

UNIVERZITA PARDUBICE
FAKULTA ELEKTROTECHNIKY A INFORMATIKY

DIPLOMOVÁ PRÁCE

2018

Ondřej Rada

Univerzita Pardubice
Fakulta elektrotechniky a informatiky

Návrh a implementace editoru železniční sítě

Ondřej Rada

Diplomová práce

2018

Univerzita Pardubice
Fakulta elektrotechniky a informatiky
Akademický rok: 2016/2017

ZADÁNÍ DIPLOMOVÉ PRÁCE

(PROJEKTU, UMĚLECKÉHO DÍLA, UMĚLECKÉHO VÝKONU)

Jméno a příjmení: **Bc. Ondřej Rada**
Osobní číslo: **I15231**
Studijní program: **N2646 Informační technologie**
Studijní obor: **Informační technologie**
Název tématu: **Návrh a implementace editoru železniční sítě**
Zadávající katedra: **Katedra softwarových technologií**

Z á s a d y p r o v y p r a c o v á n í :

V teoretické části práce bude proveden popis možností lokalizace kolejových vozidel vzhledem k infrastruktuře železniční sítě

V praktické části se diplomant zaměří na návrh a implementaci softwarového nástroje pro editaci železniční sítě. Student bude pro práci využívat především aktuální bázi dat, která je uložena v databázi Oracle Spatial.

Výsledný softwarový editor bude otestován nad reálnými daty třívrstvého modelu železniční sítě

Rozsah grafických prací:

Rozsah pracovní zprávy:

Forma zpracování diplomové práce: **tištěná**

Seznam odborné literatury:

***MURRAY, Chuck, et al. Oracle? Spatial : User?s Guide and Reference 10g Release 2 (10.2) [online]. [s.l.] : Oracle, 2006 Dostupné z WWW:**

<http://download.oracle.com/docs/cd/B19306_01/appdev.102/b14256.pdf>.

***MURRAY, Chuck, et al. Oracle? Spatial : Topology and Network Data Models 10g Release 2 (10.2) [online]. [s.l.] : Oracle, 2005. Dostupné z WWW:**

<http://download.oracle.com/docs/cd/B19306_01/appdev.102/b14256.pdf>.

Vedoucí diplomové práce:

Ing. Jan Fikejz, Ph.D.

Katedra softwarových technologií

Datum zadání diplomové práce: **31. října 2016**

Termín odevzdání diplomové práce: **17. května 2017**



Ing. Zdeněk Němec, Ph.D.
děkan

L.S.



prof. Ing. Antonín Kavička, Ph.D.
vedoucí katedry

V Pardubicích dne 15. listopadu 2016

Prohlášení autora

Prohlašuji, že jsem tuto práci vypracoval samostatně. Veškeré literární prameny a informace, které jsem v práci využil, jsou uvedeny v seznamu použité literatury.

Byl jsem seznámen s tím, že se na moji práci vztahují práva a povinnosti vyplývající ze zákona č. 121/2000 Sb., autorský zákon, zejména se skutečností, že Univerzita Pardubice má právo na uzavření licenční smlouvy o užití této práce jako školního díla podle § 60 odst. 1 autorského zákona, a s tím, že pokud dojde k užití této práce mnou nebo bude poskytnuta licence o užití jinému subjektu, je Univerzita Pardubice oprávněna ode mne požadovat přiměřený příspěvek na úhradu nákladů, které na vytvoření díla vynaložila, a to podle okolností až do jejich skutečné výše.

Souhlasím s prezenčním zpřístupněním své práce v Univerzitní knihovně.

V Pardubicích dne 20. 08. 2018

Ondřej Rada

PODĚKOVÁNÍ

Rád bych poděkoval panu Ing. Janu Fikejzovi Ph.D. za odborné vedení práce, ochotu, pomoc a poskytnuté rady a materiály, které mi pomohly tuto práci dokončit. Dále bych chtěl poděkovat rodině, přítelkyni a přátelům za podporu při mém studiu.

ANOTACE

Náplní této diplomové práce je návrh a implementace editoru železniční sítě. V teoretické části jsou popsána multidimenzionální data a datové struktury vhodné pro jejich uchování. Dále jsou popsány možnosti lokalizace kolejových vozidel.

Praktická část se zabývá implementací editoru železniční sítě, který využívá existující třívrstvý datový model uchovaný v databázi Oracle Spatial. K vizualizaci dat slouží nástroj MapViewer.

KLÍČOVÁ SLOVA

Lokalizace kolejových vozidel, editor železniční sítě, Oracle MapViwer, Oracle Spatial, multidimenzionální datové struktury

TITLE

Design and implementation of editor of the railway network

ANNOTATION

The content of this master's thesis is a design and implementation of editor of the railway network. The theoretical part describes multidimensional data and datastructures for their storing. In the next part are described possibilities of localization of rolling stock.

Practical part deals with implementation of a railway network editor using the existing three layers data model stored in Oracle Spatial database. MapViewer tool was intended for data visualization.

KEYWORDS

Localization of rolling stock, editor of the railway network, Oracle MapViwer, Oracle Spatial, multidimenzional datastructures

OBSAH

Úvod.....	15
1 Multidimenzionální bodová data	16
1.1 Datové struktury pro uchování dat.....	18
1.1.1 Range strom	19
1.1.2 Prioritní vyhledávací strom.....	20
1.1.3 K-d strom	22
1.1.4 Quadstrom.....	24
1.2 Databázové systémy pro uchování multidimenzionálních dat.....	27
1.2.1 MySQL	27
1.2.2 PostgreSQL.....	27
1.2.3 MS-SQL.....	27
1.2.4 Oracle.....	28
2 Oracle Spatial.....	29
2.1 Objektově-relační model.....	29
2.2 Datový model.....	30
2.3 Query model.....	31
2.4 Indexování prostorových dat.....	31
2.5 Prostorový datový typ	33
2.5.1 SDO_GTYPE.....	33
2.5.2 SDO_SRID	34
2.5.3 SDO_POINT.....	34
2.5.4 SDO_ELEM_INFO	35
2.5.5 SDO_ORDINATES.....	35
2.6 Metadata geometrie.....	35
2.7 Oracle Fusion Middleware MapViewer.....	36
2.7.1 Základní tok činností aplikace MapViewer	37

2.7.2	Architektura MapVieweru	37
2.7.3	Renderovací engine.....	38
2.7.4	Sada API	38
2.7.5	Map Builder	38
2.7.6	Oracle Maps	38
3	Infrastruktura železniční sítě.....	39
3.1	Systémy pro lokalizaci kolejových vozidel.....	39
3.1.1	Lokalizace vozidel bez použití satelitního navigačního systému	40
3.1.2	Lokalizace s použitím GNSS	41
3.1.3	Lokalizace s použitím GNSS a podpůrných systémů	42
3.2	Reprezentace železniční sítě v paměti počítače	42
3.2.1	Abstraktní datový typ graf	42
3.3	Topologie podle standardu M12	43
3.3.1	Železniční dopravní cesta	43
3.3.2	Železniční těleso	43
3.3.3	Trat'	43
3.3.4	Staničení.....	43
3.3.5	Staničník	43
3.3.6	Definiční úsek (DÚ).....	44
3.3.7	Trat'ový definiční úsek (TUDU).....	44
4	Implementace editoru železniční sítě.....	45
4.1	Výchozí bod a představení dat	45
4.2	Instalace databáze.....	47
4.3	Konfigurace databáze	48
4.4	Import dat do databáze	50
4.5	Databázové tabulky	52
4.5.1	Tabulka <i>RAILWAY_ABS_NODE\$</i>	52

4.5.2	Tabulka <i>RAILWAY_MEZO_NODE\$</i>	53
4.5.3	Tabulka <i>RAILWAY_ABS_LINK\$</i>	53
4.5.4	Tabulka <i>RAILWAY_MEZO_LINK\$</i>	54
4.5.5	Tabulka <i>STATION_N</i>	54
4.5.6	Bindovací tabulky	55
4.6	MapBuilder	55
4.7	Knihovny potřebné pro implementaci	56
4.8	Propojení aplikace s databází	57
4.9	Datová struktura graf	57
4.10	MapView	58
4.10.1	Inicializace MapVieweru	59
4.10.2	Nastavení parametrů žádosti MapVieweru	60
4.10.3	Vkládání vlastností	60
4.10.4	Odeslání žádosti na server	61
4.10.5	Získání odpovědi ze serveru	61
4.10.6	Kreslení na mapu	61
4.11	Celková koncepce řešení	62
4.12	Struktura projektu aplikace	63
4.13	Funkce editoru	67
4.13.1	Hrany	69
4.13.2	Makro vrstva	69
4.13.3	Mezo vrstva	69
4.13.4	Mikro vrstva	70
4.13.5	Stanice	70
4.13.6	Validace	70
4.13.7	Cache paměť	71
4.14	Zakomponování Quadstromu	71

4.15	Problém duplicitních souřadnic	72
4.15.1	První duplicita	73
4.15.2	Druhá duplicita	74
4.15.3	Třetí a čtvrtá duplicita	76
	Závěr	77
	Použitá literatura	78
	Přílohy	81

SEZNAM ILUSTRACÍ A TABULEK

Obrázek 1: Zobrazení ilustračních dat v grafu.....	18
Obrázek 2: Range strom (1 dimenze).	19
Obrázek 3: Range strom (2 dimenze).	20
Obrázek 4: Prioritní vyhledávací strom.	21
Obrázek 5: Prioritní vyhledávací strom – dekompozice prostoru.	22
Obrázek 6: 2-d strom.	23
Obrázek 7: 2-d strom – dekompozice prostoru.....	23
Obrázek 8: Quadstrom.	26
Obrázek 9: Dělení prostoru quadstromu.	26
Obrázek 10: Typy dvourozměrných geometrií.	30
Obrázek 11: Query model.....	31
Obrázek 12: Aproximace geometrie.	32
Obrázek 13: Dekompozice prostoru vrstvy geometrií a R-strom.	32
Obrázek 14: Princip činnosti MapVieweru.	37
Obrázek 15: Architektura MapVieweru.	37
Obrázek 16: ETCS Level 1.....	40
Obrázek 17: ETCS Level 2.....	41
Obrázek 18: ETCS Level 3.....	41
Obrázek 19: Koncepce třívrstvého modelu železniční sítě.....	45
Obrázek 20: Sql Developer – databázové připojení.	50
Obrázek 21: Tabulka <i>RAILWAY_ABS_NODE\$</i>	52
Obrázek 22: Tabulka <i>RAILWAY_MEZO_NODE\$</i>	53
Obrázek 23: Tabulka <i>RAILWAY_ABS_LINK\$</i>	53
Obrázek 24: Tabulka <i>RAILWAY_MEZO_LINK\$</i>	54
Obrázek 25: Tabulka <i>STATION_N</i>	54
Obrázek 26: Tabulka <i>SUPERNODE_BINDING</i>	55
Obrázek 27: Tabulka <i>STATION_BINDING</i>	55
Obrázek 28: MapBuilder.	56
Obrázek 29: Ukázka vizualizovaných dat.	60
Obrázek 30: Celková koncepce řešení.....	62
Obrázek 31: Struktura projektu aplikace.	63
Obrázek 32: Balíček nodes.	64

