhistTournament:



Public Member Functions

- WhistTournament (int order)

  Initializes a new instance of the WhistTournament class.
- WhistTournament (int order, List< List< List< int >>> rounds)

  Initializes a new instance of the WhistTournament class.
- WhistTournament (int order, List< List< List< int >>> rounds, List< int > leftOvers)

  Initializes a new instance of the WhistTournament class.

Properties

- List< int > LeftOverByRounds = new () `[get]`

  Gets the left over players.

Additional Inherited Members

7.162.1 Detailed Description

The model of the whist tournament.

7.162.2 Constructor & Destructor Documentation

7.162.2.1 WhistTournament() [1/3]

```
TournamentPlanner.Model.Tournaments.WhistTournament.WhistTournament (
                int order )
```

Initializes a new instance of the WhistTournament class.

Parameters

order	The order.

7.162.2.2 WhistTournament() [2/3]

```
TournamentPlanner.Model.Tournaments.WhistTournament.WhistTournament (
            int order,
            List< List< List< int >>> rounds )
```

Initializes a new instance of the WhistTournament class.

Parameters

order	The order.
rounds	The rounds.

7.162.2.3 WhistTournament() [3/3]

```
TournamentPlanner.Model.Tournaments.WhistTournament.WhistTournament (
            int order,
            List< List< List< int >>> rounds,
            List< int > leftOvers )
```

Initializes a new instance of the WhistTournament class.

Parameters

order	The order.
rounds	The rounds.
leftOvers	The left over players.

7.162.3 Property Documentation

7.162.3.1 LeftOverByRounds

```
List<int> TournamentPlanner.Model.Tournaments.WhistTournament.LeftOverByRounds = new ()  [get]
```

Gets the left over players.

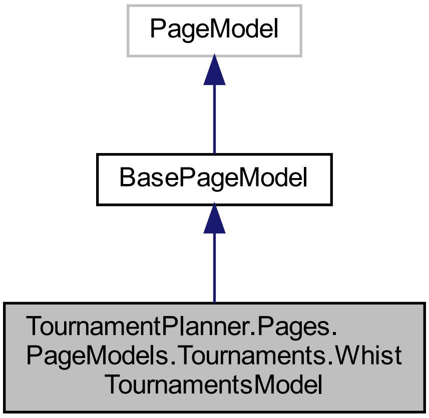
The documentation for this class was generated from the following file:

- WhistTournament.cs

7.163 TournamentPlanner.Pages.PageModels.Tournaments.Whist↩ TournamentsModel Class Reference

Page model for the Whist tournaments.

Inheritance diagram for TournamentPlanner.Pages.PageModels.Tournaments.WhistTournamentsModel:


PageModel
BasePageModel
TournamentPlanner.Pages.PageModels.Tournaments.WhistTournamentsModel

Public Member Functions

- void OnPost ()

Properties

- int NbPlayers = 9 `[get, set]`

  Gets or sets the number of players in the tournament.
- WhistTournament Whist `[get]`

  Gets the constructed tournament.

7.163.1 Detailed Description

Page model for the Whist tournaments.

7.163.2 Property Documentation

7.163.2.1 NbPlayers

```
int TournamentPlanner.Pages.PageModels.Tournaments.WhistTournamentsModel.NbPlayers = 9  [get],
[set]
```

Gets or sets the number of players in the tournament.

7.163.2.2 Whist

```
WhistTournament TournamentPlanner.Pages.PageModels.Tournaments.WhistTournamentsModel.Whist
[get]
```

Gets the constructed tournament.

The documentation for this class was generated from the following file:

- WhistTournaments.cshtml.cs

7.164 TournamentPlannerTests.Model.Tournaments.WhistTournament↩ Validator Class Reference

A validator for whist tournaments.

Static Public Member Functions

- static void ValidateWhistTournament (WhistTournament whist)

  Validate the properties of a whist tournament.

7.164.1 Detailed Description

A validator for whist tournaments.

7.164.2 Member Function Documentation

7.164.2.1 ValidateWhistTournament()

```
static void TournamentPlannerTests.Model.Tournaments.WhistTournamentValidator.ValidateWhist↩
Tournament (
             WhistTournament whist )  [static]
```

Validate the properties of a whist tournament.

Parameters

whist	The whist tournament to validate.

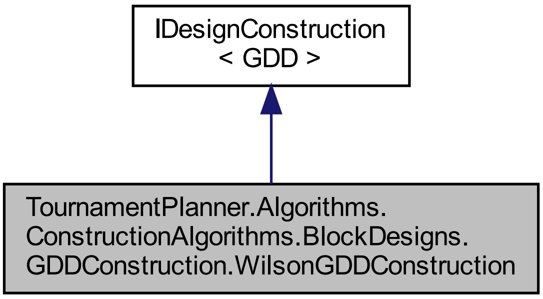
The documentation for this class was generated from the following file:

- WhistTournamentValidator.cs

7.165 TournamentPlanner.Algorithms.ConstructionAlgorithms.Block↵ Designs.GDDConstruction.WilsonGDDConstruction Class Reference

Wilsons construction methods using smaller GDDs to construct bigger ones. https://www.semanticscholar.↵ org/paper/An-Existence-Theory-for-Pairwise-Balanced-Designs-Wilson/b9c1c6bb1d7828b4f47f2ee3330dab26adb946dc

Inheritance diagram for TournamentPlanner.Algorithms.ConstructionAlgorithms.BlockDesigns.GDDConstruction.↵ WilsonGDDConstruction:


IDesignConstruction
< GDD >
TournamentPlanner.Algorithms.
ConstructionAlgorithms.BlockDesigns.
GDDConstruction.WilsonGDDConstruction

Public Member Functions

- WilsonGDDConstruction (int w, GDD initGDD)

    Initializes a new instance of the WilsonGDDConstruction class.
- GDD ConstructDesign ()

    Construct a GDD using wilsons method.

7.165.1 Detailed Description

Wilsons construction methods using smaller GDDs to construct bigger ones. https://www.semanticscholar.↵ org/paper/An-Existence-Theory-for-Pairwise-Balanced-Designs-Wilson/b9c1c6bb1d7828b4f47f2ee3330dab26adb946dc

7.165.2 Constructor & Destructor Documentation

7.165.2.1 WilsonGDDConstruction()

```
TournamentPlanner.Algorithms.ConstructionAlgorithms.BlockDesigns.GDDConstruction.Wilson↩
GDDConstruction.WilsonGDDConstruction (
              int w,
              GDD initGDD )
```

Initializes a new instance of the WilsonGDDConstruction class.

Parameters

w	The multiplier.
initGDD	The initial GDD.

7.165.3 Member Function Documentation

7.165.3.1 ConstructDesign()

```
GDD TournamentPlanner.Algorithms.ConstructionAlgorithms.BlockDesigns.GDDConstruction.Wilson↩
GDDConstruction.ConstructDesign ( )
```

Construct a GDD using wilsons method.

Returns

Constructed GDD.

The documentation for this class was generated from the following file:

- WilsonGDDConstruction.cs

7.166 TournamentPlannerTests.Algorithms.ConstructionAlgorithms.↩ BlockDesigns.GDDConstruction.WilsonGDDConstructionTests Class Reference

Tests for the WilsonGDDConstruction.

Public Member Functions

- void **GDD81ConstructionTest** ()
- void **GDD42ConstructionTest** ()
- void **GDD45Test** ()
- void **GDD78Construction** ()

7.166.1 Detailed Description

Tests for the WilsonGDDConstruction.

The documentation for this class was generated from the following file:

- WilsonGDDConstructionTests.cs

Index