UNIVERZITA PARDUBICE

DOPRAVNÍ FAKULTA JANA PERNERA

Metody určování nástupištní koleje pro zpožděný přijíždějící vlak v osobních železničních stanicích s využitím výpočetní techniky

Bc. Michal Česnek

Diplomová práce

2008

Univerzita Pardubice
Dopravní fakulta Jana Pernera
Katedra informatiky v dopravě
Akademický rok: **2007/2008**

ZADÁNÍ DIPLOMOVÉ PRÁCE

(PROJEKTU, UMĚLECKÉHO DÍLA, UMĚLECKÉHO VÝKONU)

Jméno a příjmení:	**Bc. Michal ČESNEK**
Studijní program:	**N3708 Dopravní inženýrství a spoje**
Studijní obor:	**Aplikovaná informatika v dopravě**
Název tématu:	**Metody určování nástupištní koleje pro zpožděný přijíždějící vlak v osobních železničních stanicích s využitím výpočetní techniky**

Zásady pro vypracování:

V simulačních modelech osobních železničních stanic s uvažováním příjezdu zpožděných vlaků vzniká problém s určením náhradní nástupištní koleje pro takový vlak. Pro určení nástupištní koleje lze využít metody vícekriteriálního hodnocení variant. Cílem diplomové práce je získat výsledky pro jednotlivé metody vícekriterilního hodnocení variant a tyto metody porovnat.
Vstupy:

- vlaky osobní dopravy (vstupní kolej do stanice, čas příjezdu, čas odjezdu, výstupní kolej ze stanice, délka vlaku, datumová omezení jízdy vlaků apod.),
- uspořádání kolejiště v osobní stanici (možnosti využití jednotlivých nástupištních kolejí pro vstupní, resp. výstupní koleje do/ze stanice, délky nástupištních kolejí apod.),
- staniční intervaly,
- podklady pro jednotlivé metody vícekriteriálního hodnocení variant.

Výstupy:

- grafické znázornění obsazení kolejí v osobní stanici,
- váhy jednotlivých kritérií pro rozhodování o přidělení nástupištní koleje,
- informace o určené koleji pro zpožděný přijíždějící vlak.

Rozsah grafických prací:

Rozsah pracovní zprávy: **50 normostran**

Forma zpracování diplomové práce: **tištěná/elektronická**

Seznam odborné literatury:

1. **FIALA, P., JABLONSKÝ, F., MAŇAS, M.** *Vícekriteriální rozhodování.* **Praha : Vysoká škola ekonomická v Praze, 1994. 316 s. ISBN 80-7079-748-7.**

2. **BAŽANT, M. Metodika určování náhradní nástupištní koleje pro přijíždějící zpožděný vlak. In** *INFOTRANS 2007 : Sborník příspěvků.* **Pardubice : Univerzita Pardubice, 2007, s. 57–62, ISBN 978-80-7194-989-3.**

3. **VONKA, J., MOLKOVÁ, T., ŠIROKÝ, J.** *Technologie a řízení dopravy II : GVD.* **Pardubice : Univerzita Pardubice, 2000. ISBN 80-7194-286-3.**

4. **VONKA, J. et al.** *Osobní doprava.* **1. vyd. Pardubice : Univerzita Pardubice, 2001. 170 s. ISBN 80-7194-320-7.**

Vedoucí diplomové práce: **Ing. Michael Bažant**
Katedra informatiky v dopravě

Datum zadání diplomové práce: **4. prosince 2007**

Termín odevzdání diplomové práce: **4. června 2008**

prof. Ing. Bohumil Culek, CSc.
děkan

L.S.

doc. Ing. Josef Volek, CSc.
vedoucí katedry

V Pardubicích dne 30. listopadu 2007

Souhrn

Práce se zabývá problematikou vyhledávání náhradní nástupištní koleje pro zpožděný přijíždějící vlak v osobní železniční stanici s využitím výpočetní techniky. Práce řeší problematiku hodnocení jednotlivých variant kolejí pomocí stanovených kritérií. Dále popisuje jak tato kriteria vyhodnotit a zvolit nejvhodnější variantu. Celá problematika je analyzována a v programátorské části popsána navrženými datovými strukturami pro tvorbu aplikace VNNK (Výpočet náhradní nástupištní koleje). Dále se zabývá případovou studií železniční stanice Praha hlavní nádraží, kde za pomocí aplikace VNNK vyhledá k několika zpožděným přijíždějícím vlakům náhradní koleje. Aplikace vyhledává a vybírá nejvhodnější kolej pomocí různých metod vícekriteriálního hodnocení variant, které jsou v závěrečné části expertně posouzeny z hlediska procentuální úspěšnosti.

Klíčová slova

náhradní kolej; zpožděný vlak; kritéria; vícekriteriální rozhodování; spolehlivost metod;

Title

Methods for assigning alternative platform track for delayed arriving train in passenger railway station using computer technology

Abstract

The work deals with a problem of searching an alternative platform rail for delayed arriving train in passenger railway station using computer technology. The work solves the task of evaluating individual track variations by the help of determined criterions. Further it describes how to evaluates these criterions and choose the most suitable variation. The whole problem is analyzed and described using the data structures designed for development of an application VNNK (computation of an alternative platform rail) in programming part of the work. Next the work is engaged in a case study of railway station Prague, the main station, in which it searches the alternative tracks for several delayed arriving trains with help of application VNNK. The application searches and chooses the most appropriate track using the different methods of multicriterial evaluation of variants, which are qualified by expert in aspect of hit-rate percentage in the final part of the work.

Keywords

alternative track; delayed train; criteria; multicriterial decision making; method reliability;

Poděkování.

Rád bych poděkoval především vedoucímu práce Ing. Michaelu Bažantovi za všechny rady a připomínky při zpracování diplomové práce. Děkuji všem kteří mi pomohli dobrou radou, případně jinak.

Obsah

Seznam obrázků

1 Úvod

Při tvorbě modelů osobních železničních stanic s uvažováním příjezdu zpožděných vlaků vzniká problém s určením náhradní nástupištní koleje pro takový vlak. Pro určení nástupištní koleje lze, mimo jiných metod, využít také metody vícekriteriálního hodnocení variant.

Toto téma je řešeno z důvodu automatizace podpory rozhodování o výběru z několika náhradních kolejí.

Cílem diplomové práce je získat výsledky pro jednotlivé metody vícekriteriálního hodnocení variant a tyto metody porovnat.

Aby bylo vůbec možné získat takovéto výsledky je zapotřebí vypočítat váhy jednotlivých kritérií. Toto problematikou se zabývá kapitola 2. Nejprve si zjistíme množinu přípustných kolejí a každému prvku z této množiny vypočteme čtyři posuzovací kritéria. Tato kritéria jsou označována velkými písmeny A až D. Význam jednotlivých kritérií je popsán v kapitole 2.2.

Tato kritéria umožňují aplikovat metody vícekriteriálního hodnocení variant. V kapitole 3 je vybráno pět metod, kde každá z nich přistupuje k řešení svým vlastním způsobem. Metody jsou navíc parametrizovány vstupními informacemi o vztazích mezi kritérii a své výsledky podávají dle nastavených parametrů. Aby bylo možné porovnat spolehlivost těchto metod, je třeba expertně zhodnotit výsledky a posoudit jejich procentuální úspěšnost.

Práce si klade za cíl vytvořit grafickou aplikaci, která by znázorňovala obsazení kolejí v osobní stanici. Ke splnění tohoto cíle bylo zapotřebí analyzovat problematiku a navrhnout skupinu datových struktur, které umožňují načíst vstupní data a poskytnou dostatečně vhodné rozhraní pro vykreslení časoprostorového obsazení kolejí v osobní železniční stanici. Touto problematikou se zabývá kapitola 4. Na těchto strukturách je dále postavena aplikace VNNK (**V**ypočet **N**áhradní **N**ástupištní **K**oleje), která v sobě agreguje nejen vykreslovací grafické funkce, ale i početní funkce umožňující hledat náhradní koleje kterémukoliv uživatelem zvolenému vlaku.

V kapitole 5 je popsána práce s aplikací VNNK na vstupních datech železniční stanice Praha hlavní nádraží. Je zde popsáno jak z uživatelského hlediska pracovat s aplikací.

Splněním dalšího cíle a to porovnáním metod vícekriteriálního hodnocení variant s využitím aplikace VNNK se zabývá kapitola 5.7. Je zde popsáno jak získat pomocí aplikace testovací data, které je možné expertně posoudit a zjistit tak procentuální úspěšnost jednotlivých metod.

Aby byla aplikace testována na konkrétních datech věnuje se kapitola 6 případové studii osobní železniční stanice Praha hlavní nádraží. Z této stanice byly poskytnuty vstupní potřebná data charakterizující vlaky osobní dopravy, uspořádání kolejiště v osobní stanici a staniční intervaly. Tato kapitola se dále věnuje tvorbě výstupních dat. Vybírá testovací množinu vlaků. Na této množině, za pomocí aplikace VNNK, provádí testovací výpočty a zjišťuje procentuální úspěšnost vícekriteriálních metod. Tato úspěšnost je dále zvyšována hledáním vhodnějších vah kritérií. V závěrečné části je určen žebříček úspěšnosti metod.

2 Problematika určování nástupištní koleje pro zpožděný přijíždějící vlak

Při modelování provozu osobní železniční stanice se zahrnutím příjezdu zpožděných vlaků vzniká problém s určením vhodné koleje pro takový vlak. Tato kapitola se zabývá možností řešení tohoto problému pomocí stanovení kritérií pro hodnocení jednotlivých kolejí v rámci simulačního modelu. [1]

2.1 Úvod do problematiky

Tato kapitola se zabývá výběrem koleje v těch osobních stanicích, kde je pro každý přijíždějící vlak možnost výběru z více kolejí, to znamená že pro každý vlak přichází v úvahu alespoň dvě koleje.

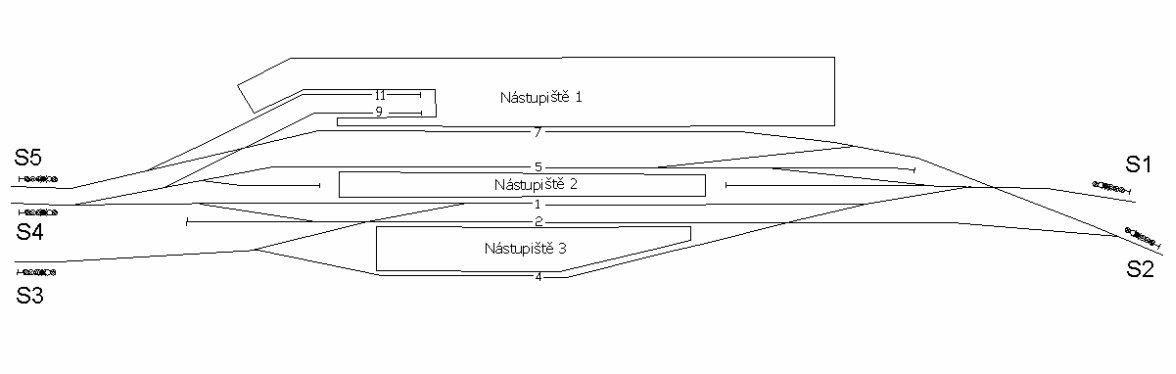
Přidělená kolej by měla odpovídat rozhodovacím mechanismům, které jsou rutinně uplatňovány na stanicích řídícími pracovníky, čímž by model osobní stanice v této oblasti do určité míry kopíroval rozhodování o řešení provozních problémů v praxi. Pro takové chování byla navrhnuta čtyři kritéria, která jsou uvedena v následující kapitole.

2.2 Kritéria pro určení náhradní nástupištní koleje

Při určování kritérií, podle kterých bude probíhat přiřazení náhradní nástupištní koleje v rámci simulačního modelu je nutné se podívat na tento proces v provozu. Pokud do stanice přijíždí zpožděný vlak, řídící pracovník musí vyhodnotit takovou kolej, která bude pro daný vlak nejvhodnější, přičemž je ve hře několik kritérií. Stejný postup bude uplatněn při určování nástupištní koleje v modelu (pro zjednodušení budeme v první fázi uvažovat s příjezdy zpožděných vlaků pouze z jednoho směru, přičemž by nemělo být problematické rozšířit úlohu na libovolný počet směrů).

Prvním krokem při výběru nástupištní koleje je určení množin přípustných kolejí pro přijíždějící vlak, které jsou určeny vjezdovou kolejí do simulačního modelu a odjezdovou kolejí ze simulačního modelu.

Množiny přípustných kolejí si označíme $K_{Si,Sj}$, kde S_i je označení vstupní koleje do modelu, S_j je označení výstupní koleje z modelu. Určení množiny přípustných kolejí si můžeme demonstrovat na následujícím příkladu viz obrázek 1. Pro příklad uvedený na obrázku lze určit např. tyto množiny přípustných kolejí: $K_{S1,S1} = \{4, 2, 1, 5, 7\}$, $K_{S1,S2} = \{4, 2, 1, 5, 7\}$, ..., $K_{S5,S5} = \{7, 9, 11\}$. Tyto množiny lze ještě dále redukovat o koleje nevhodné pro uvažovaný vlak např. z důvodu nedostatečné délky koleje apod.



Obrázek 1 Určení množiny přípustných kolejí (zdroj: [1])

Pokud je určena množina přípustných kolejí pro příjezd zpožděného přijíždějícího vlaku, je nutné z této množiny vybrat tu kolej, která bude v době skutečného příjezdu vlaku pro něj nejvýhodnější. Pro určení výhodnosti kolejí je nutné zvolit odpovídající kritéria. Kritéria pro určení koleje byla odvozena ze znalosti práce řídících pracovníků a jsou v zásadě čtyři:

A. volnost koleje,

B. doba volnosti koleje vzhledem k době pobytu přijíždějícího vlaku ve stanici,

C. obsazení koleje u stejného nástupiště přípojným vlakem,

D. preference koleje pro přijíždějící vlak.

Z definice úlohy je zřejmé, že se jedná o úlohu vícekriteriálního hodnocení variant, neboť množina rozhodovacích variant (množina přípustných kolejí), kterou budeme značit $K = (k_1, ..., k_m)$, má konečný počet prvků.

Pokud máme určena kritéria (*A, B, C, D*) a metody získání kvantitativních údajů o hodnotách těchto kritérií pro jednotlivé rozhodovací varianty, lze úlohu vícekriteriálního hodnocení variant charakterizovat tzv. kriteriální maticí.

V této matici řádky odpovídají kriteriím a sloupce hodnoceným variantám. Označíme-li prvky kriteriální matice y_{ij}, i = 1, 2, ..., 4, j = 1, 2, ..., m, můžeme kriteriální matici zapsat ve tvaru:

$$\begin{array}{c} \\ A \\ B \\ C \\ D \end{array} \begin{array}{c} \begin{array}{cccc} k_1 & k_2 & \dots & k_m \end{array} \\ \begin{pmatrix} y_{11} & y_{12} & \dots & y_{1m} \\ y_{21} & y_{22} & \dots & y_{2m} \\ y_{31} & y_{32} & \dots & y_{3m} \\ y_{41} & y_{42} & \dots & y_{4m} \end{pmatrix} \end{array} \qquad (1)$$

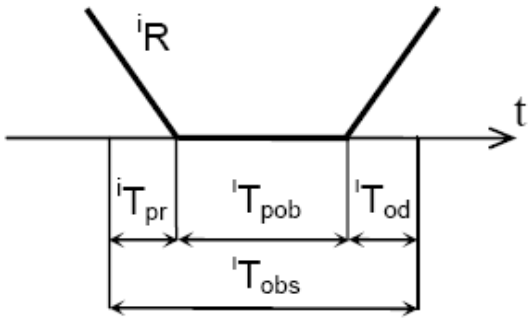
V další části bude uvedeno vyhodnocování jednotlivých kritérií, přičemž uplatníme maximalizační zásadu. To znamená, že kriteria jsou určena tak, že varianta je tím lepší, čím jsou hodnoty kritérií větší.

Výpočet hodnot kritérií *A a B* vychází z *Plánu obsazení kolejí ve stanici*, který se sestavuje pro každou větší osobní stanici. Pro každou dopravní kolej ve stanici jsou k dispozici údaje o jejím obsazení vlaky s krokem který je vhodný pro příslušný model. V tomto modelu v rámci zjednodušení uvažujme krok o délce jedné minuty.

2.2.1 Volnost koleje

Kritérium hodnotící volnost koleje by mělo logicky nabývat pouze dvou hodnot a to kolej je volná nebo je kolej obsazená. Takto postavené kritérium by ale nedokázalo činit rozdíly mezi kolejemi, které jsou v daném čase obsazené a budou obsazené na dlouhou dobu a kolejemi, které jsou v daném čase obsazené, ale v poměrně krátkém čase už by mohly být využity přijíždějícím vlakem. Z tohoto důvodu je vhodné toto kritérium rozšířit o faktor času se stanoveným výhledem do budoucna.

Pokud zpožděný vlak ${}^{j}R$ přijíždí k uvažované koleji mimo interval ${}^{i}T_{obs}$, kolej je volná a hodnota kritéria *A* je rovna jedné, tedy *A = 1*.



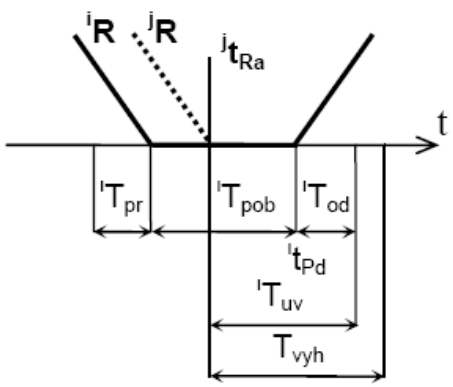
${}^{i}R$	vlak, který obsazuje uvažovanou kolej
${}^{i}T_{pob}$	doba pobytu vlaku ${}^{i}R$ na koleji
${}^{i}t_{\mathrm{Pr}}$	doba, kterou je nutné uvažovat pro příjezd vlaku ${}^{i}R$ na kolej
${}^{i}T_{od}$	doba, kterou je nutné uvažovat pro odjezd vlaku ${}^{i}R$ z koleje
${}^{i}T_{obs}$	celková doba obsazení koleje vlakem ${}^{i}R$

Obrázek 2 Celková doba obsazení koleje jedním vlakem (zdroj: [1])

Nyní se již můžeme podívat na uvedené rozšíření tohoto kritéria o faktor času, tedy o výhled na stanovenou dobu do budoucna. Tento výhled si můžeme označit T_{vyh}. Tuto informaci využijeme pro ohodnocení koleje v případě, kdy zpožděný vlak přijíždí v čase kdy je kolej obsazená. Mohou nastat dva případy:

a) Během doby výhledu dojde k uvolnění koleje (tato situace je na obrázku 3). Hodnota kritéria A se vypočte podle vztahu (3).

b) Během doby výhledu nedojde k uvolnění koleje, potom podle vztahu (2) platí, že A = 0.



^{j}R	přijíždějící zpožděný vlak
$^{j}t_{Ra}$	čas příjezdu zpožděného vlaku ^{j}R
$^{i}t_{Pd}$	čas odjezdu vlaku ^{i}R
$^{i}T_{uv}$	doba zbývající do uvolnění koleje vlakem ^{i}R
T_{vyh}	doba, kterou uvažujeme jako maximální pro výhled vzhledem k možné změně stavu obsazení koleje

Obrázek 3 Příjezd zpožděného vlaku v době obsazení uvažované koleje (zdroj: [1])

Dobu zbývající do uvolnění koleje vlakem ^{i}R lze vypočítat podle vztahu:

$$^{i}T_{uv} = {}^{i}t_{Pd} - {}^{j}t_{Ra} + {}^{i}T_{od} \qquad (2)$$

Pro výpočet hodnoty kritéria *A* platí vztah (uplatňujeme opět maximalizační zásadu):

$$A = \max\{0, 1 - \frac{{}^{i}t_{Pd} - {}^{j}t_{Ra} + {}^{i}t_{od}}{T_{vyh}}\} \qquad (3) \quad .$$

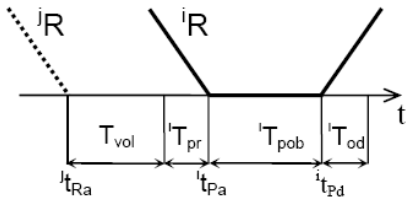
2.2.2 Doba volnosti koleje vzhledem k době pobytu vlaku ve stanici

Druhým kritériem je doba volnosti koleje vzhledem k době pobytu přijíždějícího zpožděného vlaku ve stanici. Stanovení hodnoty kritéria *B se* vztahuje k plánovanému času pobytu vlaku ^{j}R *ve* stanici. V první fázi stanovení hodnot kritérií uvažujeme s konstantní (plánovanou) dobou pobytu vlaku ve stanici, přičemž v další fázi výzkumu je možné s tímto kritériem dále pracovat v tom smyslu, že je možné u vybraných vlakových spojů uvažovat o kratší než plánované době pobytu vlaku ve stanici. Tato poznámka se týká zejména vlaků, které mají ve stanicích delší plánovanou dobu pobytu a zkrácením doby

pobytu by nevznikaly komplikace při vykonávání obsluhy podle stanovených technologických postupů.