Obrázek 33: Balíček links.....	64
Obrázek 34: Balíček graph.	65
Obrázek 35: Balíček search.	66
Obrázek 36: Balíček dao.....	66
Obrázek 37: Přesun do mezo vrstvy.	68
Obrázek 38: Porovnání datových struktur.	72
Obrázek 39: První duplicita.	73
Obrázek 40: Stav po odebrání první duplicity.	74
Obrázek 41: Druhá duplicita.....	75
Obrázek 42: Hrany k přesunu.	75
Obrázek 43: Situace po přesunu.	75
Obrázek 44: Třetí duplicita.	76
Obrázek 45: MapViewer není spuštěn.....	82
Obrázek 46: Uživatelské rozhraní.....	82
Obrázek 47: Kontextové menu.	83
Obrázek 48: Přesun existující stanice do mezo vrstvy.	83
Obrázek 49: Odebrání prvku z makro vrstvy.....	83
Tabulka 1: Dimenze ne-bodových dat.	16
Tabulka 2: Multidimenzionální data.....	17
Tabulka 3: Výčet ilustračních dat pro následující datové struktury.	18
Tabulka 4: Kvadranty korespondující s binárními hodnotami.	25
Tabulka 5: Hodnoty datového typu <i>SDO_GTYPE</i>	34
Tabulka 6: Základní charakteristika železniční sítě.....	39
Tabulka 7: Barevné zobrazení vrstev.....	68
Tabulka 8: Duplicitní prvky.....	73

SEZNAM ZKRATEK A ZNAČEK

ADT	Abstraktní datový typ
API	Application Programming Interface
CDGPS	Canadian Differencial Global Possitioning System
ČR	Česká republika
DDL	Data Definiton Language
DML	Data Manipulation Language
DÚ	Definiční úsek
FOI	Features of Interesting
GNSS	Global Navigation Satellite System
GPS	Global Possitioning System
GSM-R	Global System for Mobile Communications – Railway
JDBC	Java Database Connectivity
LEU	Lineside Electronic Unit
MBR	minimum bounding rectangle
MX	matrix
ORDBMS	Object-Relational Database Management System
PM	polygonální mapa
PR	point-region
RBC	Radio Base Station
RDBMS	Relational Database Management System
SDO	Spatial Data Option
SID	System ID
TUDU	Trat'ový definiční úsek
URL	Uniform Resource Locator

Úvod

Primárním cílem diplomové práce je návrh a implementace editoru železniční sítě, jakožto podpůrného softwarového nástroje pracujícího nad již existujícím třívrstevným modelem. Model železniční infrastruktury je reprezentován jako neorientovaný graf. Báze dat je uložena v databázi Oracle s nadstavbou Oracle Spatial, která umožňuje uchování multidimenzionálních dat.

První kapitola práce popisuje multidimenzionální data a možnosti jejich uchování. Jsou zde popsány datové struktury, které slouží pro uchování dat v paměti počítače. Pozornost je dále věnována databázovým systémům, které již podporu pro multidimenzionální data obsahují.

Jedním z těchto systémů je již výše zmíněný Oracle Spatial, jehož popisem se podrobně zabývá druhá kapitola této práce, kde je popsán Oracle Spatial jako celek, od objektově-relačního modelu přes model datový. Dále způsob indexování a datový typ až po metadata. Poslední část kapitoly je věnována Oracle Fusion Middleware a to konkrétně produktu MapViewer, jenž je použit jako nástroj k vizualizaci dat třívrstevného modelu sítě.

Ve třetí kapitole je popsána infrastruktura železniční sítě zejména z pohledu, jakým lze reprezentovat model v paměti počítače. Kapitola se dále věnuje předpisu ČD M12, jehož základní znalost je důležitá pro pochopení některých parametrů dat, ze kterých práce vychází. Poslední část kapitoly se věnuje lokalizaci kolejových vozidel.

Zavěrečná kapitola diplomové práce se týká praktické části. Je zde popsáno jakým způsobem se postupovalo při tvorbě editoru. Dále jsou podrobně představena data třívrstevného modelu. V kapitole je rovněž uveden postup instalace a konfigurace databáze a následný import dat. Následuje popis návrhu tříd editoru a výběr vhodné datové struktury na základě informací uvedených v první kapitole. Další nedílnou součástí kapitoly je představení nástroje MapBuilder pro tvorbu vizuálních schémat. Také je zde popsána instalace a inicializace nástroje MapViewer a jeho následné zakomponování do aplikace. Poslední částí kapitoly je popis funkcí editoru a dále popis problémů, které nastaly během implementace.

Přílohou práce je uživatelská příručka, která by měla usnadnit ovládání programu.

1 MULTIDIMENZIONÁLNÍ BODOVÁ DATA

Z obecného hlediska lze pojem multidimenzionální chápat jako pevný vztah dvou i více hodnot, které spolu úzce souvisí a jedna bez druhé by ztrácela význam. V informatice se jako příklad nabízí rozlišení obrazovky udávané v pixelech (šířka x výška). V elektrotechnice by to mohl být signál sinusového průběhu, kde lze v daném čase zjistit souřadnice x a y . V 3D počítačové grafice zase grafický objekt o souřadnicích x , y , z .

Ve všech výše uvedených případech se jedná vždy o nějakou lokaci (bod) v 2D nebo 3D prostoru, tedy **multidimenzionální bodová data**. Mohlo by tedy nesprávně vyplynout, že pojem multidimenzionální zahrnuje pouze číselné hodnoty. Následující tabulka relační databáze obsahující ilustrační data ukazuje, že tomu tak vždy není (Tabulka 1).

Tabulka 1: Dimenze ne-bodových dat.

Id	Model	Typ	Barva	Rok výroby
1	Škoda	Octavia	Modrá	2005
2	Audi	A3	Červená	2011
3	Fiat	Punto	Černá	1998

Zdroj: vlastní

Každý sloupec tabulky o třech záznamech představuje jednu dimenzi. Každý záznam má tedy čtyři dimenze. Informace uložené v dimenzích samy o sobě postrádají význam, pouze jako celek dávají smysl. Každý záznam v tabulce má kromě jednotlivých dimenzí také jednoznačný identifikátor (sloupec *Id*) neboli **klíč**. [1]

Diplomová práce se zabývá dvourozměrnými multidimenzionálními daty a operací s nimi v rámci desktopové aplikace, která spolupracuje s multidimenzionální databází. Data jsou představena až v Import dat do databáze. Nicméně již nyní je třeba říci, že klíčem jsou geografické souřadnice, přesněji obě složky polohy – longitude a latitude¹. Jedná se o **multidimenzionální klíč**. Vzhledem ke dvěma složkám lze hovořit o **dvousložkovém multidimenzionálním klíči**. Souřadnice jsou stejného datového typu a mají stejné jednotky. To je důležité za prvé z hlediska hledání nejbližšího souseda. Za druhé z důvodu stanovení intervalu, ve kterém je možno vyhledávat. V tomto vymezeném prostoru lze zjistit počet objektů v něm umístěných, nebo případně do tohoto prostoru pouze zasahujících. Tím se dostáváme k tomu, že prostorová data nejsou pouze body, ale mohou to být také různé objekty. Prostorové 2D objekty jsou obecně křivky a polygony. Křivky mohou představovat například

¹ longitude – představuje zeměpisnou délku, latitude – představuje zeměpisnou šířku.

silnice, řeky nebo v případě diplomové práce železniční trať. Polygony je nejjednodušší si představit jako kraje v rámci mapy ČR. Mapa ČR je také polygonem v rámci mapy Evropy. Dále to mohou být budovy, vodní plochy atd.

Příklad vybraných dat z výchozí databáze pro tuto práci (Tabulka 2):

Tabulka 2: Multidimenzionální data.

Stanice	Longitude [°]	Latitude [°]
Hradec Králové hlavní nádraží	50.21654807	15.81039444
Pardubice hlavní nádraží	50.03165545	15.75957563
Praha hlavní nádraží	50.08166278	14.43564585
Náchod	50.41828495	16.17585706
Jaroměř	50.34186654	15.91105183

Zdroj: vlastní

Multidimenzionální data lze uchovávat buď v paměti počítače (RAM nebo pevný disk), nebo v multidimenzionální databázi. V prvním případě je třeba zvolit vhodnou datovou strukturu pro efektivní uložení a rychlý přístup. [1] a [2]

Při výběru datové struktury pro uchování dat v počítači je třeba brát ohledy hlavně na:

- 1) Druh ukládaných dat – zejména, zda půjde o spojitě, nebo nespojitě hodnoty.
- 2) Velikost ukládaných dat – počet záznamů, které bude potřeba uchovávat.
- 3) Operace s daty – jaké operace s daty se budou provádět.
- 4) Časová náročnost – v jakých časových intervalech se budou data měnit.
- 5) Paměťová náročnost – kde budou data uložena, jestli v RAM nebo na pevném disku.

Vše záleží na tom, k jakému účelu bude aplikace použita. Například zda bude sloužit pouze jako nástroj vizualizační, nebo také k editaci, vkládání a mazání dat. [1] a [2]

V první podkapitole jsou nejprve popsány vhodné datové struktury pro uchování multidimenzionálních dat. V druhé podkapitole jsou představeny databázové systémy pro multidimenzionální data. Jedná se o komerční i free produkty.

