



ČESKÉ VYSOKÉ UČENÍ TECHNICKÉ V PRAZE
FAKULTA DOPRAVNÍ

VÍT ŘEZÁČ

MOŽNOSTI APLIKACE VÍCEÚČELOVÉHO MODELU
ŽELEZNIČNÍ INFRASTRUKTURY

BAKALÁŘSKÁ PRÁCE - PŘÍLOHA 2:

POPIS FUNKCÍ VYTVOŘENÝCH V RÁMCI USNADNĚNÍ TVORBY MODELU V
NÁSTROJI PRO JEHO TVORBU V NAVRŽENÉM SPECIFICKÉM ROZŠÍŘENÍ

2023

Obsah

| | | |
|----------|--|----------|
| 1 | Vzorový popis funkce | 1 |
| 2 | Popis funkcí | 2 |
| 2.1 | VložitHorizontalCurveProJednoduchouVyhybkuDSFD(zdrojHPP, referenceVychozihoBodu, vzorVyhybky, smerOdboceni, uzavritLinearNetElement) | 2 |
| 2.2 | VložitHorizontalCurveProObloukovouVyhybkuDSFD(zdrojHPP, referenceVychozihoBodu, vzorVyhybky, smerOdboceni, uzavritLinearNetElement) | 5 |
| 2.3 | VložitHorizontalCurveProKrizovatkovouVyhybkuDSFD(zdrojHPP, referenceVychozihoBodu, vzorVyhybky, uzavritLinearNetElement) | 7 |
| 2.4 | VložitHorizontalCurveProKrizeniDSFD(zdrojHPP, referenceVychozihoBodu, vzorVyhybky, uzavritLinearNetElement) | 10 |
| 2.5 | PridatPrestavnikDSFD(goIdLinearElementPripojeni, goIdPosRel_Primy, goIdPosRel_Odbocny, referenceVychozihoBodu, referenceBodu, stranaPrestavniku, jmenoPrestavniku) | 13 |
| 2.6 | ZalozitPreddefinovaneObjektyProDSFD2023() | 15 |
| 2.7 | ZalozitSignalTypeDSFD(internalDesignator, longname, signalConstructionType) | 16 |
| 2.8 | VložitHorizontalCurveProVyhybkyDSFD() | 17 |
| 2.9 | PriraditCastNastupniHranyNastupniHrane() | 18 |

1 Vzorový popis funkce

Následující část obsahuje vzorový popis typické funkce vytvořené v rámci tvorby části nástroje obsluhující navrhované specifické rozšíření VMŽI pro DSFD. Jedná se neexistující funkci. Určité části vzorového popisu mohou být doplněny o rozšiřující popis, který u běžné funkce není. Tento popis je v textu vyznačen kurzívou.

VzorováFunkceProDSFD (vstupníParametr1, vstupníParametr2, ..., vstupníParametrN)

Účel funkce:

Tato část dokumentace obsahuje popis účelu funkce, přičemž v případě existence podobné, již existující funkce v nástroji pro tvorbu VMŽI, zdůrazňuje rozdílnou funkcionalitu oproti ostatním implementovaným funkcím v rámci nástroje pro tvorbu VMŽI.

Vstupní parametry:

Tato část obsahuje seznam všech parametrů vstupujících do funkce s vysvětlením jejich účelu v rámci funkce a rozsahu přípustných hodnot parametru formou intervalu, množiny nebo seznamu přípustných hodnot. V případě seznamu je každá položka doplněna textem vysvětlujícím funkci dané hodnoty.

- Parametr1: - Funkce daného parametru v rámci dané funkce.
Přípustné hodnoty:
 - Hodnota 1 - Zadáním této hodnoty dojde k určité operaci.
 - Hodnota 2 - Zadáním této hodnoty dojde k jiné operaci v rámci funkce.
- Parametr2: - Funkce daného parametru v rámci dané funkce.
Přípustné hodnoty: $< 0, 1$) - mohou být zadány hodnoty od 0 včetně do 1 mimo.
- ParametrN: - Funkce daného parametru v rámci dané funkce.
Přípustné hodnoty:
 - Hodnota 1 - Zadáním této hodnoty dojde k určité operaci.
 - $\{1, 2, 3, \dots, N\}$ - Mohou být zadána celá čísla od 0 do N včetně.

Vnitřní popis funkce:

V této části je detailně popsána struktura funkce, její vlastnosti, očekávané chování vč. požadovaných vstupů od uživatele zadávaných v průběhu běhu funkce, formy jejich zadání apod. Tato část také obsahuje popis možných nestandardních situací, které mohou vzniknout v průběhu běhu funkce a případná opatření v těchto situacích.

Seznam užitých funkcí:

Vzhledem k pokračujícímu vývoji nástroje pro tvorbu VMŽI je v této části uveden seznam všech funkcí využitých v rámci této funkce, aby v případě nekompatibility s modernizovanou verzí nástroje pro tvorbu VMŽI bylo usnadněno obnovení kompatibility.

- Funkce1(Parametr1, Parametr2, ...)
- Funkce2(Parametr1, Parametr2, ...)
- Funkce3(Parametr1, Parametr2, ...)

Výstupní hodnota:

Tato část obsahuje popis výstupní hodnoty, kterou tato funkce vrací a je možné ji přiřadit do proměnné. V případě absence této části tato funkce žádnou hodnotu nevrací.

výstupníHodnota - Jedná se o hodnotu reprezentující zpravidla vytvořený objekt. Tato hodnota je daného typu.

2 Popis funkcí

2.1 VložitHorizontalCurveProJednoduchouVyhybkuDSFD(zdrojHPP, referenceVychozihoBodu, vzorVyhybky, smerOdboceni, uzavritLinearNetElement)

Účel funkce:

Účelem této funkce je vložení geometrické a topologické reprezentace modelu jednoduché výhybky používané na DSFD. Na rozdíl od podobně pojmenované funkce **VložitHorizontalCurveProJednoduchouVyhybku**, která je součástí jádra nástroje pro tvorbu VMŽI tato funkce nepodporuje transformaci jednoduché výhybky na obloukovou, za to však vzhledem k mírně odlišnému popisu výhybky umožňuje nad výhybkou uzavřít instance třídy **LinearNetElement**, vč. určení průjezdnosti a připojit k dané výhybce přestavník.