Stejně jako u kritéria *A* bude i u tohoto kritéria uplatněna maximalizační zásada a hodnota kritéria bude z intervalu $\langle 0, 1 \rangle$.

Výpočet probíhá na základě vyhodnocení doby, po kterou je uvažovaná kolej volná do příjezdu dalšího vlaku. Postup výpočtu pro příjezd zpožděného vlaku v čase kdy je uvažovaná kolej volná je znázorněn na následujícím obrázku.



$^{i}t_{Pa}$ čas plánovaného příjezdu vlaku ^{i}R

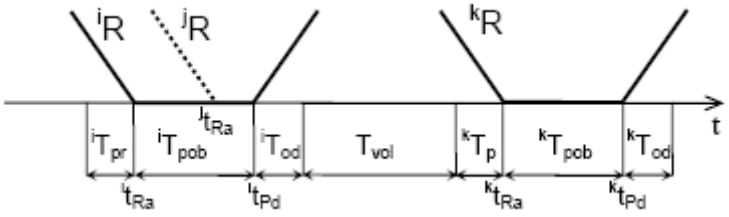
$^{j}T_{pob}$ plánovaná doba pobytu vlaku ^{j}R na koleji

Obrázek 4 Určení kritéria B v čase kdy je uvažovaná kolej volná (zdroj: [1])

V tomto případě lze hodnotu kritéria *B* vypočítat dle tohoto vztahu:

$$B = \min\{1, \frac{T_{vol}}{{}^{j}T_{pr} + {}^{j}T_{pob} + {}^{j}T_{od}}\} \qquad (4)$$

Pokud vlak ^{j}R přijede v době kdy je uvažovaná kolej obsazená, je postup výpočtů podobný, s tím že uvažujeme dobu, po kterou je kolej volná až po jejím uvolnění vlakem ^{i}R. Tento případ znázorňuje obrázek číslo 5.

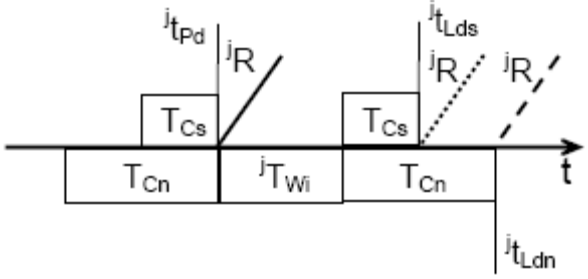


Obrázek 5 Určení kritéria B v čase když je uvažovaná kolej obsazená (zdroj: [1])

2.2.3 Obsazení koleje u stejného nástupiště přípojným vlakem

Pokud na přijíždějící zpožděný vlak čeká přípojný vlak ^{j}R, kterému by tímto čekáním mohlo vzniknout zpoždění, případně by mohlo dojít k nárůstu zpoždění, je vhodné přijíždějící zpožděný vlak umístit ke stejnému nástupišti tak, aby mohla být uplatněna kratší doba potřebná na přestup. Z tohoto důvodu je zavedeno toto kritérium označené C.

Toto kritérium se liší od předchozích kritérií tím, že nabývá pouze dvou hodnot. Na obrázku 6 je znázorněn postup výpočtu kritéria C.



^{j}R	vlak na uvažované koleji
$^{j}t_{Pd}$	plánovaný odjezd vlaku ^{j}R
T_{Cn}	normální doba potřebná na přestup
T_{Cs}	kratší doba potřebná na přestup
$^{j}T_{wi}$	čekací doba vlaku ^{j}R na přijíždějící vlak ^{i}R
$^{j}t_{Lds}$	nejzazší čas odjezdu vlaku ^{j}R při uplatnění kratší doby potřebné na přestup
$^{j}t_{Ldn}$	nejzazší čas odjezdu vlaku ^{j}R při uplatnění normální doby potřebné na přestup
$^{i}t_{Ra}$	skutečný čas příjezdu zpožděného vlaku ^{i}R

Obrázek 6 Určení časového intervalu pro ohodnocení kritéria C (zdroj: [1])

Z obrázku je patrné, že je výhodné pro příjezd zpožděného vlaku ^{i}R v časovém intervalu $\left\langle {}^{j}t_{Pd} - T_{Cn},\ {}^{j}t_{Pd} + {}^{j}T_{Wi} \right\rangle$ tento přijíždějící vlak umístit ke stejnému nástupišti, u něhož na něj čeká přípojný odjíždějící vlak.

Hodnota kritéria *C* pro sousedící koleje s kolejí, ze které odjíždí přípojný vlak:

$$\begin{aligned} C &= 1 \; pro \; {}^{i}t_{Ra} \in \left\langle {}^{j}t_{Pd} - T_{Cn},\ {}^{j}t_{Pd} + {}^{j}T_{Wi} \right\rangle \\ C &= 0 \; pro \; {}^{i}t_{Ra} \notin \left\langle {}^{j}t_{Pd} - T_{Cn},\ {}^{j}t_{Pd} + {}^{j}T_{Wi} \right\rangle \end{aligned} \qquad (5)$$

2.2.4 Preference koleje pro přijíždějící vlak

Toto kritérium představuje vhodnost koleje pro uvažovaný zpožděný přijíždějící vlak ^{j}R. Stejně jako u předchozích kritérií platí maximalizační zásada a hodnoty tohoto kritéria nabývají hodnot od nuly do jedné, D je tedy z intervalu $\langle 0, 1 \rangle$. Kolej pravidelně určená přijíždějícímu vlaku má ohodnocení D = 1, pro koleje nevhodné pro přijíždějící vlak D = 0. Ostatní koleje nabývají ohodnocení z intervalu $(0,1\rangle$, přičemž je možné měnit hodnotu tohoto kritéria pro různé časy příjezdu vlaku ^{j}R. Hodnotu kritéria D může stanovit

experimentátor na základě svých znalostí provozu zkoumané stanice, případně lze stanovit hodnotu tohoto kritéria po konzultaci s provozními zaměstnanci.

2.3 Metody stanovení vah kriterií

V předchozích kapitolách jsme si definovali úlohu a uvedli způsob výpočtu kritérií A–D. Po těchto krocích již máme kriteriální matici (1) s konkrétními hodnotami pro jednotlivé časy.

Dalším krokem je stanovení vah jednotlivých kritérií pomocí klasických metod vícekriteriálního hodnocení variant. Tyto metody jsou dále popsány v následující kapitole 3.

3 Popis metod vícekriteriálního hodnocení variant

Vícekriteriální analýza variant je charakteristická tím, že umožňuje hodnotit konečný počet variant podle konečného počtu různých kritérií (alespoň dvou).

Vybraná varianta rozhodnutí stanovená některou z metod vícekriteriálního hodnocení variant vychází z informací o preferenci jednotlivých kritérií a o preferenci jednotlivých variant podle jednotlivých kritérií a z použité metody řešení. Protože různé metody mohou poskytnout rozdílná řešení, je nutné při vícekriteriálním hodnocení variant uplatnit více metod a ověřit si citlivost výsledků vzhledem k použitým metodám.

Člověk, který není seznámen s oblastí vícekriteriálního rozhodování, činí rozhodnutí intuitivně. Tento přístup je vhodný zejména u problémů, kdy realizací jiného než nejlepšího řešení nevznikne podstatná škoda.

Naší motivací v tomto případě je automatizace procesu rozhodování a tím eliminace chyb vzniklých intuitivním lidským rozhodnutím.

3.1 Metody stanovení vah kritérií

Stanovení vah kritérií bývá výchozím krokem analýzy modelu vícekriteriální analýzy variant. Téměř výhradně je informace získaná některým z dále uvedených postupů použita ke stanovení preferenčních vztahů mezi variantami v závislosti na cílech celé analýzy.

V následujících podkapitolách jsou uvedeny nejpoužívanější metody stanovení vah mezi kritérii seřazené podle informace, jakou tyto metody požadují na vstupu. Uvedené postupy je možné i kombinovat, resp. používat vedle sebe, ale vše by mělo být podřízeno úspěšnému dosažení cílů analýzy a kritériu účelnosti. [3]

3.2 Možnosti stanovení vah kritérií bez informace o preferenci kritérii

Nemít k dispozici žádnou informaci o preferencích mezi kritérií neznamená nevědět o problému vůbec nic. Samozřejmě se předpokládá, že kriteriální matice kvantifikovaná pomocí kardinálních hodnot existuje. Problém je v tom, že řešitel vůbec neumí (nebo nechce) rozhodnout, jak je které kritérium důležité pro posouzení variant. V takovém případě je samozřejmě možné přiřadit všem kritériím váhu stejnou, vypočtenou podle vztahu, $v_j = \frac{1}{n}$, $j = 1,2,...,n$, kde n je počet kritérií. Pokud však řešitel nechce přiřadit všem kritériím stejnou váhu, může váhový vektor stanovit pomocí entropické metody viz kapitola 3.2.1.

Entropická metoda v této formě je použitelná pouze pro kriteriální matici s kladnými hodnotami, neboť musí být stanoveny pravděpodobnosti p_{ij} a jejich přirozené logaritmy.

Úprava kriteriální matice přičtením vhodné konstanty (ať už k celé matici, nebo pouze k jejímu jednomu sloupci) však může změnit nejen vypočtené váhy a poměr mezi nimi, ale někdy dokonce pořadí důležitosti kritérií.

3.2.1 Entropická metoda

Kriteriální matice $Y = (y_{ij})$ pro množinu alternativ obsahuje určité množství informace. Kritérium není příliš důležité, když hodnoty všech alternativ podle tohoto kritéria jsou podobné. Pokud jsou všechna tato ohodnocení variant podle některého kritéria dokonce všechna stejná, můžeme toto kritérium pro potřeby hodnocení variant zcela vynechat (jinak řečeno toto kritérium má nulovou váhu). Naopak, čím rozdílnější jsou ohodnocení variant podle některého kritéria, tím větší váhu je možno tomuto kritériu přisoudit. Proto je možno použít entropii (resp. míru entropie) pro určení vah kritérií.

Entropie je důležitý pojem ve společenských i přírodních vědách. V teorii informace je kritériem pro množství neurčitosti představované diskrétním rozdělením. Je mírou očekávaného informačního obsahu zprávy, která je vyjádřena takto s pravděpodobnostmi $p_j, j = 1,...,n$.

$$S(p_1, p_2, ..., p_n) = -k\sum_{j=1}^{n} p_j \ln p_j \qquad (6)$$

kde k je kladná konstanta. Pokud se všechna p_j rovnají, tedy $p_j = 1/n$, dosahuje $S(p_1, p_2, ..., p_n)$ maximální hodnoty.

Významnost kritérií tedy bude určena rozdíly velikostí jednotlivých ohodnocení všech variant podle jednotlivých kritérií. Ohodnocení y_{ij} i-té varianty podle j-tého kritéria pak můžeme převést na pravděpodobnosti diskrétní veličiny pomocí vztahu:

$$p_{ij} = \frac{y_{ij}}{\sum_{i=1}^{m} y_{ij}}, i = 1,...,m \; j = 1,...,n \qquad (7)$$

Entropii E_j množiny očekávaných výstupů j-tého kritéria potom vypočteme jako:

$$E_j = -k\sum_{i=1}^{m} p \ln p_{ij}, \forall j, kde\, k = \frac{1}{\ln m} \tag{8}$$

Tato hodnota konstanty k zajišťuje, že hodnota E_j leží v intervalu mezi nulou a jedničkou. Stupeň diversifikace d_j informace poskytované výstupy j-tého kritéria je pak definován jako:

$$d_j = 1 - E_j, j = 1,..., n. \tag{9}$$

Vektor vah potom dostaneme normalizací vektoru d, takže

$$v_j = \frac{d_j}{\sum_{j=1}^{n} d_j}, j = 1,..., n \tag{10}$$

3.3 Stanovení vah kritérií z ordinální informace o preferencích kritérií

Metody pracující s ordinální informací o kritériích předpokládají, že je řešitel schopen a ochoten vyjádřit důležitost jednotlivých kritérií tak, že přiřadí všem kritériím jejich pořadová čísla nebo při porovnání všech dvojic kritérií určí, které kritérium z aktuální dvojice je důležitější než druhé. V obou případech je přípustné označení dvou nebo více kritérií jako rovnocenných. Způsob, jak tuto skutečnost vyjádřit bude popsán u příslušných metod, z nichž uvedeme dvě nejčastěji používané:

a) metodu pořadí – viz kapitola 3.3.1

b) metodu porovnání ve Fullerově trojúhelníku – viz kapitola 3.3.2

Obě dvě metody transformují ordinální informaci do podoby váhového vektoru.

3.3.1 Metoda pořadí

K určení vah kritérií se metoda pořadí používá především v případech, že jejich důležitost hodnotí několik expertů. Každý z nich seřadí kritéria od nejdůležitějšího po nejméně důležité. Nejdůležitější kritérium bude ohodnoceno n body (n je počet kritérií), druhé nejdůležitější n-1 body, atd., až nejméně důležité kritérium dostane jen 1 bod. V případě stejné důležitosti kritérií dostanou tato kritéria body podle průměrného pořadí. Váhu každého z kritérií určíme tak, že sečteme body, které získalo od všech expertů, a vydělíme je celkovým počtem bodů, které experti rozdělili mezi všechna kritéria. Tím je zaručeno, že suma vah všech kritérií je rovna 1.

Je-li obecně *j*-té kritérium ohodnoceno b_j body (jedinou hodnotou nebo součtem hodnot při hodnocení více experty), vypočítá se jeho váha na základě vztahu

$$v_j = \frac{b_j}{\sum_{j=1}^{n} b_j}, j = 1,...,n \qquad (11)$$

Tento vzorec normalizuje informace o preferenci kritérií, postup se proto nazývá normalizace vah kritérií.

3.3.2 Metoda Fullerova trojúhelníka

Pokud ordinální informace vyjadřuje pouze vztah mezi každou dvojicí hodnocených kritérií, lze použít metodu párového porovnávání. Pokud předpokládáme, že v případě, kdy uživatel ohodnotí kritérium *j* jako důležitější než *l* zároveň platí, že kritérium *l* je považováno za méně důležité než kritérium *j*, stačí provést počet srovnání $N = \frac{n(n-1)}{2}$, kde *n* je počet porovnávaných kritérií.

Toto porovnávání se většinou provádí pomocí tzv. Fullerova trojúhelníku**.** **U** každé dvojice prvků se zakroužkuje ten prvek, který se považuje za důležitější. Označíme-li počet zakroužkování *j*-tého prvku n_j, pak ohodnocení či váhu tohoto prvku vypočteme podle vzorce:

$$v_j = \frac{n_j}{N}, j = 1,2,...,n \qquad (12)$$

Nevýhoda tohoto postupu pro výpočet vah kritérií je v tom, že při plně konzistentní informaci od uživatele je vždy hodnota n_j pro nejméně důležité kritérium rovna nule, čímž samozřejmě bude i hodnota váhy v_j tohoto kritéria rovna nule. Pokud bychom byli důslední, mohli bychom toto kritérium vyloučit z množiny kritérií a provést porovnání ve Fullerově trojúhelníku znovu. Pokud bychom tento postup opakovali *k*–1 krát a vždy by byla informace uživatele plně konzistentní, zůstalo by v množině kritérií pouze jediné – to nejdůležitější – kritérium.

Této situaci se můžeme vyhnout tak, že po ukončení porovnání a vyčíslení hodnot n_j všechny tyto hodnoty zvětšíme o hodnotu jedna (jako by bylo každé kritérium porovnáváno též samo se sebou a bylo důležitější). V tom případě budou hodnoty n_j přesně odpovídat hodnotám p_j tak, jak byly tyto hodnoty zavedeny v metodě pořadí. Navíc není jasné, zda hodnotu jedna přičítat k hodnotám n_j vždy nebo pouze v případě, že existuje n_j

rovno nule. Díky normalizace vektoru vah totiž přičtení hodnoty jedna zkreslí poměr mezi všemi dvojicemi vah, přičemž nejdůležitější informací, kterou váhový vektor poskytuje většině metod pro stanovování preferencí mezi variantami, nejsou absolutní hodnoty vektoru vah, ale právě výše uvedené poměry hodnot vah.

3.4 Stanovení vah z kardinální informace o preferencích kritérií

Metody stanovení vah kritérií z kardinální informace o jejich preferencích předpokládají, že je uživatel schopen a ochoten určit nejen pořadí důležitosti kritérií, ale také poměr důležitosti mezi všemi dvojicemi kritérií. Nejpoužívanějšími metodami této oblasti jsou **metoda bodovací** viz kapitola 3.4.1, která transformuje bodové hodnocení důležitosti kritérií do podoby váhového vektoru, a **Saatyho metoda kvantitativního párového porovnání** viz kapitola 3.4.2, která odvozuje váhový vektor z informace o odhadu poměru vah, který stanoví přímo uživatel.

3.4.1 Metoda bodovací

Důležitost každého z variant podle tohoto kritéria vyjádříme určitým počtem bodů v rámci určené bodovací stupnice. Smí se používat i desetinná čísla a více kritériím je možné přiřadit stejnou bodovou hodnotu.

Také tato metoda se pro výpočet vah kritérií používá podobně jako metoda pořadí tehdy, hodnotí-li kritéria více expertů. Každý expert ohodnotí každé kritérium určitým počtem bodů, čím je kritérium důležitější, tím více bodů dostane (při použití stupnice od 0 do 10 může mít kritérium 0 bodů od experta, podle kterého je zcela bezvýznamné, a 10 bodů od experta, který je považuje za absolutně důležité). Stupnice pro bodování může být vyjádřena i graficky pomocí úsečky. na ní jsou pak zakresleny pozice jednotlivých kritérií vzhledem ke koncům úsečky, které vyjadřují nejvyšší a nejnižší preferenci.

Výpočet vah se z bodového hodnocení provede stejně jako u metody pořadí. Hodnoty váhového vektoru se pak normalizují podle vztahu:

$$v_j = \frac{b_j}{\sum_{j=1}^{n} b_j}, kde\ j = 1, 2, ..., n \qquad (13)$$

kde b_j je součet všech bodů od jednotlivých expertů, které j-tému kritériu tito experti přidělili.

Je ovšem otázkou, zda je vždy vhodné stanovit natvrdo rozsah stupnice již na začátku hodnocení. Tento postup je možný v případě, že máme hned na počátku poměrně jasnou představu o tom, jak asi jsou ta která kritéria důležitá pro hodnocení variant. Potom je asi nejvhodnější přiřadit nejdůležitějšímu kritériu nejvyšší možný počet bodů, nejméně důležitému kritériu nejnižší možný počet bodů a všechna ostatní kritéria umístit na danou stupnici s přihlédnutím na hodnocení nejen těchto dvou kritérií, ale i na hodnocení ostatních, již dříve umístěných kritérií. Je možné postupovat i tak, že v prvním kroku provedeme jakýsi odhad bodového hodnocení kritérií, který potom ještě jednou posoudíme a případné nesrovnalosti odstraníme.

Další možností, jak k bodovému hodnocení přistupovat, je postup, kdy přiřazujeme kritériím bodové hodnocení po indexech s tím, že máme stanovený pouze řád bodů pro hodnocení důležitosti prvního kritéria. Každému dalšímu kritériu přiřazujeme bodové hodnocení opět podle hodnot přidělených předchozím kritériím. Skutečný rozsah stupnice bude tedy znám až po hodnocení posledního kritéria v množině všech kritérií.

3.4.2 Saatyho metoda kvantitativního párového porovnání

Tato metoda slouží k určení vah kritérií pomocí expertního hodnocení. V níže uvedené formě lze tuto metodu požít, pokud hodnocení provádí jediný expert.

Jde o metodu kvantitativního párového porovnávání kritérií. Pro ohodnocení párových porovnání kritérií se používá 9-ti bodové stupnice a je možné používat i mezistupně (hodnoty 2, 4, 6, 8):

1 - rovnocenná kritéria i a *j*

3 - slabě preferované kritérium i před *j*

5 - silně preferované kritérium i před *j*

7 - velmi silně preferované kritérium i před *j*

9 - absolutně preferované kritérium i před *j*

Expert porovná každou dvojici kritérií a velikosti preferencí *i*-tého kritéria vzhledem k *j*-tému kritériu zapíše do Saatyho matice $S = (s_{ij})$:,

$$S = \begin{pmatrix} 1 & s_{12} & \dots & s_{1n} \\ 1/s_{12} & 1 & \dots & s_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ 1/s_{1n} & 1/s_{2n} & \dots & 1 \end{pmatrix} \qquad (14)$$

jsou-li *i*-té a *j*-té kritérium rovnocenná, je $s_{ij} = 1$, preferuje-li slabě *i*-té kritérium před *j*-tým, je $s_{ij} = 3$, preferuje-li silně *i*-té kritérium před *j*-tým, je $s_{ij} = 5$, při velmi silné preferenci *i*-tého kritéria je $s_{ij} = 7$, při preferenci absolutní dokonce $s_{ij} = 9$.