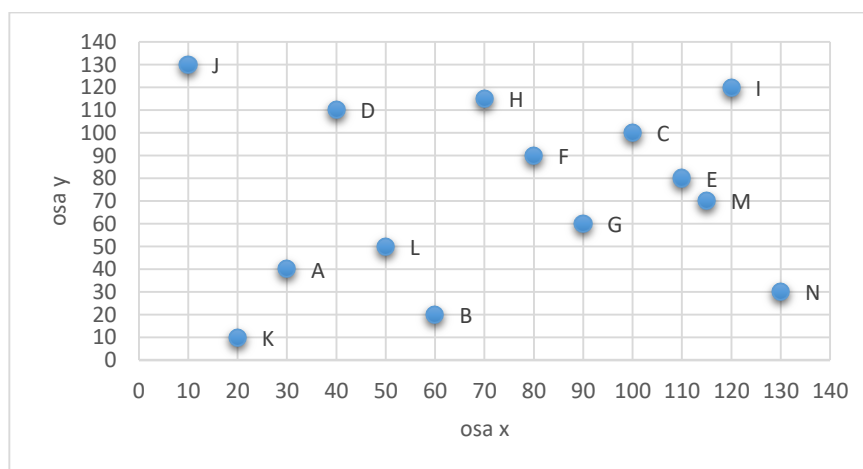
1.1 Datové struktury pro uchování dat

Jako výchozí případ pro následující vybrané datové struktury jsou použita ilustrační data (body) znázorněná v grafu. Z důvodu jednoduchosti a pro snadnější pochopení zatím nejsou uvažována reálná data. Pro přehlednost je nejprve uvedena tabulka se souřadnicemi daných bodů a následně graf, který byl z těchto hodnot vytvořen. Je zde uvedeno více bodů pro případné rozšíření práce o složitější datové struktury, kde by mohlo být vyžadováno více dat. Aktuálně používaná data jsou zvýrazněna světle modrou barvou (Tabulka 3) a je z nich vytvořen graf (Obrázek 1).

Tabulka 3: Výčet ilustračních dat pro následující datové struktury.

Název	X	Y
A	30	40
B	60	20
C	100	100
D	40	110
E	110	80
F	80	90
G	90	60
H	70	115
I	120	120
J	10	130
K	20	10
L	50	50
M	115	70
N	130	30

Zdroj: vlastní



Obrázek 1: Zobrazení ilustračních dat v grafu. *Zdroj: vlastní*

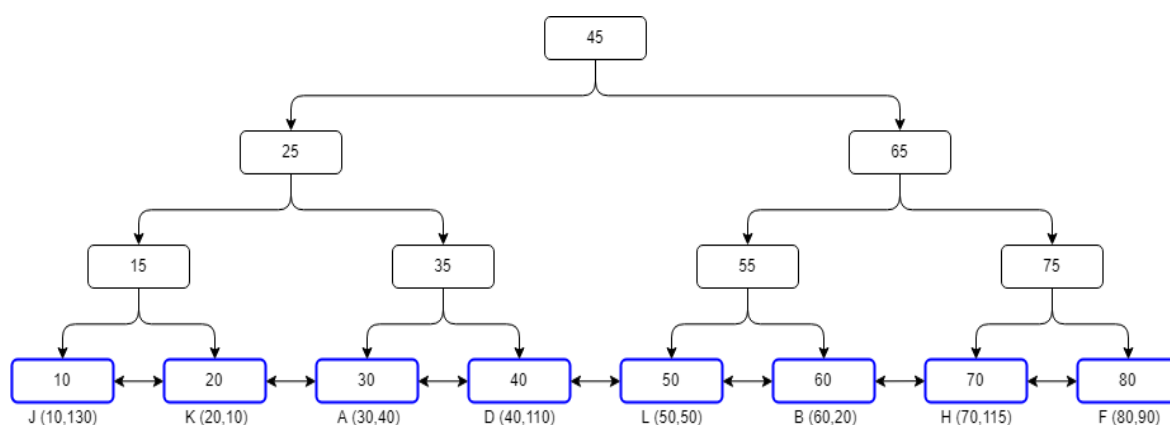
V této podkapitole bylo čerpáno převážně z [1], kde jsou také uvedeny pseudokódy k případným dalším studiím.

1.1.1 Range strom

Range strom je vyhledávací datová struktura, jejíž asymptotické složitosti základních operací jsou rychlejší než u datových struktur quadstrom nebo K-d strom, které jsou představeny dále, nicméně má však výrazně vyšší nároky na paměť. Struktura je určena převážně k rychlému vyhledávání bodů patřících do daného intervalu. Strom může mít jednu až k dimenzí.

1 - dimenzionální range strom

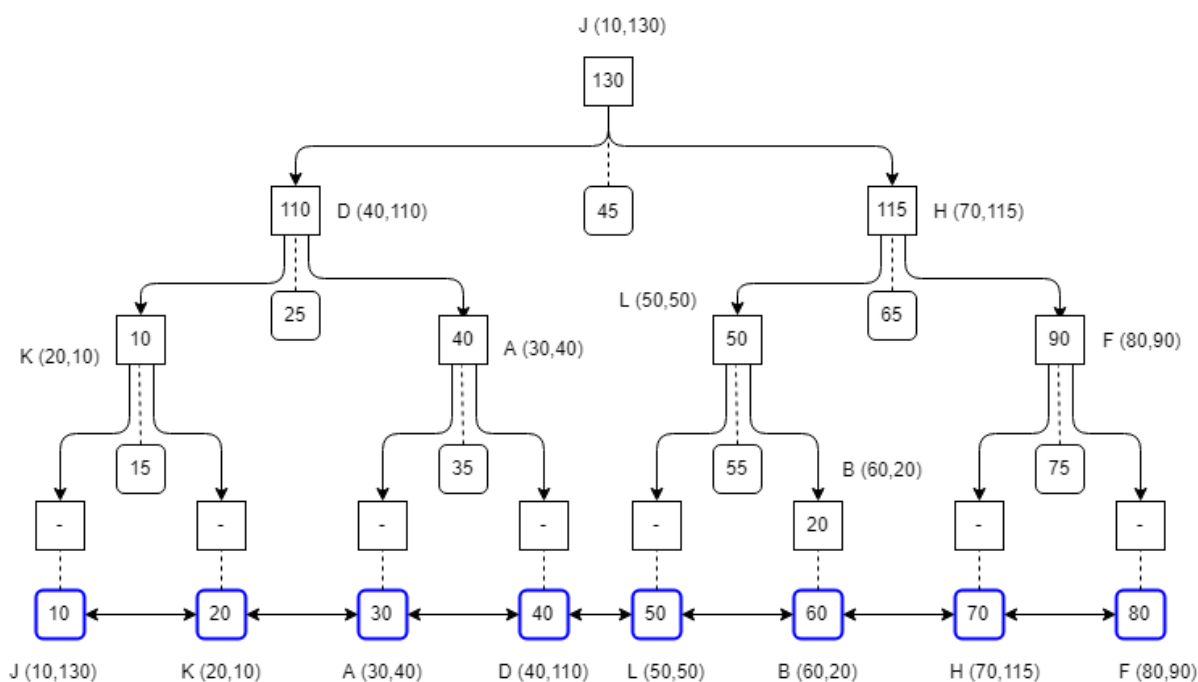
Jednorozměrný range strom je nezbytný k pochopení dvou a vícerozměrného range stromu. Jedná se o vyvážený binární vyhledávací strom (příkladem může být AVL strom), kde jsou data umístěna pouze v listech. Tyto listy jsou seřazeny podle klíče (u této varianty má klíč jen jednu složku, byla použita x souřadnice) a propojeny za pomoci obousměrného zřetěženého lineárního seznamu (na obrázcích zvýrazněno modře). Zbývající interní prvky (ne-listy) stromu slouží k navigaci ve struktuře. Navigační vrchol obsahuje střed intervalu, který dělí interval na levý a pravý podstrom. Středem intervalu se rozumí medián všech hodnot daného podstromu. Modifikací také bývá to, že místo středu intervalu jsou použity hranice intervalu, což urychluje vyhledávání daného prvku – není třeba provádět vyhledávání, je-li prvek mimo hranice. Tato modifikace také urychluje intervalové vyhledávání – je-li celý interval prvků uvnitř aktuálního intervalu, pak celý podstrom leží uvnitř. Obrázek znázorňuje range strom o jedné dimenzi (Obrázek 2).



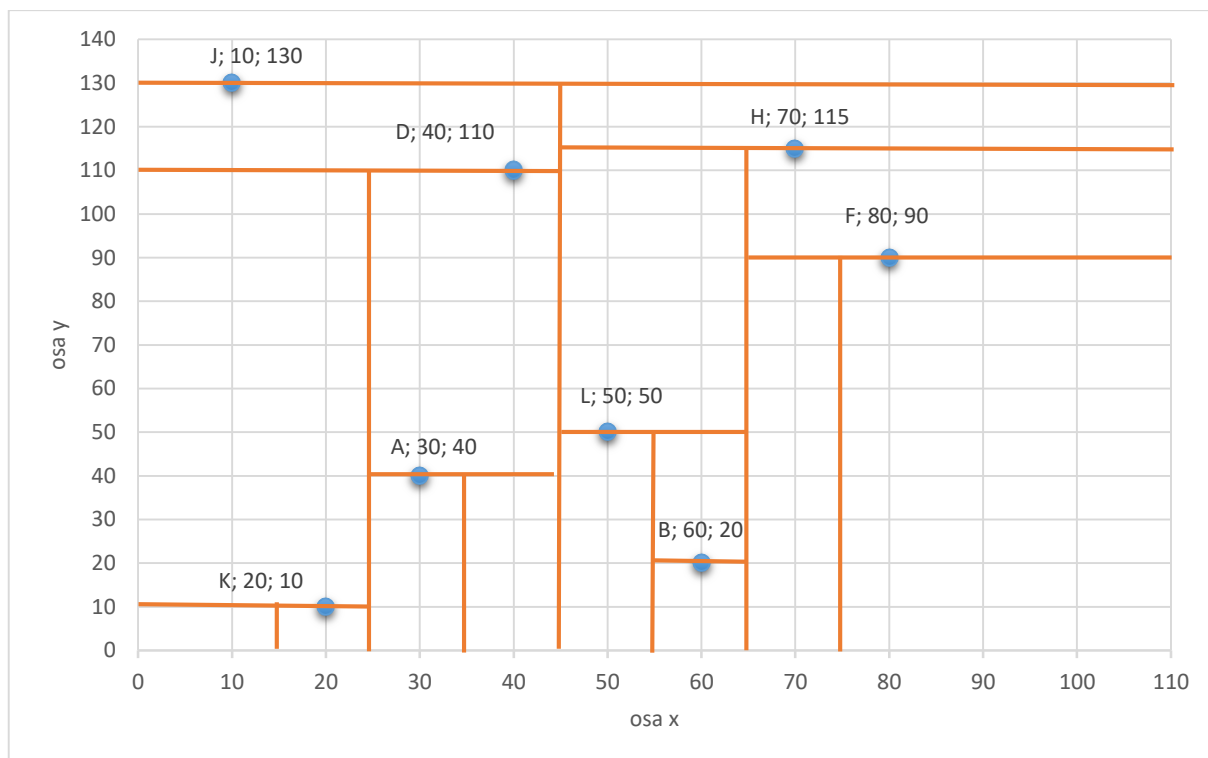
Obrázek 2: Range strom (1 dimenze). Zdroj: vlastní

k navigaci. Z této definice lze odvodit, že jako stromovou strukturu můžeme použít například výše zmíněný range strom. Lze zvolit jedno nebo více dimenzionální variantu. Můžeme také stanovit způsob, jakým bude řazena priorita (dle nejvyšší nebo nejnižší hodnoty souřadnice). Je možné místo range stromu zvolit i složitější strukturu, která je náročnější na implementaci. Může se jednat o red-black strom, o který by šla tato práce případně rozšířit.