Vstupní parametry:

- zdrojHPP: - Jedná se vektor souřadnic počátečního bodu a azimutu výhybky.
Přípustné hodnoty:
 - Null - Zadáním této hodnoty dojde k vynucení zadání bodu a azimutu výhybky uživatelem v průběhu běhu funkce.
 - Existující vektor HPP - Zadáním této hodnoty dojde přeskočení uživatelského zadání a využití daného bodu a azimutu.
- referenceVychozihoBodu: - Parametr určující ze kterého bodu a strany bude výhybka vkládána.
Přípustné hodnoty:
 - Null - Zadáním této hodnoty dojde k vynucení vybrání bodu uživatelem v průběhu běhu funkce. Ve všech ostatních případech dojde k přeskočení této volby
 - A - Zadáním této hodnoty dojde k vložení výhybky z bodu, který je společný pro oba směry (bod A - viz Obr.: 4 bakalářské práce) v orientaci proti hrotu výhybky.
 - B - Zadáním této hodnoty dojde k vložení výhybky z bodu konce přímé větve (bod B - viz Obr.: 4 bakalářské práce) v orientaci po hrotu výhybky.
 - D - Zadáním této hodnoty dojde k vložení výhybky z bodu konce odbočné větve (bod D - viz Obr.: 4 bakalářské práce) v orientaci po hrotu výhybky.
- vzorVyhybky: - Tento parametr je určen pro případ nutnosti budoucího rozšíření modelu o jiný typ jednoduché výhybky. V současné době je jeho funkce omezená existencí jediného typu jednoduché výhybky.
Přípustné hodnoty:

- Null - Zadáním této hodnoty dojde k vynucení uživatelského potvrzení vkládaného typu výhybky “JednoduchaDSFD”.
- JednoduchaDSFD - Zadáním této hodnoty dojde k přeskočení nutnosti uživatelského potvrzení.
- **směrOdboceni:** - Parametr definuje stranu, na kterou bude odbočná větev odbočovat z pohledu ve směru tvorby modelu výhybky.
Přípustné hodnoty:
 - Null - Zadáním této hodnoty dojde k vynucení uživatelského zadání směru v průběhu běhu funkce.
 - P - Zadáním této hodnoty dojde přeskočení uživatelského výběru a definování směru odbočení doprava.
 - L - Zadáním této hodnoty dojde přeskočení uživatelského výběru a definování směru odbočení doleva.
- **uzavritLinearNetElement:** - Parametr definuje, zdali dojde k automatickému zahájení uzavírání **linearNetElement** v průběhu běhu funkce.
Přípustné hodnoty:
 - Null - Zadáním této hodnoty dojde k umožnění volby uživatelem v průběhu běhu funkce.
 - 1 - Zadáním této hodnoty dojde přeskočení uživatelského výběru a zahájením automatizované tvorby instancí třídy **linearNetElement** v průběhu běhu funkce.
 - 0 - Zadáním této hodnoty dojde přeskočení uživatelského výběru a ukončením běhu funkce bez uzavření instancí třídy **linearNetElement**.

Vnitřní popis funkce:

Na počátku běhu funkce jsou vytvořeny potřebné proměnné, reprezentující jednotlivé části výhybky, polohy a azimuty významných bodů a další nutné hodnoty. V části funkce dochází k získání potřebných hodnot vstupních parametrů v případě, že na vstupu funkce jsou nastaveny na hodnotu Null. V případě vstupního vektoru HPP k tomuto dochází pomocí standardní nabídky v rámci nástroje pro tvorbu VMŽI, tj. v několika krocích s možnostmi přímého zadání, volby bodu apod. Zbylé parametry jsou získávány výběrem z předdefinovaných hodnot prostřednictvím nabídky. V této sekci je také obsažen prototyp výhybky s jeho parametry zadanými prostřednictvím parametrů popsanych v kap.: 4.3.1 bakalářské práce. Možnost dynamické tvorby prototypu výhybky uživatelem v průběhu běhu funkce není uvažována.

V následující části funkce probíhá tvorba geometrického a topologického modelu výhybky v závislosti na zadaném počátečním bodu topologického popisu výhybky. Tvorba modelů probíhá po částech s pomocí základních příkazů pro vykreslování přímek a křivek obsažených ve VMŽI, které jsou následně opět pomocí standardizovaných funkcí jádra VMŽI otáčeny do směru shodného se zadaným směrem pro dosažený konzistentního směru všech prvků v rámci tvořené výhybkové konstrukce. Jednotlivé křivky vytvořené jsou pro potřeby dalšího běhu funkce ukládány do proměnných reprezentujících danou křivku.

Další část funkce v případě, že vstupní parametr **uzavritLinearNetElement** = **Null**, vyzve uživatele, aby zvolil, jestli požaduje uzavření instancí třídy **LinearNetElement** automatizovaně v průběhu běhu funkce s pomocí volby z nabídky. V případě vstupu nebo volby hodnoty 1 (“Ano”) dojde k aktivaci další části funkce a postupnému vytvoření proměnných potřebných pro další běh funkce, získání **goId** všech geometrických

elementů a jejich převodu na **id**, což je provedeno pomocí standardizovaných postupů v rámci nástroje. Následně jsou takto vytvořené elementy propojeny s pomocí instancí třídy **PositionedRelation** a je definována sjízdnost mezi nimi. Funkce zajišťuje shodnost směru elementů s jejich geometrickou reprezentací.

V případě, že dojde k uzavření instancí třídy **NetElement** na tvořené výhybce, je uživatel vyzván, aby zvolil, jestli požaduje automatické přiřazení přestavníku k výhybce. V případě volby “Ano” je vyzván, aby zadal číslo výhybky. Pokud takto neučiní, je automaticky přiřazeno číslo 1. Následně dojde k přiřazení přestavníku. Touto operací běh funkce končí.

Hlavním problémem při běhu funkce je možnost selhání předání informace o vstupní proměnné získávané uživatelským zadáním v průběhu běhu funkce, např. přerušením nebo špatným zadáním ze strany uživatele. V tomto případě může dojít k úplnému selhání při běhu funkce, popř. k jinému atypickému chování. Jediným řešením v případě selhání je ukončení běhu skriptu pro tvorbu VMŽI a odebrání výsledků poslední akce.

Seznam užitých funkcí:

- ZjistitGoIdLevelNetworkMicro2D()
- AktualizovatVektorHPP(zdrojHPP, vnitriReference, pootoceni, textPriZadavaniVektoruSouradnic, textPriZadavaniAzimutu, aktualizovatGlobalniVektorHPP)
- ZvolitPolozkuZVyskakovaciNabidky(hlavicka_G, vektorVoleb_G)
- VypocitatPolomer(typZdrojeDelky, delka, deltaAzimuthDeg)
- VlozitPrimou(zdrojHPP, horizontalLength)
- ZjistitAzimuth0(goIdHorizontalCurve)
- ZjistitAzimuth1(goIdHorizontalCurve)
- VlozitKruznicovyOblouk(zdrojHPP, deltaAzimuth, radius)
- PrevestGoIdNaId(goIdHorizontalCurve)
- VytvoritPositionedRelation(vektorZakonceniA, vektorZakonceniB, navigability, goIdNetwork, goIdLevelNetwork)
- ZadatTextovyRetezec(identifikaceVystupu, vychoziHodnota)
- ObratitSmer(azimuthDeg)
- ZmenitSmerInstancimHorizontalCurve(vektorGoIdHorizontalCurve_S)
- PridatPrestavnikDSFD(goIdLinearElementPripojeni, goIdPosRel_Primy, goIdPosRel_Odbocky, referenceVychozihoBodu, referenceBodu, stranaPrestavniku, jmenoPrestavniku)

2.2 VložitHorizontalCurveProObloukovouVyhybkuDSFD(zdrojHPP, referenceVychozihoBodu, vzorVyhybky, smerOdboceni, uzavritLinearNetElement)

Účel funkce:

Účelem této funkce je vložení geometrické a topologické reprezentace modelu obloukové výhybky používané na DSFD, která se na rozdíl od jádra nástroje pro tvorbu VMŽI netvoří transformací jednoduché výhybky, ale pomocí této funkce. Tato funkce však vzhledem k mírně odlišnému popisu výhybky umožňuje nad výhybkou uzavřít **LinearNetElement**, vč. určení průjezdnosti a připojit k dané výhybce přestavník.

