



**ČESKÉ VYSOKÉ UČENÍ TECHNICKÉ V PRAZE**  
**Fakulta elektrotechnická**  
**Katedra ekonomiky, manažerství a humanitních věd**

**Porovnání vlastností SW produktů pro lokální virtualizaci**

**Properties comparison of SW for local virtualization**

Bakalářská práce

Studijní program: Softwarové technologie a management

Studijní obor: Manažerská informatika

Vedoucí práce: Ing. Jaroslav Šafránek, CSc.

**Viktor Tyleček**

**Praha 2017**

## I. OSOBNÍ A STUDIJNÍ ÚDAJE

Příjmení: **Tyleček** Jméno: **Viktor** Osobní číslo: **410307**  
Fakulta/ústav: **Fakulta elektrotechnická**  
Zadávací katedra/ústav: **Katedra ekonomiky, manažerství a humanitních věd**  
Studijní program: **Softwarové technologie a management**  
Studijní obor: **Manažerská informatika**

## II. ÚDAJE K BAKALÁŘSKÉ PRÁCI

Název bakalářské práce:

**Porovnání vlastností SW produktů pro lokální virtualizaci**

Název bakalářské práce anglicky:

**Properties comparison of SW for local virtualization**

Pokyny pro vypracování:

- vyhledání a popis SW pro lokální virtualizaci
- návrh typických uživatelů
- vícekriteriální porovnání produktů
- doporučení pro potenciální uživatele
- programová realizace vybrané metody vícekriteriálního hodnocení

Seznam doporučené literatury:

RUEST, D., RUEST, N. Virtualizace: Podrobný průvodce. 1. vydání. Brno: Computer Press, 2010. 408 s. ISBN 978-80-251-2676-9  
FIALA, Petr; JABLONSKÝ, Josef; MAŇAS, Miroslav. Vícekriteriální rozhodování. Praha: Vysoká škola ekonomická v Praze, 1994. 31 7079-748-7.  
KOŘENÁŘ, Václav; LAGOVÁ, Milada. Optimalizační metody. Praha: Vysoká škola ekonomická v Praze, 2003. 187 s. ISBN 80-245-0609-2  
Technická dokumentace srovnávaných virtualizačních produktů.

Jméno a pracoviště vedoucí(ho) bakalářské práce:

**Ing. Jaroslav Šafránek CSc., katedra ekonomiky, manažerství a humanitních věd FEL**

Jméno a pracoviště druhého(ho) vedoucí(ho) nebo konzultanta(ky) bakalářské práce:

Datum zadání bakalářské práce: **31.01.2017** Termín odevzdání bakalářské práce: **26.05.2017**

Platnost zadání bakalářské práce: **28.05.2018**

Podpis vedoucí(ho) práce

Podpis vedoucí(ho) ústavu/katedry

Podpis děkana(ky)

## III. PŘEVZETÍ ZADÁNÍ

Student bere na vědomí, že je povinen vypracovat bakalářskou práci samostatně, bez cizí pomoci, s výjimkou poskytnutých konzultací. Seznam použité literatury, jiných pramenů a jmen konzultantů je třeba uvést v bakalářské práci.

Datum převzetí zadání

Podpis studenta

## Prohlášení

Prohlašuji, že jsem předloženou práci vypracoval samostatně a že jsem uvedl veškeré použité informační zdroje v souladu s Metodickým pokynem o dodržování etických principů při přípravě vysokoškolských závěrečných prací.

V Praze dne 26. května 2017

.....

Podpis autora práce

## Poděkování

Děkuji vedoucímu mé bakalářské práce, Ing. Jaroslavu Šafránkovi, CSc., za odborné vedení a konzultace při jejím zpracování.

## Abstrakt

Tato bakalářská práce se zabývá programy určenými pro lokální virtualizaci a jejich vhodností pro různé potenciální uživatele.

V úvodu je vysvětlen princip virtualizace a její využití. Další část se zabývá návrhem typických uživatelů virtualizačního softwaru a jejich požadavků. Následuje popis a porovnání vlastností vybraných virtualizačních programů z různých hledisek.

Z těchto hledisek jsou sestavena kritéria, která jsou použita v provedeném vícekritériálním porovnání virtualizačních programů podle lexikografické metody, metody váženého součtu pořadí, AGREPREF, ORESTE a ELECTRE III. Metoda ELECTRE III byla také naprogramována do existující školní aplikace a v této práci popsán průběh implementace.

Poslední část práce obsahuje doporučení virtualizačních programů navrženým typickým uživatelům.

## Klíčová slova

Virtualizace, virtualizační software, lokální virtualizace, virtuální stroj, hypervizor, hardwarová virtualizace, VirtualBox, VMware Player, Windows Virtual PC, typický uživatel, vícekritériální hodnocení, doporučení virtualizačního softwaru

## Abstract

This Bachelor thesis deals with the computer programs for local virtualization and how suitable they are for different possible users.

The introduction explains the principle of virtualization and its usage. The next part is about the design of typical users who use virtualization software and their requirements. The following section describes the virtualization program's features and compares them from various aspects.

The criteria made from these aspects are then used in a multiple-criteria decision analysis by lexicographic method, weighted rank-sum method, AGREPREF, ORESTE and ELECTRE III. Method ELECTRE III was also programmed into the existing school application and the implementation process described in this thesis.

The final part of this thesis contains recommendations of virtualization programs to designed typical users.

## Key words

Virtualization, virtualization software, local virtualization, virtual machine, hypervisor, hardware virtualization, VirtualBox, VMware Player, Windows Virtual PC, typical user, multiple-criteria decision analysis, virtualization software recommendations

## Obsah

1	Úvod .....	1
1.1	Využití virtualizace .....	3
2	Návrh typických uživatelů a požadavků .....	4
2.1	První uživatel .....	4
2.2	Druhý uživatel .....	4
2.3	Třetí uživatel .....	5
2.4	Vymezení cílů .....	6
3	Porovnání vlastností virtualizačních programů .....	8
3.1	VirtualBox .....	8
3.1.1	Podpora operačních systémů (hostujících i hostitelských) .....	8
3.1.2	Pokročilost emulace .....	8
3.1.3	Správa virtuálních strojů .....	9
3.1.4	Další funkce .....	10
3.1.5	Uživatelská přívětivost .....	10
3.1.6	Hardwarové požadavky .....	10
3.1.7	Podpora od výrobce .....	11
3.1.8	Četnost aktualizací .....	11
3.1.9	Popularita a rozšířenost .....	11
3.1.10	Cena .....	12
3.2	VMware Workstation Player .....	12
3.2.1	Podpora operačních systémů (hostujících i hostitelských) .....	12
3.2.2	Pokročilost emulace .....	12
3.2.3	Správa virtuálních strojů .....	13
3.2.4	Další funkce .....	14
3.2.5	Uživatelská přívětivost .....	14
3.2.6	Hardwarové požadavky .....	14

3.2.7	Podpora od výrobce .....	15
3.2.8	Četnost aktualizací.....	15
3.2.9	Popularita a rozšířenost .....	15
3.2.10	Cena .....	16
3.3	Windows Virtual PC .....	16
3.3.1	Podpora operačních systémů (hostujících i hostitelských).....	16
3.3.2	Pokročilost emulace.....	16
3.3.3	Správa virtuálních strojů.....	17
3.3.4	Další funkce .....	18
3.3.5	Uživatelská přívětivost .....	18
3.3.6	Hardwarové požadavky .....	18
3.3.7	Podpora od výrobce .....	19
3.3.8	Četnost aktualizací.....	19
3.3.9	Popularita a rozšířenost .....	19
3.3.10	Cena .....	19
4	Vícekritériální porovnání virtualizačních programů .....	20
4.1	Vybraná kritéria .....	20
4.2	Vysvětlení kritérií a stanovení způsobů zjištění jejich hodnot .....	21
4.3	Ohodnocení virtualizačních programů.....	24
4.4	Stanovení vah kritérií pro typické uživatele .....	25
4.5	Metody vícekritériálního hodnocení .....	26
4.5.1	Lexikografická metoda .....	27
4.5.2	Metoda váženého součtu pořadí .....	28
4.5.3	Metoda AGREPREF .....	29
4.5.4	Metoda ORESTE.....	30
4.5.5	Metoda ELECTRE III .....	31
5	Implementace ELECTRE III .....	32



5.1	Realizace .....	32
6	Doporučení pro typické uživatele.....	33
6.1	První uživatel .....	33
6.2	Druhý uživatel.....	34
6.3	Třetí uživatel .....	35
7	Závěr.....	36
8	Zdroje .....	37
8.1	Internetové zdroje .....	37
8.2	Odborná literatura .....	40
9	Seznam příloh.....	41
10	Přílohy .....	42

## Seznam tabulek

<b>Tabulka 1:</b> Srovnávací kritéria .....	20
<b>Tabulka 2:</b> Ohodnocení virtualizačních programů .....	24
<b>Tabulka 3:</b> Stanovení vah kritérií pro jednotlivé typické uživatele .....	26
<b>Tabulka 4:</b> Pořadí důležitosti kritérií .....	27
<b>Tabulka 5:</b> Výsledné pořadí z lexikografické metody .....	28
<b>Tabulka 6:</b> Výsledné pořadí z metody váženého součtu pořadí .....	28
<b>Tabulka 7:</b> Výsledné pořadí z metody AGREPREF .....	29
<b>Tabulka 8:</b> Pořadí důležitosti kritérií pro jednotlivé uživatele .....	30
<b>Tabulka 9:</b> Výsledné pořadí z metody ORESTE .....	31
<b>Tabulka 10:</b> Výsledné pořadí z metody ELECTRE III .....	31
<b>Tabulka 11:</b> Výsledné pořadí virtualizačních programů pro prvního uživatele .....	33
<b>Tabulka 12:</b> Výsledné pořadí virtualizačních programů pro druhého uživatele .....	34
<b>Tabulka 13:</b> Výsledné pořadí virtualizačních programů pro třetího uživatele .....	35

## Seznam obrázků

<b>Obrázek 1:</b> Znázornění virtualizace .....	2
<b>Obrázek 2:</b> Dekompozice cílů .....	7

## 1 Úvod

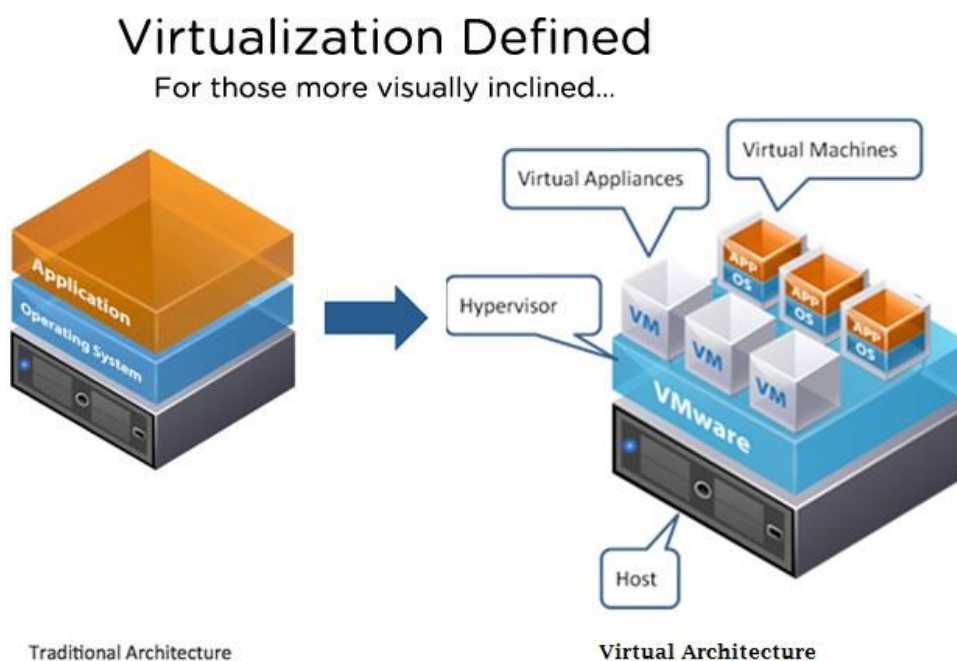
Virtualizace je dnes velmi diskutované téma na poli technologií. Existují různé druhy virtualizace. V této práci se zaměřím pouze na virtualizaci lokální. Co se vlastně pod pojmem lokální virtualizace skrývá? V tradičním modelu fungování běží na počítači operační systém, nad kterým jsou prováděny procesy jednotlivých aplikací s jejich grafickým rozhraním (user interface), které je zobrazováno na monitoru počítače.

Použitím virtualizace se přeruší přímé spojení mezi fyzickou vrstvou (hardwarem), operačním systémem, aplikací a monitorem.

Virtualizace je tedy proces oddělení (abstrakce) softwarové vrstvy počítače od fyzické vrstvy (hardwaru) počítače. Mezi ně je vložena vrstva nová, která jedná jako prostředník mezi nimi.

Tento prostředník se nazývá hypervizor (hypervisor) nebo virtuální strojový monitor (virtual machine monitor) a má podobu počítačového programu, který umožní vytvoření a správu tzv. virtuálních strojů (virtual machine). To jsou kompletní virtuální počítače včetně virtuálního hardwaru, které běží jako klasické aplikace spouštěné na reálném systému. Operačnímu systému, na kterém virtuální stroj běží, tj. ten který běží na reálném počítači, se říká hostitelský (host). Virtuálnímu operačnímu systému se říká hostující (guest). Tomuto hostujícímu systému se virtuální hardware jeví jako reálný. Virtuální hardware poskytovaný virtuálním strojem je mapován na reálný hardware na fyzickém stroji. Tento typ virtualizace se označuje jako hardwarová virtualizace.

**Obr. č. 1:** Znárodnění virtualizace



Zdroj: [1]

Rozlišuje se několik druhů hardwarové virtualizace. Prvním takovým je plná virtualizace, kdy virtuální stroj poskytuje všechny potřebný virtuální hardware pro běh nějak upraveného hostujícího operačního systému. U desktopů nejpoužívanější způsob virtualizace. Největší slabinou je vysoká náročnost vstupně-výstupních operací.

Dalším druhem je částečná virtualizace, historický předchůdce plné virtualizace. Je daleko jednodušší ji implementovat, ale neumožňuje běh celého operačního systému. Lze tedy provozovat pouze aplikace, které nevyžadují nepodporované části hardwaru. Dále následuje paravirtualizace, při níž virtuální stroj nemusí nutně simulovat hardware, ale místo toho nabídne speciální API. To bude využito hostovaným operačním systémem, který se ale napřed musí upravit, aby toto API volal namísto klasických instrukcí operačního systému (např. cli nahradit za vm\_handle\_cli()). Při tomto druhu virtualizace tedy musí mít operační systém dostupný zdrojový kód, aby úpravy mohly být provedeny a systém poté znovu zkompileován. Výhodou je ale lepší výkon než v případě plné virtualizace.