Je-li preferováno *j*-té kritérium před *i*-tým, zapíší se do Saatyho matice převrácené hodnoty ($s_{ij} = {}^1/_3$ při slabé preferenci, $s_{ij} = {}^1/_5$ při silné preferenci atd.).

Z toho již vyplývají základní vlastnosti Saatyho matice. Jedná se o matici čtvercovou velikosti $n \times n$ a reciproční, tj. platí, že $s_{ij} = {}^1/_{s_{ji}}$. Prvky matice vlastně vyjadřují odhad podílů vah *i*-tého a *j*-tého kritéria. Na diagonále Saatyho matice jsou proto vždy hodnoty jedna (každé kritérium je samo sobě rovnocenné).

Saaty proto navrhl několik početně velmi jednoduchých způsobů, pomocí kterých lze odhadnout váhy v_j. Nejčastěji se používá postup výpočtu vah jako normalizovaného geometrického průměru **řádků** Saatyho matice, postup se někdy označuje termínem "metoda logaritmických nejmenších čtverců". Vypočteme hodnoty b_i jako geometrický průměr řádků Saatyho matice

$$b_i = \sqrt[n]{\prod_{j=1}^{n} s_{ij}} \qquad (15)$$

Váhy se pak vypočtou normalizací hodnot b_i

$$v_i = \frac{b_i}{\sum_{i=1}^{n} b_i}, kde\, i = 1, 2, ..., n \qquad (16)$$

4 Popis aplikačního rozhraní *VNNK Interface* – popis struktur a metod

Aplikační rozhraní VNNK Interface se skládá ze tří vrstev:

I. aplikační vrstva,

II. grafická vrstva,

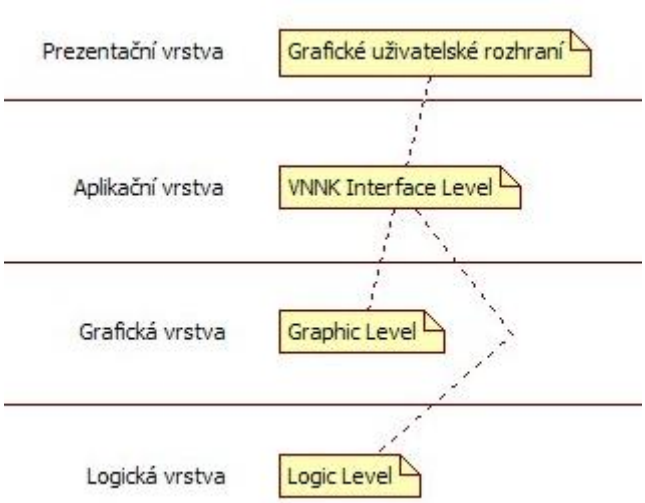
III. logická vrstva.

Každá vrstva je implementována v programovacím prostředí Turbo Delphi, jako programová jednotka (Unit). Každá tato jednotka se skládá z tříd a ty zase z vlastností a metod. Následující kapitoly se budou věnovat popisu a charakteristice těchto vrstev.

4.1 Diagram tříd

Diagram tříd je umístěn na přiloženém CD viz soubor *DiagramTrid.jpg*.

4.2 Architektura



Obrázek 7 Čtyřvrstvá architektura

4.3 Prezentační vrstva grafického uživatelského rozhraní

Prezentační vrstva grafického uživatelského rozhraní je již grafická podoba aplikace tak jak ji vidíme při spuštění aplikace VNNK. Tato vrstva využívá tříd, jejich vlastností a metod v aplikační vrstvě *VNNK Interface Level*, logické vrstvě *Logic Level* a grafické vrstvě *Graphic Level*. Uživatelský popis aplikace je dále uveden v kapitole 5.

4.4 Aplikační vrstva VNNK Interface Level

4.4.1 Třída KriterialniSada

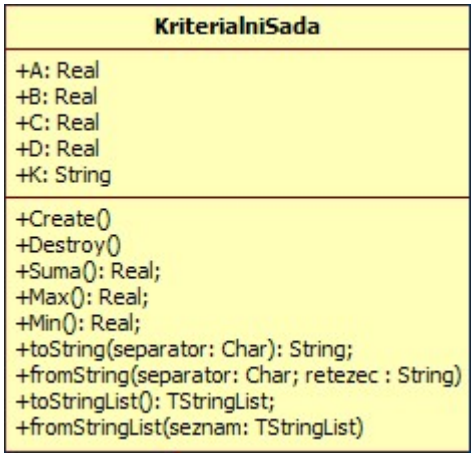
Jak bylo již zmíněno v kapitole 2.2 při určování náhradní nástupištní koleje je zapotřebí stanovit čtyři kritéria (A, B, C, D). Pro tento typ dat je zde k dispozici třída *KriterialniSada*, která disponuje čtyřmi vlastnostmi (A, B, C, D) typu *Real* a jednou vlastností typu *String*, která je výhradně určena pro uchování identifikátoru koleje, ke které se daná čtyři kritéria vztahují.

Matematické metody

Při začlenění této třídy do algoritmů pro vyhledávání nástupištní koleje se předpokládá časté hledání minima, maxima, popřípadě součtu všech vlastností. Pro tyto operace jsou zde metody *Max*, *Min* a *Suma*.

Konverzní metody

Jelikož je tento model navržen pro nasazení do prostředí, ve kterém bude zobrazován v různých uživatelských formulářích atp. je doplněn o konverzní metody, které převedou kriteriální sadu na řetězec jako např. metoda *toString* přičemž jednotlivé vlastnosti budou odděleny zadaným oddělovačem *separator*. Následující metoda *fromString* je inverzní k předešlé. Další metoda *toStringList* a inverzní *fromStringList* pracují s datovou strukturou *TStringList*, do které vloží zkonvertované vlastnosti A až D jako řetězce. Popřípadě je načtou a převedou na datový typ *Real*.



Obrázek 8 Model třídy KriterialniSada

4.4.2 Třída PripojnyVlak

Po prostudování čekacích dob a opatřeních při zpoždění vlaků osobní dopravy viz literatura [5] byl navržen model přípojného vlaku se čtyřmi klíčovými vlastnostmi. První vlastností je celočíselný identifikátor vlaku *IDvlaku*. Následující tři vlastnosti popisují

základní časové charakteristiky v celých minutách. Mezi tyto vlastnosti patří normální doba potřebná na přestup *DobaPotrebnaNaPrestupNormalni*. Tato doba je uplatněna pokud přípojný vlak není u stejného nástupiště jako vlak, od kterého má být realizován přípoj. Další vlastností je kratší doba potřebná na přestup *DobaPotrebnaNaPrestupKratsi*. Tato doba je uplatněna pokud přípojný vlak je u stejného nástupiště jako vlak od kterého má být realizován přípoj. Poslední vlastností je čekací doba *DobaCekaci*. Čekací doba je maximální doba, po kterou přípojný vlak může čekat na zpožděný vlak od kterého má být realizován přípoj.

Konverzní metody

Třída disponuje konverzními metodami pro případný výpis do formuláře textového souboru či grafické komponenty. Mezi tyto metody patří převod na seznam řetězců *toStringList* a převod na řetězec, kde jednotlivé vlastnosti jsou odděleny zadaným oddělovačem *separator*.

Obrázek 9 Model třídy PripojnyVlak

4.4.3 Třída RozsirujiciVlastnostiVlaku

Tato třída rozšiřuje vlastnosti vlaku o celočíselnou informaci počtu vozů *PocetVozu* který je potřeba pro výpočet přípustných kolejí viz kapitola 2.2. Navíc musí být doplněna o seznam přípojných vlaků *PripojneVlaky*, ke kterým se realizuje přípoj. Samozřejmě zde musí být uveden i identifikátor vlaku *IDvlaku* kterého se tyto rozšiřující vlastnosti týkají.

Třída je dále rozšířena o obyčejný a přetížený kopírovací konstruktor, který vytvoří kopii jak samotného objektu, tak i všech agregovaných přípojných vlaků v seznamu *PripojneVlaky*.

Dalšími metodami, kterými tato třída disponuje, je metoda *Count* která vrací počet agregovaných přípojných vlaků. Metodou pro odebrání ze seznamu přípojných vlaků je metoda *Delete*, která projde celý seznam a smaže příslušný přípojný vlak zadaný parametrem *IdVlaku*. Metoda *toString* jak již název napovídá převede strukturu na řetězec.

Zde je struktura oddělovačů stanovena implicitně. Pro snazší kopírování objektů je zde k dispozici metoda *copyTo*, která instanci objektu zadaného v parametru naplní totožnými daty bez nutnosti destrukce objektu a použití kopírovacího konstruktoru.

Obrázek 10 Model třídy RozsirujiciVlastnostiVlaku

4.4.4 Třída VlakyApripojneVlaky

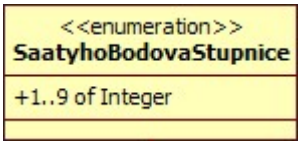
Tato třída má za úkol agregovat objekty třídy *RozsirujiciVlastnostiVlaku* viz kapitola 4.4.3.

Dále disponuje metodou *Add* pro přidání takového objektu, metodou *Delete* s parametrem *IdVlaku* pro odebrání a metodou *najdiVlak* pro vyhledání v seznamu rozšiřujících vlaků.

Obrázek 11 Model třídy VlakyApripojneVlaky

4.4.5 Výčtový typ SaatyhoBodovaStupice

Pro potřebu realizace Saatyho bodové stupnice viz kapitola 3.4.2 je navržen výčtový typ který nabývá celočíselných hodnot z intervalu $\langle 1,...,9 \rangle$.



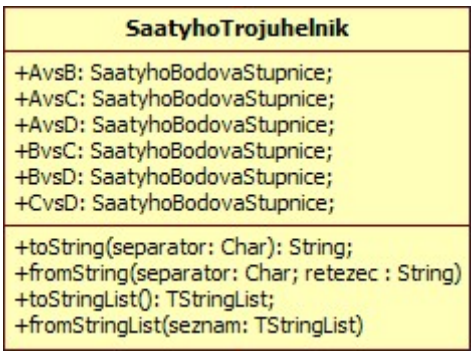
Obrázek 12 Model výčtového typu SaatyhoBodovaStupice

4.4.6 Třída SaatyhoTrojuhelnik

Pro namodelování Saatyho matice tak jak je popsána v kapitole 3.4.2 je třeba znát pouze horní polovinu matice nad diagonálou a ostatní hodnoty se již dají z těchto hodnot dopočítat. Proto je název této třídy zvolen jako Saatyho trojúhelník.

Třída disponuje šesti vlastnostmi, které charakterizují párové porovnání jednotlivých kritérií A až D viz kapitola 2.2. Například první vlastnost pod názvem *AvsB* jak již z názvu je patrné porovnává vlastnost A s B.

Dále jsou zde k dispozici čtyři konverzní metody, které mají stejnou strukturu a význam jak již bylo popsáno u třídy *KriterialniSada* viz kapitola 4.4.1



Obrázek 13 Model třídy SaatyhoTrojuhelnik

4.4.7 Třída SaatyhoMatice

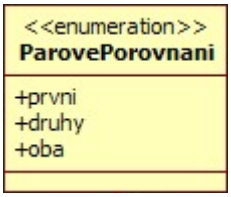
Saatyho matice je dvourozměrná matice o velikosti N, kde N je počet kritérií které se v ní porovnávají. V našem případě pracujeme se čtyřmi kritérii a proto je vlastnost *matice* dvourozměrná čtvercová matice o velikosti 4x4. Dále třída obsahuje soukromý atribut *pSaatyhoTrojuhelnik*, který obsahuje jednu instanci třídy *SaatyhoTrojuhelnik*.

Metoda pro vložení konkrétní instance třídy *SaatyhoTrojuhelnik* a naplnění vlastnosti matice z tohoto trojúhelníku je metoda *vlozDataZeSaatyhoTrojuhelniku.* Pro výpočet vektoru vah viz vzorec (16) je zde metoda *vektorVah* a zbývající čtyři konverzní metody mají stejnou strukturu a význam jak tomu již bylo u třídy *KriterialniSada* viz kapitola 4.4.1 či třídy *SaatyhoTrojuhelnik* viz kapitola 4.4.6.

Obrázek 14 Model třídy SaatyhoMatice

4.4.8 Výčtový typ ParovePorovnani

Při konstrukci Fullerova trojúhelníku se setkáme s tzv. kroužkovací metodou, kde porovnáváme dvě kriteria takovým způsobem abychom preferovali kriterium první, druhé, nebo stejnou vahou obě kritéria. Pro potřebu realizace párového porovnávání u Fullerova trojúhelníku viz kapitola 3.3.2 je navržen výčtový typ, který může nabývat těchto tří hodnot.



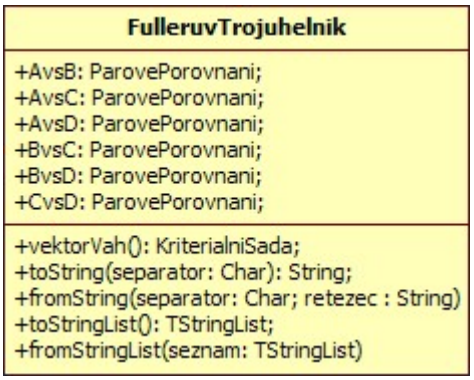
Obrázek 15 Model výčtového typu ParovePorovnani

4.4.9 Třída FulleruvTrojuhelnik

Tato třída je již realizací zmíněného Fullerova trojúhelníku z kapitoly 3.3.2 a využívá k zachycení vztahů mezi kritérii třídu *ParovePorovnani* z předešlé kapitoly. Například první vlastnost *AvsB* porovnává kritérium A s B.

Pro výpočet vektoru vah je k dispozici metoda *vektorVah* která pro návratovou hodnotu používá třídu *KriterialniSada*. Tato sada je vektorem čtyř reálných hodnot tzv. vah které se vztahují k daným čtyřem kriteriím. Metoda výpočtu viz vzorec (12).

Zbývající čtyři konverzní metody, mají stejnou strukturu a význam jak již bylo popsáno u třídy *KriterialniSada* viz kapitola 4.4.1



Obrázek 16 Model třídy FulleruvTrojuhelnik

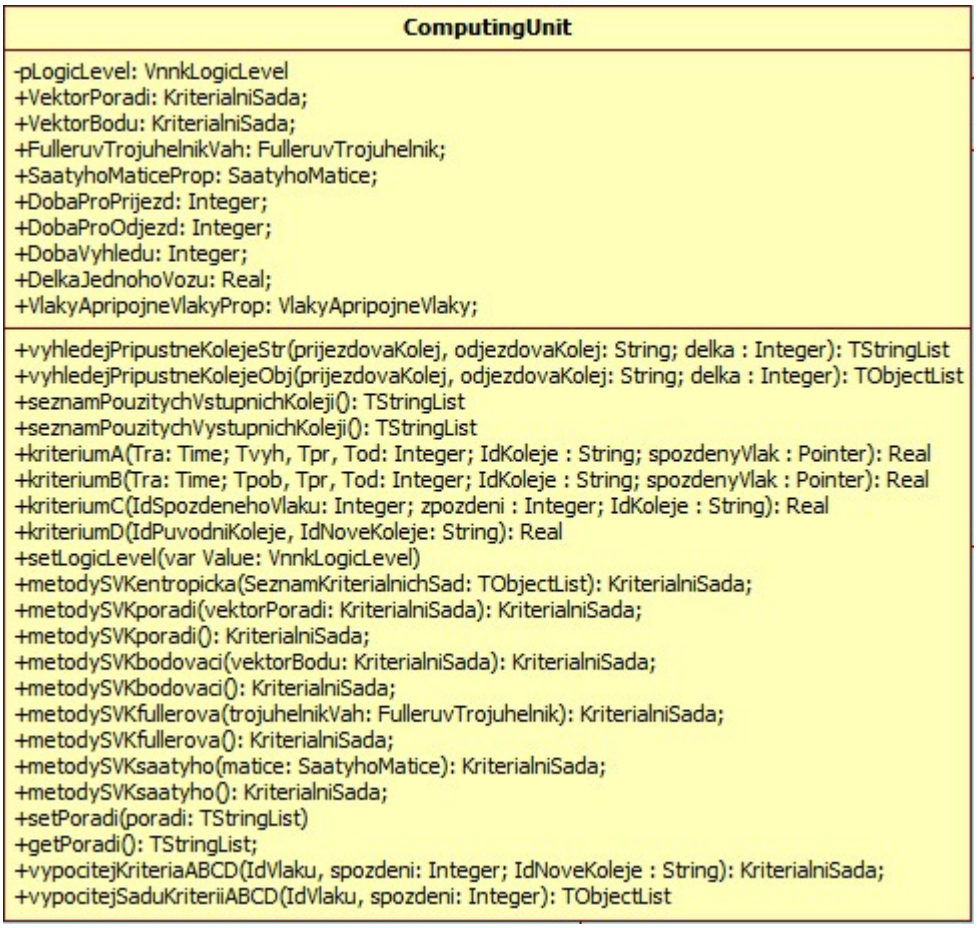
4.4.10 Třída ComputingUnit

Tato třída se zabývá vyhledáváním přípustných kolejí tak, jak je to popsáno v kapitole 2. Disponuje metodami, které počítají jednotlivá kritéria A až D (viz kapitola 2.2), dále implementuje metody pro stanovení vah kritériím viz kapitola 3.

K vlastnostem této třídy patří vektor pořadí *VektorPoradi*, který je vstupním parametrem pro stanovení vah kritérií metody pořadí (viz kapitola 3.3.1). Další metoda pro stanovení vah kritérií je metoda bodovací, která má zde k dispozici potřebný vektor bodů *VektorBodu*. Zbývající dvě metody stanovení vah kritérií jsou Fullerova a Saatyho metoda. Obě musí mít přednastavenou množinu vah (viz kapitola 3.3), které zde vystupují jako veřejné vlastnosti *FulleruvTrojuhelnikVah* a *SaatyhoMaticeProp*.

Každý vlak přijíždějící do železniční stanice potřebuje mít pro příjezd a odjezd vyhrazenou časovou rezervu. Tato časová rezerva je pro všechny vlaky stejná a je zde k dispozici vlastnost *DobaProPrijezd* resp. *DobaProOdjezd*. Pro stanovení kritéria A je zapotřebí stanovit dobu výhledu, která je zde realizována vlastností *DobaVyhledu*. Abychom mohli stanovit délku jednoho vozu a tím případně vypočítat délku soupravy je zde k dispozici vlastnost *DelkaJednohoVozu*. Tato vlastnost používá měrnou jednotku [1 m]. V kapitole 4.4.4 je popsaný model třídy *VlakyApripojneVlaky*. Zde je již tato třída použita jako vlastnost *VlakyApripojneVlakyProp*.

Metody, kterými třída *ComputingUnit* disponuje jsou vyhledání přípustných kolejí *VyhledejPripustneKoleje*, která je zde k dispozici ve dvou verzích rozlišených koncovkou *Str* a *Obj*. Metoda s koncovkou *Str* vyhledá seznam přípustných kolejí a vrátí ho jako seznam řetězců *TStringList*, kdežto metoda s koncovkou Obj vrátí objektový seznam přípustných kolejí. Pro získání seznamu použitých vstupních a výstupních kolejí jsou zde metody *seznamPouzitychVstupnichKoleji* a *seznamPouzitychVystupnichKoleji*. Pro výpočet jednotlivých kritérií A až D jsou zde k dispozici metody *kriteriumA* až *kriteriumD*. Další metody se zabývají stanovením vah kritériím či lépe řečeno výpočtem vektoru vah. Mezi tyto metody patří *metodySVKentropicka*, *metodySVKporadi*, *metodySVKbodovaci*, *metodySVKfullerova* a *metodySVKsaatyho*. Metody jsou vždy až na entropickou ve dvou přetížených verzích. Metody bez parametru používají pro výpočet příslušnou vlastnost, kdežto metody s parametrem používají k výpočtu zadaný parametr. Poslední dvě metody automatizují výpočet kritérií A až D v jediné metodě *vypocitejKriteriaABCD*. Tato metoda vyžaduje v parametru zadat výstupní kolej a tak je tu k dispozici ještě poslední metoda *vypocitejSaduKriteriiABCD*, která automatizuje výpočet v maximální možné míře a to tím, že jako vstupní parametr vyžaduje pouze číslo vlaku a míru zpoždění v celých minutách. V návratové hodnotě již předá seznam všech sad kritérií A až D ke všem přípustným kolejím.