Vstupní parametry:

- zdrojHPP: - Jedná se vektor souřadnic počátečního bodu a azimutu výhybky.
Přípustné hodnoty:
 - Null - Zadáním této hodnoty dojde k vynucení zadání bodu a azimutu výhybky uživatelem v průběhu běhu funkce.
 - Existující vektor HPP - Zadáním této hodnoty dojde přeskočení uživatelského zadání a využití daného bodu a azimutu.
- referenceVychozihoBodu: - Parametr určující ze kterého bodu a strany bude výhybka vkládána.
Přípustné hodnoty:
 - Null - Zadáním této hodnoty dojde k vynucení vybrání bodu uživatelem v průběhu běhu funkce. Ve všech ostatních případech dojde k přeskočení této volby
 - A - Zadáním této hodnoty dojde k vložení výhybky z bodu, který je společný pro oba směry (bod A - viz Obr.: 4 bakalářské práce) v orientaci proti hrotu výhybky.
 - B - Zadáním této hodnoty dojde k vložení výhybky z bodu konce přímé větve (bod B - viz Obr.: 4 bakalářské práce) v orientaci po hrotu výhybky.
 - D - Zadáním této hodnoty dojde k vložení výhybky z bodu konce odbočné větve (bod D - viz Obr.: 4 bakalářské práce) v orientaci po hrotu výhybky.
- vzorVyhybky: - Tento parametr umožňuje přímé zadání prototypu tvořené výhybky.
Přípustné hodnoty:
 - Null - Zadáním této hodnoty dojde k vynucení uživatelského výběru vkládaného typu výhybky.
 - ObloukovaMalaDSFD - Zadáním této hodnoty dojde k použití parametrů odpovídající menší z obloukových výhybek použitých na DSFD.
 - ObloukovaVelkaDSFD - Zadáním této hodnoty dojde k použití parametrů odpovídající větší z obloukových výhybek použitých na DSFD.
- smerOdboceni: - Parametr definuje stranu, na kterou bude odbočná větev odbočovat z pohledu ve směru tvorby modelu výhybky.
Přípustné hodnoty:
 - Null - Zadáním této hodnoty dojde k vynucení uživatelského zadání směru v průběhu běhu funkce.
 - P - Zadáním této hodnoty dojde přeskočení uživatelského výběru a definování směru odbočení doprava.

- L - Zadáním této hodnoty dojde přeskočení uživatelského výběru a definování směru odbočení doleva.
- uzavritLinearNetElement: - Parametr definuje, zdali dojde k automatickému zahájení uzavírání **linearNetElement** v průběhu běhu funkce.
Přípustné hodnoty:
 - Null - Zadáním této hodnoty dojde k umožnění volby uživatelem v průběhu běhu funkce.
 - 1 - Zadáním této hodnoty dojde přeskočení uživatelského výběru a zahájením automatizované tvorby instancí třídy **linearNetElement** v průběhu běhu funkce.
 - 0 - Zadáním této hodnoty dojde přeskočení uživatelského výběru a ukončením běhu funkce bez uzavření instancí třídy **linearNetElement**.

Vnitřní popis funkce:

Struktura funkce je analogická funkci pro vkládání jednoduché výhybky a liší se především ve vkládání geometrické reprezentace výhybky, která je komplikovanější než v případě výhybky jednoduché. Z tohoto vychází i vyšší komplexnost některých návazných kroků. Na počátku běhu funkce jsou vytvořeny potřebné proměnné, reprezentující jednotlivé části výhybky, polohy a azimuty významných bodů a další nutné hodnoty. V části funkce dochází k získání potřebných hodnot vstupních parametrů v případě, že na vstupu funkce jsou nastaveny na hodnotu Null. V případě vstupního vektoru HPP k tomuto dochází pomocí standardní nabídky v rámci nástroje pro tvorbu VMŽI, tj. v několika krocích s možnostmi přímého zadání, volby bodu apod. Zbylé parametry jsou získávány výběrem z předdefinovaných hodnot prostřednictvím nabídky. V této sekci jsou také obsaženy prototypy výhybky s jejich parametry zadanými prostřednictvím parametrů popsaných v kap.: 4.3.2 bakalářské práce. Možnost dynamické tvorby prototypu výhybky uživatelem v průběhu běhu funkce není uvažována.

V následující části funkce probíhá tvorba geometrického a topologického modelu výhybky v závislosti na zadaném počátečním bodu topologického popisu výhybky. Tvorba modelů probíhá po částech s pomocí základních příkazů pro vykreslování přímků a křivek obsažených ve VMŽI, které jsou následně opět pomocí standardizovaných funkcí jádra VMŽI otáčeny do směru shodného se zadaným směrem pro dosažení konzistentního směru všech prvků v rámci tvořené výhybkové konstrukce. Jednotlivé vytvořené křivky jsou pro potřeby dalšího běhu funkce ukládány do proměnných reprezentujících danou křivku.

Další část funkce v případě, že vstupní parametr **uzavritLinearNetElement** = **Null**, vyzve uživatele, aby zvolil, jestli požaduje uzavření instancí třídy **LinearNetElement** automatizovaně v průběhu běhu funkce s pomocí volby z nabídky. V případě vstupu nebo volby hodnoty 1 (“Ano”) dojde k aktivaci další části funkce a postupnému vytvoření proměnných potřebných pro další běh funkce, získání **goId** všech geometrických elementů a jejich převodu na **id**, což je provedeno pomocí standardizovaných funkcí nástrojů pro tvorbu VMŽI. Následně jsou takto vytvořené elementy propojeny s pomocí instancí třídy **PositionedRelation** a je definována sjízdnost mezi nimi. Funkce zajišťuje shodnost směru elementů s jejich geometrickou reprezentací.

V případě, že dojde k uzavření instancí třídy **NetElement** na tvořené výhybce, je uživatel vyzván, aby zvolil, jestli požaduje automatické přiřazení přestavníku k výhybce. V případě volby “Ano” je vyzván, aby zadal číslo výhybky. Pokud takto neučiní, je auto-

matically přiřazeno číslo 1. Následně dojde k přiřazení přestavníku. Touto operací běh funkce končí.

Hlavním problémem při běhu funkce je možnost selhání předání informace o vstupní proměnné získávané uživatelským zadáním v průběhu běhu funkce, např. přerušením nebo špatným zadáním ze strany uživatele. V tomto případě může dojít k úplnému selhání při běhu funkce, popř. k jinému atypickému chování. Jediným řešením v případě selhání je ukončení běhu skriptu pro tvorbu VMŽI a odebrání výsledků jeho posledního běhu.