V této práci se budu zabývat porovnáním softwaru označovaného jako virtuální strojový monitor schopného poskytnout plnou hardwarovou virtualizaci.

Umožňuje běh většiny operačních systémů nezávisle na hardwaru počítače. Lze tak třeba provozovat Windows na počítačích od firmy Apple.

Slovem lokální se rozumí, že virtualizační software běží právě na jednom konkrétním stroji, kde virtualizaci chceme provozovat. Hlavní výhodou tohoto přístupu je, že není potřeba žádné síťové spojení s vnějším světem (LAN, WAN, Internet) a virtualizace je plně v roli uživatele daného počítače.

## 1.1 Využití virtualizace

Mezi nejpobulárnější příklady patří možnost experimentovat s různými druhy operačních systémů bez nutnosti opuštění stávajícího operačního systému. To se hodí například při zvažování přechodu na jiný operační systém nebo potřebě vyzkoušet ještě neoficiální a neodladěnou verzi nového operačního systému.

S různými virtuálními operačními systémy se automaticky nabízí také možnost instalace a používání aplikací pro ně určených. Velmi užitečná věc, jelikož zdaleka ne všechny aplikace jsou dostupné na všechny operační systémy. Vývojářům umožňuje testovat software na různých platformách.

Virtuální stroje jsou izolovány od zbytku systému (sandbox), což znamená, že software instalovaný na virtuálním stroji nemůže nijak porušit reálný systém. Hodí se tedy na testování programů, kterým nedůvěřujeme a chceme je vyzkoušet bez rizika zavirování nebo poškození reálného operačního systému.

Sandboxing umožňuje bezpečně používat i dále neaktualizované a zastaralé operační systémy, které představují bezpečnostní riziko. To je užitečné například v případě, že jsme závislí na nějaké aplikaci, která přímo tento operační systém pro svůj běh vyžaduje. V neposlední řadě také použití pro oddělení soukromého prostředí od pracovního, pokud nemáme k dispozici pracovní počítač.

Dále třeba jednodušší instalace komplikovaných softwarových řešení vyžadujících od uživatele hodně konfigurování prostředí, ve kterém běží, např. řešení kompletního mailového serveru. Stačí, když výrobce vytvoří virtuální stroj, kde vše nainstaluje a nakonfiguruje a uživateli distribuuje už takovéto hotové řešení.

Virtualizace také může přispět k ušetření nákladů na hardware a poplatky za elektřinu. Dnešní počítače většinu času využívají pouze zlomek svého výkonu a běží pouze s malým zatížením dostupných systémových prostředků. Dochází tak k plýtvání potenciálu hardware stejně jako elektrické energie. Řešení v podobě virtualizace spočívá v provozování více virtuálních strojů na hrstce výkonných reálných strojích.

## 2 Návrh typických uživatelů a požadavků

V této práci se budu věnovat výběru vhodného lokálního virtualizačního softwaru pro tři typické uživatele. Vzhledem k tomu, že každý uživatel bude mít jiné potřeby, určím požadavky uživatele a na jejich základě vyberu co nejvhodnější software.

### 2.1 První uživatel

Prvním typickým uživatelem je člověk, který virtualizační software používá při každodenní práci. Má bohaté znalosti ohledně virtualizace a očekává pokročilé funkce, které mu usnadní práci. Rád si sám vše nakonfiguruje a vyladí do posledního detailu podle sebe. Pod takovýmto uživatelem si lze představit například programátora/testera, který vyvíjí/testuje programy pro rozdílné operační systémy nebo počítačového hackera, který zkoumá zranitelnost různých systémů.

Nároky na virtualizační software:

- otevřený zdrojový kód
- vlastní kompilace virtualizačního programu
- široká podpora operačních systémů (jak hostujících, tak hostitelských)
- pokročilá podpora nativního hardwaru
- časté aktualizace a vylepšení
- přenositelnost vytvořených virtuálních strojů na jiné počítače
- cena

### 2.2 Druhý uživatel

Druhým typickým uživatelem je IT nadšenec, který rád zkouší nové věci v oblasti informačních technologií. Má základní znalosti ohledně virtualizace, tj. chápe v čem spočívá hlavní princip, ale nikdy podobný software nezkoušel a nezná ani technické detaily. Chtěl by co nejintuitivnější prostředí pro začínajícího uživatele. Pokročilé funkce ho zatím nezajímají, chce si se softwarem hlavně pohrát a načerpat praktické zkušenosti do jeho všeobecného přehledu IT technologií. Nerad by do virtualizačního softwaru investoval peníze.

Nároky na virtualizační software:

- intuitivní uživatelské rozhraní
- srozumitelná dokumentace od výrobce softwaru
- široká podpora hostujících operačních systémů
- rozšířenost v IT komunitě
- cena zdarma nebo alespoň nabídka trial verze

### 2.3 Třetí uživatel

Posledním uživatelem je malá firma do 10 zaměstnanců, která chce ušetřit peníze za správu počítačů a také čas potřebný ke konfiguraci jednotlivých počítačů na firemní prostředí. Ideálně jednou nakonfigurovat a dále už pouze používat na všech počítačích. Vedení chce mít možnost dohlížet na zaměstnance, tj. vědět co na počítači dělají. Firma zaměstnancům umožňuje home office, takže by zaměstnanci měli mít možnost se vzdáleně připojit k virtuálnímu stroji z domova. Firemní data by zároveň měla být chráněna proti okopírování zaměstnancem. Ve firmě se používá Microsoft Windows, tudíž je požadována co nejlepší podpora právě tohoto operačního systému jako hostujícího. Od virtualizačního softwaru se očekává i hardwarová a prostorová nenáročnost, jelikož počítače ve firmě nejsou moc výkonné. Je vyžadováno stabilní a časem prověřené řešení.

Nároky na virtualizační software:

- podpora od výrobce, případně od komunity
- jednoduché používání
- obnovení virtuálního stroje (disaster recovery)
- přenosnost virtuálního stroje
- hardwarová nenáročnost
- prvotřídní podpora Microsoft Windows
- vzdálené připojení k virtuálnímu stroji
- správa virtuálních strojů
- zabezpečení dat
- cena

## 2.4 Vymezení cílů

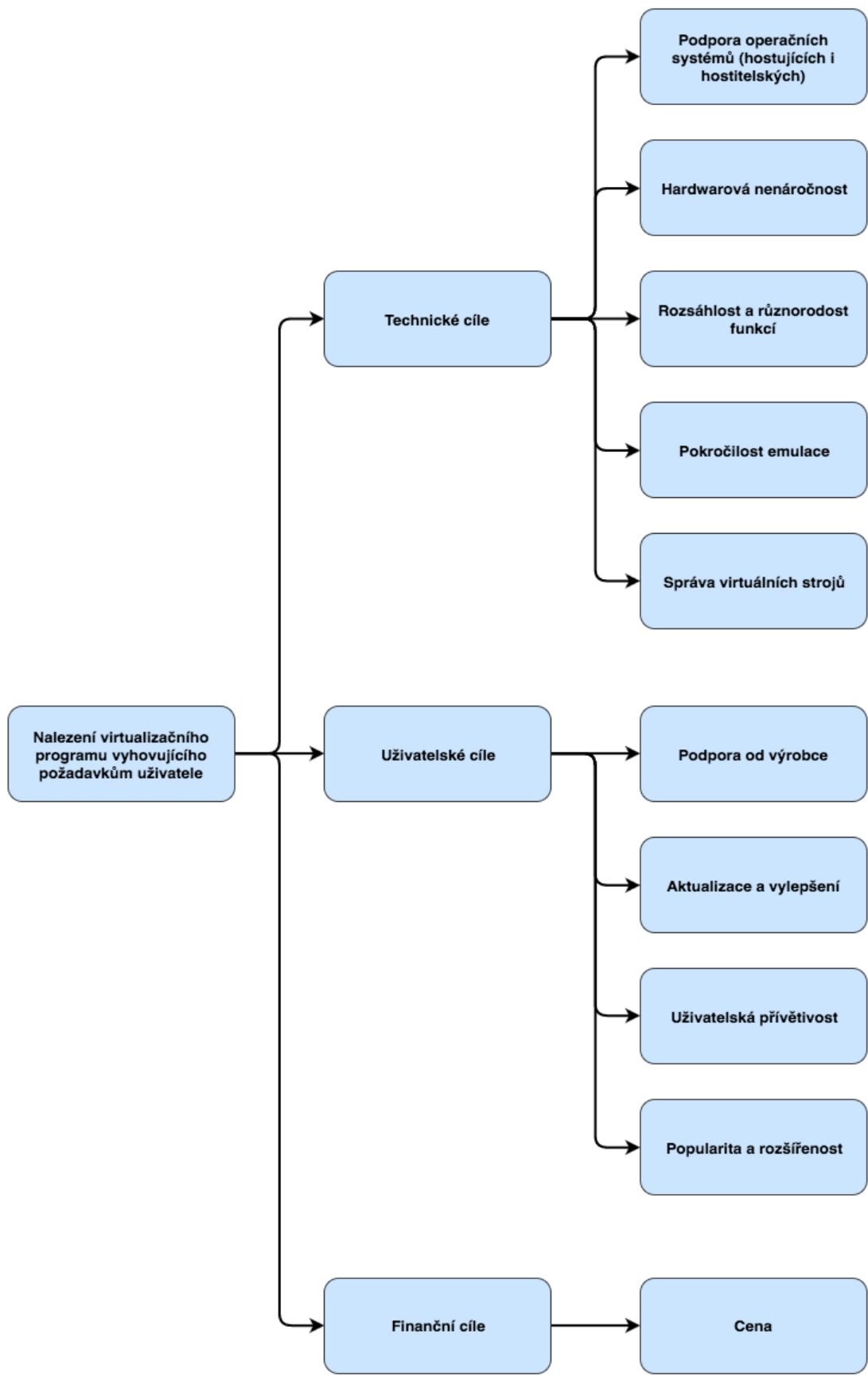
Než začnu porovnávat jednotlivé virtualizační programy, je třeba ještě stanovit cíle. Z těchto cílů budou následně vycházet kritéria potřebná k vícekritériálnímu porovnání. Hlavním cílem je nalezení co nejvhodnějšího virtualizačního programu pro každého navrženého typického uživatele dle jeho požadavků. Tento cíl se dále dekomponuje na dílčí cíle, které nyní stručně popíšu.

Virtualizační program by měl podporovat minimálně nejrozšířenější operační systémy dneška (Windows, Linux, macOS), a to jako hostitelské i hostující. Důležitá je pro uživatele také pokročilost emulace na virtualizovaném stroji, tj. jak moc se virtualizované prostředí dokáže přiblížit nativnímu. Dalším cílem je umožnění správy virtuálních strojů, která zahrnuje například možnost vytváření a spravování skupin, přenositelnost, vzdálené připojení, obnovení nebo nastavení práv k užívání virtuálního stroje. Program by toho obecně měl hodně umět a zároveň zůstat hardwarově nenáročný. Mimo toho jde uživateli také o intuitivní ovládání programu, které bude přispívat k jednoduchému používání bez nutnosti nejdříve pročítat dokumentaci.

Uživatel požaduje i podporu od výrobce virtualizačního programu. Ta může mít formu kvalitně a detailně zpracované dokumentace, fóra pro uživatele, nabídku školení nebo poskytnutí tutoriálů. Výrobce by měl program nadále vyvíjet a opravovat chyby, což se projeví ve formě častých aktualizací. Pokud je program navíc populární a rozšířený, uživatel bude mít i podporu od komunity, která program používá, a nebude tak muset spoléhat s podporou pouze na výrobce. Posledním velmi důležitým cílem je, aby byl program co nejlevnější, ideálně zdarma.



Obr. č. 2: Dekompozice cílů



Zdroj: vlastní zpracování

## 3 Porovnání vlastností virtualizačních programů

Mezi nejznámější a nejpoužívanější softwarová řešení pro lokální virtualizaci patří Windows Virtual PC od společnosti Microsoft, VirtualBox od společnosti Oracle a VMware Workstation Player od společnosti VMware.

VirtualBox v sobě ukrývá softwarové balíčky zvané VirtualBox Guest Additions, které je možné volitelně nainstalovat. Tyto balíčky slouží k lepší integraci a komunikaci s hostitelským systémem. K VirtualBoxu je nabízen i rozšiřující balíček VirtualBox Extension Pack přímo od společnosti Oracle. Při porovnávání budu předpokládat nainstalování obou rozšiřujících doplňků. V programu Workstation Player je možné při vytváření každého virtuálního stroje nainstalovat VMware Tools, což je sada nástrojů poskytující lepší integraci hostovaného operačního systému a správu virtuálního stroje. V porovnání budu předpokládat její nainstalování. U Windows Virtual PC zase existuje možnost nainstalovat ve virtuálním stroji Integration Components, balíček vylepšující interakci mezi virtuálním strojem a hostitelským systémem. Budu uvažovat jeho instalaci. Nyní postupně rozeberu vlastnosti uvedených programů detailněji.

### 3.1 VirtualBox

#### 3.1.1 Podpora operačních systémů (hostujících i hostitelských)

Program může být nainstalován na většinu dnes používaných operačních systémů: Windows, Linux, OS X (macOS), Solaris a OpenSolaris. Zmíněné operační systémy podporuje i jako hostující. Od verze 5 již není podporován Windows XP jako hostitelský operační systém a uživatel tedy musí použít starší verzi VirtualBoxu.