Pro ilustraci je zde zvolen range strom o jedné dimenzi. Prvky znázorněné oblými čtverci představují navigační strukturu range stromu – v řetěženém lineárním seznamu jsou seřazeny dle souřadnice x . Tento seznam je zvýrazněn modrou barvou. Druhou částí je halda, jejíž prvky jsou znázorněny ostrými čtverci, souřadnice y jsou seřazeny podle priority (od největšího). Postupně od kořene je vrcholu přiřazena maximální y hodnota z jeho podstromu, která nebyla použita v předchozích úrovních. Pokud taková hodnota neexistuje, vrchol zůstane prázdný (značeno pomlčkou). Pro pochopení, jakým způsobem funguje navigační struktura, jsou vedle těchto obdélníků zobrazeny názvy bodů z tabulky (Tabulka 3). Je to pouze pro názornost, data jsou uložena v listech (Obrázek 4). Dále je zobrazena dekompozice prostoru (Obrázek 5).



Obrázek 4: Prioritní vyhledávací strom. Zdroj: vlastní



Obrázek 5: Prioritní vyhledávací strom – dekompozice prostoru. Zdroj: vlastní

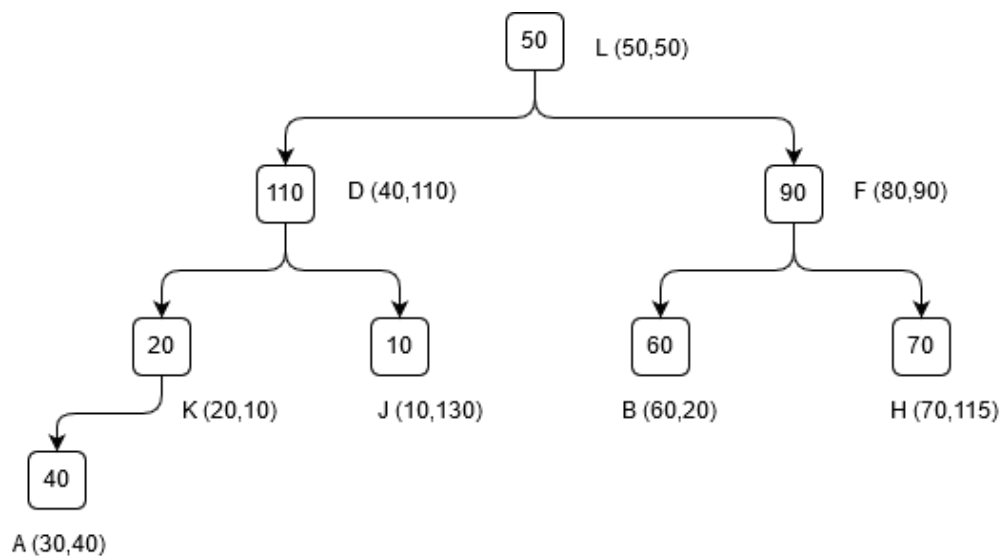
Jako další možnost může být použit inverzní prioritní vyhledávací strom. Jedná se range strom o jedné dimenzi, kde je priorita haldy řazena opačným způsobem než u předchozího případu. Kombinací range stromu o dvou dimenzích a právě inverzního prioritního vyhledávacího stromu dostaneme prioritní vyhledávací range strom.

1.1.3 K-d strom

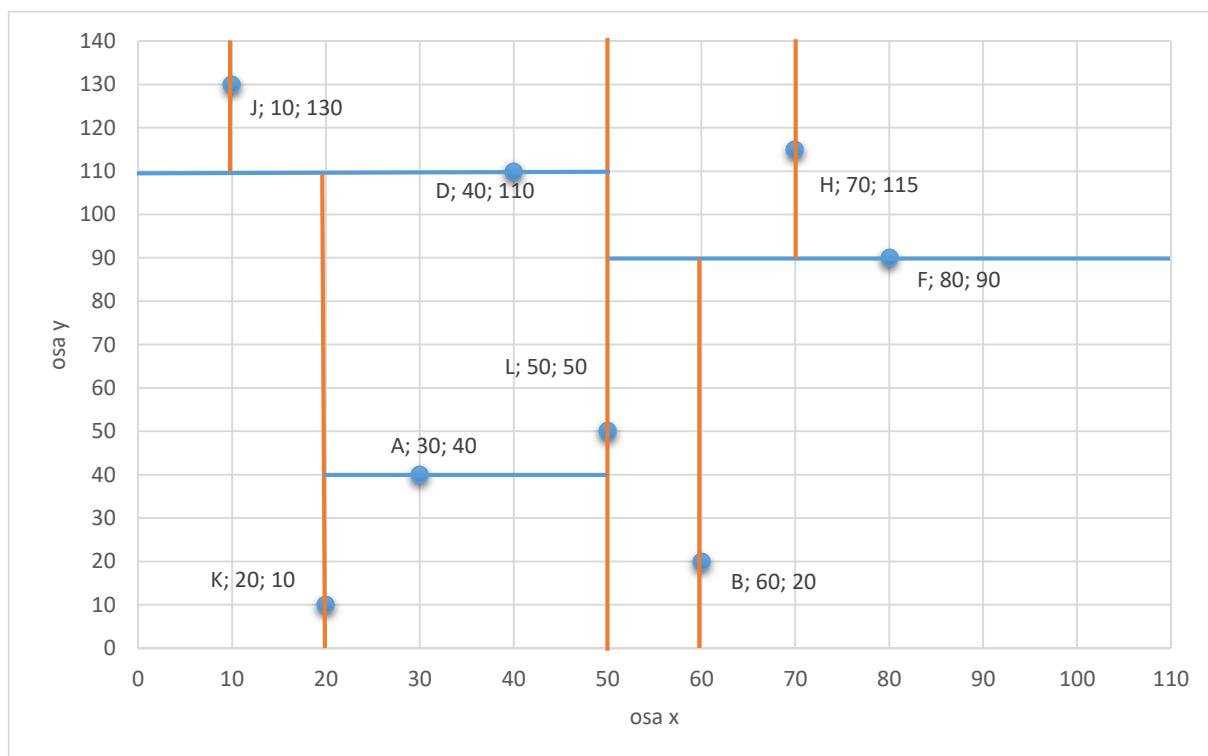
K-d stromy jsou ve své podstatě vyvážené binární stromy. Obecně platí, že písmeno k značí počet dimenzí stromu, ve kterém každý uzel tohoto stromu představuje k -rozměrný bod. Existuje celá řada typů K-d stromů. Od jednoduché bodové varianty K-d stromu až po složitější založené na Trie, jako jsou například PR, Bucket PR, PMR K-d stromy. V této části kapitoly je představena pouze základní varianta. Jedná se o bodový dvourozměrný strom, který je v literatuře označován jako 2-d strom.

Princip dělení prostoru spočívá ve střídání souřadnic pro každou úroveň stromu. Pro zaručení vyváženosti stromu je potřeba hodnoty, ze kterých má být strom budován, seřadit podle jedné složky klíče. Je stanovena souřadnice x . Ze seřazených hodnot je určen medián a daný prvek vložen jako kořen stromu. Tento prvek rozdělí stávající data do dvou podstromů. Dojde-li k posunu na druhou úroveň, jsou data seřazena tentokrát podle druhé složky klíče (v našem případě y) v každém podstromu. Opět se z hodnot podstromu určí medián a dochází k dalšímu

větvení. Tento postup, kdy se střídají souřadnice, se neustále rekurzivně opakuje do doby, než je vybudován celý strom a nezbyvá žádný další prvek. Tímto způsobem vznikne vyvážený strom, který je znázorněn na obrázku (Obrázek 6). Na dalším obrázku je zobrazeno dělení prostoru (Obrázek 7).



Obrázek 6: 2-d strom. Zdroj: vlastní



Obrázek 7: 2-d strom – dekompozice prostoru. Zdroj: vlastní

1.1.4 Quadstrom

U této datové struktury jsou kromě základního popisu popsány i jednotlivé operace. A to z důvodu, že tato datová struktura byla použita jako stavební kámen praktické části úlohy. Je tedy vhodné právě tuto strukturu představit ve větším rozsahu.

Stejně jako u 2-d stromu existuje několik variant této struktury. Základní variantou je bodový quadstrom. Dále můžeme narazit na regionální variantu quadstromu nebo quadstromy s označením MX. Existují také PM quadstromy, které vychází z MX a slouží pro uchovávání polygonálních map. Ty mají též několik variant, např. PM1 až PM4 a mnohé další. V této podkapitole je představena bodová a pseudo varianta. Dále podrobně popsána varianta regionální, označována jako PR quadstrom.

Datová struktura quadstrom je založena na rekurzivním dělení prostoru na čtyři kvadranty. Každý vrchol stromu obsahuje referenci na jeho čtyři potomky. Ty reprezentují právě zmíněné čtyři kvadranty, které jsou označovány jako severovýchodní, severozápadní, jihovýchodní a jihozápadní.