Seznam užitých funkcí:

- `ZjistitGoIdLevelNetworkMicro2D()`
- `AktualizovatVektorHPP(zdrojHPP, vnitriReference, pootoceni, textPriZadavaniVektoruSouradnic, textPriZadavaniAzimutu, aktualizovatGlobalniVektorHPP)`
- `ZvolitPolozkuZVyskakovaciNabidky(hlavicka_G, vektorVoleb_G)`
- `VypocitatPolomer(typZdrojeDelky, delka, deltaAzimuthDeg)`
- `VlozitPrimou(zdrojHPP, horizontalLength)`
- `ZjistitAzimuth0(goIdHorizontalCurve)`
- `ZjistitAzimuth1(goIdHorizontalCurve)`
- `VlozitKruznicovyOblouk(zdrojHPP, deltaAzimuth, radius)`
- `PrevestGoIdNaId(goIdHorizontalCurve)`
- `VytvoritPositionedRelation(vektorZakonceniA, vektorZakonceniB, navigability, goIdNetwork, goIdLevelNetwork)`
- `ZadatTextovyRetezec(identifikaceVystupu, vychoziHodnota)`
- `ObratitSmer(azimuthDeg)`
- `ZmenitSmerInstancimHorizontalCurve(vektorGoIdHorizontalCurve_S)`
- `PridatPrestavnikDSFD(goIdLinearElementPripojeni, goIdPosRel_Primy, goIdPosRel_Odbocky, referenceVychozihoBodu, referenceBodu, stranaPrestavniku, jmenoPrestavniku)`

2.3 `VlozitHorizontalCurveProKrizovatkovouVyhybkuDSFD(zdrojHPP, referenceVychozihoBodu, vzorVyhybky, uzavritLinearNetElement)`

Účel funkce:

Účelem funkce je vkládání geometrické a topologické reprezentace křížovatkové výhybky používané na DSFD. Před vytvořením této funkce neexistovala žádná funkce pro tvorbu křížovatkových výhybek v rámci nástroje pro tvorbu VMŽI. Implementovaná funkce umožňuje automatizované vložení geometrického popisu křížovatkové výhybky používané v kolejišti DSFD, případně uzavření instancí třídy **LinearNetElement** a připojení přestavníků vč. jejich logického propojení.

Vstupní parametry:

- zdrojHPP: - Jedná se vektor souřadnic počátečního bodu a azimutu křížovatky.
Přípustné hodnoty:
 - Null - Zadáním této hodnoty dojde k vynucení zadání bodu a azimutu křížovatkové výhybky uživatelem v průběhu běhu funkce.
 - Existující vektor HPP - Zadáním této hodnoty dojde přeskočení uživatelského zadání a využití daného bodu a azimutu.
- referenceVychozihobodu: - Parametr určující ze kterého bodu a strany bude konstrukce vkládána.
Přípustné hodnoty:
 - Null - Zadáním této hodnoty dojde k vynucení vybrání bodu uživatelem v průběhu běhu funkce. Ve všech ostatních případech dojde k přeskočení této volby
 - A - Zadáním této hodnoty dojde k vložení křížovatkové výhybky z bodu A (viz Obr.: 4 bakalářské práce) v orientaci všech elementů v pokračování z daného bodu směrem do konstrukce.
 - B - Zadáním této hodnoty dojde k vložení křížovatkové výhybky z bodu B (viz Obr.: 4 bakalářské práce) v orientaci všech elementů v pokračování z daného bodu směrem do konstrukce.
 - C - Zadáním této hodnoty dojde k vložení křížovatkové výhybky z bodu C (viz Obr.: 4 bakalářské práce) v orientaci všech elementů v pokračování z daného bodu směrem do konstrukce.
 - D - Zadáním této hodnoty dojde k vložení křížovatkové výhybky z bodu D (viz Obr.: 4 bakalářské práce) v orientaci všech elementů v pokračování z daného bodu směrem do konstrukce.
- vzorVyhybky: - Tento parametr umožňuje přímé zadání prototypu tvořené křížovatkové výhybky. V současné verzi není tento parametr funkční a zadáním libovolné hodnoty, vč. **Null** dojde k přeskočení volby a vytvoření standardní křížovatkové výhybky na DSFD. Tento parametr je ponechán pro případ rozšíření modelu o jiné typy křížovatkových výhybek.
Přípustné hodnoty: Libovolná hodnota je přípustná. Je doporučeno používat hodnotu **Null**
- uzavritLinearNetElement: - Parametr definuje, zdali dojde k automatickému zahájení uzavírání **linearNetElement** v průběhu běhu funkce.
Přípustné hodnoty:
 - Null - Zadáním této hodnoty dojde k umožnění volby uživatelem v průběhu běhu funkce.
 - 1 - Zadáním této hodnoty dojde přeskočení uživatelského výběru a zahájením automatizované tvorby instancí třídy **linearNetElement** v průběhu běhu funkce.
 - 0 - Zadáním této hodnoty dojde přeskočení uživatelského výběru a ukončením běhu funkce bez uzavření instancí třídy **linearNetElement**.

Vnitřní popis funkce:

Struktura funkce vychází z funkcí pro vkládání jednoduchých a obloukových výhybek, od nich se však liší vyšší komplexitou dané výhybkové konstrukce a s tím spojenou složitostí

funkce. Po zahájení běhu funkce jsou vytvořeny proměnné reprezentující jednotlivé části konstrukce, polohy a směry začátků a konců jednotlivých prvků popisů konstrukce a další nutné proměnné, všechny v této fázi běhu nastavené na hodnotu **Null**. V následující části funkce dochází k získání parametrů jako je poloha, počáteční azimut a bod zahájení kreslení konstrukce, pokud nebyly vloženy pomocí vstupních proměnných. K provedení těchto vstupů je využito standardních funkcí implementovaných v nástroji pro tvorbu VMŽI. Tato část funkce také obsahuje prototyp křížovatkové výhybky popsany způsobem uvedeným v kap.: 4.3.3 bakalářské práce. Ten je v současné verzi implementován jako pevný, bez možnosti uživatelské změny nebo výběru vzoru křížovatkové výhybky za běhu funkce. Je zde však ponechaný kód, potřebný pro umožnění uživatelské volby vzoru konstrukce, který je však převeden na komentář a není tedy v průběhu funkce prováděn.

Na základě zadaných hodnot dojde v dalším běhu funkce k vytvoření geometrického a topologického modelu konstrukce, v závislosti na zadaném vstupním bodu. Postup je vždy stejný, avšak vzhledem k fixnímu pojmenování částí konstrukce pro potřeby navazujících částí funkce je tvorba rozdělena na čtyři možnosti podle bodu vstupu. Části konstrukce jsou vždy vytvářeny následujícím způsobem: nejprve je vytvořen úsek před jazyky u zadaného vstupního bodu a navazující přímé úseky až po opačný konec konstrukce, následně je z bodu křížení **X** tvořeny oba přímé úseky vedoucí do bodu sousedícího se zadaným vstupním bodem, tyto dva úseky jsou následně otočeny, aby měly stejnou orientaci s úseky vedoucími ze zadaného bodu (a tedy mířily směrem k bodu křížení), dále jsou vytvořeny zbývající přímé úseky z bodu křížení do posledního nevytvořeného koncového bodu konstrukce, nakonec jsou vytvářeny obě obloukové větve.