#### 3.1.2 Pokročilost emulace

Téměř všechny hardware na hostitelském počítači je programem virtualizován. Viditelnost hardwaru záleží na konfiguraci virtuálního stroje. Co se týká vstupních zařízení, tak je standardně emulována klávesnice i myš připojené pomocí PS/2 a to u všech operačních systémů bez ohledu na jejich verzi. VirtualBox navíc poskytuje i podporu pro USB zařízení, a to včetně rozhraní USB 3.0 u novějších verzí operačních systémů. Data z virtuálního stroje se defaultně ukládají na virtuální pevný disk, což je ve skutečnosti soubor s příponou VDI (VirtualBox Disk Image). Je poskytována také

podpora pro pevné disky vytvořené v konkurenčních programech – VMware Player (formát VMDK) a Windows Virtual PC (formát VHD). Existuje i možnost nastavení přístupu k fyzickému pevnému disku. VirtualBox umí emulovat dnes nejběžnější řadiče pevných disků (IDE, SATA, SCSI, NVMe), takže případná migrace virtuálních strojů na jiné hypervizory by měla být bezproblémová. K účelu připojení do sítě VirtualBox nabídne až 8 virtuálních PCI Ethernet karet pro každý virtuální stroj. Na každé síťové kartě individuálně je možné vybrat fyzický hardware, který se má virtualizovat, stejně jako virtualizační mód, ve kterém bude virtuální síťová karta operovat. Přes grafické rozhraní programu jdou ale nastavit „pouze“ 4 karty, všech 8 se dá nastavit přes příkazovou řádku. Pro emulaci zvukové karty stačí v nastavení programu povolit viditelnost připojené zvukové karty pro virtuální stroj a vybrat si její ovladač, který bude k emulaci použit. Virtuální grafická karta použitá ve VirtualBoxu není založena na svém fyzickém protějšku, ale jedná se o čistě syntetické zařízení poskytující kompatibilitu se standardem VGA. Vzhledem k nainstalovaným Guest Additions je podporována i hardwarově akcelerovaná 3D grafika.

### 3.1.3 Správa virtuálních strojů

Uživatel může virtuální stroje organizovat do skupin a poté celé skupiny ovládat jako jednotný funkční celek. Virtuální stroj v tomto programu vytvořený lze jednoduše přenášet na další počítače, na kterých běží jakýkoliv virtualizační software podporující standardizovaný formát OVA (Open Virtualization Format). Aktuálně používaná virtuální záznamová zařízení (pevné disky, CD/DVD, diskety) lze přehledně spravovat ve Virtual Media Manageru. Virtuální pevný disk lze dokonce zašifrovat algoritmem AES (šifrovací klíč 128 nebo 256 bitů). Program dále umožňuje vytvářet takzvané snapshoty, což je uložení aktuálního stavu virtuálního stroje pro možnost se k tomuto stavu kdykoliv později vrátit. V případě potřeby se uživatel může vzdáleně připojit k virtuálnímu stroji a ovládat jej stejným způsobem jako by byl lokální. Komunikace je realizována pomocí VirtualBox Remote Display Protocolu (VRDP), který je rozšířením Remote Display Protocolu (RDP) od Microsoftu. Spojení je šifrované s možností autentizace.

#### 3.1.4 Další funkce

Díky Guest Additions nabízí VirtualBox další zajímavé funkce. Mezi ně patří integrace kurzoru myši, aby již nebylo nutné zmáčknout klávesu pro „osvobození“ kurzoru z hostujícího operačního systému. Přejechy kurzoru z hostujícího na hostitelský systém, a naopak, jsou tak plynulé a ničím nenarušené. S myší souvisí také obousměrná podpora drag and drop, kdy lze kopírovat nebo otevírat soubory a složky z jednoho systému na druhém jako by se jednalo o drag and drop operaci na cílovém operačním systému. Další funkcí jsou sdílené složky, které poskytují snadný způsob, jak sdílet soubory mezi hostujícím a hostitelským operačním systémem. Složka na hostitelském systému určená ke sdílení je poté dostupná v hostujícím systému jako síťové úložiště. Zajímavé je i zobrazení aplikace běžící na hostovaném systému jako by běžela přímo nativně na hostitelském systému (seamless windows). Dále také synchronizace času hostujícího systému na základě hostitelského, sdílená schránka pro kopírování, spouštění aplikací na hostovaném systému z hostitelského a memory ballooning – přidělení další paměti běžícímu virtuálnímu stroji, kterému už jeho alokovaná paměť nestačí, od jiného virtuálního stroje, a to vše za běhu bez nutnosti vypnutí.

#### 3.1.5 Uživatelská přívětivost

Uživatelské grafické rozhraní programu hodnotím kladně. Je srozumitelné i pro začátečníka díky dostatečně velkým grafickým ikonám s popisky akce a přehlednému uspořádání potřebných informací. Pro zkušenější uživatele je možnost ovládat program přes příkazovou řádku a detailně si vyladit pokročilé nastavení, které není přes grafické rozhraní dostupné, podle sebe. Program jsem si zkušebně nainstaloval a zkusil v něm zprovoznit virtuální stroj se systémem Linux. Ovládání mi přišlo přirozené a nemusel jsem ani studovat dokumentaci.

#### 3.1.6 Hardwarové požadavky

Ke spuštění VirtualBoxu by měl stačit „rozumně“ výkonný procesor s typem architektury x86 (citováno z [18]). Tuto podmínku splňuje prakticky každý dnes dostupný procesor, ať už Intel nebo AMD. Plusem je, že procesor v sobě nemusí mít podporu pro hardwarovou virtualizaci (u Intelu VT-x, u AMD AMD-V). VirtualBox totiž zvládá čistě softwarovou virtualizaci, takže může fungovat i na starších procesorech. Paměti typu RAM je zapotřebí nejméně 512 MB. To je ale opravdové minimum. Potřebnou paměť

pro optimální běh lze snadno zjistit tak, že sečteme hodnoty paměti, které požaduje hostitelský a hostující operační systém. Místa na pevném disku zabere samotný VirtualBox málo, typická instalace pouze okolo 30 MB. Aby mohl být vůbec nainstalován, musí na hostitelském systému běžet podporovaný operační systém.

### 3.1.7 Podpora od výrobce

Oracle nabízí na webových stránkách VirtualBoxu velmi dobrou dokumentaci k tomuto programu ve formě přehledně členěného uživatelského manuálu. V manuálu je uživatel veden od počátečních kroků jako instalace, spuštění a vysvětlení základní funkcionality programu až po detailní popisy jednotlivých funkcí nebo možnosti pokročilé konfigurace virtuálních strojů. Manuál je tedy vhodný jak pro začátečníky, tak i pokročilé uživatele. Dále jsou na stránkách k nalezení často kladené otázky uživatelů (FAQ) nebo tutoriály zaměřené spíše na pokročilejší uživatele. Případně lze využít i oficiální fórum, které má aktivní a početnou (přes 90 000) komunitu uživatelů. Kdyby chtěl uživatel ještě víc, může si zakoupit oficiální podporu přímo od Oraculu. Musí mít ale pořízenou licenci na VirtualBox Extension Pack, jelikož je s ní podpora svázaná. Tato placená podpora se vztahuje nejen na Extension Pack, ale i na celý VirtualBox. Zahrnuje 24/7 pomoc přes telefon nebo web od Oracle odborníků, dále přístup k nejnovějším patchům, aktualizacím a releasům.

### 3.1.8 Četnost aktualizací

VirtualBox je aktivně vyvíjen a opravován. Aktuálně poslední hlavní (major) update na verzi 5.1 byl 12.7.2016 a od té doby k březnu 2017 vyšlo už 9 menších (minor) updatů. Tyto menší updaty opravující nalezené chyby v programu (bugy) vycházejí většinou každý měsíc. Díky tomu, že je VirtualBox open-source, mohou na chyby upozornit (nebo je dokonce opravit) i sami uživatelé.

### 3.1.9 Popularita a rozšířenost

VirtualBox existuje již od roku 2007 a za tu dobu si dokázal získat početnou uživatelskou základnu. Konkrétní čísla nejsou k dispozici, ale jeho popularita je vysoká už jen podle vyhledávání na Internetu. Na jeho rozšíření má podíl i společnost Oracle, která je velkým hráčem na poli technologií a pod kterou program spadá. S jistotou ho lze zařadit

do top 3 programů pro daný typ virtualizace. Jenom v ČR má ke dni 5.4.2017 podle serverů slunecnice.cz a stahuj.cz dohromady 263 764 stažení.

#### 3.1.10 Cena

Samotný VirtualBox (včetně Guest Additions) je zdarma a open-source. Rozšiřující balíček VirtualBox Extension Pack je zdarma pouze pro nekomerční účely. V opačném případě musí být zakoupena Oracle VM VirtualBox Enterprise licence. Navíc pouze k licenci je možné zakoupit i podporu přímo od firmy Oracle. Licenci je nutné pořídit alespoň pro 100 uživatelů, pro každého stojí 50\$ a není časově nijak omezená. Minimální cena pouze za licenci bude tedy 5000\$, což zejména pro malé firmy není moc výhodné. Podpora vyjde za první rok na 11\$/uživatel, tedy 1100\$ pro minimální počet uživatelů. Po uplynutí prvního roku stojí podpora 1199\$ na další rok nebo 3597\$ na další tři roky. Uvedené údaje jsem čerpal přímo z oficiálního Oracle e-shopu.

## 3.2 VMware Workstation Player

### 3.2.1 Podpora operačních systémů (hostujících i hostitelských)

Jako hostitelské operační systémy program podporuje Linux a Microsoft Windows od verze 7. Podporovány jsou ale pouze 64 bitové verze zmíněných operačních systémů. V případě potřeby použít program na 32 bitové verzi je nutné použít jeho starou verzi – Player 6 Plus. Mezi podporované hostující operační systémy patří Microsoft Windows 2000 a vyšší, Linux, Solaris a FreeBSD. Hostující operační systémy mohou být i 32 bitové.

### 3.2.2 Pokročilost emulace

Všechny funkce procesoru i jeho výkon jsou plně dostupné i ve virtuálním stroji a pokud má procesor alespoň 2 logická jádra, je možné virtuálnímu stroji přidělit až 16 virtuálních jader. Paměť typu RAM je také celá k dispozici. V případě 64 bitového hostitelského systému lze přidělit maximálně 64 GB RAM každému virtuálnímu stroji, v případě 32 bitového 8 GB RAM. Data z virtuálního stroje se ukládají na virtuální disk, který se skládá z 1 nebo více speciálních souborů umístěných na fyzickém pevném disku. Virtuálnímu stroji může být namapován i fyzický harddisk, ale pouze ve verzi Workstation Player, která je součástí Workstation Pro. Podporovány jsou běžné řadiče

IDE, SCSI a SATA. Výhodou je, že zvolená kapacita virtuálního disku, která může být až 8 TB, není prealokována na fyzickém disku a narůstá postupně. Program podporuje více připojených monitorů, a to s vysokým rozlišením až 4K (3840 x 2160). Umožňuje i akcelеровanou 3D grafiku s použitím DirectX 9 a 10 nebo OpenGL 3.3. Dále lze nastavit přidělení grafické paměti virtuálnímu stroji až do výše 2 GB. Virtuální stroj používá virtuální zvukovou kartu, kterou si lze zvolit. Ovladače potom mapují zvuk z této karty na reálnou. Je podporována i pulzně kódová modulace (PCM), což umožňuje např. přehrávání WAV a MP3 souborů nebo RealMedia formátu. Workstation Player poskytuje virtuálním strojům skrze virtuální síťové adaptéry 3 typy připojení k síti – bridge, překlad síťových adres (NAT) a host-only (uzavřená síť kompletně uvnitř hostitelského systému). Každý virtuální stroj může obsahovat až 10 virtuálních síťových adaptérů. Potřebný software k síťové konfiguraci je nainstalován na hostitelském systému již při instalaci Workstation Player. Zařízením včetně myši nebo klávesnice připojeným přes USB jsou ovladače automaticky nainstalovány. Program podporuje USB 1.1 a u novějších verzí operačních systémů i USB 3.0 zařízení. Připojený USB flash disk je viditelný buď pro hostovaný nebo hostitelský systém, ale ne pro oba zároveň. Emulace PS/2 hardwaru je pouze částečná. K použití připojené tiskárny ve virtuálním stroji není nutné instalovat žádné další ovladače a může se ihned tisknout. Přímá komunikace hostovaného operačního systému s CD nebo DVD jednotkou umožňuje např. vypalování CD a DVD rovnou z virtuálního stroje. Zajímavostí je i podpora disketové jednotky.

### 3.2.3 Správa virtuálních strojů

Při otevření souboru s virtuálním strojem nebo vytvoření nového stroje je tento virtuální stroj přidán do knihovny obsahující všechny programem evidované virtuální stroje, které odtud lze spouštět a konfigurovat. Vytvořený virtuální stroj je přenositelný na jiné počítače. Kromě virtuálních strojů vytvořených ve Workstation Player (formát VMX) lze importovat také virtuální stroje vytvořené ve Windows Virtual PC (formát VMC), platformě nezávislé (formát OVF, OVA) nebo obrazy systému z programu Symantec Backup Exec System Recovery (formát SV2I). Programátoři mohou využít VMware VIX API k napsání vlastních programů, které automatizují operace virtuálního stroje. Workstation Player neumožňuje vytváření zašifrovaných virtuálních strojů. Při licencování programu je možné pouze jejich spouštění, navíc mohou mít i omezený přístup, jehož účelem je ochrana firemních dat proti odcizení. To je dosaženo díky zákazu funkce drag and drop, kopírování a vkládání nebo připojení USB zařízení. Nelze ani

uložit aktuální stav stroje pro možnost pozdější obnovy (snapshoty). Vzdálené připojení k virtuálnímu stroji není oficiálně podporováno, ale v programu je zabudovaný modul, který toto umožňuje, akorát je zapotřebí technicky pokročilého uživatele schopného funkcionalitu zprovoznit.

#### 3.2.4 Další funkce

Program nabízí díky nainstalování VMware Tools rozšiřující funkce. Mezi ně patří ovládání drag and drop, kdy je tímto způsobem umožněno přesouvat soubory, složky, e-mailové přílohy, text a obrázky mezi hostitelským systémem a virtuálním strojem. Používat lze i kopírování a vkládání textu, obrázků a e-mailových příloh mezi oběma systémy. Funkce s názvem Unity mód zobrazuje aplikace spuštěné na virtuálním stroji jako by běžely nativně na hostitelském systému. V operačních systémech Microsoft Windows je podporováno grafické rozhraní Aero. Sdílené složky poslouží pro výměnu souborů mezi virtuálními stroji navzájem nebo virtuálním strojem a hostitelským systémem. Z dalších funkcí stojí za zmínku běh virtuálního stroje ve fullscreen režimu, volný pohyb kurzoru myši mezi systémy bez nutnosti přepínání nebo synchronizace času hostovaného operačního systému s hostitelským.

#### 3.2.5 Uživatelská přívětivost

Program mi přijde opravdu hodně jednoduchý na použití, což bylo i cílem jeho vývojářů. Je velmi vhodný pro neznalé uživatele nebo začátečníky, jelikož je nezatěžuje přílišnou konfigurací. Na své si ale přijdou i pokročilí uživatelé, jelikož možnosti nastavení virtuálních strojů jsou bohaté. Líbilo se mi intuitivní ovládání a minimalistický design.