Obrázek 17 Model třídy ComputingUnit

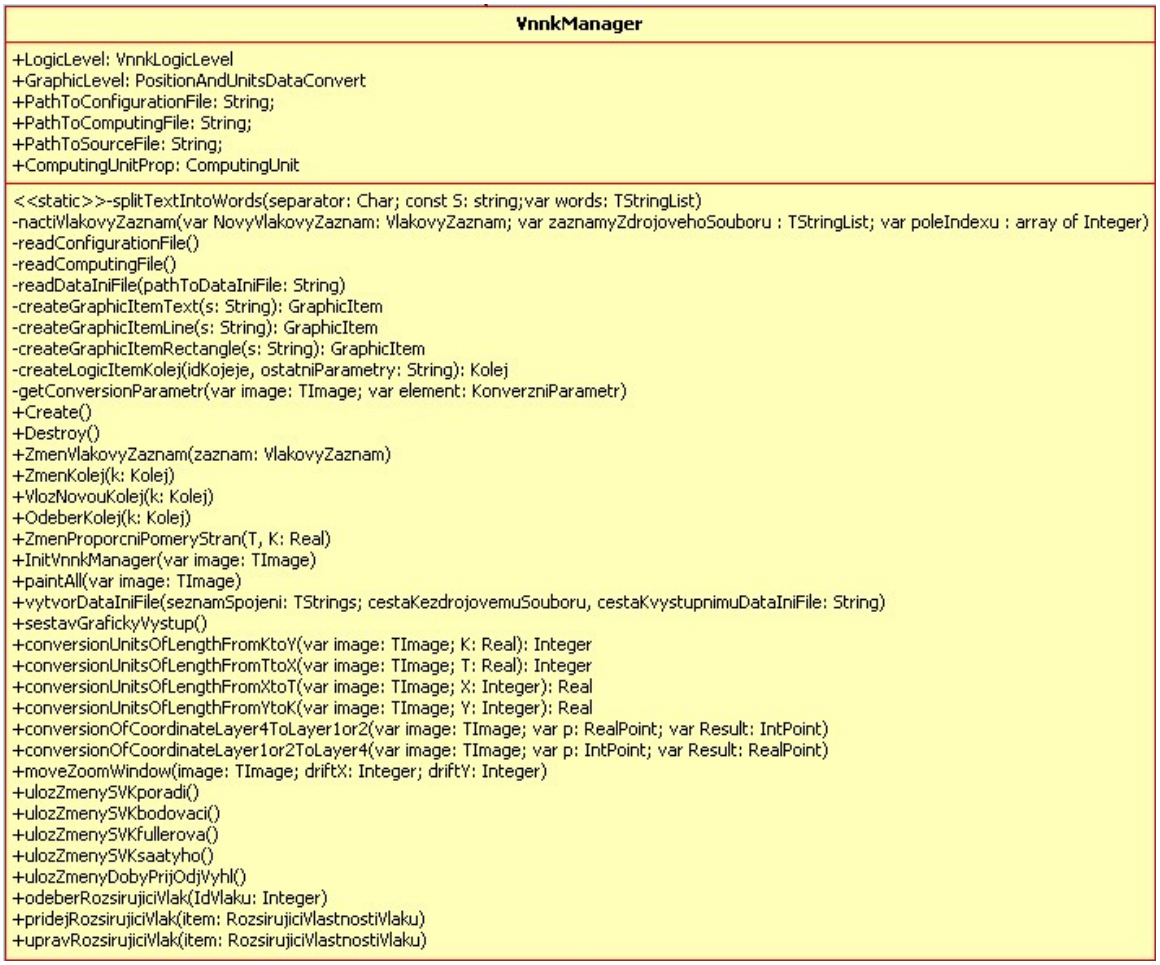
4.4.11 Třída VnnkManager

Třída VnnkManager je hlavní třídou aplikačního rozhraní VNNK Interface Level. Agreguje do sebe logickou i grafickou úroveň, umí pracovat s konfiguračními soubory a ukládat do nich případné změny. Pracuje s grafickým výstupním zařízením a usnadňuje celkovou práci s ostatními třídami všech úrovní.

Třída disponuje vlastností *LogicLevel* pro práci s logickou vrstvou aplikačního rozhraní. Dále má k dispozici vlastnost *GraphicLevel* pro práci s grafickou vrstvou a vlastnost *ComputingUnitProp* pro práci s výpočty náhradních nástupištních kolejí. Tato třída pracuje s konfiguračním souborem, kde informace o cestě k tomuto souboru slouží vlastnost *PathToConfigurationFile*. Pro potřeby výpočtů náhradních nástupištích kolejí slouží konfigurační soubor *PathToComputingFile*. Cesta ke zdrojovému souboru s vlakovými záznamy zastupuje vlastnost *PathToSourceFile*.

Jako zdrojový či konfigurační soubor je použit textový formát *ini*, tudíž se musí číselné hodnoty převést na řetězec a zpět. V mnoha případech se nejedná o osamocenou hodnotu ale o skupinu uspořádaných hodnot. Aby bylo možné tuto skupinu hodnot načíst je zde

k dispozici statická metoda *splitTextIntoWords*, která rozdělí řetězec na seznam dílčích řetězců. Tyto dílčí řetězce je již možné převést na příslušné číselné hodnoty pomocí předdefinovaných funkcí prostředí Turbo Delphi. Pro načtení vlakových záznamů je k dispozici metoda *nactiVlakovyZaznam*. Konfigurační, početní a zdrojový soubor je načítán metodami *readConfigurationFile*, *readComputingFile* a *readDataIniFile*. Pro snadné vytváření instancí grafických objektů typu text, čára a obdélník jsou zde metody *createGraphicItemText*, *createGraphicItemLine* a *createGraphicItemRectangle*. Tvorba logických objektů typu *Kolej* je zabezpečena metodou *createLogicItemKolej*. Konverzní parametr (viz kapitola 4.6.4) je vytvářen metodou *getConversionParametr*. Pokud provedeme změnu vlakového záznamu, kterou je třeba zaznamenat do zdrojového souboru je tu k dispozici metoda *ZmenVlakovyZaznam*. Podobnou funkci poskytují i metody *ZmenKolej* a *ZmenProporcniPomeryStran*. Pro přidání resp. odebrání koleje z konfiguračního souboru je tu metoda *VlozNovouKolej* resp. *OdeberKolej*. Inicializaci provedeme spuštěním funkce *InitVnnkManager* se vstupním parametrem *image* typu *TImage*. Pro vykreslení všech grafických elementů je zde metoda *paintAll*. Aby mohlo aplikační rozhraní pracovat s různými typy zdrojových souborů je zde metoda *vytvorDataIniFile*, která takový zdrojový soubor vytvoří. Pro kompletní sestavení grafického výstupu je zde metoda *sestavGrafickyVystup*. Následujících šest konverzních metod je pouze delegováno ze třídy *PositionAndUnitsDataConvert* viz kapitola 4.6.11. Snadné posunutí grafického okna čtvrté virtuální vrstvy zabezpečuje veřejná metoda *moveZoomWindow*. Pro uložení změn v parametrech pro stanovení vah kritérií a dob potřebných pro příjezd, odjezd či doby výhledu jsou k dispozici metody *ulozZmenySVKporadi/bodovaci/fullerova/saatyho* a *ulozZmenyDobyPrijOdjVyhl*. Pro manipulaci s rozšiřujícími vlastnostmi vlaků jsou k dispozici metody pro přidávání *pridejRozsirujiciVlak*, odebírání *odeberRozsirujiciVlak* a úpravy *upravRozsirujiciVlak*.



Obrázek 18 Model třídy VnnkManager

4.5 Logická vrstva „Logic Level“

Tato vrstva definuje třídy pro práci s časem, časovými omezeními, časovou řadou, vlakovými záznamy a kolejemi které organizuje a řídí centrální třída *VnnkLogicLevel*.

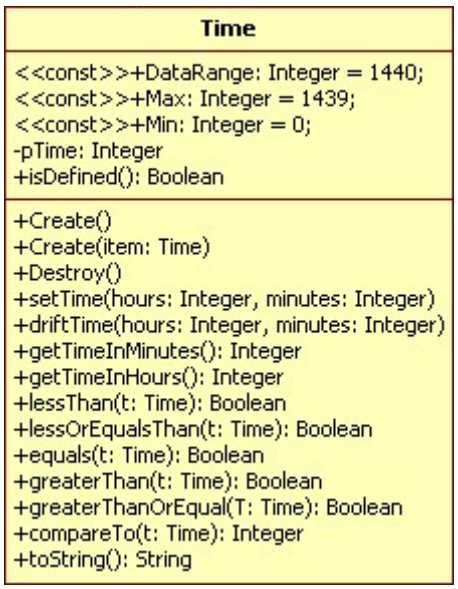
4.5.1 Třída Time

Pro definici času na železnici je zde třída, která pracuje s časem v celých minutách. Maximální, minimální a datový rozsah času je definován celočíselnými konstantami *Max*, *Min* a *DataRange*. Datový rozsah času vychází z počtu minut obsažených v jednom dni tj. 1440. Minimální hodnota je rovna nule, a tedy maximum je rovno 1349. Hodnota času se ukládá do soukromého celočíselného atributu *pTime*. Hodnota času může nabývat ještě nedefinované hodnoty kterou zastupuje logická vlastnost *isDefined*.

Pro nastavení hodnoty času je k dispozici metoda *setTime*. Pokud potřebujeme posunout čas o zadaný počet hodin a minut můžeme jednoduše použít funkci *driftTime*. Pro zjištění hodnoty času převedenou na minuty je k dispozici funkce *getTimeInMinutes*. Pokud nás

minuty nezajímají ale potřebujeme znát počet celých hodin, pak je zde funkce *getTimeInHours*, která odstraní přebytečné minuty a vrátí hodnotu v celých hodinách.

Pro potřebu logického porovnávání jednotlivých instancí třídy *Time* jsou tu k dispozici funkce menší než (*lessThan*), menší nebo rovno (*lessOrEqualsThan*), rovno (*equals*), větší než (*greaterThan*), větší nebo rovno (*greaterThanOrEqual*). Poslední porovnávací metodou je zde funkce *compareTo*, která pokud hodnota času vstupního parametru *t* je rovna vrací hodnotu nula, pokud je větší vrací hodnotu větší než nula a pokud je menší vrací hodnotu menší než nula. Poslední metodou třídy *Time* je funkce *toString*, která převede celočíselnou hodnotu času na řetězec.



Obrázek 19 Model třídy Time

4.5.2 Třída TimeDefinition

Při analýze obsazení dopravních kolejí v železniční stanici nastala otázka jak jednotlivým vlakovým záznamům přiřadit časové omezení (např. jede jen v pracovní dny). Otázku řeší tato třída, která je složena z konstant definujících jednotlivá omezení. Konstanty jsou reprezentovány celočíselnými hodnotami od jedné do devíti. V následující tabulce 1 je seznam předdefinovaných konstant.

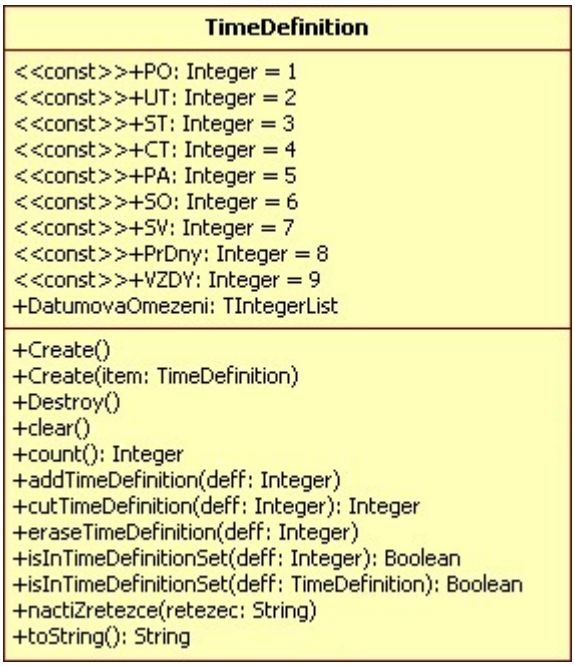
Tabulka 1 Seznam předefinovaných konstant

Název	Celočíselná hodnota	Význam
PO	1	Vlaky které jedou pouze v pondělí.
UT	2	Vlaky které jedou pouze v úterý.
ST	3	Vlaky které jedou pouze ve středu.
CT	4	Vlaky které jedou pouze ve čtvrtek.
PA	5	Vlaky které jedou pouze v pátek.
SO	6	Vlaky které jedou pouze v sobotu.
SV	7	Vlaky které jedou pouze v neděli a státem uznaný svátek.
PrDny	8	Vlaky které jedou pouze v pracovních dnech.
VZDY	9	Vlaky které jedou vždy.

Pro uchování celočíselných hodnot je tu vlastnost *DatumovaOmezeni*, která je seznamem celočíselných hodnot typu *TIntegerList*.

Pro vyprázdnění seznamu omezení je zde k dispozici metoda *clear*. Počet datumových omezení jde snadno zjistit pomocí metody *count*. Přidávat, vyjímat a mazat datumová omezení je možné metodami *addTimeDefinition*, *cutTimeDefinition* a *eraseTimeDefinition*. Pro zjišťování zda objekt obsahuje příslušné datumové omezení zadané parametrem deff je zde metoda *isInTimeDeffinitionSet* s parametrem typu *Integer*. Pokud porovnáváme mezi sebou jednotlivé instance této třídy a zjišťujeme zda obsahují alespoň jedno společné omezení můžeme použít přetíženou metodu *isInTimeDeffinitionSet* s parametrem typu *TimeDefinition*. Pokud je snaha o převedení objektu na řetězec, pak použijeme metodu *toString*[1] a naopak z řetězce metodu *nactiZretezce*.

[1] Při převodu datumového omezení na řetězec se implicitně použije jako oddělovač znak roura „|“.



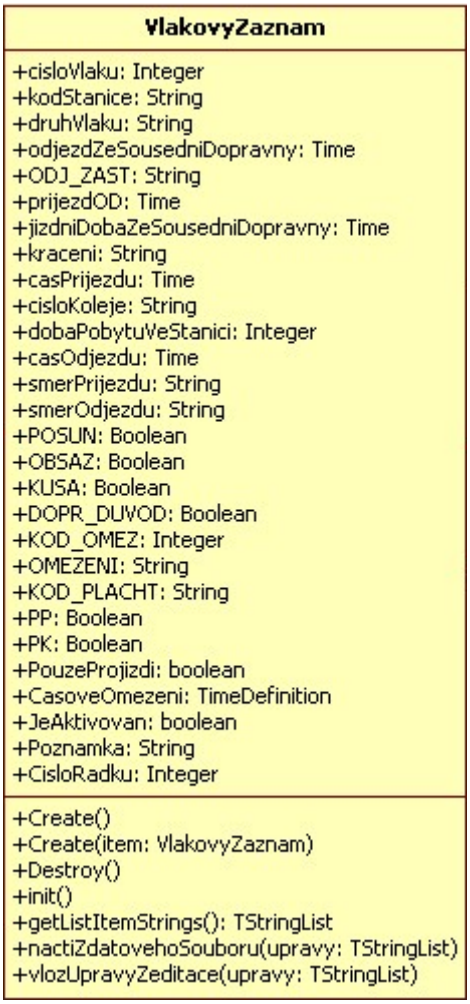
Obrázek 20 Model třídy TimeDefinition

4.5.3 Třída VlakovyZaznam

Vlakový záznam jako množina atributů patří mezi nejrozsáhlejší třídy této logické vrstvy. Obsahuje 28 různých vlastností. Vlastnosti jsou z větší části odvozeny od struktury datového vstupního souboru *vlaky.csv*, který je uložen na přiloženém CD. Tato třída obsahuje mnoho vlastností, kterými lze popsat vlakový záznam. Význam jednotlivých vlastností lze intuitivně odvodit od samotného názvu, leč pro zpřesnění nyní uveďme jejich význam. Vlastnost pro jednoznačné identifikování vlaku *cisloVlaku*, je celočíselná celá hodnota. Další vlastností *kodStanice* je kód stanice do které má daný vlak plánovaný příjezd/odjezd. Následující uvedené příklady užitých vlastností jsou vybrány ze zdrojového souboru *vlaky.csv*. Kód uzlové železniční stanice Praha hlavní nádraží je například *Ph*. Pro rozlišení různých druhů vlaku, jako je například osobní, spěšný či rychlík (Os, Sp či R) je k dispozici vlastnost *druhVlaku*. Další vlastnost *odjezdZeSousedniDopravny* charakterizuje čas odjezdu vlaku ze sousední dopravny. Jízdní dobu ze sousední dopravny popisuje vlastnost typu *Time jizdniDobaZeSousedniDopravny*. Dále pokud dochází ke krácení jízdní doby můžeme tuto informaci zachytit do vlastnosti *kraceni*. Každý vlak, který má plánovaný čas příjezdu do stanice musí obsahovat tomu odpovídající vlastnost *casPrijezdu*. Na železnici je možné, že tato vlastnost není uvedena a má tedy *null* hodnotu. Toto je možné v případech, kdy je po příjezdu do stanice vlak/souprava přečíslována a tak obsahuje pouze informaci o čase odjezdu. Nutno dodat, že vlakový záznam s původním číslem analogicky neobsahuje informaci o čase odjezdu vlaku/soupravy (vlastnost

casOdjezdu je tedy *null*). Plánovanou dobu pobytu ve stanici popisuje vlastnost *dobaPobytuVeStanici*. Tato doba pobytu je uvažována v celých minutách. Každý vlakový záznam by měl obsahovat platnou hodnotu vlastnosti číslo koleje *cisloKoleje*. Z názvu je patrné že jde o číslo, leč kolej označená např. *13a* již číslem v žádném případě není. Proto je datový typ této vlastnosti řetězec *String*. Následující dvě popisované vlastnosti směr příjezdu *smerPrijezdu* a směr odjezdu *smerOdjezdu* charakterizují ze kterého směru přijeli a kterým směrem odjedou. Jelikož není prakticky možné mít velkou železniční stanici ve které je možné se ze všech příjezdových/odjezdových směrů dá přijet/odjet na kteroukoliv kolej. Všechny koleje ve stanici obsahují množinu možných směrů pro příjezd/odjezd. Z těchto kolejí se pomocí množinových operací vyberou jen ty, které umožní vlaku přijet/odjet ze/do stejného směru. Nyní si popišme vlastnosti, které nejsou uvedeny ve zdrojovém souboru *vlaky.csv*, leč jsou důležité pro další zpracování. Pokud vlak pouze projíždí je při vykreslování na grafický výstup zpracováván přednostně. Takto by byl označen vlak s parametrem *true* ve vlastnosti *PouzeProjizdi*. Další vlastností je časové omezení *CasoveOmezeni*, které je již popsáno v kapitole 4.5.2. Pokud se z nějakého důvodu rozhodnete deaktivovat vlakový záznam můžete takto učinit nastavením vlastnosti *JeAktivovan* na hodnotu *false*. Pro přidání textové poznámky je zde vlastnost *Poznamka*. Při tvorbě datového souboru (**.dataIni*) ze zdrojového (**.csv*) se ukládá do vlastnosti *CisloRadku* číslo řádku. Tato informace je pro možnost zpětné identifikace vlakového záznamu ve zdrojovém souboru.

Mezi metody této třídy patří inicializační *init*. Inicializuje všechny vlastnosti na implicitní hodnoty. Pro převod na posloupnost řetězců je zde metoda *getListItemStrings*. Pro načítání dat z datového souboru je tu metoda *nactiZdatovehoSouboru*. Pro vkládání úprav je k dispozici metoda *vlozUpravyZeditace*.



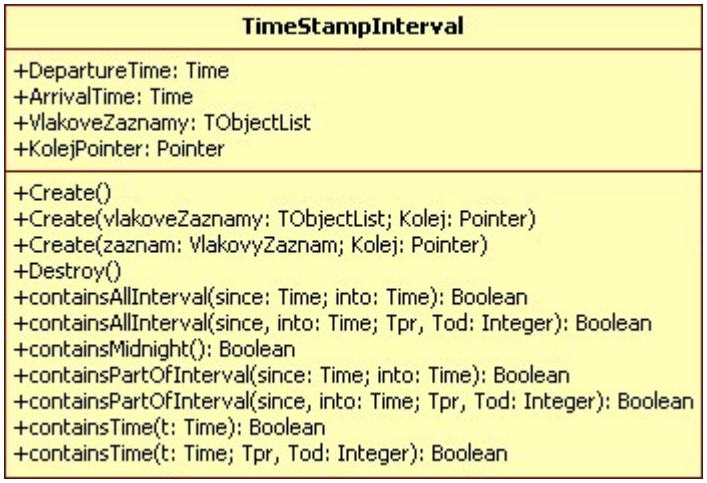
Obrázek 21 Model třídy VlakovyZaznam

4.5.4 Třída TimeStampInterval

Pro reprezentaci časového intervalu, po který je železniční kolej obsazena je zde k dispozici třída, která ohraničuje tento interval dvěma časovými okamžiky, a to časem příjezdu *ArrivalTime* a časem odjezdu *DepartureTime*. Dále je zde k dispozici seznam vlakových záznamů *VlakoveZaznamy*, které přísluší k danému intervalu. Poslední vlastností je *KolejPointer*, což je ukazatel na kolej, které interval přísluší.