V dalším běhu funkce je v případě hodnoty **Null** ve vstupním parametru uživatel dotázán na požadavek uzavření instancí třídy **LinearNetEntity** nad vytvořenou konstrukcí. V případě vybraní volby “Ano” (odpovídající hodnotě **1**) běh funkce pokračuje dále, v případě opačné volby je běh funkce ukončen. V další části běhu funkce dochází k vytvoření proměnných potřebných pro další běh funkce, získání **gold** všech geometrických elementů a jejich převodu na **id**, což je provedeno pomocí běžných funkcí jádra nástroje pro tvorbu VMŽI. Následně jsou takto vytvořené elementy v bodech **AR**, **BR**, **CR**, **DR** a **X** (viz Obr.: 4 bakalářské práce) propojeny s pomocí instancí třídy **PositionedRelation** a je definována sjízdnost mezi nimi. Funkce zajišťuje shodnost směru elementů s jejich geometrickou reprezentací.

Následně je od uživatele vyžádána volba, jestli chce k takto vytvořeným instancím třídy **PositionedRelation** vložit přestavníky. Pokud zvolí hodnotu **1** (volba “Ano”) běh funkce pokračuje, pokud zvolí možnost **0** (volba “Ne”), běh funkce končí. V dalším běhu funkce je vyžádáno zadání čísla výhybky, založení potřebných proměnných a vygenerování názvu přestavníků na základě jejich polohy ve výhybce ze zadaného čísla výhybky. Následně jsou přidány čtyři přestavníky do bodů **AR**, **BR**, **CR** a **DR** v tomto pořadí, přičemž při volbě strany umístění přestavníku je vždy nutno posuzovat stranu z pohledu do křížovatkové výhybky z vnější strany konstrukce (pro přestavník v bodě **AR** je nutno se dívat ve směru z bodu **A** do bodu **X**). Na konci běhu funkce jsou dvojice přestavníků spojeny tak, aby reprezentovaly skutečné přestavníky křížovatkové výhybce. Po propojení přestavníků běh funkce končí.

Nejčastějším problémem, který může při běhu funkce nastat je chybné zadání hodnoty uživatelem (vzniklé např. v důsledku přerušování zadávání), které však neukončí běh funkce. V takovémto případě může dojít k nestandardnímu chování, nebo úplnému selhání

dalšího běhu funkce. Jediným způsobem řešení nastalé chyby je ukončení běhu nástroje a odebrání výsledků posledního běhu skriptu.

Seznam užitých funkcí:

- ZjistitGoIdLevelNetworkMicro2D()
- AktualizovatVektorHPP(zdrojHPP, vnitriReference, pootoceni, textPriZadavaniVektoruSouradnic, textPriZadavaniAzimutu, aktualizovatGlobalniVektorHPP)
- ZvolitPolozkuZVyskakovaciNabidky(hlavicka_G, vektorVoleb_G)
- VypocitatPolomer(typZdrojeDelky, delka, deltaAzimuthDeg)
- VlozitPrimou(zdrojHPP, horizontalLength)
- ZjistitAzimuth0(goIdHorizontalCurve)
- ZjistitAzimuth1(goIdHorizontalCurve)
- VlozitKruznicovyOblouk(zdrojHPP, deltaAzimuth, radius)
- PrevestGoIdNaId(goIdHorizontalCurve)
- VytvoritPositionedRelation(vektorZakonceniA, vektorZakonceniB, navigability, goIdNetwork, goIdLevelNetwork)
- ZadatTextovyRetezec(identifikaceVystupu, vychoziHodnota)
- ObratitSmer(azimuthDeg)
- ZmenitSmerInstancimHorizontalCurve(vektorGoIdHorizontalCurve_S)
- PridatPrestavnikDSFD(goIdLinearElementPripojeni, goIdPosRel_Primy, goIdPosRel_Odbocny, referenceVychozihoBodu, referenceBodu, stranaPrestavniku, jmenoPrestavniku)
- PriraditSitovouEntituSitoveEntite(tridaA, goIdNetEntityA, tridaB, goIdNetEntityB, relationType)

2.4 VlozitHorizontalCurveProKrizeniDSFD(zdrojHPP, referenceVychozihoBodu, vzorVyhybky, uzavritLinearNetElement)

Účel funkce:

Účelem této funkce je vytvoření topologické a geometrické reprezentace prostého křížení dle vzoru užívaného na kolejišti DSFD, případné uzavření instancí třídy **LinearNetElement**. V rámci nástroje pro tvorbu VMŽI se jedná o první implementaci funkce pro automatické vkládání kolejového křížení do modelu. V rámci nástrojů existovala možnost vložení dvojité kolejové spojky, která prosté křížení obsahuje, nešlo však o samostatnou funkci pro tvorbu této kolejové konstrukce.

Vstupní parametry:

- zdrojHPP: - Jedná se vektor souřadnic počátečního bodu a azimutu křížení.
Přípustné hodnoty:

- Null - Zadáním této hodnoty dojde k vynucení zadání bodu a azimutu křížení uživatelem v průběhu běhu funkce.
 - Existující vektor HPP - Zadáním této hodnoty dojde přeskočení uživatelského zadání a využití daného bodu a azimutu.
- referenceVychozihobodu: - Parametr určující ze kterého bodu a strany bude křížení vkládáno.
Přípustné hodnoty:
 - Null - Zadáním této hodnoty dojde k vynucení vybrání bodu uživatelem v průběhu běhu funkce. Ve všech ostatních případech dojde k přeskočení této volby
 - A - Zadáním této hodnoty dojde k vložení křížení z bodu A (viz Obr.: 4 bakalářské práce) v orientaci všech elementů v pokračování z daného bodu směrem do konstrukce.
 - B - Zadáním této hodnoty dojde k vložení křížení z bodu B (viz Obr.: 4 bakalářské práce) v orientaci všech elementů v pokračování z daného bodu směrem do konstrukce.
 - C - Zadáním této hodnoty dojde k vložení křížení z bodu C (viz Obr.: 4 bakalářské práce) v orientaci všech elementů v pokračování z daného bodu směrem do konstrukce.
 - D - Zadáním této hodnoty dojde k vložení křížení z bodu D (viz Obr.: 4 bakalářské práce) v orientaci všech elementů v pokračování z daného bodu směrem do konstrukce.
 - vzorVyhybky: - Tento parametr umožňuje přímé zadání prototypu tvořeného křížení. V současné verzi není tento parametr funkční a zadáním libovolné hodnoty, vč. **Null** dojde k přeskočení volby a vytvoření standardního křížení užívaného v kolejišti DSFD. Tento parametr je ponechán pro případ rozšíření modelu o jiné typy kolejového křížení.
Přípustné hodnoty: Libovolná hodnota je přípustná. Je doporučeno používat hodnotu **Null**
 - uzavritLinearNetElement: - Parametr definuje, zdali dojde k automatickému zahájení uzavírání **LinearNetElement** v průběhu běhu funkce.
Přípustné hodnoty:
 - Null - Zadáním této hodnoty dojde k umožnění volby uživatelem v průběhu běhu funkce.
 - 1 - Zadáním této hodnoty dojde přeskočení uživatelského výběru a zahájením automatizované tvorby instancí třídy **linearNetElement** v průběhu běhu funkce.
 - 0 - Zadáním této hodnoty dojde přeskočení uživatelského výběru a ukončením běhu funkce bez uzavření instancí třídy **linearNetElement**.