#### 3.2.6 Hardwarové požadavky

Minimální požadavky na spuštění samotného programu jsou 2 jádrový procesor Intel nebo AMD s taktem alespoň 1,3 GHz, 2 GB paměti typu RAM a 200 MB volného místa na disku pro instalaci. Pro fungování 64 bitového hostujícího operačního systému je nutné, aby procesor podporoval VT-x technologii určenou k hardwarové virtualizaci v případě Intelu a segmentaci v 64 bitovém adresním módu v případě AMD.



### 3.2.7 Podpora od výrobce

Společnost VMware nabízí při zakoupení programu 3 různé stupně podpory. Nejobyčejnější je Complimentary podpora, která zahrnuje třicetidenní e-mailovou podporu pouze ohledně instalace programu a začíná běžet ode dne zakoupení programu. Žádost o pomoc lze podat buď telefonicky nebo e-mailem a odpověď by měla přijít do jednoho pracovního dne od nahlášení. Vyšším stupněm je Basic podpora slibující komplexní technickou podporu od pondělí do pátku a to 12 hodin pro každý z těchto dnů. Odpověď na kritický požadavek je do čtyř pracovních hodin od podání. Nejvyšším stupněm je Production podpora fungující 24 hodin 7 dní v týdnu s odpovědí do 30 minut na kritický požadavek. Každý stupeň podpory zahrnuje aktualizace zakoupené verze programu, ale pouze poslední dva typy zahrnují i upgrade na vyšší verzi obohacenou o nové funkce. Na stránkách výrobce je zdarma k dispozici dokumentace k programu, která je dostatečně podrobná a srozumitelná. Dále se zde nachází znalostní databáze (knowledge base) obsahující různé tutoriály a návody na řešení problémů. Využit lze i uživatelské fórum, kde se stačí pouze zaregistrovat. Zajímavá je možnost přidat se do VMware User Group (VMUG) a potkávat se na lokálně organizovaných shromážděních uživatelů, kteří si zde vyměňují své vědomosti a zkušenosti. Členství ve skupině je zdarma a podporované společností VMware ve formě slev na produkty nebo pořádáním vzdělávacích webinářů pro členy.

### 3.2.8 Četnost aktualizací

Program je aktualizován nepravidelně. Od doby vydání verze 12 do aktuální verze 12.5.5 k březnu 2017 vyšlo 9 aktualizací, což v průměru vychází na vydání jedné aktualizace za 2 měsíce. Číslování aktualizací je navíc trochu zvláštní, jelikož dochází k velkým skokům např. z verze 12.1.1 rovnou na verzi 12.5, přitom v porovnání s menšími aktualizacemi obsahuje srovnatelné množství a významnost změn.

### 3.2.9 Popularita a rozšířenost

Společnost VMware je již dlouholetým hráčem na poli virtualizace a její řešení jsou využívána po celém světě. Program byl uveden v roce 2008 ještě pod bývalým názvem VMware Player. Problémy ale nastaly v lednu 2016, kdy byl propuštěn celý tým vývojářů stojící za tímto programem a jeho budoucnost byla nejistá. Neukončení vývoje bylo potvrzeno až v dubnu 2016 vydanou aktualizací a ujištěním dalšího pokračování prací na

programu. Workstation Player je rozšířený hlavně v korporátním prostředí. V ČR má ke dni 5.4.2017 podle serverů stahuj.cz a slunecnice.cz dohromady 135 985 stažení.

#### 3.2.10 Cena

Workstation Player je zdarma pro osobní a nekomerční použití. Pro komerční účely je nutné zakoupit licenci. Ta je nabízena i s povinnou podporou a to ve 3 verzích – Complimentary, Basic a Production. Na Basic a Production podporu je potřeba nákup minimálně deseti licencí. Při úvaze deseti licencí vyjde Complimentary na 1 530\$ s danou podporou po dobu třiceti dní, Basic na 1 850\$ s dobou podpory 1 rok, 2 370\$ s podporou na 3 roky a Production na 1 910\$/1 rok podpory a 2 530\$/3 roky podpory. Program i po vypršení podpory zůstane licencován pro komerční účely. Obsažené údaje pochází z oficiálního e-shopu firmy VMware.

### 3.3 Windows Virtual PC

#### 3.3.1 Podpora operačních systémů (hostujících i hostitelských)

Z hostujících operačních systémů jsou podporovány Windows XP Service Pack 3 Professional, Windows Vista a Windows 7. U hostitelských to jsou pouze Windows 7. Linux není oficiálně podporován, což znamená nedostupnost potřebných virtualizačních ovladačů, ale rozběhnout by měl jít i když s omezenou funkcionalitou.

#### 3.3.2 Pokročilost emulace

Tento virtualizační program je nastaven tak, aby emuloval konkrétní a pevně daný všeobecně použitelný hardware kvůli zajištění maximální kompatibility s hostujícím operačním systémem. Jedná se o pouze jedno jádrový procesor a zastaralou grafickou kartu bez podpory 3D grafiky. Virtuálnímu stroji lze přidělit veškerou volnou paměť typu RAM. Komunikace mezi virtuálním strojem a hostitelem probíhá způsobem klient-server, kde jako server vystupuje virtuální stroj a jako klient hostitel. Aplikace tento způsob komunikace používají k přenosu dat mezi hostem a hostitelem. Všechny fyzické pevné disky na hostiteli jsou přístupné i ve virtuálním stroji a je tak možné jednoduše přistupovat ke všem jejich datům z hostujícího operačního systému. Je také možné vytvoření až tří virtuálních harddisků, což jsou soubory ve formátu VHD. Jeden takový disk může mít kapacitu maximálně 127 GB. Existují tři typy virtuálních disků – fixované,

keré si už předem alokují svou nastavenou kapacitu, dynamické, které se postupně rozšiřují až do dané kapacity a rozdílové mající svůj rodičovský disk a ukládající pouze změny oproti rodičovskému disku. Windows Virtual PC podporuje USB zařízení, a to maximálně standardu USB 2.0. Zařízení mohou být připojena ve dvou módech. Prvním je Sharing mód, kdy jsou zařízení sdílena s hostitelským systémem. Tento mód lze použít jen u úložných zařízení, tiskáren a chytrých karet. Druhým módem, který neomezuje použitelná zařízení, je Redirection, v němž je zařízení viditelné pouze pro virtuální stroj. U zvukového vstupu a výstupu virtuálního stroje je na výběr přesměrování na hostitelské zvukové zařízení nebo zpracování emulovaným zvukovým zařízením. Z virtuálního stroje lze také přistupovat k fyzickým CD a DVD jednotkám hostitele. Připojení k síti je možné přes virtuální Ethernet adaptéry, které jdou namapovat na fyzické síťové karty na hostiteli. K dispozici jsou maximálně 4 virtuální adaptéry a každý má svou unikátní MAC adresu. V případě, že je hostitel připojen k Internetu přes Wi-Fi adaptér, musí být použit překlad síťových adres (NAT) na virtuálním stroji. Další podporované módy pro konfiguraci síťového připojení jsou Bridge, Internal Network a Not connected.

### 3.3.3 Správa virtuálních strojů

K virtuálním strojům jde přistupovat a spravovat je skrze speciální složku Virtuální Stroje. Tato složka je velmi podobná klasickým složkám v operačním systému. Může odtud být vytvořen také nový virtuální stroj nebo importován již existující. Importovat lze jenom stroje, které byly vytvořeny ve virtualizačním programu od společnosti Microsoft (formát VMC). Naráz může běžet i více virtuálních strojů, pokud pro ně má hostitelský systém dostatek volné paměti typu RAM. Konfigurace virtuálního stroje je bohužel velmi omezená. Programátoři a IT profesionálové mohou virtuální stroj ovládat přes Windows PowerShell a psát si své vlastní skripty, jelikož mají k dispozici COM API. Díky tomu mohou provádět i věci, které nejsou použitím konvenčního uživatelského rozhraní možné. K virtuálnímu stroji je umožněno vzdálené připojení zajišťované službou Remote Desktop Connection. Program nenabízí uložení aktuálního stavu virtuálního stroje pro pozdější možnost obnovy (snapshot). K docílení stejného efektu se doporučuje jednoduše zkopírovat a zálohovat jednotlivé soubory virtuálního stroje (.VHD, .VUD, .VMC) a při potřebě je potom znovu obnovit.

### 3.3.4 Další funkce

Velkou výhodou je pokročilá integrace aplikací, instalovaných na virtuálním stroji, do hostitelského operačního systému. To v praxi vypadá tak, že aplikace se chová jako by byla nainstalována na hostitelském operačním systému – má ikonu na ploše, odkaz v Nabídce Start a jdou k ní asociovat typy otevíraných souborů. Spuštění aplikace vyvolá start virtuálního stroje (pokud už neběží v pozadí) a automaticky uživatele přihlásí do hostovaného systému. Kurzorem myši se dá plynule přecházet z hostovaného systému na hostitelský a naopak, ale podpora pro drag and drop chybí. Sdílení konkrétních složek také není podporováno, lze sdílet pouze celý disk. Toto omezení se dá obejít, ale vyžaduje to technicky pokročilého uživatele. Zajímavou funkcí je Undo Disks, která od doby svého zapnutí ukládá všechna data (soubory, updaty, aplikace) z virtuálního stroje na oddělený virtuální disk. Uživatel má možnost kdykoliv všechny provedené změny zrušit nebo naopak potvrdit, čímž dojde k jejich přesunutí na virtuální disky připojené k virtuálnímu stroji. Mezi další funkce patří sdílená schránka pro kopírování dat, běh virtuálního stroje ve fullscreen režimu nebo synchronizace času hostovaného operačního systému s hostitelským.

### 3.3.5 Uživatelská přívětivost

Vzhled je jasně cílený na uživatele Microsoft Windows 7, pro které je tento program také určen. Integrace do operačního systému je opravdu vynikající. Dominuje velmi jednoduché použití. Program se prakticky nemusí manuálně konfigurovat a pokud uživatel např. vytváří nový virtuální stroj, je všemi kroky postupně proveden a maximálně odstíněn od zabíhání do technických podrobností. Problém jsem měl ale s bootováním virtuálního stroje, které bylo velmi pomalé. Za to, že je program velmi úzce zaměřený, bych čekal vysokou odladěnost, což se mi nepotvrdilo a virtuální stroj občas dokonce i „zamrzl“.

### 3.3.6 Hardwarové požadavky

Na rozběhnutí programu je potřeba procesor Intel, AMD nebo VIA s taktem alespoň 1 GHz a podporou hardwarové virtualizace (AMD-V, Intel-VT). Toto omezení bylo později odstraněno vydaným updatem, který ale omezení odstraňuje pouze pro Windows XP. Paměti typu RAM je zapotřebí 1,25 GB, ale doporučeny jsou 2 GB. Na disku zabere instalace samotného programu 20 MB.

### 3.3.7 Podpora od výrobce

Na stránkách Microsoftu se nachází podpůrné materiály k programu, které se převážně skládají z návodů, jak něco v programu udělat. Členění jednotlivých sekcí mi přijde trochu nepřehledné, materiály jsou roztráštěné na různých místech. Chybí mi technicky založená dokumentace pro pokročilejší uživatele, i když na ně Microsoft s tímto programem ani necílí. Dále jsou k dispozici tutoriály na oficiálním blogu, některé jdou i technicky do hloubky, což hodnotím pozitivně. Bohužel není k dispozici ani žádné oficiální fórum pro uživatele, pouze neoficiální.

### 3.3.8 Četnost aktualizací

Poslední aktualizace pochází ze 14.2.2011 a pravděpodobně je také finální, jelikož již existuje nástupce Microsoft Client Hyper-V.

### 3.3.9 Popularita a rozšířenost

Windows Virtual PC nebyl standardně součástí Windows 7, možná tak v nějaké OEM verzi tohoto operačního systému, takže si ho všichni uživatelé museli stáhnout a nebyl masově šířený. Vzhledem k popularitě a významnosti firmy Microsoft a oblíbenosti systému Windows XP se však stal hojně používaným řešením pro udržení běhu tohoto již nepodporovaného operačního systému. Navíc jeho předchůdce Microsoft Virtual PC 2007 si vybudoval na poli virtualizace dobré jméno. Windows Virtual PC ale moc oblíbený není a mnoho uživatelů je zklamaných a našťvaných, což je vidět hlavně na diskuzních fórech. Nejvíce rozšířený je hlavně u malých firem.

### 3.3.10 Cena

Program je licencován jako freeware, tudíž je zdarma k použití pro všechny účely.

## 4 Vícekriteriální porovnání virtualizačních programů

### 4.1 Vybraná kritéria

Pro vícekriteriální porovnání jednotlivých virtualizačních programů jsem na základě stanovených cílů vybral následující kritéria uvedená v tabulce. Všechna kritéria jsou hodnotící.

**Tabulka 1:** Srovnávací kritéria

Označení	Kritérium	Typ kritéria	Stupnice
K <sub>1</sub>	Podpora hostitelských operačních systémů	Maximalizační	Poměrová
K <sub>2</sub>	Podpora hostujících operačních systémů	Maximalizační	Poměrová
K <sub>3</sub>	Pokročilost emulace	Maximalizační	Pořadová
K <sub>4</sub>	Správa virtuálních strojů	Maximalizační	Pořadová
K <sub>5</sub>	Další funkce	Maximalizační	Pořadová
K <sub>6</sub>	Uživatelská přívětivost	Maximalizační	Pořadová
K <sub>7</sub>	Hardwarové požadavky	Minimalizační	Pořadová
K <sub>8</sub>	Podpora od výrobce	Maximalizační	Pořadová
K <sub>9</sub>	Četnost aktualizací	Maximalizační	Pořadová
K <sub>10</sub>	Popularita a rozšířenost	Maximalizační	Pořadová
K <sub>11</sub>	Cena	Minimalizační	Poměrová

*Zdroj: vlastní zpracování*

## 4.2 Vysvětlení kritérií a stanovení způsobů zjištění jejich hodnot

Podpora hostitelských, respektive hostujících operačních systémů označuje počet podporovaných operačních systémů v každé z těchto kategorií. Různé linuxové distribuce se budou počítat jako jeden operační systém – Linux. Cena virtualizačních programů se v případě komerčních verzí uvádí v dolarech a pro porovnání se použije nejmenší možná cena za 10 licencí.