Bodový quadstrom

V případě, že má být vybudován strom o předem známých datech, jsou data seřazena primárně podle souřadnice x a sekundárně podle souřadnice y . Je vybrán medián a vložen jako kořen stromu. Zbytek dat je rozdělen podle souřadnic do kvadrantů a postup je rekurzivně opakován. Tato varianta je vhodná pro rychlé vyhledávání. Nevýhodou je nevyváženost struktury při předem neznámých datech a složitost při odebírání prvku, který není listem.

Pseudo quadstrom

Pseudo quadstrom je modifikace bodového quadstromu, která spočívá v tom, že data jsou umístěna pouze v listech. To velice zjednodušuje odebírání prvků. Nicméně zůstává nevyváženost struktury při předem neznámých datech. Princip budování je stejný s tím rozdílem, že pokud už v listu existují data, je tento list dále rozdělen a stává se navigačním vrcholem. Data se přesunou do jeho nově vytvořených listů.

Regionální quadstrom

Jak již bylo výše řečeno, tato varianta je použita v praktické části práce. Její výhodou je, že nemusí být předem známa data. Pokud by známa byla, tak jejich případné seřazení nebude mít na následné vybudování vliv. Tímto odpadá řazení dat. Další výhodou je, že data jsou uložena

pouze v listech, takže není složité prvky odebírat. Při vkládání prvek traverzuje stromem, dokud nedorazí na místo vložení. Pokud zde nejsou žádná data, prvek je úspěšně vložen. Pokud zde již data jsou, musí dojít k rozdělení na čtyři stejné oblasti (kvadranty). Dělení probíhá do té doby, dokud každý z kvadrantů neobsahuje pouze jeden prvek. V případě dělení na stejné kvadranty je potřeba dopředu znát definovanou oblast a její hranice. U obrazovky to mohou být souřadnice bodů $[0, 0]$ $[0, \text{šířka}]$ $[0, \text{výška}]$ a $[\text{šířka}, \text{výška}]$. U mapy zase nejjižnější, nejsevernější, nejzápadnější a nejvýchodnější bod.

Při použití rekurzivního dělení se v každé iteraci půlí souřadnice dané oblasti. Při rekurzivním volání funkce uvnitř sebe sama jsou jako parametry použity nové souřadnice, tedy polovina předchozího kvadrantu.

Druhou možností je využít principu znakového stromu, kdy jsou souřadnice oblasti převedeny na binární formu. To probíhá tím způsobem, že x souřadnice je vydělena maximální hodnotou x dané oblasti. Tento výsledek je předveden na binární kód. To samé je provedeno se souřadnicí y . Z následujícího příkladu by mělo být patrné, jak to celé funguje.

Př. Mějme bod $Z = [25, 55]$, maximální hodnota $x_{\max} = 50$ a maximální hodnota $y_{\max} = 70$. Bod Z má po převodu souřadnice $Z = [0, \mathbf{10010010} \dots, 0, \mathbf{1000000000}]$.

Každá dvojice binárních čísel odpovídá hodnotě kvadrantu, viz (Tabulka 4). Postupné traverzování stromem by probíhalo postupně po souřadnicích. Nejprve souřadnice **11** což odpovídá JV kvadrantu. Dále **10** což je SV kvadrant. Následuje souřadnice **00**, která odpovídá SZ kvadrantu atd. Tento způsob dělení je použit v praktické části.

Tabulka 4: Kvadranty korespondující s binárními hodnotami.

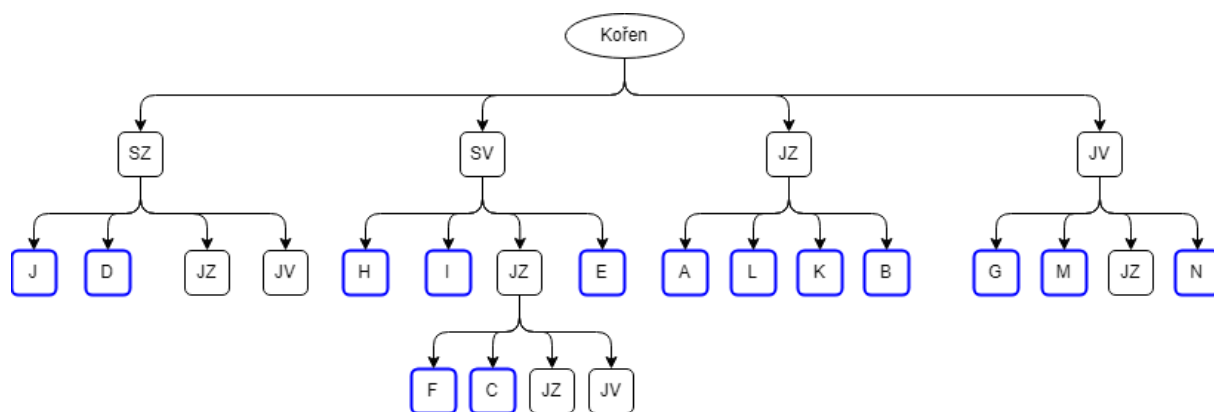
SZ	SV
JZ	JV

00	10
01	11

Zdroj: vlastní

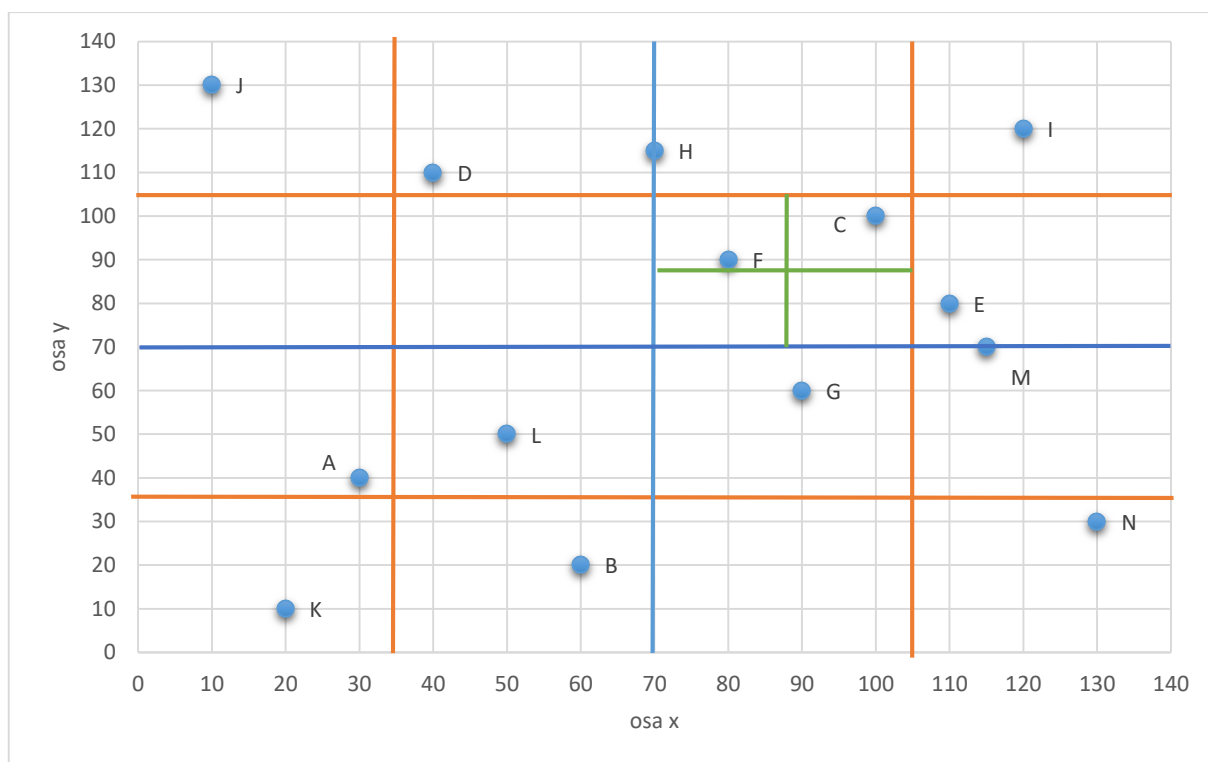
Tohoto principu se využívá jak při vyhledávání prvků, tak při jejich vkládání i odebírání. Při vyhledávání je postupně traverzováno na danou pozici, a jestliže prvek existuje, je vrácen. Při vkládání se prvek po traverzování buď na pozici vloží, nebo je dále rekurzivně děleno, jak bylo popsáno výše. V případě odebírání je traverzováno stromem a po dokončení je prvek odebrán. Jestliže již v jeho listech nejsou žádná data, je odebrán i rodič tohoto prvku.

Regionální quadstrom viz (Obrázek 8) a dělení prostoru na obrázku následujícím (Obrázek 9).



Obrázek 8: Quadstrom. Zdroj: vlastní

Leží-li bod přímo na vertikálním dělení prostoru, je přesunut doprava. V případě bodu *H* je to *SV* kvadrant při prvním dělení prostoru (modrá barva). Za situace, že bod leží na horizontálním dělení kvadrantu, je přesunut do jeho dolní poloviny. V případě bodu *M* je počítáno, že patří do *JV* kvadrantu při prvním dělení prostoru.



Obrázek 9: Dělení prostoru quadstromu. Zdroj: vlastní

1.2 Databázové systémy pro uchování multidimenzionálních dat

Moderní databázové systémy již obsahují podporu pro prostorová data a jsou hojně využívány pro práci s nimi. Databáze nejčastěji slouží k uchování geografických dat a složitých polygonů. Lze samozřejmě ukládat i jednoduché primitivní elementy jako body a úsečky. Každý z těchto systémů využívá různé datové struktury, ale mají společný cíl. Tímto cílem je kromě uchování dat i efektivní práce s nimi. Může se jednat o optimalizované vyhledávání pomocí prostorových indexů, různou manipulaci, určování nejbližších sousedů vrcholů, průniky přímek a mnoho dalších funkcí.