Mezi základní metody této třídy patří tři přetížené konstruktory a virtuální destruktor. Pokud vznikne potřeba pro zjištění zda interval je podmnožinou jiného intervalu, pak je zde k dispozici metoda *containsAllInterval*, která je ve dvou přetížených variantách. První varianta metody pracuje se vstupními parametry intervalu zadanými dvěma časovými okamžiky od (*since*) do (*into*). Druhá varianta rozšiřuje porovnávaný interval o čas pro příjezd (*Tpr*) a odjezd (*Tod*). Další dvě přetížené metody *containsPartOfInterval* jsou velice podobné těm předcházejícím jen s tím rozdílem, že vracejí hodnotu *true* už když se intervaly překrývají byť jen částečně. Pro potřebu zjištění zda interval obsahuje půlnoc je

tu metoda *containsMidnight*. Poslední dvě přetížené metody *containsTime* zjišťují, zda interval obsahuje časový okamžik zadaný parametrem *t*, přičemž druhá metoda rozšiřuje interval o dobu pro příjezd (*Tpr*) a dobu potřebnou pro odjezd (*Tod*).

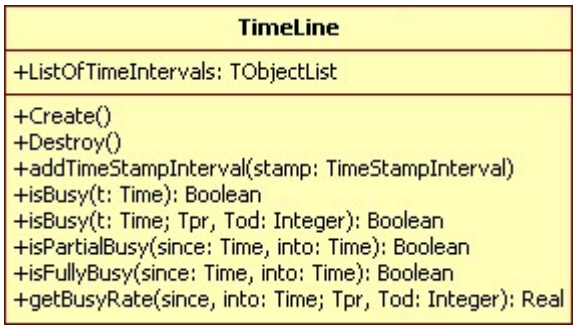


Obrázek 22 Model třídy TimeStampInterval

4.5.5 Třída TimeLine

Pro potřebu seřazení množiny časových intervalů viz kapitola 4.5.4 do časové osy je tu třída *TimeLine*. Tato třída uchovává jednotlivé instance třídy *TimeStampInterval* v objektovém seznamu, který prezentuje jako vlastnost pod názvem *ListOfTimeIntervals*.

Třída disponuje metodou pro přidávání intervalů *addTimeStampInterval*. Následující čtyři metody slouží pro posuzování, zda v časovém okamžiku či intervalu je časová řada obsazena s přihlédnutím na normální či rozšířený stav intervalů o dobu pro příjezd (*Tpr*) a dobu potřebnou pro odjezd (*Tod*). Poslední metoda *getBusyRate* vrací hodnotu z intervalu $\langle 0, 1 \rangle$ a vyjadřuje poměr zaneprázdnění časovými intervaly na časové řadě v zadaném intervalu od (*since*) – do (*into*) s přihlédnutím na dobu potřebnou pro příjezd (*Tpr*) a dobu potřebnou pro odjezd (*Tod*).



Obrázek 23 Model třídy TimeLine

4.5.6 Třída MnozinaRetezcu

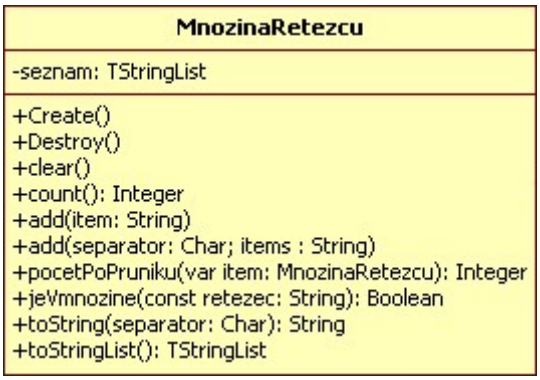
Tato třída má za úkol reprezentovat seznam řetězců a umožnit s tímto seznamem pracovat jako s množinou.

Mezi základní metody patří získání velikosti množiny metoda count. Pro přidání jednoho prvku do množiny slouží metoda *add* s parametrem *item* typu *String*. Tato metoda *add* je zde ještě v přetížené verzi a slouží k hromadnému vložení prvků uložených v řetězci oddělených oddělovačem *separator*.

Disponuje základní metodou pro práci s množinami *pocetPoPruniku*, která provede průnik vlastní množiny s množinou uvedenou v parametru *item* a v návratové hodnotě poskytuje informaci o počtu prvků zbylých po průniku.

Metoda *jeVmnozine* informuje o existenci prvku zadaného parametrem *retezec* a pro existenci resp. neexistenci vrací hodnotu *true* resp. *false*.

Konverzní metoda *toString* slouží pro konverzi množiny na jeden ucelený řetězec, kde jednotlivé prvky jsou odděleny vstupním parametrem *separator*. Poslední metoda *toStringList* provede kopii soukromého atributu *seznam*.



Obrázek 24 Model třídy MnozinaRetezcu

Využití této třídy je popsáno v následující kapitole 4.5.7.

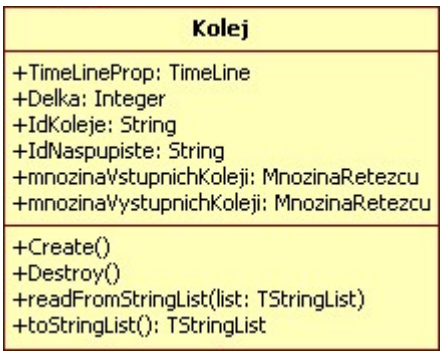
4.5.7 Třída Kolej

Touto třídou je reprezentována kolej v železniční stanici. Takováto kolej má svoji určitou délku, je charakterizována specifickým identifikátorem a může přilehat k nějakému nástupišti. Tyto uvedené vlastnosti jsou popsány třídními vlastnostmi jako *Delka*, *IdKoleje* a *IdNastupiste*.

Každá takováto kolej je dále charakteristická tím, že disonuje množinou vstupních resp. výstupních kolejí ze kterých je na tuto kolej možný příjezd resp. odjezd. Tyto vlastnosti jsou popsány třídními vlastnostmi *mnozinaVstupnichKoleji* a *mnozinaVystupnichkoleji*.

Tato kolej je navíc rozšířena o časovou osu (viz kapitola 4.5.5) na kterou můžeme umístit časové intervaly (viz kapitola 4.5.4), které charakterizují příjezd a odjezd vlaku či vlakové soupravy. Touto vlastností je *TimeLineProp*.

Třída disponuje dvěma konverzními metodami, přičemž první z nich *readFromStringList* načte vlastnosti objektu ze seznamu řetězců a druhá k ní inverzní *toStringList* převede objekt na seznam řetězců.



Obrázek 25 Model třídy Kolej

4.5.8 Třída VnnkLogicLevel

Poslední třídou logické úrovně je třída *VnnkLogicLevel*, která používá či agreguje všechny výše uvedené třídy této logické úrovně.

Mezi její vlastnosti patří seznam vlakových záznamů, seznam filtrovaných vlakových záznamů, seznam kolejí, což je objektové seznamy instancí příslušných tříd. Poslední vlastností je datumové omezení, kterým si může uživatel nastavit které vlaky chce promítnout do filtrovaných vlakových záznamů a které nikoliv. Seznam filtrovaných vlakových záznamů je tudíž podmnožinou seznamu vlakových záznamů.

První metoda *sestavGrafickyVystup* sestaví seznam kolejí podle seznamu filtrovaných vlakových záznamů, který získá za pomocí seznamu vlakových záznamů a datumových omezení. Vlakové záznamy načtené ze vstupního datového souboru byly ošetřeny identifikátorem čísla řádku a tak je tu k dispozici metoda *najdiVlakovyZaznam*, která podle tohoto parametru tyto vlakové záznamy vyhledává. Mnohem intuitivnější metoda která vyhledává podle čísla vlaku resp. jeho identifikátoru *IdVlaku* je metoda *najdiVlakovyZaznamDleID*.

Obrázek 26 Model třídy VnnkLogicLevel

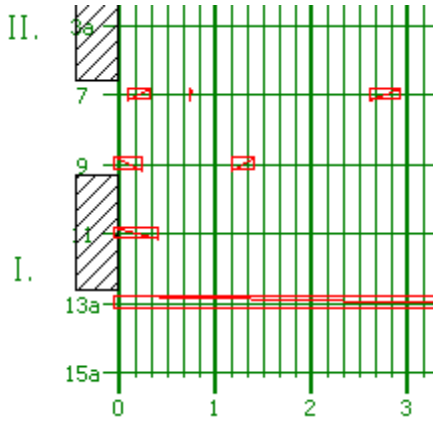
4.6 Grafická vrstva „Graphic Level“

Grafická vrstva sestává z programové jednotky pojmenované „VnnkGraphicLevelUnit“ a „VnnkGraphicConversionUnit“. V následujících kapitolách je detailní popis jednotlivých složek těchto tříd.

4.6.1 Vykreslovací strategie pomocí virtuálních vrstev

Problematika vykreslování grafických objektů na výstupní grafické zařízení, které si žádá, aby se dal výstup přibližovat, oddalovat posouvat atd. byla podrobena analýze, ze které vychází následující strategie vykreslování pomocí virtuálních vrstev.

V tomto konceptu jsou navrženy čtyři vrstvy které jsou potřeba k vizualizaci obsazení železničních kolejí ve stanici.



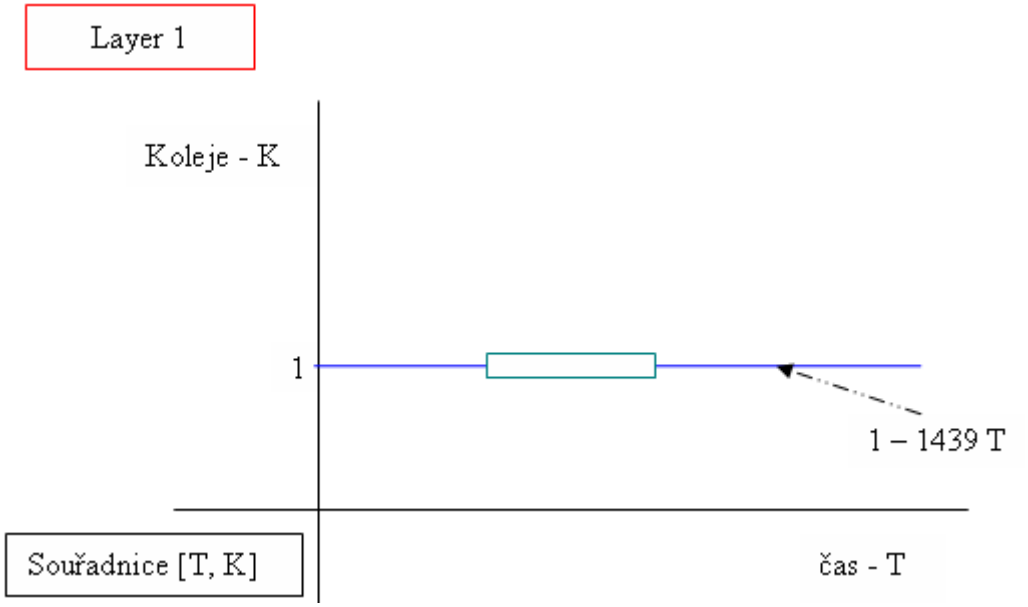
Obrázek 27 Ukázka vizualizace obsazení železničních kolejí ve stanici

Vrstva 1

První a zároveň nejnižší vrstva, která je koncipována jako dvourozměrný souřadnicový systém s vertikální osou *K* a horizontální *T*. Do tohoto souřadnicového systému je možné

vkládat grafické prvky, jejichž souřadnice jsou popsány pomocí dvou reálných čísel. Souřadnice v této vrstvě si označme jako [T,K].

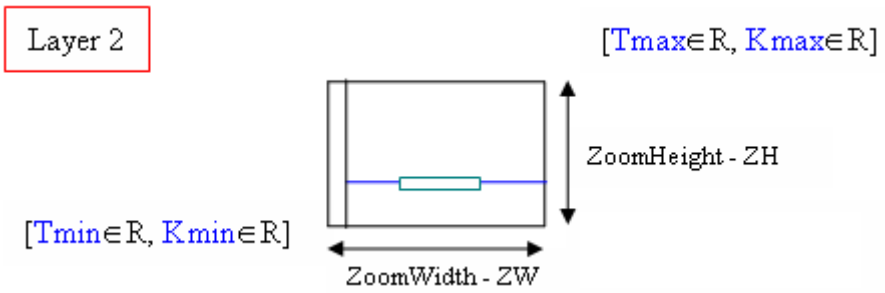
Například můžeme do této vrstvy vkládat grafické prvky typu čára, text či obdélník.



Obrázek 28 Nákres první vrstvy

Vrstva 2

Drhá vrstva je taková, která definuje obdélníkový výřez z vrstvy první. Tento výřez je definován pomocí levého dolního [Tmin, Kmin] a pravého horního rohu [Tmax, Kmax]. Souřadnice v této vrstvě si opět označme jako [T,K].



Obrázek 29 Nákres druhé vrstvy

Tato vrstva je důležitým prvkem pro přibližování, oddalování či pohybu a proto je pojmenována jako *Zoom Layer*. Šířka tohoto okna je označena **ZW** a vypočítá se pomocí vzorce (17). Výška tohoto okna je označena **ZH** a vypočítá se pomocí vzorce (18).

$$ZW = T\max - T\min \tag{17}$$

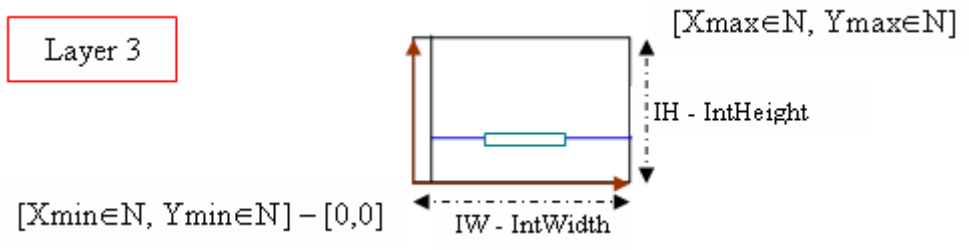
$$ZH = K\max - K\min \tag{18}$$

$$\begin{aligned} X &= Round((T - T\min) * (IW / ZW)) \\ Y &= Round((K - K\min) * (IH / ZH)) \end{aligned} \tag{19}$$

Souřadnice mezi touto druhou resp. první vrstvou a následující třetí lze převézt pomocí vzorce (19).

Vrstva 3

Třetí vrstva je již velice úzce spjata s výstupním grafickým zařízením a proto je zde nutné zavést nové dvě konstanty a to **IW** a **IH**. Zde předpokládáme, že výstupní grafické zařízení má souřadnicový systém celočíselný a jeho horizontální resp. vertikální rozsah je definován právě hodnotami **IW** resp. **IH** viz vzorec (20) resp. (21). Dále předpokládejme, že počátek souřadnicového systému je v levém dolním rohu a začíná souřadnicemi [0, 0] a končí [**IW**-1, **IH**-1]. Souřadnice v této vrstvě si označme jako [X,Y].



Obrázek 30 Nákres třetí vrstvy

$$IW = X\max - X\min \tag{20}$$

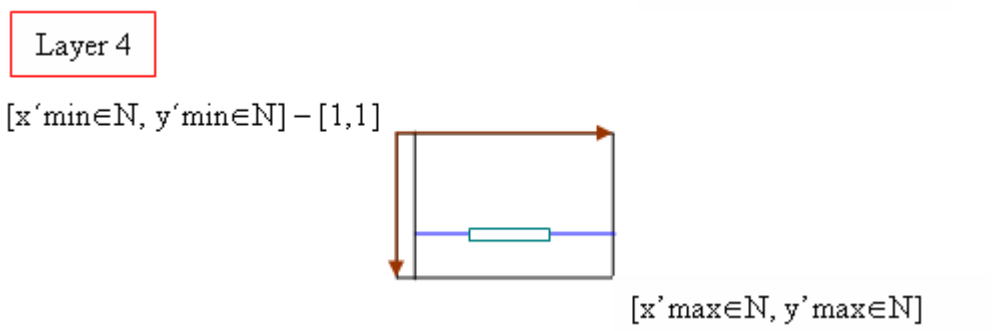
$$IH = Y\max - Y\min \tag{21}$$

$$\begin{aligned} T &= T\min + ((X * ZW) / IW) \\ K &= K\min + ((Y * ZH) / IH) \end{aligned} \tag{22}$$

Vrstva 4

Implementovat grafický výstup tak jak je popsán ve třetí vrstvě v prostředí Turbo Delphi by bylo značně pracné. Vhodnější je použít již existující komponentu *TImage* která je sice odlišná, ale pomocí jednoduché konverzní metody, kterou poskytuje tato čtvrtá vrstva, ji můžeme snadno použít. Komponenta *TImage* má počátek souřadnicového systému v levém horním rohu a začíná souřadnicemi [1,1]. Jelikož šířka a výška této vrstvy je totožná

s šířkou a výškou vrstvy třetí použijeme zde stejné označení **IW** a **IH**. Souřadnice v této vrstvě si označme jako [x´,y´].



Obrázek 31 Nákres čtvrté vrstvy

$$X = x' - 1$$
$$Y = IH - y' \tag{23}$$

$$x' = X + 1$$
$$y' = IH - Y \tag{24}$$

Převody jednotek délky

Dlužno ještě doplnit, že pro práci s těmito vrstvami budeme potřebovat metody pro konverze mezi jednotkami délky. Tyto metody využijeme při pohybu okna výřezu druhé virtuální vrstvy zadaných v jednotkách délky vrstvy čtvrté a naopak. Vzorec (25) definuje převod jednotky y' na K. Dále vzorec (26) převádí jednotky x' na T. Následující dva vzorce jsou inverzní tj. převod jednotky T na x' a K na y' vyjadřují vzorce (27) a (28).

$$K = (y' * ZH) / IH \tag{25}$$

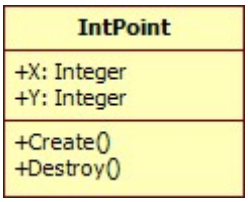
$$T = (x' * ZW) / IW \tag{26}$$

$$x' = Round((T * IW) / ZW \tag{27}$$

$$y' = Round((K * IH) / ZH \tag{28}$$

4.6.2 Třída IntPoint

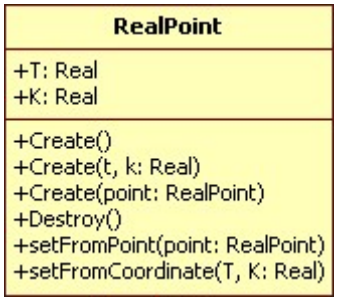
Třída *IntPoint* reprezentuje grafický dvojrozměrný bod. Souřadnice jsou pouze celočíselné. Tato třída je vhodná pro vyjádření souřadnic na vykreslovacím plátně výstupního grafického zařízení. Instance této třídy jsou použity jako body ve třetí a čtvrté virtuální vrstvě viz kapitola 4.6.1.



Obrázek 32 Model třídy IntPoint

4.6.3 Třída RealPoint

Tato třída reprezentuje grafický dvojrozměrný bod, kde souřadnice jsou v reálných číslech. Instance této třídy jsou použity v první a druhé virtuální vrstvě viz kapitola 4.6.1.



Obrázek 33 Model třídy RealPoint

4.6.4 Třída KonverzniParametr

Pro potřebu konverze souřadnic tak jak je to popsáno v kapitole 4.6.1 potřebujeme informace o výstupním grafickém zařízení či komponentě na kterou budeme vykreslovat grafické objekty. Všechny tyto potřebné informace jsou sloučeny do jednoho konverzního parametru.

Vlastnosti které obsahuje tato třída jsou *ImageHeight* resp. *ImageWidth* což je výška resp. šířka v bodech výstupního grafického zařízení. Další čtyři vlastnosti se týkají druhé virtuální vrstvy viz kapitola 4.6.1. Vlastnost *zoomBottomLeft_K* a *zoomBottomLeft_T* reprezentují souřadnice levého dolního rohu. Výšku resp. šířku této vrstvy zastupují poslední dvě vlastnosti, a to *zoomHeight* resp. *zoomWidth*.

Obrázek 34 Model třídy KonverzniParametr

4.6.5 Třída GraphicItem

Tato třída představuje abstraktní grafický prvek jehož potomci budou již konkrétní grafické prvky. Třída obsahuje virtuální metodu *paint*, která jej vykreslí na grafický výstup zadaný parametrem *image*. Tato metoda také obsahuje vstupní parametr *element*, který reprezentuje konverzní parametr potřebný pro převod souřadnic mezi virtuálními vrstvami viz kapitola 4.6.4.