Vnitřní popis funkce:

Struktura a běh funkce je z velké části shodná s výše popsanou funkcí **VlozitHorizontalCurveProKrizovatkovouVyhybkuDSFD**, z níž tato funkce vznikla vypuštěním částí specifických pro úplnou křížovatkovou výhybku v porovnání s prostým kolejovým křížením. Běh funkce začíná vynucením uživatelského doplnění chybějících parametrů pro další běh funkce, pokud byla jejich hodnota na vstupu do funkce **Null**. Toto doplnění je provedeno s pomocí standardních nástrojů implementovaných v rámci nástroje pro tvorbu VMŽI. Shodně s funkcí pro tvorbu křížovatkové výhybky se v této části funkce nachází popis křížení, které je touto funkcí vkládáno. Na rozdíl od funkce z níž byla odvozena, nenachází se v této části kód pro umožnění uživatelské volby vzoru křížení (a to ani ve formě

komentáře), a to z důvodu jeho shody s kódem nacházejícím se ve funkci pro vkládání křížovatkové výhybky.

Vkládání topologického a geometrického popisu probíhá z uživatelem zadaného bodu vkládáním přímých úseků spojujících tento bod s bodem křížení **X** (viz Obr.: 4 bakalářské práce) a s bodem na opačné straně konstrukce. Následně je vložena přímá část z bodu křížení do bodu sousedícího se vstupním bodem, jejíž orientace je následně otočena tak, aby souhlasila s orientací první vkládané přímé (tj. aby mířila z krajního bodu křížení do bodu **X**). Nakonec je vkládána přímá navazující na naposledy vkládanou část a mířící z bodu křížení do posledního krajního bodu konstrukce.

V další části běhu funkce je (v případě hodnoty **Null** ve vstupním parametru **uzavritLinearNetElement**) uživatel vyzván, jestli požaduje uzavření instancí třídy **LinearNetElement** na právě vytvářené konstrukci. V případě odpovědi **0** (“Ne”) běh funkce končí, v případě odpovědi **1** (“Ano”) dochází k vytvoření potřebných proměnných a postupnému uzavírání jednotlivých elementů nad všemi částmi konstrukce. Nakonec dochází k propojování jednotlivých elementů v bodě křížení s pomocí instancí tříd **PositionedRelation** a určování jejich průjezdnosti. Na rozdíl od křížovatkové výhybky nelze, vzhledem k charakteru konstrukce, provést přiřazení přestavníků, a proto po ukončení tvorby **PositionedRelation** běh funkce končí.

Nejčastějším problémem, který může při běhu funkce nastat je chybné zadání hodnoty uživatelem (vzniklé např. v důsledku přerušení zadávání), které však neukončí běh funkce. V takovémto případě může dojít k nestandardnímu chování, nebo úplnému selhání dalšího běhu funkce. Jediným způsobem řešení nastalé chyby je ukončení běhu nástroje a odebrání výsledků posledního běhu skriptu.

Seznam užitých funkcí:

- ZjistitGoIdLevelNetworkMicro2D()
- AktualizovatVektorHPP(zdrojHPP, vnitriReference, pootoceni, textPriZadavaniVektoruSouradnic, textPriZadavaniAzimutu, aktualizovatGlobalniVektorHPP)
- ZvolitPolozkuZVyskakovaciNabidky(hlavicka_G, vektorVoleb_G)
- VlozitPrimou(zdrojHPP, horizontalLength)
- ZjistitAzimuth0(goIdHorizontalCurve)
- ZjistitAzimuth1(goIdHorizontalCurve)
- PrevestGoIdNaId(goIdHorizontalCurve)
- VytvoritPositionedRelation(vektorZakonceniA, vektorZakonceniB, navigability, goIdNetwork, goIdLevelNetwork)
- ZadatTextovyRetezec(identifikaceVystupu, vychoziHodnota)
- ObratitSmer(azimuthDeg)
- ZmenitSmerInstancimHorizontalCurve(vektorGoIdHorizontalCurve_S)

2.5 PridatPrestavnikDSFD(goIdLinearElementPripojeni, goIdPosRel_Privy, goIdPosRel_Odbocny, referenceVychozihoBodu, referenceBodu, stranaPrestavniku, jmenoPrestavniku)

Účel funkce:

Účelem této funkce je vytvoření instance třídy **SwitchMachine**, její lokalizace a přiřazení vytvořeného přestavníku k instancím třídy **PositionedRelation**. V rámci funkce je vytvořena příprava pro jednoduchou volbu strany přestavníku na výhybkových konstrukcích DSFD, která však v současné době není v provozu z důvodu chybějící návaznosti v jádře nástroje pro tvorbu VMŽI. Funkce slouží pro automatizované vkládání a není určena pro ruční vkládání, pro které je nutné použít standardní funkce nástroje.

Vstupní parametry:

- goIdLinearElementPripojeni - Jedná se o goId elementů, na němž má být daný přestavník lokalizován.
Přípustné hodnoty:
 - Null - Zadáním této hodnoty dojde k vynucení vybrání elementu uživatelem v průběhu běhu funkce.
 - Existující goId - Zadáním této hodnoty dojde přeskočení uživatelského zadání a využití daného elementu.
- goIdPosRel_Privy - Jedná se od goId instance třídy PositionedRelation reprezentující přímý směr.
 - Null - Zadáním této hodnoty dojde k vynucení vybrání instance PositionedRelation uživatelem v průběhu běhu funkce.
 - Existující goId - Zadáním této hodnoty dojde přeskočení uživatelského zadání a využití dané instance.
- goIdPosRel_Odbocny - Jedná se od goId instance třídy PositionedRelation reprezentující odbočný směr.
 - Null - Zadáním této hodnoty dojde k vynucení vybrání instance PositionedRelation uživatelem v průběhu běhu funkce.
 - Existující goId - Zadáním této hodnoty dojde přeskočení uživatelského zadání a využití dané instance.
- referenceVychozihoBodu - Parametr určující ze kterého bodu a strany byla obecná výhybková konstrukce vkládána.
Přípustné hodnoty:
 - A - Zadáním této hodnoty dojde k zorientování zvolené strany pro vkládání dle orientace pro konstrukci vkládanou z bodu A.
 - B - Zadáním této hodnoty dojde k zorientování zvolené strany pro vkládání dle orientace pro konstrukci vkládanou z bodu B.
 - C - Zadáním této hodnoty dojde k zorientování zvolené strany pro vkládání dle orientace pro konstrukci vkládanou z bodu C.
 - D - Zadáním této hodnoty dojde k zorientování zvolené strany pro vkládání dle orientace pro konstrukci vkládanou z bodu D.