1.2.1 MySQL

MySQL je jeden z nejoblíbenějších nekomerčních databázových produktů. Pro prostorová data realizuje rozšíření a využívá při tom systému OGC. Do verze MySQL 5.0.16 jsou prostorové funkce dostupné jen pro tabulky MyISAM. Tento typ tabulek je nejpoužívanější v MySQL databázi. Tabulky jsou optimalizovány pro *SELECT* dotazy, nicméně nepodporují transakce. Od verze 5.0.16 jsou již podporovány i InnoDB tabulky. Lze tedy používat příkazy *BEGIN*, *COMMIT* a *ROLLBACK*. InnoDB jsou dále optimalizovány pro velké množství dat. Pro indexování se používá R-strom index. Pro skenování rozsahu B-strom index, který je užitečný pro nalezení přesné hodnoty. [3] a [4]

1.2.2 PostgreSQL

PostgreSQL je stejně jako MySQL nekomerční open-source produkt. Jedná se o objektově-relační databázový systém (ORDBMS). Pro podporu prostorových dat je realizováno rozšíření PostGIS. Databáze pro indexování používá R-strom index. [5]

1.2.3 MS-SQL

MS-SQL je produkt společnosti Microsoft. Tady se již nejedná o open-source řešení jako v předchozích dvou případech, ale řešení komerční. Při vývoji Microsoft zpočátku spolupracoval se společností Sybase. První verze, kterou vydala společnost Microsoft v roce 1995, nesla označení MS-SQL Server 6.0. V roce 2008 s příchodem nového vydání verze MS-SQL Server 2008 byla přidána i podpora pro prostorová data. Server podporuje dva typy prostorových dat: **geometry** a **geography**. Datový typ geometry reprezentuje data euklidovského souřadnicového systému, zatímco typ geography reprezentuje stejná data jako typ geometry, avšak v rámci geografického souřadnicového systému. V této databázi jsou prostorové indexy realizovány pomocí B-stromů. [6] a [7]

1.2.4 Oracle

Firma Oracle vydala svou první verzi RDMS již v roce 1979. Část databáze pro podporu prostorových dat se nyní nazývá Oracle Spatial a je k dispozici pouze v edici Enterprise. U verze Oracle4, kde začaly pokusy o přidání podpory pro prostorová data, byla Oracle Spatial známa jako MultiDimension. Později s příchodem verze Oracle7 nesla název SDO. S následnou verzí Oracle8 byl představen prostorový datový typ a název od té doby nesl označení Oracle Spatial. Aktuální verze databáze je 12c, přesněji *Oracle Database 12c Release 2 v 12.2.0.2*. V letošním roce 2018 Oracle mění značení verzí a toto označení lze spíše dohledat pod názvem *Oracle Database 18c*, kde číslo 18 značí rok vydání verze databáze. V praktické části je využívána databáze Oracle, proto se následující kapitola zabývá podrobným představením právě Oracle Spatial. [8], [9] a [10]

2 ORACLE SPATIAL

Báze dat, která byla využita v praktické části, je uložena v databázi Oracle. Z tohoto důvodu nebyla možnost volby jiné databáze. Následující kapitola čerpá převážně z oficiální dokumentace pro vývojáře [11] a [12]. Dalším zdrojem této kapitoly je kniha [13].

Oracle Spatial (dále jen Spatial) je integrovaná sada funkcí a procedur, které umožňují rychle a efektivně ukládat, přistupovat a analyzovat prostorová data v Oracle databázi (dále jen databáze). Prostorový prvek, jehož tvar lze reprezentovat pomocí určitého souřadnicového systému, nazýváme **geometrie**. Geometrie je uspořádaná sekvence vrcholů, které jsou spojeny úsečkami nebo kruhovými oblouky. Spatial jako celek obsahuje následující:

- Schéma (**MDSYS**), které popisuje úložiště, syntaxi a sémantiku podporovaných geometrických datových typů.
- Mechanismus prostorového indexování.
- Operátory, funkce a procedury pro provádění dotazů, prostorového spojení (spatial join) a různých analytických operací.
- Funkce a procedury pro optimalizaci výkonu.
- **Datový model topologie** pracuje s daty jako s uzly, hranami a plochami v topologii.
- **Sít'ový datový model** reprezentuje schopnosti objektů, které jsou vytvořeny jako uzly nebo hrany v síti.
- **GeoRaster** je funkce, která umožňuje ukládat, indexovat, dotazovat, analyzovat a zobrazovat GeoRaster data, což jsou rastrové obrázky geografických objektů. Tato funkce se využívá při pořizování satelitních snímků.

2.1 Objektově-relační model

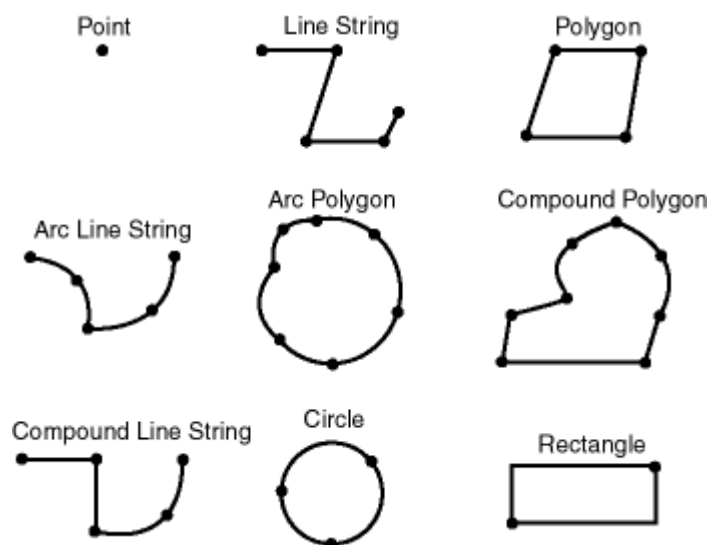
Spatial podporuje objektově-relační model pro reprezentaci geometrií, definuje nativní datový typ pro vektorová data *SDO_GEOMETRY*. Tento datový typ tedy slouží k uložení geometrií. Spatial tabulka může obsahovat jeden nebo více sloupců těchto datových typů.

Výhody poskytované objektově-relačním modelem:

- Geometrie lze uchovávat v jediném sloupci.
- Optimální výkon.
- O údržbu indexů se stará databáze.
- Snadné použití při vytváření a udržování indexů a dále při provádění prostorových dotazů.

- Podpora mnoha typů geometrií včetně oblouků, kružnic, optimalizovaných obdélníků, složených polygonů a řetězců.

Na následujícím obrázku lze vidět základní typy geometrií (Obrázek 10).



Obrázek 10: Typy dvourozměrných geometrií. Zdroj: [11]

2.2 Datový model

Spatial datový model je hierarchická struktura skládající se z elementů, geometrií a vrstev.

Element je základní kámen geometrie. Podporované typy jsou body, zřetěžené hrany a polygony. **Geometrie** je uspořádaná sada elementů. Může se skládat z jednoho elementu nebo kolekce různých elementů. **Vrstva** je sada geometrií, kde všechny geometrie patří do stejné množiny dat (např. jedna vrstva popisuje řeky na mapě a druhá popisuje silnice). Geometrie mají pro příslušnou vrstvu přidružený prostorový index a jsou uloženy ve standartních tabulkách databáze.

Do datového modelu je dále řazen souřadnicový systém a jakákoliv data mající souřadnicový systém. Prostorová data mohou být spojena s kartézským, geografickým nebo lokálním souřadnicovým systémem.

Kartézská soustava souřadnic je soustava, u které jsou souřadné osy vzájemně kolmé přímky. Ty se protínají v jednom bodě, který je označován jako počátek souřadnic. V rovině má soustava dvě souřadnice, v prostoru souřadnice tři. Není-li souřadnicový systém explicitně spojen s geometrií, předpokládá se, že se jedná o tuto soustavu. [14]

Geografické souřadnice jsou úhly souřadnic (zeměpisná šířka a délka) a jsou definovány vzhledem k určitému vztažnému bodu země.

Lokální souřadnice jsou kartézské souřadnice v určitém lokálním systému, jímž může být například CAD aplikace.

V neposlední řadě je třeba brát v potaz také vlastnost zvanou **tolerance**. Tato vlastnost určuje míru vzdálenosti, kdy se body umístěné na mapě považují za stejné. Hodnota tolerance musí být kladné číslo. Minimální hodnota pro geografická data je 0.05 (což odpovídá 5cm). Pro ostatní prostorová data je to 0.005 (což odpovídá 1 / 200 míle).

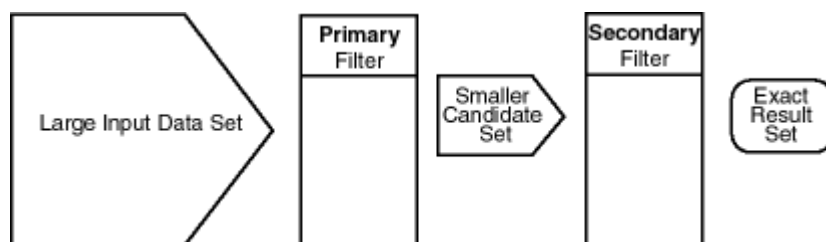
2.3 Query model

Spatial používá dvouvrstvý dotazový model (query model) k vyřešení prostorových dotazů a spojení. Query model provádí dvě odlišné operace, ty jsou označeny jako primární a sekundární filtrační operace. Výsledkem obou je přesný výsledek.

Primární filtr – do primárního filtru vstupuje množina záznamů. Primární filtr je považován za filtr s nižšími náklady. Vrací pouze geometrické aproximace výsledků (kandidáty na přesný výsledek), které postupují do sekundárního filtru. Pro implementaci tohoto filtru je použit prostorový index.

Sekundární filtr – filtr na aproximované výsledky z primárního filtru aplikuje operace pro získání přesných výsledků. Operace sekundárního filtru jsou výpočetně náročné, ale jsou aplikovány pouze na výsledky z primárního filtru.

Příklad filtru dat lze vidět na obrázku (Obrázek 11).



Obrázek 11: Query model. Zdroj: [11]

Účelem primárního filtru je rychle vytvořit podmnožinu dat a snížit zatížení pro zpracování sekundárním filtrem. V některých případech postačuje pouze aplikace primárního filtru.

2.4 Indexování prostorových dat

Indexování prostorových dat je klíčová funkce celého systému Oracle Spatial. Prostorový index, jako každý index, poskytuje mechanismus pro rychlé vyhledávání. V tomto případě je založen na prostorových kritériích, jako jsou hranice a průsečíky.