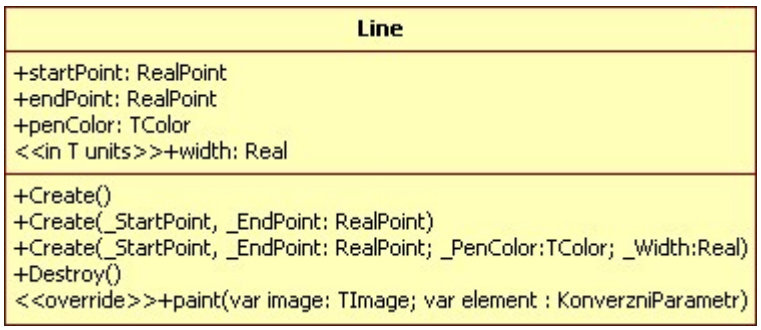
Obrázek 35 Model třídy GraphicItem

4.6.6 Třída Line

Tato třída je potomkem třídy *GraphicItem* viz kapitola 4.6.5. Třída Line reprezentuje grafický objekt, jak již je z názvu patrné, typu čára resp. *Line*.Tento objekt je definován dvěma body, a to počátečním *startPoint* a koncovým *endPoint*. Dalšími vlastnostmi je barva *penColor* a šířka čáry *width*.

Šířka čáry musí být ve specifikovaných jednotkách. Jednotka šířky čáry je tedy stanovena v jednotkách první virtuální vrstvy (viz kapitola 4.6.1) osy T.

Metody této třídy jsou tři přetížené konstruktory, destruktor a zděděná virtuální metoda *paint* která jej vykreslí na grafický výstup zadaný parametrem *image*. Tato metoda také obsahuje vstupní parametr *element*, který reprezentuje konverzní parametr potřebný pro převod souřadnic mezi virtuálními vrstvami viz kapitola 4.6.4.



Obrázek 36 Model třídy Line

4.6.7 Třída TextGO

Pro reprezentaci textu na grafickém výstupu je zde třída *TextGO* (Text Graphic Object). Tento objekt je vymezen vlastností *text* definující obsah, dále umístěním *place*[2] na první virtuální vrstvě. Dalšími dvěma vlastnostmi je barva *fontColor* a styl textu *fontStyle*. Poslední vlastností *width* je velikost textu v jednotkách první virtuální vrstvy (viz kapitola 4.6.1) osy T.

Metodami této třídy jsou dva konstruktory, destruktor a zděděná virtuální metoda *paint* která jej vykreslí na grafický výstup zadaný parametrem *image*. Tato metoda také obsahuje vstupní parametr *element*, který reprezentuje konverzní parametr potřebný pro převod souřadnic mezi virtuálními vrstvami viz kapitola 4.6.4.

Obrázek 37 Model třídy TextGO

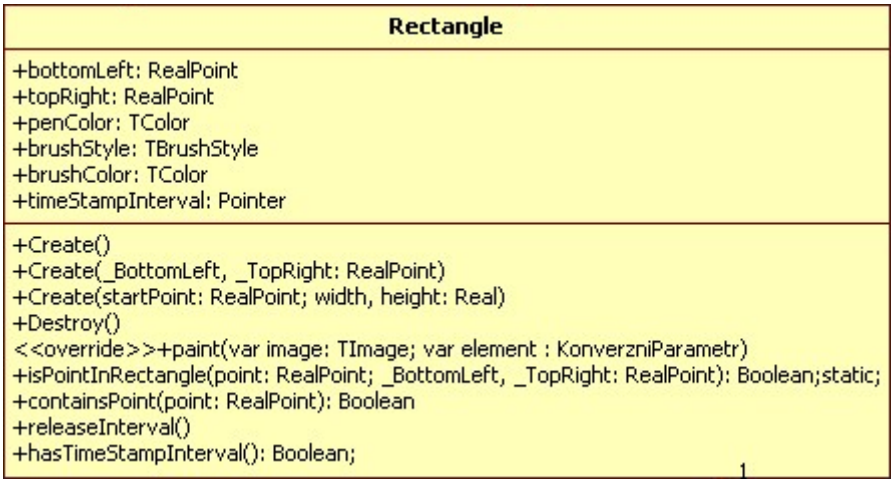
4.6.8 Třída Rectangle

Poslední třída z řady potomků abstraktní třídy *GraphicItem* je třída *Rectangle*. Tato třída je již sofistikovanější grafický objekt se kterým je možné provádět celou řadu operací. Popišme si nejprve vlastnosti kterými tato třída disponuje. První dvě vlastnosti vymezují velikost obdélníku dvěma souřadnicemi *bottomLeft* resp. *topRight*, které znázorňují souřadnice levého dolního resp. pravého horního rohu. Další vlastnost *penColor* reprezentuje barvu obvodové čáry a následující dvě vlastnosti *brushStyle* resp. *brushColor* znamenají styl resp. barvu výplně obdélníka. Poslední vlastností *timeStampInterval* je ukazatel na instanci třídy typu *TimeStampInterval* (viz kapitola 4.5.4) přetypovaný na *Pointer*.

Metody této třídy jsou tři konstruktory a virtuální destruktor[3]. Opět je zde metoda *paint*, která jej vykreslí na grafický výstup zadaný parametrem *image*. Tato metoda také obsahuje

[2] Text je vždy orientován vodorovně, a to zleva doprava. Souřadnice je prostřední bod v textu a to jak vertikálně, tak horizontálně. Např. v textu „nemocnice“ by byla souřadnice uprostřed písmena „o“.
[3] Nejen tato třída, ale všechny třídy disponují virtuálním destruktorem, který při použití polymorfismu a následné destrukci objektu zabezpečí zavolání příslušného destruktoru, který uvolní veškerou paměť alokovanou objektem.

vstupní parametr *element*, který reprezentuje konverzní parametr potřebný pro převod souřadnic mezi virtuálními vrstvami viz kapitola 4.6.4. Pro obecné zjištění zda v obdélníku zadaným dvěma vrcholy se nachází příslušný bod je zde metoda *isPointInRectangle.* Tato metoda je autonomní tj. není závislá na konkrétním stavu objektu, a proto je navržena jako veřejná, statická. Pro zjištění zda konkrétní instance třídy obsahuje příslušný bod je zde metoda *containsPoint*. Odebrání ukazatele resp. nastavení hodnoty *nil* ve vlastnosti *timeStampInterval* je k dispozici metoda *releaseInterval*. Pro zjištění zda tento ukazatel obsahuje hodnotu *nil* je tu metoda *hasTimeStampInterval*.



Obrázek 38 Model třídy Rectangle

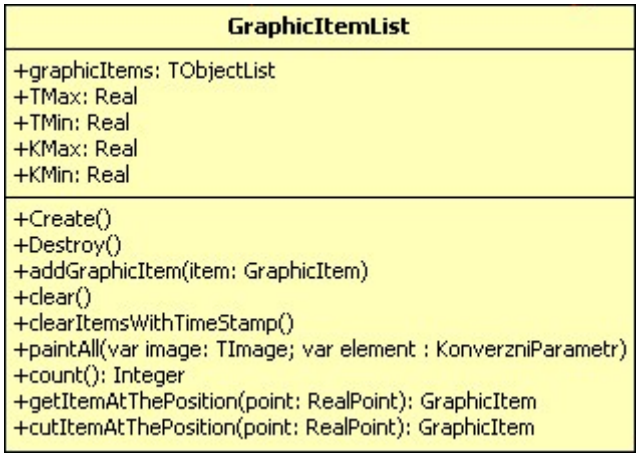
4.6.9 Třída GraphicItemList

Pro potřebu uspořádání grafických objektů do seznamu je zde třída GraphicItemList, která se stará o uchovávání, vkládání, vykreslování či mazání grafických objektů. Zároveň při každém vložení grafického objektu si zjistí zda jeho poloha na první virtuální vrstvě nevytváří nové osové maximum či minimum.

První vlastnost *graphicItems* je seznam již zmíněných grafických objektů. Další čtyři vlastnosti vyjadřují maxima či minima dosažená na jednotlivých osách vloženými grafickými objekty. Tato maxima či minima budou posléze potřeba při konstruování metod typu *setZoomForFullView* (viz kapitola 4.6.11), které nastaví velikost *zoom* okna takovým způsobem, aby byly viditelné všechny grafické objekty obsažené v seznamu *graphicItems* a zároveň byl zachován poměr stran výstupního grafického okna.

Třída disponuje metodami pro vkládání grafických objektů *addGraphicItem*, metodou pro smazání všech objektů ze seznamu *clear*, metodou *clearItemsWithTimeStamp* pro smazání všech objektů jejichž vlastnost *timeStampInterval* není *nul*. Další metoda paintAll projde

všechny grafické objekty v seznamu a zavolá jejich metodu *paint*. Počet grafických objektů se zjistí voláním metody count. Dvě poslední metody jsou důležité pro interakci s grafickým uživatelským rozhraním. Pokud zjišťujeme zda pod určitou souřadnicí existuje nějaký grafický prvek, pak je zde k dispozici metoda getItemAtThePosition. Podobnou funkci má i metoda cutItemAtThePosition jen s tím rozdílem, že tato metoda pokud najde prvek v seznamu, pak ho vyjme a vrátí v návratové hodnotě.

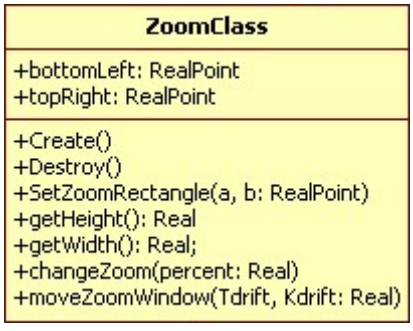


Obrázek 39 Model třídy GraphicItemList

4.6.10 Třída ZoomClass

Tato třída reprezentuje druhou virtuální vrstvu (viz kapitola 4.6.1). Její vlastnosti jsou souřadnice dolního levého *bottomLeft* a horního pravého *topRight* rohu *zoom* okna.

Veškeré její metody slouží k získání informací o šířce či výšce okna popřípadě k nastavení příslušných vlastností.



Obrázek 40 Model třídy ZoomClass

4.6.11 Třída PositionAndUnitsDataConvert

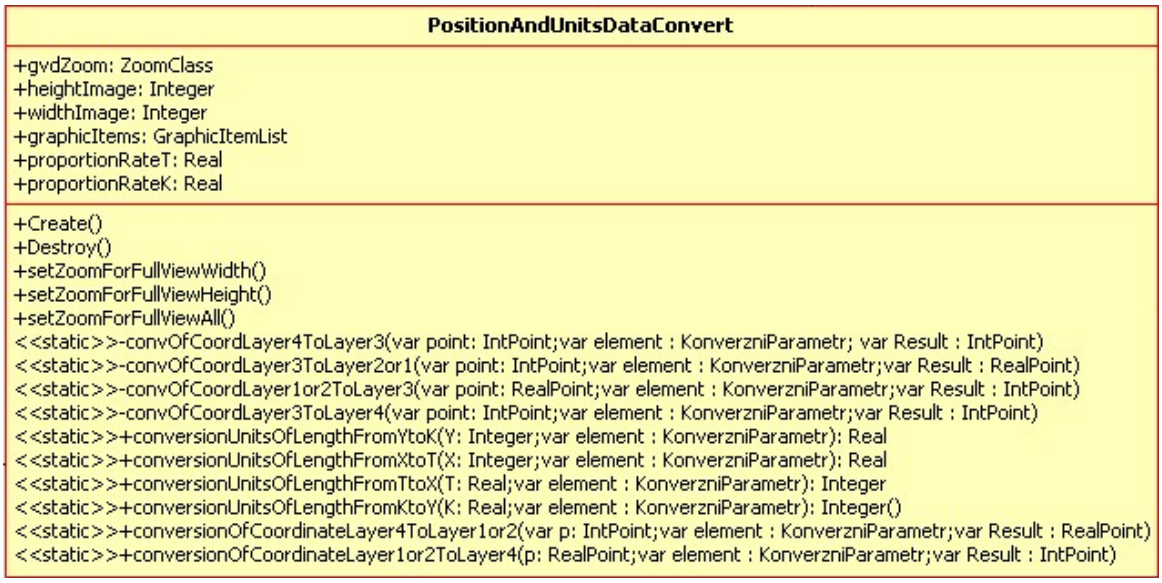
Tato třída představuje špičku ledovce celé grafické vrstvy. Třída agreguje či používá ve svých metodách všechny výše zmíněné třídy této kapitoly „grafická vrstva“.

Mezi její vlastnosti patří *gvdZoom* viz kapitola 4.6.10, dále výška resp. šířka výstupního grafického zařízení *heightImage* resp. *widthImage* a seznam grafických prvků *graphicItems*. Poslední dvě vlastnosti reprezentují proporční poměr stran první virtuální vrstvy. Jelikož tato vrstva má na svých osách různé jednotky, pak je třeba určit v jakém poměru tyto jednotky jsou.

Metody této třídy se dají rozdělit do dvou skupin první skupina *sezZoomForFullView* nastavuje velikost *zoom* okna s přihlédnutím na velikost výstupního grafického okna a rozložením grafických prvků na první virtuální vrstvě.

První metoda z této skupiny *setZoomForFullViewWidth* nastavuje šířku okna podle nejníže resp. nejvýše umístěného grafického prvku osy T, první virtuální vrstvy, a výšku okna podle proporčního poměru (viz vlastnost *ProportionRateT/K*) s přihlédnutím na velikost výstupního grafického okna.

Druhá metoda z této skupiny *setZoomForFullViewHeight* nastavuje výšku okna podle nejníže resp. nejvýše umístěného grafického prvku osy K, první virtuální vrstvy, a šířku okna podle proporčního poměru (viz vlastnost *ProportionRateT/K*) s přihlédnutím na velikost výstupního grafického okna.



Obrázek 41 Model třídy PositionAndUnitsDataConvert

Druhou skupinou metod jsou konverzní soukromé a veřejné metody. Mezi ty soukromé patří konverze souřadnic mezi jednotlivými virtuálními vrstvami. Veřejné metody jsou také konverzní, ale první čtyři z šesti uvedených na obrázku 41 převádí jednotky délky mezi první a poslední čtvrtou virtuální vrstvou (viz kapitola 4.6.1 vzorce (25) až (28)).

Posledními dvěma metodami provádíme konverze souřadnic vždy z první virtuální vrstvy do čtvrté a naopak.

5 Popis a práce s aplikací VNNK pracující s API VNNK Interface

Aplikace VNNK (**V**ýpočet **N**áhradní **N**ástupištní **K**oleje) je napsána v programovacím jazyku Delphi. Aplikace využívá aplikační rozhraní VNNK Interface které je popsáno v kapitole 4.

5.1 Obsazení dopravních kolejí

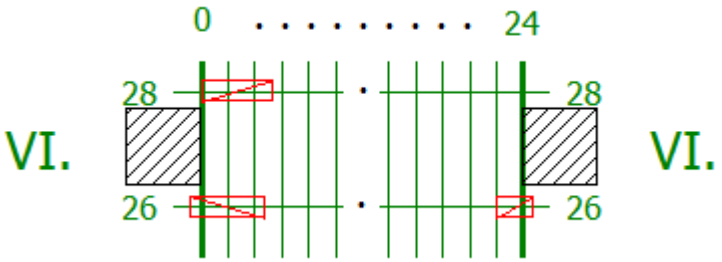
Po spuštění programu se zobrazí první záložka **Obsazení dopravních kolejí**. Jak již název napovídá, zde je možné zobrazit obsazení dopravních kolejí. Pro otevření datového (**.dataIni*), konfiguračního (**.ini*) a výpočtového (**.compIni*) souboru klikněte na tlačítko **Open**. Tlačítkem **Paint** načtete otevřené soubory a vykreslíte obsazení kolejí. Nyní již můžete používat navigační tlačítka, přibližovat, oddalovat či měnit způsoby zobrazení.

Pro snazší práci je umožněna i navigace pomocí myši. Pomocí myši je možné posouvat s plánem metodou uchop, suň a pusť. Dvojitým kliknutím na levé tlačítko myši plán přiblížíte a jedním kliknutím na pravé tlačítko plán oddálíte.

Diagonálně proškrtnuté obdélníky, viz obrázek 42, znázorňují vlakové spoje a jimi obsazené koleje v čase. Při kliknutí na ně se v prostředním okně na horní liště zobrazí seznam vlakových záznamů vybraného vlakového spoje. Dvojitým kliknutím na tyto záznamy se zobrazí vlastnosti s možností úprav. Dále při kliknutí na spoj se v pravém okně na horní liště zobrazí příslušná kolej na kterou má vybraný spoj plánovaný příjezd. I v tomto pravém okně je možné, po dvojitém kliknutí na záznam, prohlížet či editovat detailní vlastnosti koleje.

Na obrázku 42 si můžete všimnout, že spoj na koleji 26 je vykreslen už před půlnocí a ten další až po půlnoci. Tato anomálie se vyskytuje tehdy, když má spoj plánovaný příjezd před půlnocí a odjezd až druhý den po půlnoci. Oba úhlopříčně proškrtnuté obdélníky tedy patří ke stejnému spoji.

Úhlopříčně šrafované čtverce a obdélníky značí nástupiště. Nástupiště jsou označena římskými čísli tak, jak je tomu například na obrázku 42.



Obrázek 42 Ukázka plánu obsazení kolejí

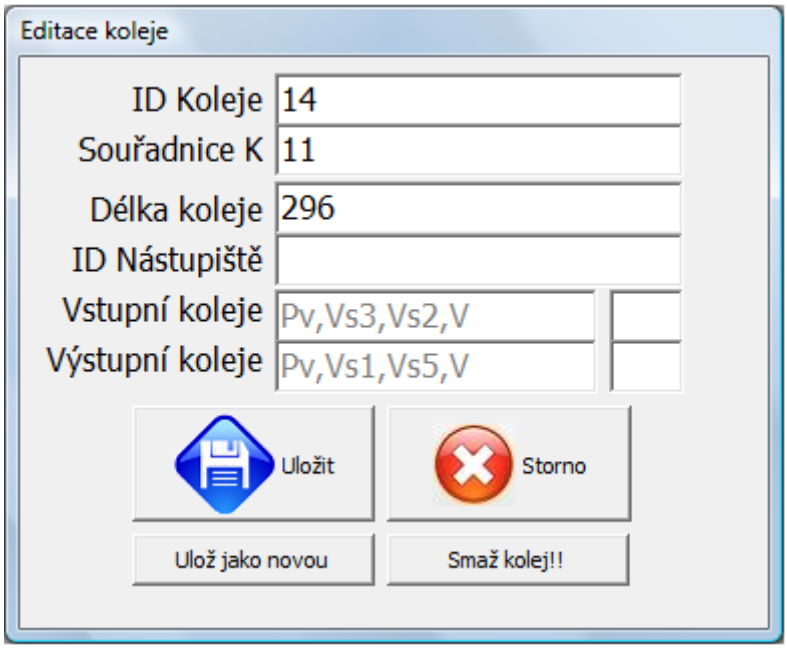
5.2 Seznam vlakových záznamů

Kompletní seznam vlakových záznamů najdete v záložce **Seznam vlakových záznamů**. Pro načtení tohoto seznamu klikněte na zelené tlačítko umístěné vlevo na horní liště. Záznamy je možné seřadit kliknutím na hlavičku vybraného sloupce. Opětovným kliknutím seřadíte záznamy v opačném pořadí. Záznamy jsou vždy řazeny podle datového typu sloupce.

Dvojitým kliknutím či označením a stisknutím tlačítka **Enter** otevřete okno **Editace vlakového záznamu**. Jak již název napovídá můžete zde editovat vlastnosti vlakového záznamu.

5.3 Seznam kolejí

Kompletní seznam kolejí najdete v záložce Seznam Kolejí. Pro načtení tohoto seznamu klikněte na zelené tlačítko umístěné vlevo na horní liště. Dvojitým kliknutím či označením a stisknutím tlačítka **Enter** otevřete okno **Editace koleje** viz obrázek 43. Zde je možné editovat vlastnosti koleje, uložit ji jako novou či smazat ze seznamu kolejí.



Obrázek 43 Okno Editace koleje

5.4 Tvorba datového souboru

Při prvním použití programu na nová testovací data musíte vytvořit ze zdrojového souboru (**.csv*) datový soubor (**.dataIni*). Spustě tedy program a přejděte do záložky **Tvorba datového souboru**.

Zde klikněte na tlačítko **Otevři soubor**. V dialogovém okně vyhledejte a otevřete zdrojový soubor (**.csv*).

V prvním sloupci zleva se nachází seznam položek vlakového záznamu. Ve druhém sloupci je seznam sloupců zdrojových dat (hlaviček sloupců). Nyní je potřeba vybrat z levého a prostředního slupce vždy jednu položku a propojit tlačítkem **Propojit vybranou položku vlakového záznamu se sloupcem**. Dbejte prosím na typovou kompatibilitu položek. Po každém propojení se vždy vybrané položky odeberou, aby nemohly být propojeny vícekrát.

Po vybrání všech položek vlakového záznamu se zaktivuje tlačítko **Ulož datový soubor**. V dialogovém okně zvolte umístění, doplňte název datového souboru a klikněte na tlačítko **Uložit**. Pokud se soubor uložil bez chybových hlášení, typová kompatibilita položek je v pořádku. Pokud se vyskytlo chybové hlášení, pak některá propojená položka není s vybranou datovou položkou typově kompatibilní.