Hodnoty kvalitativních kritérií jsou definovány následujícími bodovými stupnicemi:

### *Pokročilost emulace*

**1 bod** – základní = velmi omezené možnosti nastavení uživatelem, mizivé využití potenciálu hostitelského stroje

**2 body** – pokročilá = uživatel si může nastavit důležité virtualizované komponenty (procesor, grafická karta, RAM, ...), většina hardwaru hostitelského stroje je alespoň částečně emulována

**3 body** – špičková = uživatel má možnost velmi detailního nastavení většiny hardwarových komponent virtuálního stroje, výkon hostitelského stroje je prakticky plně k dispozici a virtuální hardware podporuje většinu funkcí hostitelského hardwaru (např. akcelerovaná 3D grafika)

### *Správa virtuálních strojů*

**1 bod** – virtuální stroj lze pouze vytvořit a smazat, další správa není dostupná, virtuální stroj nelze přenášet na jiná PC a není ani umožněn import cizích virtuálních strojů

**2 body** – program podporuje většinu standardních funkcí (přehledná správa virtuálních strojů na jednom místě, běh více virtuálních strojů naráz, přenositelnost virtuálních strojů na jiná PC, import cizích virtuálních strojů, možnost psaní vlastních skriptů, vzdálené připojení k virtuálnímu stroji)

**3 body** – dále umožněno ukládat aktuální stav virtuálního stroje (snapshot), šifrování virtuálního disku nebo organizování virtuálních strojů do skupin a jejich ovládání jako celku

### *Další funkce*

**0 bodů** – virtualizační program nenabízí kromě základních žádné další funkce

**1 bod** – program podporuje tyto další funkce: synchronizace času, sdílená schránka pro kopírování dat, běh virtuálního stroje ve fullscreen režimu, volný pohyb kurzoru mezi hostem a hostitelem, zobrazení virtuálních aplikací jako nativních

**2 body** – uživatel má navíc ještě k dispozici funkce jako sdílené složky a drag and drop mezi hostem a hostitelem

### *Uživatelská přívětivost*

**0 bodů** – velmi špatná ovladatelnost a složité použití, absolutně nevhodný pro neznalé uživatele, chybějící nápověda, pomalý běh programu

**1 bod** – intuitivní ovládání, jednoduchý na použití i pro začátečníka, přehledné uspořádání jednotlivých prvků a informací, nápověda

**2 body** – nejen dobrá ovladatelnost, ale i povedený design, dále také rychlý a bezproblémový běh programu

### *Hardwarové požadavky*

**1 bod** – nízké = procesor s jedním jádrem, méně než 1 GB RAM, instalace programu do 100 MB

**2 body** – střední = procesor s minimálně jedním jádrem a podporou hardwarové virtualizace, více jak 1 GB RAM

**3 body** – vysoké = procesor s minimálně dvěma jádry, 2 GB RAM a více, instalace samotného programu větší než 100 MB

### *Podpora od výrobce*

**0 bodů** – prakticky žádná podpora, uživatel je odkázán na jiné neoficiální zdroje

**1 bod** – výrobce poskytuje základní neplacenou podporu v podobě dokumentace nebo uživatelského manuálu, různé tutoriály nebo fórum pro uživatele programu

**2 body** – výrobce poskytuje i profesionální placenou podporu



### *Četnost aktualizací*

**0 bodů** – program nebyl už více než 1 rok aktualizován

**1 bod** – program je aktualizován alespoň jednou za čtvrt roku

**2 body** – program je většinou aktualizován jednou za měsíc nebo častěji

### *Popularita a rozšířenost*

**0 bodů** – program není celosvětově rozšířen, a navíc je neoblíbený mezi uživateli

**1 bod** – program hojně rozšířený po celém světě s početnou uživatelskou základnou nebo alespoň velmi oblíbený mezi uživateli

**2 body** – program je globálně rozšířený, na trhu 10 a více let, stojí za ním velká a známá technologická společnost

### 4.3 Ohodnocení virtualizačních programů

**Tabulka 2:** Ohodnocení virtualizačních programů

		<b>VirtualBox</b>	<b>VMware Workstation Player</b>	<b>Windows Virtual PC</b>
Podpora hostitelských operačních systémů		9	6	1
Podpora hostujících operačních systémů		14	14	3
Pokročilost emulace		3 b	3 b	1 b
Správa virtuálních strojů		3 b	2 b	2 b
Další funkce		2 b	2 b	1 b
Uživatelská přívětivost		2 b	2 b	1 b
Hardwarové požadavky		1 b	3 b	2 b
Podpora od výrobce		2 b	2 b	1 b
Četnost aktualizací		2 b	1 b	0 b
Popularita a rozšířenost		2 b	1 b	1 b
Cena	osobní použití	0	0	0
	komerční použití	5 000\$	1 530\$	0

*Zdroj: vlastní zpracování*

#### 4.4 Stanovení vah kritérií pro typické uživatele

Váhy kritérií  $v_k$  ( $k = 1, \dots, n$ ) vyjadřují relativní důležitost kritérií pro rozhodovatele.

Obvykle se používají normované váhy kritérií, pro které platí:

$$\sum_{k=1}^n v_k = 1 \quad ; \quad v_k \geq 0$$

Pro stanovení vah kritérií jsem si vybral bodovací metodu, která důležitost kritéria vyjadřuje číslem ve vhodně definované bodovací stupnici. Doporučuje se, aby stupnice začínala nulou, měla lichý počet stupňů (existence přirozeného středu) a byla slovně popsána. Ke znormování získaných vah se použije vztah:

$$v_k = \frac{b_k}{\sum_{i=1}^n b_i}$$

$b_k$  – počet bodů přiřazených  $k$ -tému kritériu

Bodovací stupnice vyjadřující důležitost kritérií vypadá následovně:

- 0** – naprosto nedůležité kritérium
- 1** – málo důležité kritérium
- 2** – středně důležité kritérium
- 3** – hodně důležité kritérium
- 4** – extrémně důležité kritérium

**Tabulka 3:** Stanovení vah kritérií pro jednotlivé typické uživatele

	První uživatel		Druhý uživatel		Třetí uživatel	
	Body	Váha	Body	Váha	Body	Váha
Podpora hostitelských operačních systémů	3	0,111	2	0,077	1	0,036
Podpora hostujících operačních systémů	3	0,111	3	0,115	1	0,036
Pokročilost emulace	4	0,149	2	0,077	1	0,036
Správa virtuálních strojů	3	0,111	2	0,077	4	0,143
Další funkce	3	0,111	1	0,038	2	0,071
Uživatelská přívětivost	2	0,074	4	0,154	3	0,107
Hardwarové požadavky	1	0,037	2	0,077	3	0,107
Podpora od výrobce	1	0,037	2	0,077	4	0,143
Četnost aktualizací	3	0,111	1	0,038	2	0,071
Popularita a rozšířenost	2	0,074	3	0,115	3	0,107
Cena	2	0,074	4	0,154	4	0,143

*Zdroj: vlastní zpracování*

#### 4.5 Metody vícekriteriálního hodnocení

Pro vícekriteriální hodnocení různých variant virtualizačních programů jsem zvolil 2 jednodušší a 3 složitější metody. Vzhledem k převaze kvalitativních kritérií jsou metody lexikografická, váženého součtu pořadí a ze složitějších AGREPREF, ELECTRE III a ORESTE.

Vyhodnocení všech metod jsem si nejprve namodeloval v MS Excelu, což mi pomohlo porozumět, jak daná metoda funguje. Výsledky jsem potom ověřoval v již naprogramovaném softwaru, konkrétně:

- VHV z katedry ekonomiky, manažerství a humanitních věd FEL ČVUT – lexikografická metoda, metoda váženého součtu pořadí
- IZAR 1.4.1 od FIT ČVUT – AGREPREF, ELECTRE III
- SANNA z fakulty informatiky a statistiky, VŠE – ORESTE, ELECTRE III

Nejlépe se mi pracovalo s programem IZAR, který je velmi intuitivní a uživatelsky přívětivý. Také se mi na něm líbilo, že zobrazoval i hodně různých mezivýpočtů u jednotlivých metod. Bohužel u excelového doplňku SANNA jsem u metody ORESTE zjistil, že při zadání stejných vah kritérií se u jejich pořadí nespočítá průměr, ale jsou řazeny sestupně podle pořadí kritérií.

#### 4.5.1 Lexikografická metoda

Tato metoda vyžaduje jednoznačně určené pořadí důležitosti kritérií, tj. žádná kritéria nemohou mít stejnou důležitost. Abych mohl metodu použít, musel jsem tedy důležitost kritérií pro každého typického uživatele upravit:

**Tabulka 4:** Pořadí důležitosti kritérií

	1. uživatel	2. uživatel	3. uživatel
Podpora hostitelských operačních systémů	5	9	11
Podpora hostujících operačních systémů	2	3	10
Pokročilost emulace	1	5	9
Správa virtuálních strojů	3	6	2
Další funkce	4	10	8
Uživatelská přívětivost	8	1	5
Hardwarové požadavky	10	7	4
Podpora od výrobce	11	8	3
Četnost aktualizací	6	11	7
Popularita a rozšířenost	9	4	6
Cena	7	2	1

*Zdroj: vlastní zpracování*

Metodu jsem používal ve striktní verzi, tj. bez uvolnění lexikografické diktatury. Výsledky jsou uvedeny v následující tabulce:

**Tabulka 5:** Výsledné pořadí z lexikografické metody

	1. uživatel	2. uživatel	3. uživatel
VirtualBox	1.	1.	3.
VMware Workstation Player	2.	2.	2.
Windows Virtual PC	3.	3.	1.

*Zdroj: vlastní zpracování*

Podrobnější výsledky jsou k nalezení v příloze.

#### 4.5.2 Metoda váženého součtu pořadí

Jako vstupní údaje do metody jsem použil hodnocení virtualizačních programů dle jednotlivých kritérií a normované váhy pro každého typického uživatele. Pro výběr optimální varianty byl použit tento vztah:

$$\min_i \sum_{j=1}^n v_j q_{i,j}$$

$v_j$  – váha  $j$ -tého kritéria

$q_{i,j}$  – hodnota pořadové funkce  $i$ -té varianty dle  $j$ -tého kritéria

$q_{i,j} = 1$  pro nejlepší variantu,  $n$  pro nejhorší variantu

Vyhodnocení dopadlo následovně:

**Tabulka 6:** Výsledné pořadí z metody váženého součtu pořadí

	1. uživatel	2. uživatel	3. uživatel
VirtualBox	1.	1.	1.
VMware Workstation Player	2.	2.	2.
Windows Virtual PC	3.	3.	3.

*Zdroj: vlastní zpracování*

Pořadová funkce a detailnější výsledky se nachází v příloze.

#### 4.5.3 Metoda AGREPREF

Nevýhodou metody je, že kvalita výsledků závisí na prazích citlivosti, pro jejichž určení neexistuje jednoznačné doporučení. Je proto nutné vyzkoušet různé prahové hodnoty, čímž se testuje citlivost daného problému a jeho řešení, a nakonec zvolit ty nejhodnější. Já jsem si nejprve u každého typického uživatele vypočítal hodnoty pro  $S_{i=j}$ ,  $S_{i,j} - S_{j,i}$  a  $S_{j,i} - S_{i,j}$ , abych jasně viděl, jak bude vypadat preferenční relace pro různé hodnoty prahů a mohl tak vybrat ty optimální. Pro výběr prahu indiference  $b$  jsem uvažoval pouze hodnoty nad 0,5:

**0,556** u programů VirtualBox a VMware Workstation Player, první typický uživatel

**0,615** u programů VirtualBox a VMware Workstation Player, druhý typický uživatel

Nejnižší možná hodnota prahu preference  $c$ , která by v porovnávání hrála roli, byla **0,25** u programů VMware Workstation Player a Windows Virtual PC. Pokud bych prah zvolil na této hranici, byly by varianty nesrovnatelné.

Dále jsem si určil průměrnou váhu kritéria – **0,091**, která mi také pomáhala ve volbě hodnot prahů.

Po provedené analýze jsem dospěl k těmto hodnotám:

$$b = 0,6$$

$$c = 0,2$$

Vyhodnocení s takto zvolenými prahy dopadlo následovně:

**Tabulka 7:** Výsledné pořadí z metody AGREPREF

	1. uživatel	2. uživatel	3. uživatel
VirtualBox	1.	1. - 2.	1.
VMware Workstation Player	2.	1. - 2.	2.
Windows Virtual PC	3.	3.	3.

*Zdroj: vlastní zpracování*

Jednotlivé mezivýsledky včetně ukázkového screenshotu z programu IZAR s vyhodnocením pro prvního typického uživatele se nachází v příloze.

#### 4.5.4 Metoda ORESTE

Pro vyhodnocení touto metodou jsem nejdříve sestavil pořadí variant dle jednotlivých kritérií, což je stejné jako pořadová funkce u metody váženého součtu pořadí, která se nachází v příloze. Dále jsem udělal pořadí důležitosti kritérií:

**Tabulka 8:** Pořadí důležitosti kritérií pro jednotlivé uživatele

	1. uživatel	2. uživatel	3. uživatel
Podpora hostitelských operačních systémů	4	7	10
Podpora hostujících operačních systémů	4	3,5	10
Pokročilost emulace	1	7	10
Správa virtuálních strojů	4	7	2
Další funkce	4	10,5	7,5
Uživatelská přívětivost	8	1,5	5
Hardwarové požadavky	10,5	7	5
Podpora od výrobce	10,5	7	2
Četnost aktualizací	4	10,5	7,5
Popularity a rozšířenost	8	3,5	5
Cena	8	1,5	2

*Zdroj: vlastní zpracování*

Dalšími vstupy jsou prahy citlivosti, které jsem zvolil následovně:

$\alpha = 0,25$  (maximální možný)

$\beta = 0,045$  (maximální možný)

$\gamma = 2,25$  (minimální možný) – s tímto prahem ani nemělo smysl experimentovat, jeho zvýšení by nemělo na výsledky žádný vliv



Výsledky u této metody dopadly takto:

**Tabulka 9:** Výsledné pořadí z metody ORESTE

	1. uživatel	2. uživatel	3. uživatel
VirtualBox	1.	1.	1.
VMware Workstation Player	2.	2.	2.
Windows Virtual PC	3.	3.	3.

*Zdroj: vlastní zpracování*

Mezivýsledky jednotlivých částí metody pro každého typického uživatele se nachází v příloze.

#### 4.5.5 Metoda ELECTRE III

K vyhodnocení stačilo do softwaru IZAR i SANNA zadat pouze hodnocení virtualizačních programů dle jednotlivých kritérií spolu s typem kritéria a normovanými váhami. Výhodou je, že u této metody rozhodovatel nemusí zadávat prahy citlivosti.

U obou použitých programů dopadlo vyhodnocení následovně:

**Tabulka 10:** Výsledné pořadí z metody ELECTRE III

	1. uživatel	2. uživatel	3. uživatel
VirtualBox	1.	1.	1.
VMware Workstation Player	2.	2.	2.
Windows Virtual PC	3.	3.	3.

*Zdroj: vlastní zpracování*

Výsledky sestupného i vzestupného uspořádání pro každého typického uživatele se nachází v příloze. Je přidán i screenshot z programu IZAR a SANNA s vyhodnocením pro prvního uživatele.