- referenceBodu - Parametr určující do kterého bodu výhybkové konstrukce je přestavník vkládán.
Přípustné hodnoty:
 - AR - Zadáním této hodnoty dojde k zorientování zvolené strany pro vkládání pro bod AR.
 - BR - Zadáním této hodnoty dojde k zorientování zvolené strany pro vkládání pro bod AR.
 - CR - Zadáním této hodnoty dojde k zorientování zvolené strany pro vkládání pro bod AR.
 - DR - Zadáním této hodnoty dojde k zorientování zvolené strany pro vkládání pro bod AR.
- stranaPrestavniku - Parametr určující stranu od osy koleje, s níž bude přestavník lokalizován na základě jeho polohy na konkrétní výhybkové konstrukci. Tento parametr slouží jako příprava pro jednoduchou volbu strany přestavníku bez ohledu na specifika konkrétní výhybkové konstrukce a v současné době není využíván.
Přípustné hodnoty: - Následující seznam je pouze seznamem zamýšlených hodnot a není v průběhu běhu využíván.
 - Null - Zadáním této hodnoty dojde k vynucení vybrání strany uživatelem v průběhu běhu funkce. Ve všech ostatních případech dojde k přeskočení této volby
 - Pravá - Zadáním této hodnoty by mělo v budoucnosti dojít k lokalizaci přestavníku na pravou stranu od osy koleje z pohledu proti hrotu srdcovky.
 - Levá - Zadáním této hodnoty by mělo v budoucnosti dojít k lokalizaci přestavníku na levou stranu od osy koleje z pohledu proti hrotu srdcovky.
- jmenoPrestavniku - Parametr určující hodnotu atributu `internalDesignator` sloužící k pojmenování daného přestavníku.
Přípustné hodnoty:
 - Null - Zadáním této hodnoty dojde k vynucení vybrání bodu uživatelem v průběhu běhu funkce.
 - obecný textový řetězec - Zadáním této hodnoty dojde k přeskočení uživatelského zadání v průběhu běhu funkce a pojmenování přestavníku daným textovým řetězcem.

Vnitřní popis funkce:

V první části běhu funkce dochází k tvorbě potřebných proměnných a výběru strany vkládaného přestavníku z pohledu proti hrotu výhybky s pomocí standardní metody nástroje. Dále je v této části získána pozice počátku a konce daného elementu, na němž má být přestavník lokalizován.

V následující části dochází na základě **referenceVýchozíhoBodu** a **referenceBodu** k výběru pozice přestavníku na elementu vzhledem k jeho poloze ve výhybkové konstrukci. V rámci této části bude výhledově též probíhat výběr parametrů předávaných dále do funkce **VložitAPridruzenoLokalizovatNetEntity** tak, aby nebylo nutno zadávat stranu méně intuitivním způsobem s pomocí standardních funkcí nástroje.

V poslední části dojde k převodu všech **goId** na **Id**, vytvoření **lokálníMaticyPHA** uchovávající vkládané informace o instanci a **MaticyOOP** sloužící k lokalizaci, dále vytvoření **goIdAssociatedLocation** definující lokalitu umístění a způsob lokalizace. Na

základě těchto proměnných je následně vytvořena a lokalizována požadovaná instance, jejíž **goId** je následně předáno jako výstup z funkce, čímž její běh končí. V této části kódu byly v průběhu tvorby nástroje provedeny změny v návaznosti na rozšíření funkcí dostupných v jádru nástroje pro tvorbu VMŽI. Součástí nových funkcí je i **VlozitAPridruzeneLokalizovatNetEntity**, která nahrazuje předchozí několikakrokový postup, který zde byl použit dříve. Původní kód byl přeměněn na komentář, aby jej bylo možno případně obnovit (např. z důvodu kompatibility se staršími verzemi jádra nástroje).

Běžnými problémy, které mohou nastat v průběhu běhu funkce, je nezadání hodnoty uživatelem. To zpravidla způsobí úplné selhání běhu funkce, výjimečně pak může běh funkce pokračovat, avšak s nestandardním průběhem a možným chybným vytvořením vkládané instance. Dalším problémem může být chybná volba strany vkládání přestavníku při zadávání uživatelem. Tato chyba nevyvolá nestandardní běh nebo pád skriptu, ale přestavník se vloží chybně (na opačnou stranu nebo přímo do osy koleje). Jediným řešením obou situací je ukončení běhu skriptu, vrácení výsledků posledního běhu tohoto skriptu a nové zadání.

Seznam užitých funkcí:

- `VytvoritVektorGoIdAssociatedPositionPodleVektoruGoIdPositioningNetElement(vektorGoIdPositioningNetElement)`
- `FiltrovatVektorGoIdPodleHodnotyAtributu(vektorGoId, tabulka, atribut, vektorHodnotAtributu_R)`
- `ZvolitPolozkuZVyskakovaciNabidky(hlavicka_G, vektorVoleb_G)`
- `PrevestGoIdNaId(goIdHorizontalCurve)`
- `VytvoritAssociatedLocationSPrirazenimPridruzenymPrvkum (maticeOPP, associatedLocationType_I)`
- `VlozitAPridruzeneLokalizovatNetEntity(trida_I, goIdAssociatedLocation_I, lokalniMaticePHA_NetEntity)`

Výstupní hodnota:

`goIdSwitchMachine` - Funkce vrací `goId` nově vytvořené instance pro další využití, např. pro propojení přestavníků s pomocí třídy **NetEntityRelation**.

2.6 ZaložitPreddefinovaneObjektyProDSFD2023()

Účel funkce:

Účelem této funkce je automatizovat zakládání instancí třídy **SignalType** při zakládání projektu ve specifickém rozšíření pro DSFD. Tato funkce není přímo dostupná běžným uživatelům, ale je automaticky volána jako poslední krok při zakládání každého nového projektu.

Vstupní parametry:

Tato funkce nemá vstupní parametry.

Vnitřní popis funkce:

Funkce začíná definicí jediné proměnné sloužící k ukládání uživatelské odpovědi, kterou nastaví v této fázi na hodnotu **Null**. Dále funkce zkontroluje, zda již instance třídy **SignalType** nejsou založeny. Pokud nejsou, zobrazí funkce uživateli dotaz, jestli požaduje založení předdefinovaných instancí třídy **SignalType**. V případě kladné odpovědi (hodnota **1**) zavolá opakovaně funkci **ZalozitSignalTypeDSFD** s pevně nastavenými hodnotami parametrů odpovídajícími jednotlivým typům návěstidel popsaných v kap.: 4.4 bakalářské práce. Po proběhnutí posledního volání funkce končí. Vzhledem k tomu, že se jedná o funkci probíhající pouze při zakládání nového projektu v navrhovaném SR běžící bez zásahu uživatele, nejsou očekávány chyby běhu, protože každá chyba běhu, která může nastat, indikuje zásadní problém v předcházejících krocích zakládání projektu vyžadující zcela nové založení.

Seznam užitých funkcí:

- `Rhino.IsLayerEmpty(layerName)`
- `ZalozitSignalTypeDSFD(internalDesignator, longname, signalConstructionType)`

2.7 `ZalozitSignalTypeDSFD(internalDesignator, longname, signalConstructionType)`

Účel funkce:

Účelem této funkce je vytvoření instance třídy **SignalType** sloužící k definici typu návěstidla, který je dále přiřazován každé instanci třídy **TracksideSignal**. Tato funkce není dostupná koncovému uživateli a vyžaduje zadání všech hodnot již na vstupu.