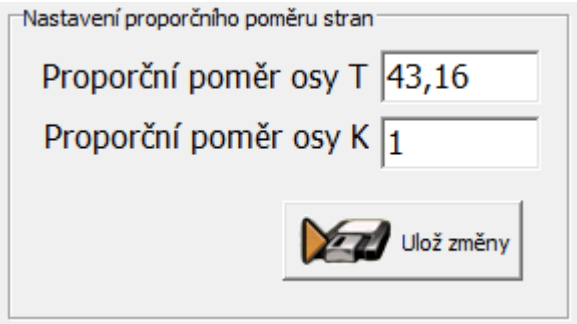
Pokud se chcete pokusit znovu vytvořit nový datový soubor, klikněte na tlačítko **Storno**.

5.5 Nastavení programu

V záložce **Nastavení** je možnost nejrůznějších nastavení, které se týkají jak způsobu vykreslení plánu obsazení kolejí, tak i výpočtu náhradní nástupištní koleje pro zpožděný vlak. Toto nastavení je možné měnit až po prvním vykreslení plánu tj. kliknutím na tlačítko **Paint** v záložce **Obsazení dopravních kolejí**.

5.5.1 Nastavení proporčního poměru stran grafikonu

Smysl proporčního poměru stran najdete v kapitole 4.6.11. Pro uložení změn klikněte na tlačítko **Ulož změny**.



Obrázek 44 Nastavení proporčního poměru stran

5.5.2 Nastavení datumových omezení

Datumovým omezením se nastavuje které vlakové záznamy mají být zahrnuty pro sestavení plánu obsazení kolejí v záložce **Obsazení dopravních kolejí**. Více o datumových omezeních viz kapitola 4.5.2.

5.5.3 Stanovení vah kritérií

U různých metod se stanovují váhy kritérií vždy odlišným způsobem.

Pro nastavení metody pořadí (viz obrázek 53) se používají tlačítka **A+** resp. **A** pro zvýšení resp. snížení priority kritéria A. Priorita kriterií B, C a D se nastavuje analogicky jako u kritéria A. Detailní informace o metodě viz kapitola 3.3.1.

U bodovací metody (viz obrázek 54) se nastavují priority kritérií pomocí počtu bodů zadaných v celých číslech. Bližší informace o metodě viz kapitola 3.4.1.

Fullerova metoda párově porovnává všechny kombinace kritérií. Detailní informace o této metodě najdete v kapitole 3.3.2. Nastavení vah kritérií Fullerovy metody je zobrazeno na obrázku 55, kde jsou nad sebou vždy posuzovány kombinace dvou kritérií.

Saatyho metoda kvantitativního párového porovnáni je detailně popsána v kapitole 3.4.2. Její nastavení je znázorněno na obrázku 56. Tato metoda také používá párové porovnání

kritérií, ale počet nastavitelných stupňů je u každého páru 9. Tato metoda umožňuje nastavit vztahy mezi kritérii velice přesně.

Pro uložení provedených změn vždy u příslušné metody stiskněte tlačítko **Ulož změny**.

5.5.4 Nastavení parametrů pro výpočty

Tyto parametry jsou nezbytné pro výpočty náhradních nástupištních kolejí.

Doba potřebná pro příjezd resp. odjezd se nastavuje v celých minutách. Tyto doby jsou potřebné pro výpočet kritérií A i B. Pro stanovení kritéria A je dále potřeba nastavit dobu výhledu, která je také v celých minutách. Posledním parametrem je délka jednoho vozu, která se nastavuje jako reálné číslo v metrech. Tato délka se používá pro výpočet celkové délky vlaku/soupravy. Více o těchto parametrech najdete v kapitole 2.2.

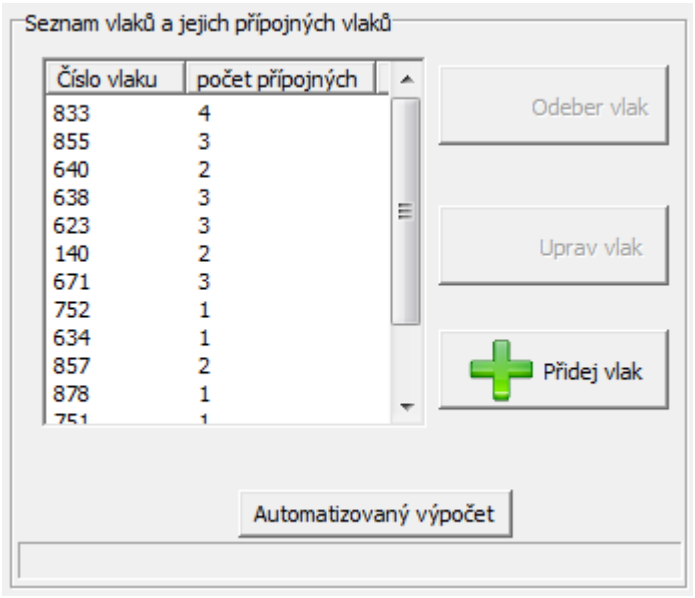
5.5.5 Nastavení přípojných vlaků

Pro účely čekacích dob je považován za první vlak ten vlak, od něhož je zajištěn přípoj. Vlak, na který je zajištěn přípoj, je považován za druhý vlak.

Přípojnými vlaky jsou vlaky, u nichž interval mezi pravidelným příjezdem prvního vlaku a pravidelným odjezdem druhého vlaku je shodný nebo větší než doba potřebná na přestup mezi těmito vlaky ve stanici. Bližší informace o tom, které vlaky jsou přípojné najdete v literatuře [5].

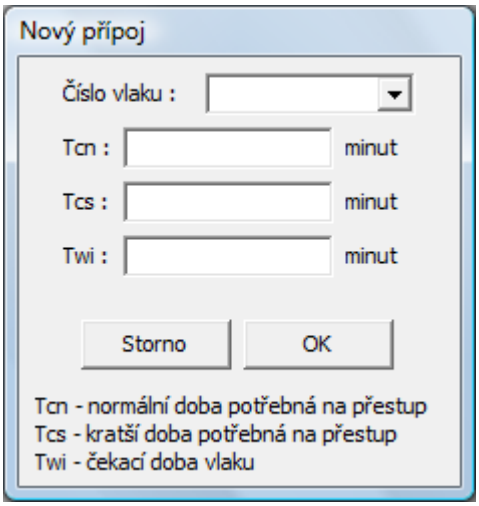
V programu VNNK se v záložce **Nastavení** přípojné vlaky registrují. Touto registrací se myslí výběr vlaku a k němu přiřazení jednoho či více přípojných vlaků.

Jak takovéto nastavení vypadá znázorňuje obrázek 45.



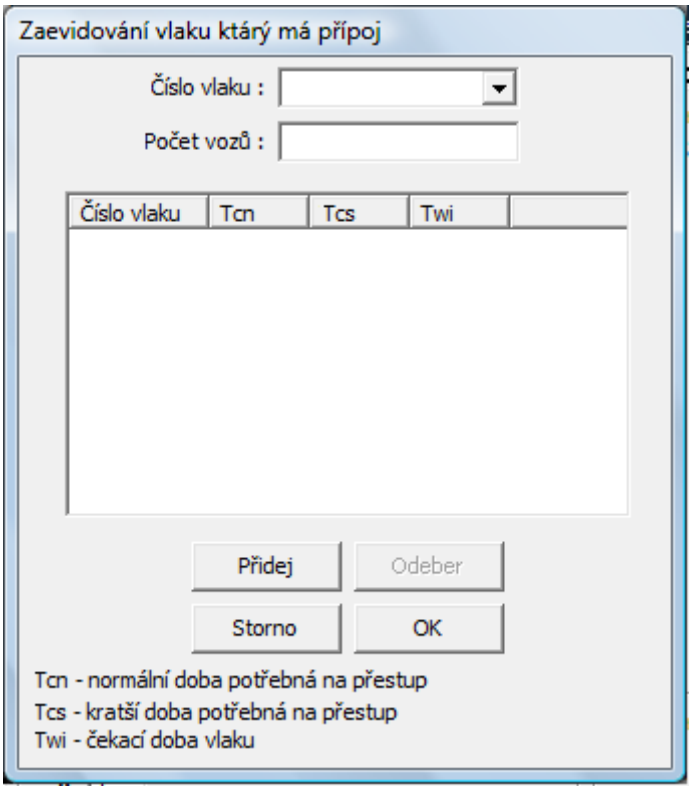
Obrázek 45 Nastavení seznamu vlaků a jejich přípojných vlaků

Přidání resp. zaregistrování takového vlaku, kterému chcete přiřadit jeden či více přípojných vlaků, začněte kliknutím na tlačítko **Přidej vlak**. Tímto se zobrazí okno viz obrázek 47. V tomto okně vyplňte políčko **Číslo vlaku** a **Počet vozů**. Nyní můžete začít přidávat přípojné vlaky pomocí tlačítka **Přidej**. Stisknutím tohoto tlačítka vyvoláte okno **Nový přípoj**, které je znázorněno na obrázku 46.



Obrázek 46 Okno pro registraci nového přípojného vlaku

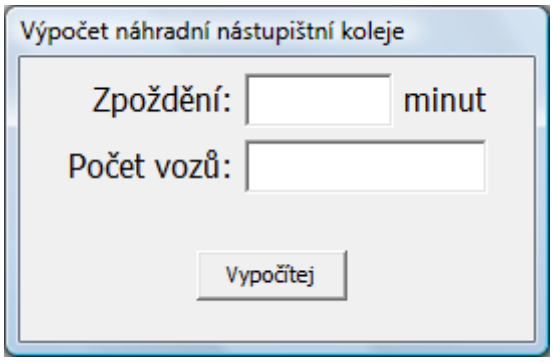
V tomto dialogovém okně musíte zadat normální dobu potřebnou na přestup *Tcn*, kratší dobu potřebnou na přestup *Tcs* a čekací dobu vlaku *Twi*. Tyto časy najdete například v literatuře [5].



Obrázek 47 Okno pro zaevidování vlaku který má přípoj

5.6 Výpočet náhradní nástupištní koleje u vybraného vlaku

Vlak je možné vybrat buď přímo ze záložky **Seznam vlakových záznamů**. Zde klikněte na vybranou položku seznamu pravým tlačítkem myši a vyberte z nabídky **Vypočítej náhradní kolej**.



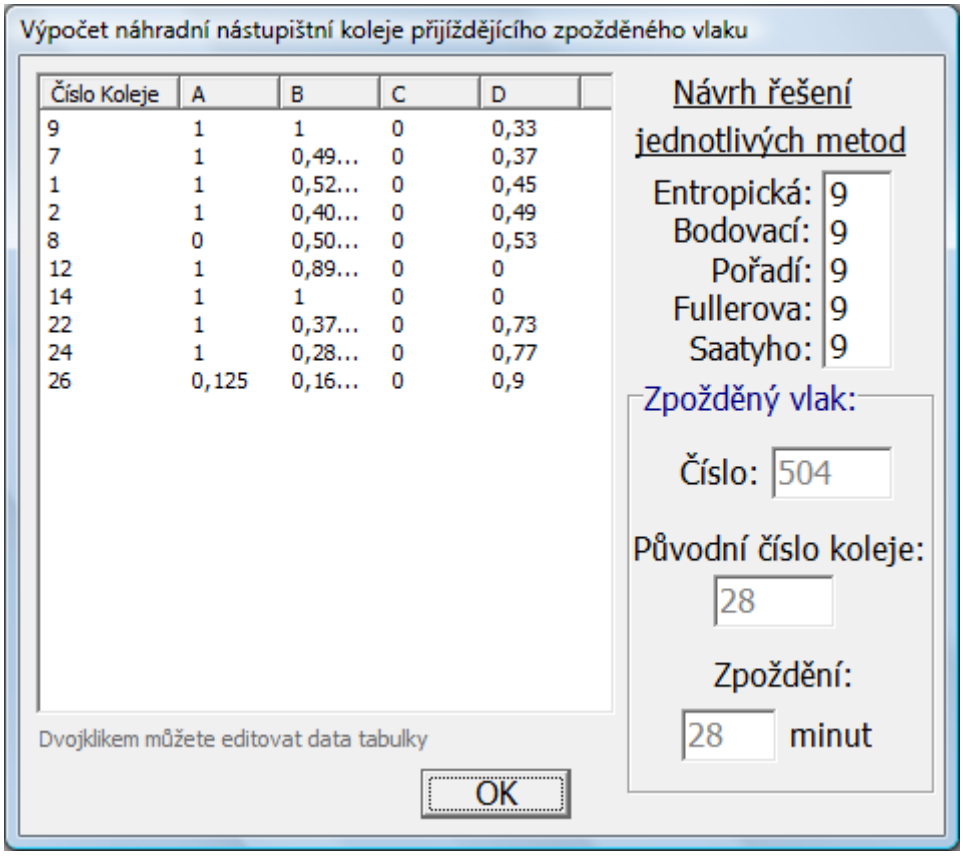
Obrázek 48 Okno pro zadání vstupních parametrů výpočtu náhradní koleje

Zobrazí se okno znázorněné na obrázku 48 pro zadání vstupních parametrů výpočtu náhradní nástupištní koleje. Zde musíte zadat zpoždění a počet vozů. Obě hodnoty musí být zadány celočíselně.

Existuje ještě jeden způsob, jak vybrat vlak u kterého chceme spočítat zpoždění. V záložce **Obsazení dopravních kolejí** klikněte na vybraný spoj.

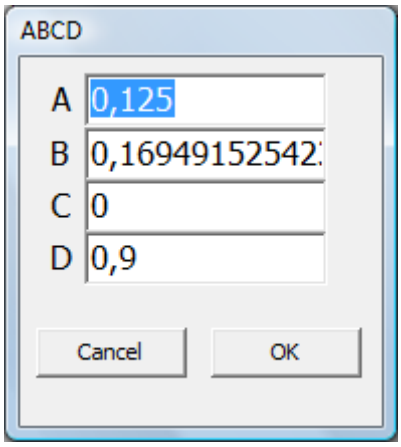
V prostředním okně na horní liště se zobrazí seznam vlakových záznamů vybraného vlakového spoje. V tomto seznamu vyberte libovolný záznam, klikněte na něj pravým tlačítkem a vyberte z nabídky **Vypočítej náhradní kolej**. Všechny záznamy v seznamu patří ke stejnému spoji a proto můžete vybrat kterýkoli z nich a dojede ke stejnému výsledku.

Nyní přejděme zpět k tomu co se stane po kliknutí na tlačítko **Vypočítej** znázorněné na obrázku 48. Pro tento příklad je zvolen vlak číslo 504, zpoždění 28 minut a počet vozů 11. Po kliknutí na již zmíněné tlačítko **Vypočítej** se zobrazí okno znázorněné na obrázku 49.



Obrázek 49 Okno s výsledkem výpočtu náhradní koleje

V tomto okně jsou návrhy řešení jednotlivých metod. V případě potřeby editovat data tabulky je zde možnost dvojitým kliknutím na záznam vyvolat okno **ABCD** znázorněný na obrázku 50. V tomto okně můžete měnit jednotlivá kritéria A, B, C i D.



Obrázek 50 Okno pro editaci kritérií A, B, C a D

5.7 Porovnání metod vícekrit. hodnocení variant s využitím aplikace VNNK

Tato kapitola se zabývá jak pomocí aplikace VNNK porovnat jednotlivé vícekriteriální metody z hlediska procentuální úspěšnosti. Tato úspěšnost se týká výběru náhradní nástupištní koleje pro zpožděný vlak. Touto procentuální úspěšností je tedy myšlena četnost správných rozhodnutí vydělená počtem všech rozhodnutí. O tom zda rozhodnutí bylo správné či nikoliv posuzuje znalec (expert).

Nejprve je třeba v aplikaci VNNK nastavit seznam vlaků a jejich přípojných vlaků. O tomto nastavení pojednává kapitola 5.5.5.

Nyní stisknutím tlačítka **Automatizovaný výpočet** v záložce **Nastavení** spustíte proces série výpočtů náhradních nástupištních kolejí. Výpočet se provádí pro všechny vlaky v seznamu vlaků a jejich přípojných vlaků. Pro každý takovýto vlak se postupně počítá zpoždění v intervalu od 1 do 60 minut s krokem 1 minuta. Výpočty jsou uloženy do (**.csv*) souborů umístěných v aktuálním adresáři spuštěného programu.

Nyní je třeba projít všechny soubory a expertně doplnit u každé matice nejvhodnější kolej. Na konci každého souboru je k dispozici souhrnná statistika, jak úspěšně si metody vedly.

Abychom se dozvěděli souhrnné statistiky musíme ke každé metodě zprůměrňovat hodnoty úspěšnosti všech souborů.

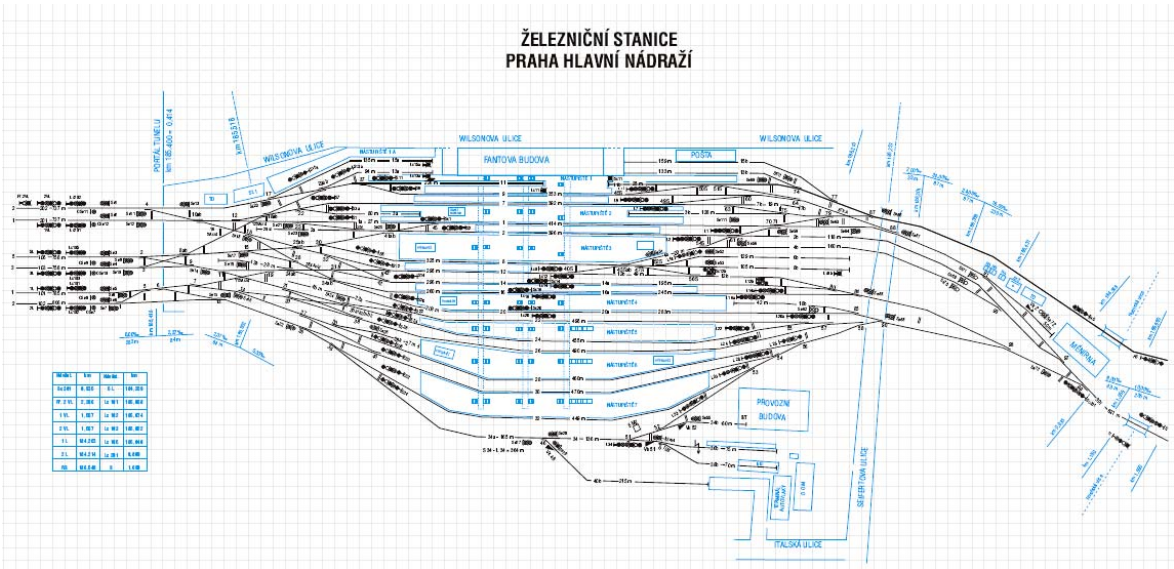
6 Případová studie ŽST Praha hlavní nádraží

Tato kapitola se věnuje nasazení aplikace na konkrétní železniční stanici Praha hlavní nádraží. Na datech z této stanice jsou testovány všechny činnosti spjaté s výpočtem náhradní nástupištní koleje pro zpožděný přijíždějící vlak. Následně jsou expertně vybrány nejlepší varianty náhradních kolejí. V závislosti na tomto výběru je vypočtena procentuální úspěšnost jednotlivých metod. Tato úspěšnost je dále optimalizována hledáním vhodnějších vah kritérií vícekriteriálních metod. V závěrečné části jsou tyto optimalizované hodnoty procentuální úspěšnosti vícekriteriálních metod uvedeny.

6.1 Vstupní data případové studie

Vstupními daty pro tuto studii jsou:

- Seznam vlaků (vstupní kolej do stanice, čas příjezdu, čas odjezdu, výstupní kolej ze stanice, apod.) viz soubor na přiloženém CD *Vlaky.csv*.
- Sešitový jízdní řád (datumová omezení jízdy vlaků) [7].
- Řazení vlaků pro obvod Praha hl.n. (délka vlaku) [8].
- Čekací doby a opatření při zpoždění vlaků osobní dopravy [5].
- Uspořádání kolejiště v osobní stanici (možnosti využití jednotlivých nástupištních kolejí pro vstupní, resp. výstupní koleje do/ze stanice, délky nástupištních kolejí apod.) viz obrázek 51 či soubor *uzstPraha.hl.nadr.pdf* na přiloženém CD.
- Staniční intervaly (doba potřebná pro příjezd resp. odjezd vlaku apod.).



Obrázek 51 Plán rozmístění nástupišť a kolejí v ŽST Praha hl. nádraží

6.2 Výstupní data případové studie

Tato kapitola se zabývá zpracováním vstupních dat a následné tvorbě výstupů, které vedou k dosažení stanovených cílů.

6.2.1 Výběr testovací množiny vlaků

Pro samotný výpočet náhradní nástupištní koleje bylo třeba určit testovací množinu vlaků u nichž se bude opakovaně spouštět výpočet náhradní koleje, ale vždy s jinou hodnotou doby zpoždění vlaku.