Prostorový index je potřebný pro:

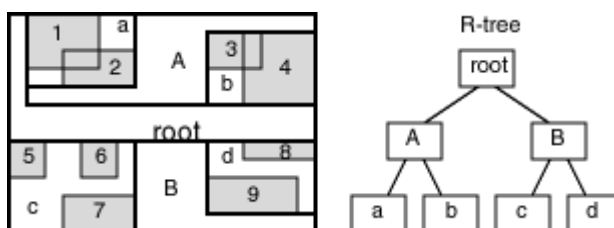
- Vyhledání objektů v indexovaném prostoru dat, které souvisí s daným bodem nebo oblastí (window query).
- Vyhledávání páru objektů ze dvou indexovaných prostorů dat, které spolu vzájemně prostorově souvisí (spatial join).

Z předchozí kapitoly je zřejmé, že u všech představených typů databází, kromě databáze Oracle, byl zmíněn používaný prostorový index. Spatial využívá R-strom index, který může indexovat prostorová data až do čtyř dimenzí. R-strom index aproximuje každou geometrii tak, že je obklopena minimálním obdélníkem (MBR - minimum bounding rectangle) (Obrázek 12).



Obrázek 12: Aproximace geometrie. Zdroj: [11]

Následující obrázek znázorňuje hierarchii indexů a R-strom. Vrstva geometrií je v levé části obrázku a R-strom je zobrazen v pravé části obrázku (Obrázek 13).



Obrázek 13: Dekompozice prostoru vrstvy geometrií a R-strom. Zdroj: [11]

- Objekty označeny čísly 1 až 9 jsou geometrie ve vrstvě.
- *a*, *b*, *c*, *d* jsou listy R-stromu. Obsahují MBR s ukazatelem na příslušnou geometrii. Např. *a* obsahuje odkaz na geometrie 1 a 2, *b* obsahuje odkaz na geometrie 3 a 4 atd.
- *A* je navigační uzel, obsahuje MBR s ukazatelem na *a* a *b*. Podobně jako *B* na *c* a *d*.
- Kořen obsahuje MBR s ukazatelem na *A* a *B* (tzn. celý zobrazovaný prostor).

Prostorový index R-stromu je uložen v tabulce prostorových indexů *SDO_INDEX_TABLE*.

2.5 Prostorový datový typ

Spatial se skládá ze sady objektových datových typů. Dále z metod, operátorů, funkcí a procedur, které tyto typy používají. Geometrie je uložena jako objekt v jediném řádku, ve sloupci typu *SDO_GEOMETRY*. Tvorba a údržba prostorových indexů je prováděna pomocí DDL (*CREATE*, *ALTER*, *DROP*) a DML (*INSERT*, *UPDATE*, *DELETE*) příkazů.

SDO_GEOMETRY

SDO_GEMOETRY je objektový datový typ pro geometrie. Každá tabulka, která má sloupec tohoto typu, musí mít další sloupec (nebo skupinu sloupců), který definuje primární klíč pro tuto tabulku. Tabulky tohoto druhu bývají označovány jako prostorové tabulky, nebo jako tabulky prostorové geometrie.

Objekt *SDO_GEOMETRY* je definován jako:

```
CREATE TYPE sdo_geometry AS OBJECT (  
  SDO_GTYPE NUMBER,  
  SDO_SRID NUMBER,  
  SDO_POINT SDO_POINT_TYPE,  
  SDO_ELEM_INFO SDO_ELEM_INFO_ARRAY,  
  SDO_ORDINATES SDO_ORDINATE_ARRAY);
```

V následujících podkapitolách jsou popsány atributy, které jsou součástí objektu *SDO_GEOMETRY* a jsou tudíž potřeba k jeho vytvoření.

2.5.1 SDO_GTYPE

SDO_GTYPE je atributem *SDO_GEOMERTY* a označuje typ geometrie. Platné typy geometrií odpovídají těm, které jsou uvedeny v OpenGIS Simple Features³. Číselné hodnoty neodpovídají těm v OpenGIS, ale existuje přímá vazba mezi názvy a sémantikou.

Hodnota *SDO_GTYPE* má 4 číslice ve formátu *DLTT*, kde:

- *D* – označuje počet rozměrů (2, 3 nebo 4).
- *L* – určuje lineární referenční měřítko pro 3D lineární referenční systém (*LRS*) geometrií, když dimenze (3 nebo 4) obsahuje hodnotu měřítka. (0 pro ne-LRS geometrii).
- *TT* – identifikuje typ geometrie (00 až 09, 10 až 99 vyhrazeno pro budoucí použití).

³ Standard, který specifikuje uložení geografických dat v digitální podobě.

Následující tabulka obsahuje hodnoty *SDO_GTYPE* (Tabulka 5).

Tabulka 5: Hodnoty datového typu *SDO_GTYPE*.

Typ geometrie	Hodnota [DLTT]
Neznámá geometrie	DL00
Bod	DL01
Úsečka nebo křivka	DL02
Polygon nebo povrch	DL03
Kolekce	DL04
Multibod	DL05
Úsečka nebo multikřivka	DL06
Multipolygon nebo multipovrch	DL07
Solid	DL08
MultiSolid	DL09

Zdroj: [11] – upraveno

2.5.2 SDO_SRID

SDO_SRID je atribut, jehož hodnota popisuje souřadnicový systém spojený geometrií. Pokud má být na sloupci vybudován prostorový index, musí mít všechny geometrie ve sloupci tabulky stejnou hodnotu *SDO_SRID*. Může mít i hodnotu *null*, ale geometrie nebude mít přiřazena souřadnicový systém. Pokud hodnota není *null*, musí obsahovat hodnotu ze sloupce *SRID* z tabulky *SDO_COORD_REF_SYS*. A tato hodnota musí být vložena do sloupce *SRID* tabulky metadat *USER_SDO_GEOM_METADATA* (viz kapitola 2.6). Toto je velice důležité pro správnou funkčnost prostorového indexu. Bez metadat není schopen prostorový index pracovat.

2.5.3 SDO_POINT

Tento atribut je definován pomocí objektového typu *SDO_POINT_TYPE*. Ten má atributy *x*, *y* a *z*, které jsou typu *NUMBER*. Jak bylo definováno, objekt *SDO_GEOMETRY* má dvě pole *SDO_ELEM_INFO* a *SDO_ORDINATES*. Pokud jsou tyto pole obě *null* a atribut *SDO_POINT* není *null*, pak jsou souřadnice (*x*, *y* a *z*) považovány za souřadnice bodu geometrie. V opačném případě je *SDO_POINT* atribut ignorován. Pro optimální uchování by se měly body ukládat do tohoto atributu. Pokud ve vrstvě existují pouze geometrie bodů, je to důrazně doporučováno.

2.5.4 SDO_ELEM_INFO

Atribut je definován jako číselné pole proměnné délky. Říká nám jakým způsobem interpretovat souřadnice, které jsou uloženy v *SDO_ORDINATES*.

Je složen minimálně z jedné trojice čísel, jejichž význam je následující:

- *SDO_STARTING_OFFSET* – označuje posun v poli *SDO_ORDINATES*, číslování je od hodnoty 1 (nikoliv 0).
- *SDO_ETYPE* – označuje typ prvku.
- *SDO_INTERPRETATION* – závisí na atributu *SDO_ETYPE*. Buďto slouží k určení, kolik následných hodnot je součástí prvku, nebo určuje jakým způsobem je interpretována posloupnost souřadnic.

Hodnot atributu *SDO_ELEM_INFO* je mnoho. Podrobnosti lze vyčíst z oficiálních stránek.⁴

2.5.5 SDO_ORDINATES

Atribut je stejně jako *SDO_ELEM_INFO* definován jako číselné pole proměnné délky. Má za úkol uchovávat souřadnice, které tvoří hranici prostorového objektu. Toto pole musí být používáno vždy výhradně s atributem *SDO_ELEM_INFO*. Hodnoty v poli jsou seřazeny podle dimenzí.

2.6 Metadata geometrie

Metadata geometrie popisují její dimenzi, horní a dolní hranici a toleranci. Metadata každé z dimenzí jsou uložena v globální tabulce jmenného prostoru *MDSYS*. Tuto tabulku by neměli uživatelé přímo aktualizovat.

Každý uživatel má ve svém schématu k dispozici následující pohledy:

- *USER_SDO_GEOM_METADATA* – obsahuje informace o metadatech všech prostorových tabulek vlastněné uživatelem. Pouze tento pohled umožňuje uživateli aktualizovat a vkládat metadata.
- *ALL_SDO_GEOM_METADATA* – obsahuje informace o metadatech všech prostorových tabulek na které má uživatel oprávnění *SELECT*. Oracle automaticky zajišťuje, aby se řádky, které jsou vloženy do *USER_SDO_GEOM_METADATA*, promítly i do tohoto pohledu.

⁴ <https://docs.oracle.com/en/database/oracle/oracle-database/18/spatl/spatial-datatypes-metadata.html#GUID-270AE39D-7B83-46D0-9DD6-E5D99C045021>

Každý pohled metadat má následující definici:

```
(  
  TABLE_NAME    VARCHAR2 (32) ,  
  COLUMN_NAME     VARCHAR2 (32) ,  
  DIMINFO         SDO_DIM_ARRAY ,  
  SRID            NUMBER  
) ;
```

- *TABLE_NAME* – tento sloupec obsahuje název tabulky, která má sloupec datového typu *SDO_GEOMETRY*.
- *COLUMN_NAME* – sloupec obsahující název sloupce, který je typu *SDO_GEOMETRY*.
- *DIMINFO* – je pole objektů proměnné délky, kde jsou objekty seřazeny podle dimenze. Pole má jeden vstup pro každou položku.
SRID – obsahuje buď typ souřadnicového systému společný pro všechny geometrie, nebo hodnotu *null*. To značí, že souřadnicový systém není definován.

2.7 Oracle Fusion Middleware MapViewer

Oracle Fusion Middleware je sbírka standardizovaných softwarových produktů zahrnující řadu nástrojů a služeb. Mezi tyto nástroje patří vývojářské utility, integrační služby, management zdrojů a nástroje bussiness intelligence. Celá sada tedy nabízí kompletní podporu pro vývoj, nasazení a správu software. [15]

V diplomové práci je věnována pozornost pouze části zvané Oracle Fusion Middleware MapViewer (zkráceně MapViewer).