Vstupní parametry:

- `internalDesignator` - Slouží k definici hodnoty v rámci číselníku třídy `SignalType`.
Přípustné hodnoty: $\{1, 2, 3, \dots, N\}$
- `longname` - Slouží k pojmenování dané instance.
Přípustné hodnoty: obecný textový řetězec
- `signalConstructionType` - Slouží k definice konstrukčního typu návěstidel dle specifikací railML. Z těchto specifikací vychází i přípustné hodnoty.¹⁹
Přípustné hodnoty:
 - `board` - hodnota definovaná v railML¹⁹
 - `light` - hodnota definovaná v railML¹⁹
 - `pole` - hodnota definovaná v railML¹⁹
 - `semaphore` - hodnota definovaná v railML¹⁹
 - `virtual` - hodnota definovaná v railML¹⁹

Vnitřní popis funkce:

Jedná se o jednoduchou funkci, která na počátku svého běhu vytvoří potřebné proměnné, převezme definované vstupy, které do jedné z vytvořených proměnných uloží ve

¹⁹Vychází z: `TSignalConstructionType: Infrastructure3. railML Wiki` [online]. 2022, 5. 2. 2022 [cit. 2023-07-11]. Dostupné z: <https://wiki3.railml.org/DataModel/3.2/IS/EARoot/EA414.htm>.

správném formátu. Následně vyžádá vytvoření instance požadované třídy, získá její **gold**, které předá jako výstup. Tím běh této funkce končí. Vzhledem k tomu, že se jedná o funkci probíhající pouze při zakládání nového projektu v navrhovaném SR běžící bez zásahu uživatele, nejsou očekávány chyby běhu, protože každá chyba běhu, která může nastat, indikuje zásadní problém v předcházejících krocích zakládání projektu vyžadující zcela nové založení.

Seznam užitých funkcí:

- `PridatInstanci(trida_C, lokalniVektorPGA, lokalniMaticePHA)`

Výstupní hodnota:

`goldSignalType` - Funkce vrací `gold` nově vytvořené instance.

2.8 `VlozitHorizontalCurveProVyhybkyDSFD()`

Účel funkce:

Tato funkce obsluhuje menu sloužící pro vyvolání funkcí pro vkládání výhybkových konstrukcí způsobem implementovaným pro potřeby specifického rozšíření pro DSFD. Tato funkce není určena pro volání v rámci funkcí implementovaných případnými uživateli.

Vstupní parametry:

Tato funkce nemá vstupní parametry.

Vnitřní popis funkce:

Funkce začíná vytvořením a naplněním proměnných sloužících k zobrazení požadovaných textů v rámci vytvářené nabídky. Následně s využitím standardní funkce nástroje pro tvorbu VMŽI tuto nabídku zobrazí. Dále na základě výstupu z nabídky definující odpověď uživatele zavolá uživatelem požadovanou funkci, které nepředá žádné parametry (všechny parametry jsou nastaveny na hodnotu **Null**). Možnou chybou v případě běhu funkce je vrácení neplatné hodnoty (popř. nenavrácení žádné hodnoty) ze standardní funkce generující menu. V tomto případě je možné očekávat nestandardní běh funkce. Jediným způsobem řešení problému je ukončení skriptu.

Seznam užitých funkcí:

- `ZvolitPolozkuZVyskakovaciNabidky(hlavicka, vektorPripustnychHodnot)`
- `VlozitHorizontalCurveProJednoduchouVyhybkuDSFD(zdrojHPP, referenceVychozihoBodu, vzorVyhybky, smerOdboceni, uzavritLinearNetElement)`
- `VlozitHorizontalCurveProObloukovouVyhybkuDSFD(zdrojHPP, referenceVychozihoBodu, vzorVyhybky, smerOdboceni, uzavritLinearNetElement)`
- `VlozitHorizontalCurveProKrizovatkovouVyhybkuDSFD(zdrojHPP, referenceVychozihoBodu, vzorVyhybky, uzavritLinearNetElement)`
- `VlozitHorizontalCurveProKrizeniDSFD(zdrojHPP, referenceVychozihoBodu, vzorVyhybky, uzavritLinearNetElement)`

2.9 PriraditCastNastupniHranyNastupniHrane()

Účel funkce:

Účelem funkce je vytvoření instance třídy **NetEntityRelation** pro potřebu případu propojování dvojice nástupních hran, z nichž jedna je zcela součástí té druhé (např. v situacích nástupních hran dělených na více částí cestovými navěstidly).

Vstupní parametry:

Tato funkce nemá vstupní parametry.

Vnitřní popis funkce:

Funkce začíná definicí proměnných potřebných pro běh funkce, jmenovitě dvojice proměnných pro uložení vybraných nástupních hran a dvojice proměnných pro uložení vektorů pro vygenerování seznamu přípustných voleb nástupních hran.

Ve druhé části běhu programu je uživatel vyzván, aby vybral ze seznamu existujících instancí třídy **PlatformEdge**, přičemž jako první je vybírána celá nástupní hrana (obsahující dílejší nástupní hrany). K tomuto výběru je využita standardní funkce nástroje pro tvorbu VMŽI. Uživatelova volba je uložena do jedné z proměnných. Následně je vytvořen vektor všech existujících instancí této třídy, z něhož je následně odebrána již zvolená instance, čímž je zabráněno tvoření chybných přiřazení. Následně je uživateli nabídnut výběr části nástupní hrany z vektoru všech instancí s výjimkou té již zvolené. Výsledky obou výběrů jsou následně předány funkci vytvářející generické propojení dvojice instancí specializací třídy **NetEntity** s napevno nastavenou vlastností propojení **aIsPartOfB**. Tímto běh funkce končí.

Možné chyby v průběhu běhu funkce jsou neexistence dvou různých instancí třídy **PlatformEdge**, chybné zadání hodnoty uživatelem (vzniklé např. v důsledku přerušování zadávání) apod. Druhá z uvedených možností způsobí nestandardní chování funkce nebo pád skriptu a jediným řešením je ukončení běhu nástroje. První z popsaných chyb je ještě závažnější, protože standardní funkce jádra nástroje využitá pro výběr instancí vynucuje výběr možnosti, přestože žádnou možnost k výběru nenabídne, ale místo toho pouze zobrazí varování o absenci možnosti výběru, které se však po potvrzení z důvodu vynucení výběru znovu zobrazí. Tato chyba znemožňuje ukončení běhu funkce a jediným způsobem řešení je vynucené ukončení SW Rhinoceros z prostředí operačního systému.

Seznam užitých funkcí:

- `VybratInstanciTridy(identifikaceVystupu, trida_S, vektorVybratelnychGoId, filtrTypuObjektu, nabizetVytvoreniNoveInstance, nabizetVyberGeometrickehoObjektu, maticePHA_S)`
- `VytvoritVektorVsechGoIdTridy(trida_S)`
- `OdebratPrvkyZVektoru (vektorA, vektorB)`
- `PriraditSitovouEntituSitoveEntite(tridaA, goIdNetEntityA, tridaB, goIdNetEntityB, relationType)`