## 5 Implementace ELECTRE III

Cílem bylo implementovat a zaintegrovat metodu ELECTRE III do již existující webové aplikace VHV z katedry ekonomiky, manažerství a humanitních věd FEL ČVUT. Aplikace byla vyvinuta v roce 2010 a je z technologického hlediska už značně zastaralá. To se projevuje hlavně na její architektuře. Než jsem proto začal se samotnou implementací metody ELECTRE III, rozhodl jsem se aplikaci částečně předělat, aby alespoň trochu odpovídala standardům dnešního vývoje webových aplikací. Původně jsem ji chtěl předělat celou, ale odhadnutá doba úprav převyšovala můj časový limit, který jsem chtěl tomuto bodu bakalářské práce věnovat a vznikl tudíž kompromis. Strukturu aplikace jsem se snažil co nejvíce přiblížit návrhovému vzoru MVC, který je už léty prověřený. Také jsem sjednotil a mírně upravil kód v JavaScriptu pro lepší přehlednost. Do logiky samotných metod jsem nezasahoval, protože jsem nechtěl nic pokazit. Jako vývojové prostředí jsem použil VS Code a PHP server, který je součástí PHP 5.4+.

### 5.1 Realizace

Při implementaci metody ELECTRE III jsem se snažil co nejvíce využít již udělaných věcí jako grafické rozhraní pro zadávání vstupních dat a třídy pro práci s těmito daty, které umožňují např. jejich maticovou reprezentaci, normalizaci a ukládání/nahrávání do/ze souboru. Tím pádem jsem se mohl soustředit čistě na implementaci logiky samotné metody. Po vzoru ostatních se jedná o jednu třídu – *ElectreIII*, která obsahuje vše potřebné pro získání výsledků. Při implementaci hlavní části algoritmu jsem použil rekurzi, která se v tomto případě jevila jako jasná volba. Správné naprogramování metody jsem ověřoval na různých vstupních datech včetně dat z mého porovnávání virtualizačních programů. Výsledky jsem vždy porovnal s programem IZAR. U jednodušších případů jsem si celý výpočet namodeloval v Excelu. Nevýhodou IZARU totiž je, že varianty vyhodnocuje pouze jedním směrem (od nejlepší k nejhorší). Princip metody ELECTRE III je popsán v příloze a je zde přiložen také zdrojový kód třídy *ElectreIII*. Samotná aplikace je dostupná na adrese <https://ekonom.feld.cvut.cz/vhv>, dokumentace jednotlivých tříd na <https://ekonom.feld.cvut.cz/vhv/docs>.

## 6 Doporučení pro typické uživatele

Vzhledem k tomu, že jsem porovnával pouze tři virtualizační programy, tak jsem žádný z nich na základě výsledků vícekriteriálního hodnocení nevyločil. Akorát jsem uspořádal jejich pořadí dle vhodnosti pro každého z typických uživatelů. Dále jsem přidal i svůj subjektivní komentář k výsledkům – zdali s nimi souhlasím nebo bych dal jiné doporučení.

### 6.1 První uživatel

Pořadí srovnávaných virtualizačních programů dle jednotlivých metod dopadlo takto:

**Tabulka 11:** Výsledné pořadí virtualizačních programů pro prvního uživatele

	<b>VirtualBox</b>	<b>VMware Workstation Player</b>	<b>Windows Virtual PC</b>
Lexikografická metoda	1.	2.	3.
Metoda váženého součtu pořadí	1.	2.	3.
AGREPREF	1.	2.	3.
ORESTE	1.	2.	3.
ELECTRE III	1.	2.	3.

*Zdroj: vlastní zpracování*

U tohoto uživatele se výsledky všech metod shodují a není tedy co řešit. Já s nimi souhlasím a doporučil bych programy ve stejném pořadí.

## 6.2 Druhý uživatel

Pro druhého uživatele byly výsledky vícekritériálního porovnání virtualizačních programů následující:

**Tabulka 12:** Výsledné pořadí virtualizačních programů pro druhého uživatele

	<b>VirtualBox</b>	<b>VMware Workstation Player</b>	<b>Windows Virtual PC</b>
Lexikografická metoda	1.	2.	3.
Metoda váženého součtu pořadí	1.	2.	3.
AGREPREF	1. - 2.		3.
ORESTE	1.	2.	3.
ELECTRE III	1.	2.	3.

*Zdroj: vlastní zpracování*

Zde je výsledné pořadí stejné až na metodu AGREPREF, kde vychází VirtualBox a VMware Workstation Player jako indiferentní varianty. Osobně se také přikláním k této anomálii, jelikož myslím, že potřeby druhého uživatele splňují oba programy stejně dobře.

### 6.3 Třetí uživatel

Vyhodnocení jednotlivými metodami vícekriteriálního hodnocení pro třetího typického uživatele shrnuje následující tabulka:

**Tabulka 13:** Výsledné pořadí virtualizačních programů pro třetího uživatele

	<b>VirtualBox</b>	<b>VMware Workstation Player</b>	<b>Windows Virtual PC</b>
Lexikografická metoda	3.	2.	1.
Metoda váženého součtu pořadí	1.	2.	3.
AGREPREF	1.	2.	3.
ORESTE	1.	2.	3.
ELECTRE III	1.	2.	3.

*Zdroj: vlastní zpracování*

Lexikografická metoda jako jediná vyhodnotila jiné pořadí než ostatní metody. To je dáno tím, že nejdůležitějším kritériem pro třetího uživatele byla cena a u ní se porovnávání zastavilo. S vyššími doporučeními jinak souhlasím.

## 7 Závěr

V této práci jsem se zaměřil na porovnání tří neznámějších virtualizačních programů pro lokální virtualizaci a jejich doporučení třem typickým uživatelům. Většinu informací jsem čerpal přímo z oficiální dokumentace k programu poskytnuté výrobcem na jeho internetových stránkách. Programy jsem i sám osobně testoval na svém počítači a zkoušel v nich provádět typické uživatelské akce.

Identifikoval jsem typické uživatele a jejich požadavky na virtualizační software, ze kterých jsem následně vytvořil hodnotící kritéria pro vícekriteriální porovnání. To bylo provedeno pomocí pěti metod – lexikografická metoda, metoda váženého součtu pořadí, AGREPREF, ORESTE, ELECTRE III. Posledně zmíněnou metodu jsem i naprogramoval do školní webové aplikace a rozšířil tak portfolio již podporovaných metod. Tato část mě bavila nejvíc a rád jsem si připomněl po téměř dvou letech nepoužívání jazyk PHP.

Nakonec jsem na základě výsledků vícekriteriálního hodnocení doporučil typickým uživatelům nejvhodnější virtualizační software.

Virtualizace je zajímavá technická oblast a jsem rád, že jsem mohl načerpat teoretické i praktické poznatky a přiučit se něco pro mě nového. To samé platí i o vícekriteriálním hodnocení, se kterým jsem neměl vůbec žádné předchozí zkušenosti.

## 8 Zdroje

### 8.1 Internetové zdroje

- [1] SaranSaro. In: *SaranSaro.in* [online]. prosinec 2014 [cit. 2017-03-12]. Dostupné z: <http://saransaro.in/wp-content/uploads/2014/12/vmw-virtualization-defined2.jpg>
- [2] Oracle. Product Details - Oracle VM Premier Support. *Oracle Store* [online]. [cit. 2017-03-30]. Dostupné z: [https://shop.oracle.com/apex/f?p=DSTORE:PRODUCT:::NO:RP,6:P6\\_LPI:4510280076791805728470](https://shop.oracle.com/apex/f?p=DSTORE:PRODUCT:::NO:RP,6:P6_LPI:4510280076791805728470)
- [3] Oracle. Oracle VM VirtualBox - Support. *Oracle Technology Network* [online]. [cit. 2017-03-30]. Dostupné z: <http://www.oracle.com/technetwork/server-storage/virtualbox/support/index.html>
- [4] COEKAERTS, Wim. Oracle Linux and Oracle VM pricing guide. In: *Wim Coekaerts Blog* [online]. 30.10.2013 [cit. 2017-03-30]. Dostupné z: [https://blogs.oracle.com/wim/entry/oracle\\_linux\\_and\\_oracle\\_vm](https://blogs.oracle.com/wim/entry/oracle_linux_and_oracle_vm)
- [5] Microsoft. Download Windows Virtual PC from Official Microsoft Download Center. *Microsoft Download Center: Windows, Office, Xbox & More* [online]. ©2017 [cit. 2017-03-31]. Dostupné z: <https://www.microsoft.com/en-us/download/details.aspx?id=3702>
- [6] TechNet Archive. Windows Virtual PC. *TechNet Blogs* [online]. ©2017 [cit. 2017-03-31]. Dostupné z: [https://blogs.technet.microsoft.com/windows\\_vpc/](https://blogs.technet.microsoft.com/windows_vpc/)
- [7] Softpedia. Windows Virtual PC Download. *Softpedia - Free Downloads Encyclopedia* [online]. ©2001-2017 [cit. 2017-03-31]. Dostupné z: <http://www.softpedia.com/get/System/OS-Enhancements/Microsoft-Virtual-PC.shtml>
- [8] BRINK, Shawn. How to Install, Enable, and Disable Windows Virtual PC Integration Features. In: *Windows 7 Help Forums* [online]. 4.8.2010 [cit. 2017-03-31]. Dostupné z: <http://www.sevenforums.com/tutorials/101888-windows-virtual-pc-integration-features-install-enable-disable.html>
- [9] BRINK, Shawn. How to Create a Virtual Machine in Windows Virtual PC. In: *Windows 7 Help Forums* [online]. 1.5.2009 [cit. 2017-03-31]. Dostupné z: <http://www.sevenforums.com/tutorials/8254-windows-virtual-pc-create-virtual-machine.html>
- [10] Microsoft. Windows Virtual PC Evaluation Guide. In: *TechNet Library* [online]. ©2017. Poslední změna 16.9.2009. [cit. 2017-03-31]. Dostupné z: [https://technet.microsoft.com/en-us/library/dd744684\(v=ws.10\).aspx](https://technet.microsoft.com/en-us/library/dd744684(v=ws.10).aspx)

- [11] Microsoft. Windows Virtual PC Help. In: *TechNet Library* [online]. ©2017. Poslední změna 25.6.2009. [cit. 2017-04-01]. Dostupné z: [https://technet.microsoft.com/en-us/library/ee449411\(v=ws.10\).aspx](https://technet.microsoft.com/en-us/library/ee449411(v=ws.10).aspx)
- [12] Microsoft. About integration features. In: *TechNet Library* [online]. ©2017. Poslední změna 17.9.2009. [cit. 2017-04-01]. Dostupné z: [https://technet.microsoft.com/en-us/library/ee449432\(WS.10\).aspx](https://technet.microsoft.com/en-us/library/ee449432(WS.10).aspx)
- [13] Microsoft. Error message about hardware-assisted virtualization for Windows XP Mode in Windows Virtual PC on a computer that is running Windows 7. In: *Microsoft Support* [online]. ©2017. Poslední změna 6.3.2012. [cit. 2017-04-01]. Dostupné z: <https://support.microsoft.com/en-us/kb/977206>
- [14] POMAŽAL, Jiří. Virtualizace v kostce. In: *SystemOnLine.cz* [online]. ©2001-2017. červen 2010 [cit. 2017-03-15]. Dostupné z: <http://www.systemonline.cz/clanky/virtualizace-v-kostce.htm>
- [15] HOFFMAN, Chris. Beginner Geek: How to Create and Use Virtual Machines. In: *How-To Geek* [online]. ©2006-2017. 8.9.2014 [cit. 2017-03-15]. Dostupné z: <http://www.howtogeek.com/196060/beginner-geek-how-to-create-and-use-virtual-machines/>
- [16] HOFFMAN, Chris. What Is a Virtual Machine?. In: *MakeUseOf* [online]. ©2017. 18.7.2012 [cit. 2017-03-16]. Dostupné z: <http://www.makeuseof.com/tag/virtual-machine-makeuseof-explains/>
- [17] ROUSE, Margaret a BIGELOW, Steve. Hardware virtualization. In: *SearchVMware* [online]. ©2007-2017. Poslední změna říjen 2015. [cit. 2017-03-16]. Dostupné z: <http://searchvmware.techtarget.com/definition/hardware-virtualization>
- [18] Oracle. End-user documentation. *Oracle VM VirtualBox* [online]. [cit. 2017-03-28]. Dostupné z: [https://www.virtualbox.org/wiki/End-user\\_documentation](https://www.virtualbox.org/wiki/End-user_documentation)
- [19] Oracle. User Manual. *Oracle VM VirtualBox* [online]. ©2004-2017. [cit. 2017-03-28]. Dostupné z: <https://www.virtualbox.org/manual/UserManual.html>
- [20] Oracle. Changelog. *Oracle VM VirtualBox* [online]. [cit. 2017-03-28]. Dostupné z: <https://www.virtualbox.org/wiki/Changelog>
- [21] Oracle. Licensing: Frequently Asked Questions. *Oracle VM VirtualBox* [online]. Poslední změna 17.11.2016. [cit. 2017-03-28]. Dostupné z: [https://www.virtualbox.org/wiki/Licensing\\_FAQ](https://www.virtualbox.org/wiki/Licensing_FAQ)
- [22] VMware. Workstation for Windows. *VMware.com* [online]. ©2017. [cit. 2017-03-29]. Dostupné z: <http://www.vmware.com/products/workstation.html>