Tabulka 2 Vybrané vlaky pro testování úspěšnosti metod

číslo vlaku	čas příjezdu	směr příjezdu	čas odjezdu	směr odjezdu	plánovaná kolej	číslo nástupiště
833	17:51	Pv	19:45	V	14	-
855	16:45	Pv	17:10	V	8	III.
640	15:39	Vs2	16:14	V	28	VI.
638	14:39	Vs2	16:23	Vs1	30	VII.
623	15:45	Pv	16:10	Hr	2	III.
140	16:48	V	17:10	Vs5	26	VI.
671	17:50	V	18:10	Hr	7	II.
752	8:57	V	9:05	Pv	8	III.
634	9:39	Vs2	10:23	Vs1	30	VII.
857	18:45	Pv	19:10	V	8	III.
878	14:43	V	14:50	Pv	2	III.
751	17:45	Pv	18:10	V	8	III.
72	17:55	V	18:16	Vs5	24	V.
142	14:48	V	15:46	Vs5	26	VI.

Aby byla plně využita všechna čtyři posuzovací kriteria A až D, bylo třeba vybrat takové vlaky, na které ve stanici čeká přípojný vlak a tím pokryjeme i posuzovací kriterium C (obsazení koleje u stejného nástupiště přípojným vlakem), které by jinak bylo zanedbáno. Ostatní tři kriteria (A, B, D) mají vždy, ať už v souvislosti s kterýmkoliv vlakem odpovídající hodnotu a nejsou zanedbávána. K účelu výběru takovýchto vlaků posloužila publikace [5] (verze GVD 2004/2005), ze které bylo vybráno z 14 vlaků viz tabulka 2. Jak tyto vlaky nastavit v aplikaci VNNK najdete v kapitole 5.5.5.

6.2.2 Výpočet kriteriálních matic na testovací množině vlaků

Každý vlak z tabulky 2 byl testován pro zpoždění od 1 do 60 minut s krokem jedné minuty. Tímto způsobem vzniklo 14 (**.csv*) souborů, kde v každém je 60 kriteriálních matic.

Celkový počet matic je tedy 840. Každá matice prošla všemi pěti vícekriteriálními metodami (viz kapitola 3) a výsledek byl ke každé matici doplněn. Tento postup byl realizován za pomocí aplikace VNNK. Detailní popis jak používat aplikaci, pro automatizovaný výpočet kriteriálních matic, je objasněn v kapitole 5.7.

6.2.3 Optimalizace procentuální úspěšnosti vícekriteriálních metod

Výsledek procesu zjišťování úspěšnosti vícekriteriálních metod je závislý na dvou vzájemně nezávislých veličinách.

První veličinou je intuitivní výběr nejlepších variant expertem. Zde čistě záleží na schopnostech a zkušenostech experta.

Druhou veličinou je výběr vstupních parametrů tj. stanovení vah kritérií vícekriteriálních metod. Jedinou metodou, kde není třeba žádné parametry nastavovat je metoda Entropická viz kapitola 3.2.1.

Expertní analýza variant

Všech 840 kriteriálních matic (viz kapitola 6.2.2) muselo být expertně posouzeno. V každé matici byla vybrána právě jedna kolej, kterou považoval expert za nejlepší variantu. V závislosti na tom, zda byla či nebyla vybrána totožná kolej, kterou vybrala některá z metod, rostla či klesala procentuální úspěšnost metod.

Stanovení nejlepší koleje je v některých případech velice komplikované a může se stát, že se zmýlí všechny vícekriteriální metody. Tento případ je znázorněn v tabulce číslo 3.

Tabulka 3 Příklad vícekriteriální matice

Vlak č:	**671**						
Kolej č:	**7**						
	Zpoždění:	31minut					
	Číslo koleje:	**9**	7	*1*	2	8	12
	A	**1,00**	1,00	*0,88*	1,00	1,00	1,00
	B	**1,00**	0,05	*0,90*	1,00	0,57	1,00
	C	**0,00**	1,00	*1,00*	0,00	0,00	0,00
	D	**0,81**	1,00	*0,90*	0,73	0,69	0,00
Nejlepší kolej podle metody Entropické:	1	0,76	0,75	0,91	0,74	0,61	0,54
Nejlepší kolej podle metody Bodovací:	1	0,86	0,62	0,90	0,85	0,67	0,70
Nejlepší kolej podle metody Pořadí:	1	0,78	0,62	0,91	0,77	0,60	0,70
Nejlepší kolej podle metody Fullerovy:	1	0,80	0,68	0,91	0,79	0,64	0,67
Nejlepší kolej podle metody Saatyho:	1	0,86	0,78	0,90	0,85	0,75	0,79
Nejlepší kolej podle Experta:	**9**						

Nyní si popišme z jakého důvodu expert zvolil kolej číslo 9.

- Hodnota kritéria *A=1* napovídá, že kolej číslo 9 bude v době příjezdu volná.
- Hodnota kritéria *B=1* naznačuje, že po celou dobu pobytu vlaku ve stanici bude kolej volná.
- Hodnota kritéria C=0 značí, že u *I.* nástupiště nestojí žádný přípojný vlak. Přípojný vlak stojí u nástupiště *II.*
- Hodnota kritéria D=0,81 vypovídá o tom, že kolej číslo 9 není plánovaná kolej, není u stejného nástupiště *II*, ale není od něj příliš daleko.

Kolej číslo 9 je velmi dobrou volbou. Naproti tomu je zde kolej číslo 1, kterou jednotně vybraly všechny metody.

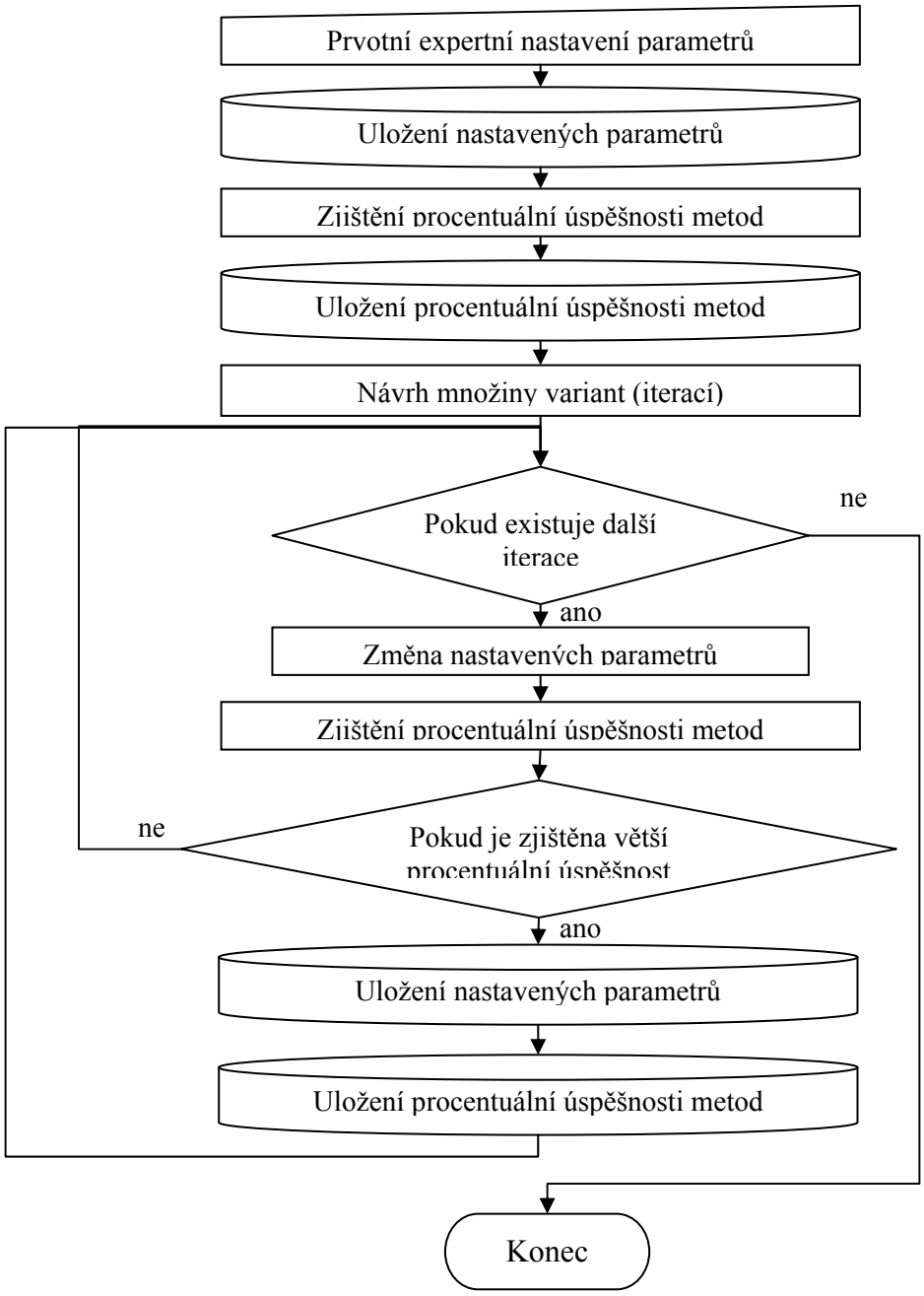
- Hodnota kritéria A=0,88 naznačuje, že v době příjezdu bude kolej obsazena. Leč za nedlouho bude uvolněna. Uvolněna bude přesně za 3 minuty.
- Hodnota kritéria B=0,9 vypovídá o tom, že z větší části doby pobytu ve stanici je kolej volná.
- Hodnota kritéria C=1 napovídá, že na koleji číslo 1 stojí přípojný vlak a bylo by možné uplatnit u přípojného vlaku kratší dobu na přestup.
- Hodnota kritéria D=0,9 značí, že kolej číslo 1 je u stejného nástupiště jako původně plánovaná kolej číslo 7.

Kolej číslo jedna je také dobrou variantou. Tato varianta sice nahrává přípojnému vlaku, jenže aby mohla být zrealizována, museli bychom vlaku, který tuto kolej blokuje přidělit jinou kolej. Touto změnou bychom ale mohli zapříčinit znevýhodnění pozice tohoto vlaku. Dalším řešením by bylo, aby vlak počkal 3 minuty před vjezdem do stanice. To by ale musel být vjezd patřičně přizpůsoben a navíc by narůstalo zpoždění, které patří k největším problémům železniční dopravy.

Z těchto důvodů byla expertem vybrána kolej číslo 9. Uvedenou logiku rozhodování je složité vyjádřit pomocí vah kritérií, což je slabina metod vícekriteriálního hodnocení. Jistě by ale bylo zajímavé takovou metodu navrhnout.

Stanovení vah kritérií (vstupních parametrů vícekriteriálních metod)

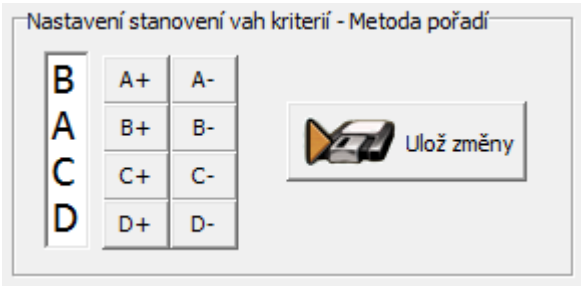
Dalším krokem je nastavení vstupních parametrů vícekriteriálních metod. Nejdůležitější je první expertní nastavení parametrů, který se dále iterativním způsobem optimalizuje. Celý optimalizační proces je popsán vývojovým diagramem na obrázku 52.



Obrázek 52 Vývojový diagram procesu optimalizace procentuální úspěšnosti

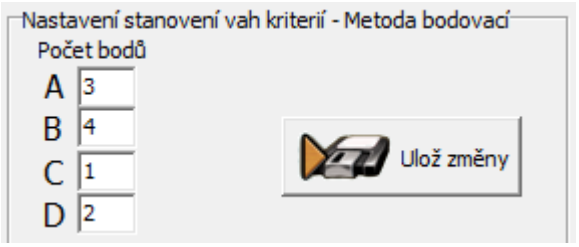
Po ukončení procesu optimalizace procentuální úspěšnosti již jsou k dispozici suboptimální vstupní parametry vícekriteriálních metod.

Jak byly tyto vstupní parametry resp. váhy jednotlivých kritérií nastaveny je znázorněno na následujících obrázcích 53 až 56.



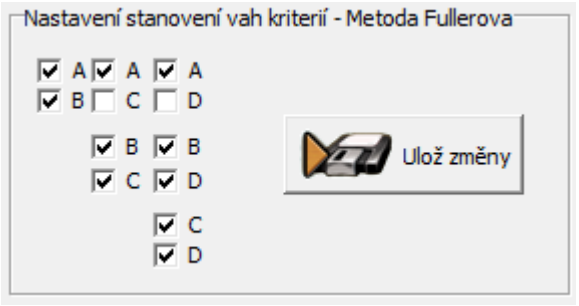
Obrázek 53 Stanovené váhy kritérií metody pořadí

Detailní popis metody pořadí najdete v kapitole 3.3.1. Jedná se o nejsnáze parametrizovatelnou metodu. Celkový počet všech možných variant je 24.



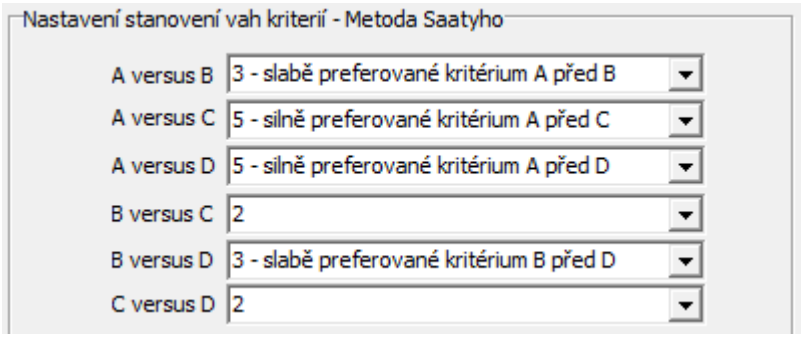
Obrázek 54 Stanovené váhy kritérií metody bodovací

Detailní popis metody bodovací najdete v kapitole 3.3.2. Tato metoda je již obtížněji parametrizovatelná. Pokud bychom si zvolili stupnici bodů od 1 do 4, pak by celkový počet všech možných variant je 256.



Obrázek 55 Stanovené váhy kritérií metody Fullerovy

Detailní popis metody Fullerovy najdete v kapitole 3.4.1. Celkový počet variant je 729.



Obrázek 56 Stanovené váhy kritérií metody Saatyho

Detailní popis metody Saatyho najdete v kapitole 3.4.2. Celkový počet všech možných variant je 531441.

6.3 Procentuální úspěšnost vícekriteriálních metod

Tato kapitola se věnuje porovnáním jednotlivých vícekriteriálních metod z hlediska procentuální úspěšnosti.

Obecně jsme si již naznačili v kapitole 5.7 jak metody porovnat. V kapitole 6.2.1 jsme si vybrali testovací množinu vlaků. Dále jsme si na této množině spočítali kriteriální matice. Provedli jsme optimalizaci procentuální úspěšnosti a nyní již můžeme uvést jak jednotlivé metody v testování obstály.

6.3.1 Porovnání vícekriteriálních metod – závěrečné vyhodnocení

Na posledním, tj. pátém místě se umístila metoda Entropická. Dosáhla procentuální úspěšnosti 84,52%.

Na čtvrtém místě se umístila metoda pořadí. Tato metoda dosáhla úspěšnosti 91,31%.

Třetí místo obsadila metoda Bodovací. Tato metoda dosáhla úspěšnosti 92,38%.

Druhé místo obsadila metoda Fullerova. Tato metoda dosáhla úspěšnosti 93,21%.

Na prvním místě se umístila metoda **Saatyho**. Tato metoda dosáhla úspěšnosti **95,24%**.

Tyto souhrnné statistické údaje najdete na přiloženém CD ve složce *Statistiky* v souboru *SouhrnneInformace.xls*.

Hodnoty procentuální úspěšnosti jednotlivých metod nejsou však nijak závazné. Pokud bychom zvolili jinou testovací množinu vlaků a varianty by posuzoval jiný expert dojdeme k jiným výsledkům.

Další věcí která stojí za povšimnutí je fakt, že umístění metod na žebříčku procentuální úspěšnosti je ve stejném pořadí, jako bychom je seřadili podle velikosti variability vah kritérií. Je tedy možné, že tato variabilita vah přímo souvisí s jejich úspěšnosti. Bylo by jistě zajímavé tuto domněnku ověřit, leč toto je již nad rámec diplomové práce.

7 Závěr

Tato práce si kladla za cíl vytvořit funkční aplikaci, která by mohla pomoci řídícím pracovníkům v železničních stanicích s výběrem náhradní nástupištní koleje pro zpožděný přijíždějící vlak. Tento cíl byl splněn a byla vytvořena aplikace VNNK (**V**ýpočet **N**áhradní **N**ástupištní **K**oleje).

Aby mohl být tento cíl splněn bylo třeba se hlouběji informovat o řídících procesech probíhajících na železničních stanicích. K tomuto účelu dobře posoužila publikace [2].

Dále bylo třeba prostudovat a osvojit si metodiku přidělování náhradní nástupištní koleje. Tato metodika je popsána v kapitole 2 a vychází z publikace [1].

Nedílnou součástí pro splnění cíle bylo nastudování vhodných metod pro vícekriteriální rozhodování. Tomuto účelu výborně posloužil zdroj [3], zejména pak část zabývající se metodami stanovení vah kritérií. Touto problematikou se zabývá kapitola 3.

Dále se práce zabývá tvorbou objektových modelů datových struktur pro aplikační rozhraní VNNK Interface. Zabývá se návrhem řešení, jak vykreslit obsazení plánu kolejí pomocí virtuálních vrstev. Tyto modely jsou popsány v kapitole 4.

V kapitole 5 je popsána aplikace VNNK pracující s aplikačním rozhraním VNNK Interface. Je zde popsáno, jak s aplikací pracovat a jak ji použít pro potřeby řídících pracovníků železničních stanic. Aplikace je naprogramována v prostředí Turbo Delphi 2006. Objektový model je navržen v Open Source modelovacím nástroji StarUML.

Kapitola 6 se zabývá případovou studií železniční stanice Praha hlavní nádraží. Aplikuje na tomto příkladu metodiku vyhledávání náhradní nástupištní koleje uplatněním aplikace VNNK. Dalším cílem této práce je otestovat metody vícekriteriálního hodnocení variant. Tyto metody byly testovány z hlediska procentuální úspěšnosti viz kapitola 6.3. Vítězem testování se stala Saatyho metoda s hodnotou 95,24%.

Pro další zvyšování úspěšnosti vícekriteriálních metod se nabízí další vhodná řešení. Jednak by bylo vhodné rozšířit množinu testovacích vlaků. Dále by bylo vhodné provézt expertní analýzu variant vícero zkušenými experty. Na konec by bylo možné prohledat všechny kombinace vah kritérií jednotlivých metod (u Saatyho metody je jich 531441).

Práce přinesla řešení automatizace podpory rozhodování o výběru z několika náhradních kolejí za pomocí aplikace VNNK. Toto řešení je vhodné zejména pro řídící pracovníky železničních stanic.

8 Soupis bibliografických citací

[1] BAŽANT, Michael, ŽARNAY, Michal. Formalizace řešení přidělení náhradní nástupištní koleje pro zpožděný vlak. In *Sborník příspěvků konference INFOTRANS 2005*. Pardubice : Univerzita Pardubice, 2005. s. 29-34. ISBN 80-7194-792-X.

[2] VONKA, Jaroslav, MOLKOVÁ, Tatiana, ŠIROKÝ, Jaromír. *Technologie a řízení dopravy 2. - GVD*. 1. vyd. Pardubice : Univerzita Pardubice, 2000. 112 s. ISBN 80-7194-286-3.

[3] KOS, Petr. *Vícekriteriální rozhodování* [online]. 2003 [cit. 2008-02-02]. Dostupný z WWW: <http://etext.czu.cz/php/skripta/skriptum.php?titul_key=79>.

[4] ELLER, Frank, BRÁZA, Jiří. *Delphi 6 : příručka programátora*. 1. vyd. Praha : Grada, 2002. 272 s. ISBN 80-247-0303-3.

[5] *Čekací doby a opatření při zpoždění vlaků osobní dopravy, platný od 12. 12. 2004*. Praha: Odbor 16, GŘ České dráhy a. s., 2004.

[6] ARLOW, Jim, NEUSTADT, Ila. *UML a unifikovaný proces vývoje aplikací : průvodce analýzou a návrhem objektově orientovaného softwaru*. 1. vyd. Brno : Computer Press, 2003. 387 s. ISBN 80-7226-947-X.

[7] *Seznam Vlaků pro staniční zaměstnance pro obvod Praha hl. n., platný od 12. 12. 2004*. Interní materiál. Praha : České dráhy a. s., 2004.

[8] *Seznam Vlaků pro staniční zaměstnance pro obvod Praha hl. n., platný od 12. 12. 2004. : DÍL II. - řadění vlaků*. Interní materiál. Praha : České dráhy a. s., 2004. 230 s.