- [23] VMware. Workstation Player FAQs. *VMware.com* [online]. ©2017. [cit. 2017-03-29].  
Dostupné z: <http://www.vmware.com/products/player/faqs.html>
- [24] VMware. VMware Workstation Player Documentation. *VMware.com* [online]. ©2017.  
[cit. 2017-03-29]. Dostupné z:  
[https://www.vmware.com/support/pubs/player\\_pubs.html](https://www.vmware.com/support/pubs/player_pubs.html)
- [25] VMware. *Using VMware Workstation Player for Windows* [online]. VMware, Inc., 2016  
[cit. 2017-03-29]. Dostupné z: <http://pubs.vmware.com/player-12-windows/topic/com.vmware.ICbase/PDF/workstation-player-12-windows-user-guide.pdf>
- [26] VMware. *Installing and Configuring VMware Tools* [online]. VMware, Inc., září 2012 [cit. 2017-03-29]. Dostupné z: <http://www.vmware.com/pdf/vmware-tools-installation-configuration.pdf>
- [27] SHARWOOD, Simon. VMware axes Fusion and Workstation US devs. In: *The Register* [online]. 27.1.2016 [cit. 2017-03-29]. Dostupné z: [http://www.theregister.co.uk/2016/01/27/vmware\\_fusion\\_and\\_workstation\\_development\\_team\\_fired](http://www.theregister.co.uk/2016/01/27/vmware_fusion_and_workstation_development_team_fired)
- [28] NANNI, Dan. How to enable VNC remote console in VMware Player. In: *Xmodulo* [online]. 21.5.2013 [cit. 2017-03-29]. Dostupné z: <http://xmodulo.com/how-to-enable-vnc-remote-console-in-vmware-player.html>
- [29] ČVUT. Systémové inženýrství a rozhodování. *Katedra ekonomiky, manažerství a humanitních věd* [online]. [cit. 2017-04-17]. Dostupné z: <https://ekonom.feld.cvut.cz/cs/student/predmety/systemove-inzenyrstvi-a-rozhodovani#prednasky>
- [30] KOUBOVÁ, Kateřina. *Vícekritériální hodnocení variant za jistoty – metody rozhodování založené na párovém srovnávání variant* [online]. Univerzita Pardubice, 2008 [cit. 2017-04-19]. Dostupné z: [http://dspace.upce.cz/bitstream/handle/10195/28622/KoubovaK\\_Vicekriterialni;jsessionid=569F6147E3CAAD606D4BF65EE3DCD472?sequence=1](http://dspace.upce.cz/bitstream/handle/10195/28622/KoubovaK_Vicekriterialni;jsessionid=569F6147E3CAAD606D4BF65EE3DCD472?sequence=1)
- [31] SEKNIČKOVÁ, Jana. Kardinální informace o kritériích (část 2). In: *Předmět: 4EK415 – Teorie rozhodování – cvičení* [online]. [cit. 2017-04-19]. Dostupné z: <http://jana.kalcev.cz/vyuka/kestazeni/EKO422-Kardinalni2.pdf>
- [32] FRIEBELOVÁ, Jana. Vícekritériální rozhodování za jistoty. In: *Tvorba a softwarová podpora projektů* [online]. 29.10.2008 [cit. 2017-04-19]. Dostupné z: <http://www2.ef.jcu.cz/~jfrieb/tspp/data/teorie/Vicekritko.pdf>

## 8.2 Odborná literatura

- [1] RUEST, Danielle a RUEST, Nelson. *Virtualizace: podrobný průvodce*. 1. vyd. Brno: Computer Press, 2010. ISBN 978-80-251-2676-9.
- [2] JABLONSKÝ, Josef, MAŇAS, Miroslav a FIALA, Petr. *Vícekritériální rozhodování*. 1. vyd. Praha: Vysoká škola ekonomická v Praze, 1994. ISBN 80-7079-748-7.

## 9 Seznam příloh

<b>Příloha 1:</b> Detaily k jednotlivým metodám vícekritériálního hodnocení .....	42
<b>Příloha 2:</b> Kód třídy ElectreIII z implementace metody ELECTRE III .....	53

## 10 Přílohy

### Příloha 1 – Detaily k jednotlivým metodám vícekriteriálního hodnocení

#### Lexikografická metoda

##### První uživatel

Výsledné pořadí	Virtualizační program	Rozhodující kritérium
1.	VirtualBox	Správa virtuálních strojů
2.	VMware Workstation Player	Pokročilost emulace
3.	Windows Virtual PC	-

*Zdroj: vlastní zpracování*

##### Druhý uživatel

Výsledné pořadí	Virtualizační program	Rozhodující kritérium
1.	VirtualBox	Popularita a rozšířenost
2.	VMware Workstation Player	Uživatelská přívětivost
3.	Windows Virtual PC	-

*Zdroj: vlastní zpracování*

##### Třetí uživatel

Výsledné pořadí	Virtualizační program	Rozhodující kritérium
1.	Windows Virtual PC	Cena
2.	VMware Workstation Player	Cena
3.	VirtualBox	-

*Zdroj: vlastní zpracování*

## Metoda váženého součtu pořadí

### Pořadová funkce

	VirtualBox	VMware Workstation Player	Windows Virtual PC
Podpora hostitelských operačních systémů	1	2	3
Podpora hostujících operačních systémů	1,5	1,5	3
Pokročilost emulace	1,5	1,5	3
Správa virtuálních strojů	1	2,5	2,5
Další funkce	1,5	1,5	3
Uživatelská přívětivost	1,5	1,5	3
Hardwarové požadavky	1	3	2
Podpora od výrobce	1,5	1,5	3
Četnost aktualizací	1	2	3
Popularita a rozšířenost	1	2,5	2,5
Cena	osobní použití	2	2
	komerční použití	3	1

Zdroj: vlastní zpracování

### Vážený součet pořadí (menší je lepší)

	První uživatel	Druhý uživatel	Třetí uživatel
VirtualBox	1,315	1,384	1,483
VMware Workstation Player	1,889	1,941	2,036
Windows Virtual PC	2,797	2,670	2,482

Zdroj: vlastní zpracování

## Metoda AGREPREF

První uživatel

*Výsledná preferenční relace*

	<b>VirtualBox</b>	<b>VMware Workstation Player</b>	<b>Windows Virtual PC</b>
<b>VirtualBox</b>	0	1	1
<b>VMware Workstation Player</b>	0	0	1
<b>Windows Virtual PC</b>	0	0	0

*Zdroj: vlastní zpracování*

*Výsledné pořadí*

VirtualBox	1.
VMware Workstation Player	2.
Windows Virtual PC	3.

*Zdroj: vlastní zpracování*

Druhý uživatel

*Výsledná preferenční relace*

	<b>VirtualBox</b>	<b>VMware Workstation Player</b>	<b>Windows Virtual PC</b>
<b>VirtualBox</b>	0	0	1
<b>VMware Workstation Player</b>	0	0	1
<b>Windows Virtual PC</b>	0	0	0

*Zdroj: vlastní zpracování*

### *Výsledné pořadí*

VirtualBox	1.
VMware Workstation Player	
Windows Virtual PC	2.

*Zdroj: vlastní zpracování*

### Třetí uživatel

### *Výsledná preferenční relace*

	<b>VirtualBox</b>	<b>VMware Workstation Player</b>	<b>Windows Virtual PC</b>
<b>VirtualBox</b>	0	1	1
<b>VMware Workstation Player</b>	0	0	1
<b>Windows Virtual PC</b>	0	0	0

*Zdroj: vlastní zpracování*

### *Výsledné pořadí*

VirtualBox	1.
VMware Workstation Player	2.
Windows Virtual PC	3.

*Zdroj: vlastní zpracování*

## Vyhodnocení programem IZAR pro prvního typického uživatele

```
1. MY.idm
Summary Model AGREPREF (1)
max Criterium2;
max Criterium3;
max Criterium4;
max Criterium5;
max Criterium6;
min Criterium7;
max Criterium8;
max Criterium9;
max Criterium10;
min Criterium11;

alternatives
  Alternative1 = [9, 14, 3, 3, 2, 2, 1, 2, 2, 2, 0];
  Alternative2 = [6, 14, 3, 2, 2, 2, 2, 3, 2, 1, 1, 0];
  Alternative3 = [1, 3, 1, 2, 1, 1, 1, 2, 1, 0, 1, 0];

--- method: AGREPREF ---
Y after conversion::
  9 14 3 3 2 2 2 2 2 2 0
  6 14 3 2 2 2 0 2 1 1 0
  1 3 1 2 1 1 1 1 0 1 0

weights = [(1/9) (1/9) (4/27) (1/9) (1/9) (2/27) (1/27) (1/27) (1/9) (2/27) (2/27) ]
alpha = (3/5)
beta = (1/5)
II:
  1 (5/9) (2/27)
  0 1 (7/27)
  0 0 1
S:
  0 (4/9) (25/27)
  0 0 (19/27)
  0 (1/27) 0
PREFERENCE = 1
DISPREFERENCE = -1
INDIFFERENCE = (1/2)
INCOMPARABILITY = 0
X:
  (1/2) 1 1
  -1 (1/2) 1
  -1 -1 (1/2)
Preference matrix:
  0 1 1
  0 0 1
  0 0 0
d = [2 0 -2 ]
P
  0 1 1
  0 0 1
  0 0 0
-----
Alternatives sorted
1: Alternative1 [9, 14, 3, 3, 2, 2, 1, 2, 2, 2, 0]
2: Alternative2 [6, 14, 3, 2, 2, 2, 2, 3, 2, 1, 1, 0]
3: Alternative3 [1, 3, 1, 2, 1, 1, 1, 2, 1, 0, 1, 0]

Solve
```

Zdroj: vlastní zpracování



## Metoda ORESTE

První uživatel

*Výsledná preferenční relace*

	<b>VirtualBox</b>	<b>VMware Workstation Player</b>	<b>Windows Virtual PC</b>
<b>VirtualBox</b>	0	1	1
<b>VMware Workstation Player</b>	0	0	1
<b>Windows Virtual PC</b>	0	0	0

*Zdroj: vlastní zpracování*

*Výsledné pořadí*

VirtualBox	1.
VMware Workstation Player	2.
Windows Virtual PC	3.

*Zdroj: vlastní zpracování*

Druhý uživatel

*Výsledná preferenční relace*

	<b>VirtualBox</b>	<b>VMware Workstation Player</b>	<b>Windows Virtual PC</b>
<b>VirtualBox</b>	0	1	1
<b>VMware Workstation Player</b>	0	0	1
<b>Windows Virtual PC</b>	0	0	0

*Zdroj: vlastní zpracování*

### *Výsledné pořadí*

VirtualBox	1.
VMware Workstation Player	2.
Windows Virtual PC	3.

*Zdroj: vlastní zpracování*

### Třetí uživatel

### *Výsledná preferenční relace*

	<b>VirtualBox</b>	<b>VMware Workstation Player</b>	<b>Windows Virtual PC</b>
<b>VirtualBox</b>	0	1	1
<b>VMware Workstation Player</b>	0	0	1
<b>Windows Virtual PC</b>	0	0	0

*Zdroj: vlastní zpracování*

### *Výsledné pořadí*

VirtualBox	1.
VMware Workstation Player	2.
Windows Virtual PC	3.

*Zdroj: vlastní zpracování*

### Teorie

Cílem metody je rozklad množiny všech variant  $\mathbf{X}$  na disjunktní indifferenční třídy  $\mathbf{X}_1, \mathbf{X}_2, \dots, \mathbf{X}_t$ , přičemž počet indifferenčních tříd  $t$  není předem znám. Pro každou variantu z  $\mathbf{X}_1$  pak platí, že je ostře preferována před všemi variantami z rozdílů množin  $\mathbf{X}, \mathbf{X}_1$ . Vztah preference existuje pouze mezi různými indifferenčními třídami, v jednotlivých třídách jsou varianty hodnoceny stejně. Nejprve se stanoví indexní množiny kritérií a stupně preference. Pořadí libovolné varianty  $x$  uvnitř množiny všech variant  $\mathbf{X}$  lze definovat funkcí:

$$q(x) = p(x) - r(x)$$

$p(x)$  – počet variant, vůči nimž je varianta  $x$  preferována

$r(x)$  – počet variant, které jsou preferovány před variantou  $x$

Práh preference u této metody nezadává uživatel, ale je generovaný automaticky na základě vstupních dat. Nejprve je určen nejvyšší stupeň preference, druhý nejvyšší stupeň preference poslouží jako práh preference  $c^l$ . Pro každou variantu  $x_i$  se určí hodnota  $d_i^1$ :

$$d_i^1 = p_i^1 - q_i^1$$

$p_i^1$  – počet variant, před kterými je varianta  $x_i$  preferována s prahem preference  $c^l$

$q_i^1$  – počet variant, které jsou preferovány před variantou  $x_i$  s prahem preference  $c^l$

Dále se vybere varianta s maximální hodnotou  $d_i^1$ . Pokud variant s maximální hodnotou bude víc, oddělí se podmnožina všech těchto variant a opakuje se celý proces od výběru prahu preference. Konec nastane při nalezení jediné varianty s maximální hodnotou nebo když už nelze zmenšit práh preference, tj. je roven nule. V druhém případě pak varianty tvoří víceprvkovou indifferenční třídu. Po nalezení indifferenční třídy se varianty z této třídy odeberou z matice stupňů preferencí a při výpočtu dalších indifferenčních tříd se pokračuje obdobně.

Varianty je možné uspořádat sestupně (od nejlepší k nejhorší) nebo naopak vzestupně. Popsaný postup byl pro sestupné uspořádání, při vzestupném se akorát místo maximální hodnoty  $d_i^k$  vybírá minimální. Po provedení obou uspořádání se porovnají výsledky pořadí variant a pokud se pořadí liší nepatrně, udělá se průměrné pořadí. Když se pořadí varianty liší hodně, je možné variantu vyhodnotit jako nesrovnatelnou.

### První uživatel

	↓	↑	Výsledné pořadí
VirtualBox	1.	1.	1.
VMware Workstation Player	2.	2.	2.
Windows Virtual PC	3.	3.	3.

*Zdroj: vlastní zpracování*

### Druhý uživatel

	↓	↑	Výsledné pořadí
VirtualBox	1.	1.	1.
VMware Workstation Player	2.	2.	2.
Windows Virtual PC	3.	3.	3.

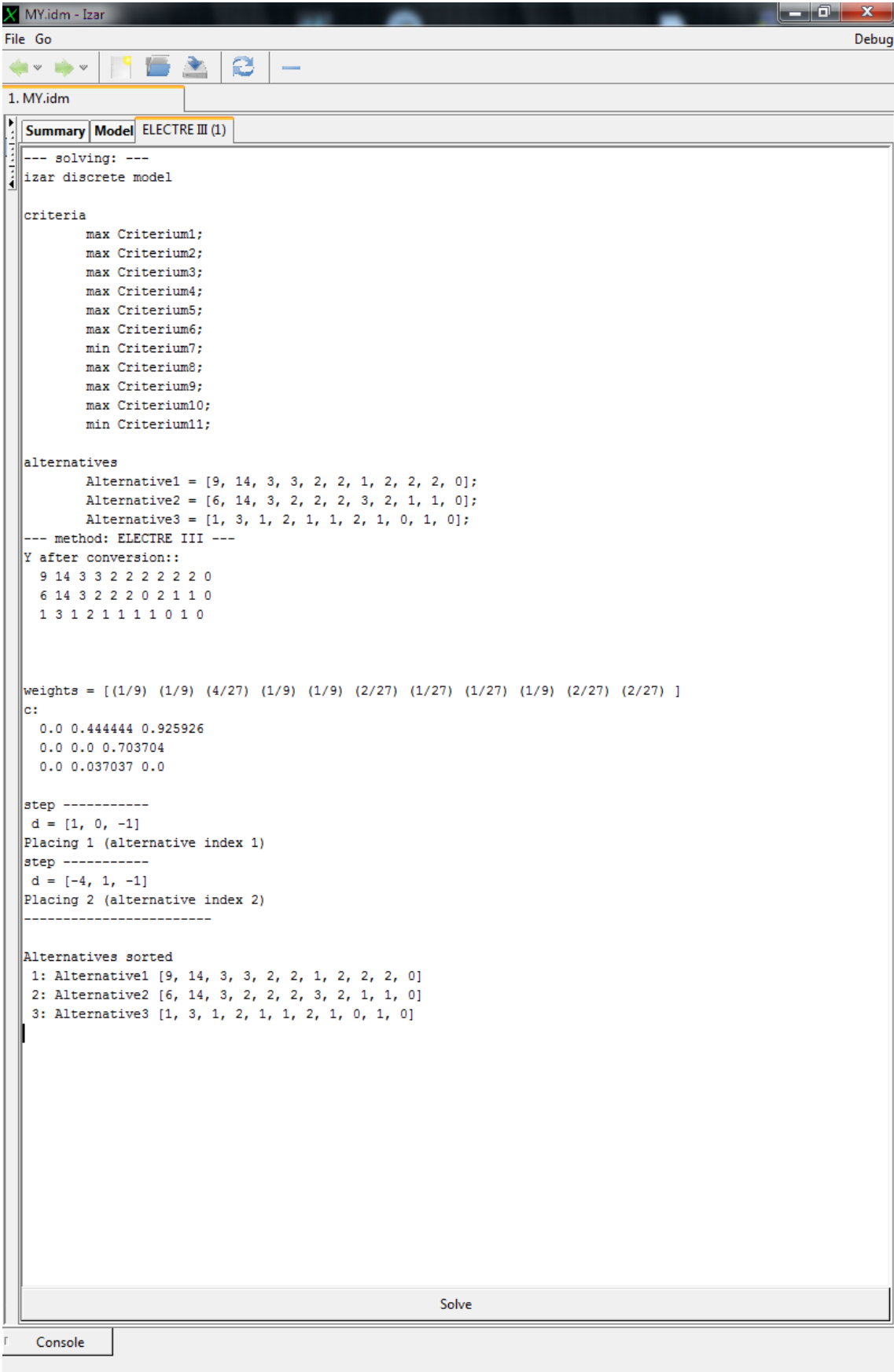
*Zdroj: vlastní zpracování*

### Třetí uživatel

	↓	↑	Výsledné pořadí
VirtualBox	1.	1.	1.
VMware Workstation Player	2.	2.	2.
Windows Virtual PC	3.	3.	3.

*Zdroj: vlastní zpracování*

## Vyhodnocení programem IZAR pro prvního typického uživatele



The screenshot shows the IZAR software interface with a console window displaying the following output:

```
MY.idm - Izar
File Go Debug
1. MY.idm
Summary Model ELECTRE III (1)
--- solving: ---
izar discrete model

criteria
  max Criterium1;
  max Criterium2;
  max Criterium3;
  max Criterium4;
  max Criterium5;
  max Criterium6;
  min Criterium7;
  max Criterium8;
  max Criterium9;
  max Criterium10;
  min Criterium11;

alternatives
  Alternative1 = [9, 14, 3, 3, 2, 2, 1, 2, 2, 2, 0];
  Alternative2 = [6, 14, 3, 2, 2, 2, 3, 2, 1, 1, 0];
  Alternative3 = [1, 3, 1, 2, 1, 1, 1, 2, 1, 0, 1, 0];

--- method: ELECTRE III ---
Y after conversion:
  9 14 3 3 2 2 2 2 2 2 0
  6 14 3 2 2 2 0 2 1 1 0
  1 3 1 2 1 1 1 1 0 1 0

weights = [(1/9) (1/9) (4/27) (1/9) (1/9) (2/27) (1/27) (1/27) (1/9) (2/27) (2/27) ]
c:
  0.0 0.444444 0.925926
  0.0 0.0 0.703704
  0.0 0.037037 0.0

step -----
d = [1, 0, -1]
Placing 1 (alternative index 1)
step -----
d = [-4, 1, -1]
Placing 2 (alternative index 2)
-----

Alternatives sorted
1: Alternative1 [9, 14, 3, 3, 2, 2, 1, 2, 2, 2, 0]
2: Alternative2 [6, 14, 3, 2, 2, 2, 3, 2, 1, 1, 0]
3: Alternative3 [1, 3, 1, 2, 1, 1, 1, 2, 1, 0, 1, 0]
```

Solve

Console

Zdroj: vlastní zpracování

## Vyhodnocení pomocí SANNA pro prvního uživatele

	A	B	C	D	E	F	G	H	I	J	K	L	M
1	<b>Metoda ELECTRE III</b>												
2	Aktuální úloha pro 3 varianty a 11 kritérií												
3	dne 18.4.2017 - 16:31:40												
4													
5													
6													
7	Vstupní data:												
8		MIN	MAX	MAX	MAX	MAX	MAX	MAX	MIN	MAX	MAX	MAX	MIN
9		1.Kritérium	2.Kritérium	3.Kritérium	4.Kritérium	5.Kritérium	6.Kritérium	7.Kritérium	8.Kritérium	9.Kritérium	10.Kritérium	11.Kritérium	
10	1.Varianta	9	14	3	3	2	2	1	2	2	2	2	0
11	2.Varianta	6	14	3	2	2	2	3	2	1	1	1	0
12	3.Varianta	1	3	1	2	1	1	2	1	0	1	1	0
13	Váhy	0,11111	0,11111	0,14815	0,11111	0,11111	0,07407	0,03704	0,03704	0,11111	0,07407	0,07407	
14													
15													
16	Upravené vstupní data:												
17		MAX	MAX	MAX	MAX	MAX	MAX	MAX	MAX	MAX	MAX	MAX	MAX
18		1.Kritérium	2.Kritérium	3.Kritérium	4.Kritérium	5.Kritérium	6.Kritérium	7.Kritérium	8.Kritérium	9.Kritérium	10.Kritérium	11.Kritérium	
19	1.Varianta	0	14	3	3	2	2	2	2	2	2	2	0
20	2.Varianta	3	14	3	2	2	2	0	2	1	1	1	0
21	3.Varianta	8	3	1	2	1	1	1	1	0	1	1	0
22	Váhy	0,11111	0,11111	0,14815	0,11111	0,11111	0,07407	0,03704	0,03704	0,11111	0,07407	0,07407	
23													
24													
25	Matice S:												
26		1.Varianta	2.Varianta	3.Varianta									
27	1.Varianta	0,00000	0,33333	0,81481									
28	2.Varianta	0,11111	0,00000	0,59259									
29	3.Varianta	0,11111	0,14815	0,00000									
30													
31													
32	Indiferenční třídy:												
33		Ind. Třída	Varianta										
34	1.	1.Varianta											
35	2.	2.Varianta											
36	3.	3.Varianta											
37													
38													
39													
40													

Zdroj: vlastní zpracování

## Příloha 2: Kód třídy ElectreIII z implementace metody ELECTRE III

```
class ElectreIII {

    private $header;
    private $data;

    public function __construct(DataHeader $header, Data $data) {
        $this->header = $header;
        $this->data = $data;
    }

    public function getResults() {
        $preferenceDegreeMatrix = $this->getPreferenceDegreeMatrix();
        $ascendingOrder = $this->getVariantsOrder($preferenceDegreeMatrix, 'ASCENDING');
        $descendingOrder = $this->getVariantsOrder($preferenceDegreeMatrix, 'DESCENDING');
        $variantsWithBothOrders = $this->mapOrdersToVariants($ascendingOrder, $descendingOrder);
        $results = $this->addDeviationAndAverageOrder($variantsWithBothOrders);
        return $results;
    }

    private function addDeviationAndAverageOrder($variantsWithBothOrders) {
        for($i = 0; $i < count($variantsWithBothOrders); $i++) {
            list($ascendingOrder, $descendingOrder) = $variantsWithBothOrders[$i];
            $deviation = abs($ascendingOrder - $descendingOrder);
            $averageOrder = ($ascendingOrder + $descendingOrder) / 2;
            array_push($variantsWithBothOrders[$i], $deviation, $averageOrder);
        }
        return $variantsWithBothOrders;
    }

    private function mapOrdersToVariants($ascendingOrder, $descendingOrder) {
        $result = array();

        for($i = 0; $i < count($ascendingOrder); $i++) {
            $variant = $ascendingOrder[$i];
            $variantPlace = $i + 1;
            if(is_array($variant)) {
```

```

    foreach($variant as $v) {
        $result[$v] = array($variantPlace);
    }
    continue;
}
$result[$variant] = array($variantPlace);
}

for($j = count($descendingOrder) - 1, $variantPlace = 1; $j >= 0; $j--, $variantPlace++) {
    $variant = $descendingOrder[$j];
    if(is_array($variant)) {
        foreach($variant as $v) {
            array_push($result[$v], $variantPlace);
        }
        continue;
    }
    array_push($result[$variant], $variantPlace);
}

return $result;
}

private function getPreferenceDegreeMatrix() {
    $preferenceDegreeMatrix = array();
    $variantsCount = $this->data->getSize();
    for($i = 0; $i < $variantsCount; $i++) {
        for($j = $i; $j < $variantsCount; $j++) {
            if($i === $j) {
                $preferenceDegreeMatrix[$i][$j] = 0;
            }
            else {
                $preferenceDegrees = $this->getPreferenceDegrees($i, $j);
                $preferenceDegreeMatrix[$i][$j] = $preferenceDegrees[0];
                $preferenceDegreeMatrix[$j][$i] = $preferenceDegrees[1];
            }
        }
    }
}
}

```



```

return $preferenceDegreeMatrix;
}

private function getPreferenceDegrees($a, $b) {
    $preferenceDegrees = array(0, 0);
    for($i = 0; $i < $this->header->getSize(); $i++) {
        $weight = $this->header->getRelAt($i);
        $aCriterionValue = $this->data->getAt($a, $i + 1);
        $bCriterionValue = $this->data->getAt($b, $i + 1);
        if($aCriterionValue > $bCriterionValue) {
            $preferenceDegrees[0] += $weight;
        }
        elseif($bCriterionValue > $aCriterionValue) {
            $preferenceDegrees[1] += $weight;
        }
    }
}

return $preferenceDegrees;
}

private function getVariantsOrder($preferenceDegreeMatrix, $type, $variantsMapping = NULL) {
    if(!$variantsMapping) {
        $defaultOrder = array_keys($preferenceDegreeMatrix);
        $variantsMapping = array_combine($defaultOrder, $defaultOrder);
    }

    list($max, $secondMax) = $this->getTwoMaxElements($preferenceDegreeMatrix);
    $variantsCount = count($preferenceDegreeMatrix);

    if($variantsCount === 1) {
        return array_values($variantsMapping);
    }

    $alphaClasificationWinners = array();

    for($i = 0; $i < $variantsCount; $i++) {
        $alphaClasification = 0;

```

```

for($j = 0; $j < $variantsCount; $j++) {
    if($preferenceDegreeMatrix[$i][$j] > $secondMax) {
        $alphaClasification++;
    }
    if($preferenceDegreeMatrix[$j][$i] > $secondMax) {
        $alphaClasification--;
    }
}

$leadingAlphaClasification = key($alphaClasificationWinners);

if($leadingAlphaClasification === $alphaClasification) {
    array_push($alphaClasificationWinners[$leadingAlphaClasification], $i);
}
elseif(empty($alphaClasificationWinners) || !$this->checkLeader($leadingAlphaClasification,
$alphaClasification, $type)) {
    $alphaClasificationWinners = array($alphaClasification => array($i));
}

}

$winners = array_pop($alphaClasificationWinners);
$winnersCount = count($winners);

if($winnersCount === 1) {
    $winningVariant = $variantsMapping[$winners[0]];
    $newPreferenceDegreeMatrix = $this->removeVariants($preferenceDegreeMatrix, $winners);
    array_splice($variantsMapping, $winners[0], 1);
    return array_merge(array($winningVariant), $this->getVariantsOrder($newPreferenceDegreeMatrix,
$type, $variantsMapping));
}
elseif($winnersCount === $variantsCount) {
    $remainingVariants = array_values($variantsMapping);
    return array($remainingVariants);
}
else {
    $semiResult = array();
    $otherVariants = array_diff(array_keys($preferenceDegreeMatrix), $winners);

```

```

$newPreferenceDegreeMatrix = $this->removeVariants($preferenceDegreeMatrix, $otherVariants);
$newVariantsMapping = array();

for($i = 0; $i < count($variantsMapping); $i++) {
    if(in_array($i, $winners)) {
        array_push($newVariantsMapping, $variantsMapping[$i]);
    }
}

$semiResult = $this->getVariantsOrder($newPreferenceDegreeMatrix, $type,
$newVariantsMapping);

if(empty($otherVariants)) {
    return $semiResult;
}

$newPreferenceDegreeMatrix = $this->removeVariants($preferenceDegreeMatrix, $winners);
$newVariantsMapping = array();

for($i = 0; $i < count($variantsMapping); $i++) {
    if(!in_array($i, $winners)) {
        array_push($newVariantsMapping, $variantsMapping[$i]);
    }
}

return array_merge($semiResult, $this->getVariantsOrder($newPreferenceDegreeMatrix, $type,
$newVariantsMapping));
}

private function getTwoMaxElements($arr) {
    $max = NULL;
    $secondMax = NULL;
    $arr = call_user_func_array('array_merge', $arr);
    rsort($arr);

    for($i = 0; $i < count($arr); $i++) {

```

```

if(!$max) {
    $max = $arr[0];
}
elseif(!$secondMax && $arr[$i] < $max) {
    $secondMax = $arr[$i];
    break;
}
}

return array($max, $secondMax);
}

private function checkLeader($leader, $rival, $type) {
    switch($type) {
        case 'ASCENDING':
            return $leader > $rival;
        case 'DESCENDING':
            return $leader < $rival;
    }
}

private function removeVariants($preferenceDegreeMatrix, $indexes) {
    $newPreferenceDegreeMatrix = array();

    for($i = 0, $variantsCount = count($preferenceDegreeMatrix); $i < $variantsCount; $i++) {
        if(in_array($i, $indexes)) {
            continue;
        }
        $row = array();
        for($j = 0; $j < $variantsCount; $j++) {
            if(!in_array($j, $indexes)) {
                array_push($row, $preferenceDegreeMatrix[$i][$j]);
            }
        }
        array_push($newPreferenceDegreeMatrix, $row);
    }

    return $newPreferenceDegreeMatrix;
}

```

}

}