MapViewer je programovatelný nástroj pro vykreslování prostorových dat uložených v databázi Oracle Spatial. Jeho hlavním přínosem je integrace s Oracle Spatial. Obsahuje následující hlavní části:

- Renderovací engine
- Sadu API (XML, Java, JavaScript)
- Nástroj pro vytváření map
- Oracle Maps

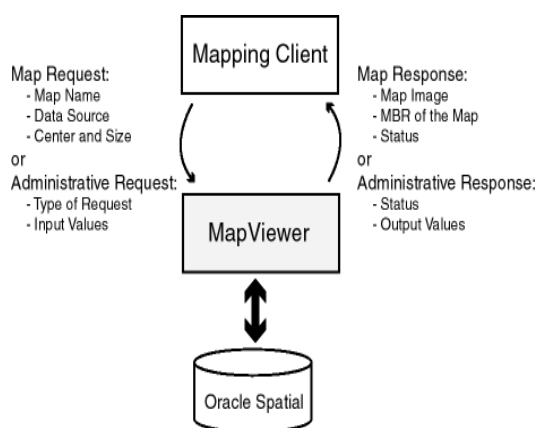
Jednotlivé části jsou popsány podrobněji v dalších částech kapitoly.

2.7.1 Základní tok činností aplikace MapViewer

Pomocí aplikace MapViewer je základní tok činností řízen principem request-response. Ať už klient požaduje mapu, nebo nějakou administrativní činnost MapVieweru.

Pro požadavek na mapu:

- Klient požaduje mapu – pošle jméno mapy, zdroj dat, střed lokace, velikost mapy a případně další data, která mají být na mapě vykreslena.
- Server vrátí obrázek mapy (případně URL adresu), MBR mapy a stav požadavku.

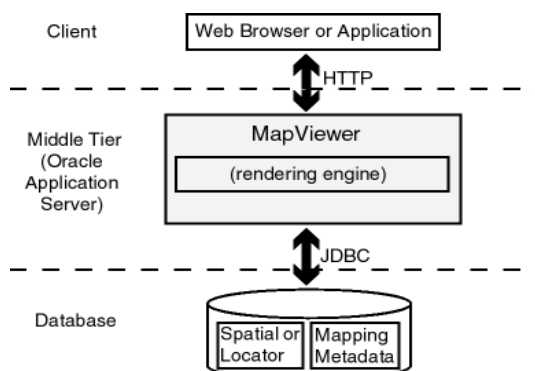


Obrázek 14: Princip činnosti MapVieweru. Zdroj: [15]

2.7.2 Architektura MapVieweru

Následující obrázek vyobrazuje architekturu MapVieweru a lze z něj vyčíst tyto informace:

- MapViewer je součástí prostřední vrstvy Oracle Fusion Middleware.
- MapViewer obsahuje renderovací engine.
- MapViewer provádí přístup k prostorovým datům v databázi prostřednictvím volání JDBC.
- Databáze obsahuje kromě Oracle Spatial také mapování metadat.



Obrázek 15: Architektura MapVieweru. Zdroj: [15]

2.7.3 Renderovací engine

Renderovací engine, zvaný *SDOVIS* provádí kartografické vykreslování. Jedná se o knihovnu Java. K databázi se připojuje pomocí JDBC. Také slouží ke čtení metadat mapy z databáze a aplikuje je na načtená prostorová data během vykreslování.

2.7.4 Sada API

Sada umožňuje programování funkcí MapVieweru. XML API poskytuje vývojářům všestranné rozhraní pro odesílání žádosti (request) MapVieweru a následný příjem (response) mapy. Java rozhraní poskytuje přístup k vykreslovacím funkcím MapVieweru. Javascriptové API umožňuje vytvářet interaktivní webové aplikace, které používají funkci Oracle Maps.

2.7.5 Map Builder

Map Builder je samostatná aplikace pro vytváření map. Umožňuje definovat vlastní pravidla, symboly a upravovat objekty MapVieweru. Pomocí tohoto nástroje je možno například vytvořit vlastní styl nebo upravit definici stylu. Nástroj také umožňuje náhled metadat (např. jak se na mapě zobrazí styl čáry). Instalace a použití nástroje je popsáno v praktické části práce.

2.7.6 Oracle Maps

Oracle Maps jsou postavené na funkcích MapVieweru. Umožňují stavět dynamické a složité mapové aplikace. Používají **server mapové cache paměti**, který ukládá mapy pomocí tzv. dlaždic do cache paměti. Dále **FOI server**, který prochází data v databázi a zobrazuje je jako interaktivní prvky na mapě.

Instalace MapVieweru, jeho konfigurace a také to jakým způsobem je realizováno propojení s desktopovou aplikací, je popsáno v praktické části práce.

V této kapitole je čerpáno ze zdrojů [12] a [15].

3 INFRASTRUKTURA ŽELEZNIČNÍ SÍTĚ

Česká republika je zemí s velmi vysokou hustotou sítě železnic. Na 1 km² připadá 0.12 km železnic, tím se řadí mezi nejvyšší na světě. Železniční síť se v uplynulých cca dvaceti letech nerozšiřovala, nicméně se zvyšovala její kvalita. A to nejen z hlediska stavebního, ale také z hlediska výpočetních technologií. Příkladem jsou systémy pro lokalizaci kolejových vozidel, kterými se zabývá první část této kapitoly. V druhé části kapitoly je představeno, jakým způsobem lze železniční síť reprezentovat v paměti počítače. Pozornost je také věnována standardu ČD M12, který je potřebný pro pochopení parametrů dat, ze kterých práce vychází. [16]

V následující tabulce jsou uvedeny základní informace týkající se železniční sítě v ČR. Hodnoty jsou aktuální k 31. 12. 2016 (Tabulka 6). [17]

Tabulka 6: Základní charakteristika železniční sítě.

Ukazatel	Jednotky	Hodnota
Délka tratí celkem	km	9463
Stavební délka kolejí celkem	km	15435
Počet výhybkových jednotek	v.j.	23278

Zdroj: [17]

3.1 Systémy pro lokalizaci kolejových vozidel

Je-li položena otázka, jakým způsobem by bylo možné lokalizovat kolejové vozidlo na trati, jsou k dispozici prakticky dvě základní řešení.

První možností je použít k lokalizaci vozidla satelitní navigační systém GNSS (Global Navigation Satellite System). Druhé řešení se týká zařízení na trati, které je schopné notifikovat polohu vozidla. A to za předpokladu, že kolem tohoto zařízení vozidlo projede. Druhá možnost je podstatně nákladnější, jelikož je potřeba trať o zařízení, které budou sledovat polohu vozidla, doplnit. Existuje však ještě třetí možnost. Satelitní navigační systém lze o další podpůrné systémy doplnit. [18]

Systémy pro lokalizaci lze tedy v zásadě rozdělit do tří základních kategorií:

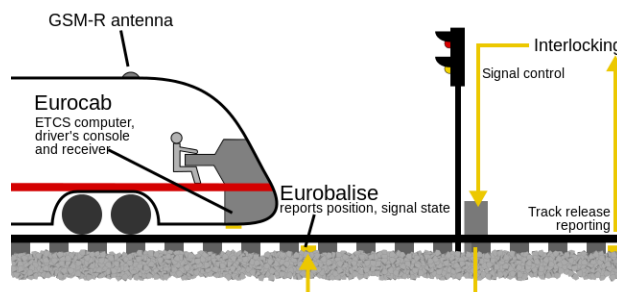
- bez použití satelitního navigačního systému.
- s použitím satelitního navigačního systému (GNSS).
- s použitím satelitního navigačního systému a podpůrných systémů.

3.1.1 Lokalizace vozidel bez použití satelitního navigačního systému

Jak již bylo zmíněno v úvodu podkapitoly, tento způsob je nákladný, nicméně poměrně spolehlivý a přesný. Do této kategorie lze zařadit například evropský zabezpečovací systém *ETCS* (European Train Control System). Úkolem zabezpečovacího systému je zaručení bezpečnosti vlakové soupravy. Systém sleduje maximální rychlost vlaku a maximální traťovou rychlost v daném úseku. Dále zjišťuje, jestli vlak dodržuje trasu, směr jízdy a další bezpečnostní kritéria. Součástí systému jsou informační body, které jsou součástí tratě a informují o poloze vozidla. Tyto informační body se nazývají **balízy**. V systému *ETCS* jsou definovány tři úrovně. [18], [19] a [20]

ETCS Level 1

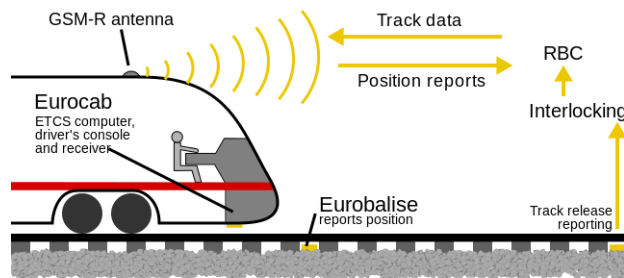
V první úrovni jsou na trati umístěny tzv. **přepínatelné balízy**. Ty slouží jako referenční body a předávají vozidlu informace o návěstidle (signal control), k němuž vlak přijíždí. Balízy jsou propojeny s jednotkami *LEU* (Lineside Electronic Unit), které komunikují se zabezpečovacím zařízením. Na vozidle je umístěn otáčkoměr, který je za pomoci balíz neustále synchronizován. Za jeho pomoci lze zjišťovat přesnou polohu vozidla. [18], [19] a [20]



Obrázek 16: ETCS Level 1. Zdroj: [20]

ETCS Level 2

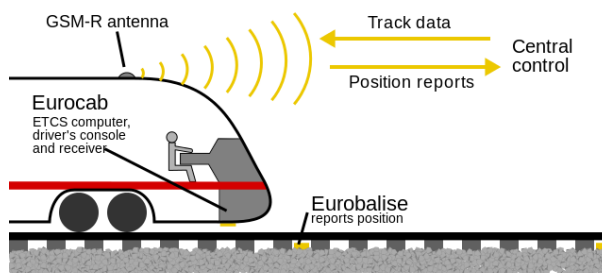
V druhé úrovni jsou **nepřepínatelné balízy**, ty slouží pouze jako referenční body k synchronizaci otáčkoměru a určení polohy vozidla. Řídící jednotkou pro daný traťový úsek je tzv. rádio-bloková centrála (RBC), která přijímá informace a komunikuje s vozidlem za pomoci GSM-R antény. Palubní počítač vozidla nepřetržitě sleduje přenášovaná data a maximální přípustnou rychlost. [18], [19] a [20]



Obrázek 17: ETCS Level 2. Zdroj: [20]

ETCS Level 3

Třetí úroveň je zatím ve fázi teorií, testování a hledání možných řešení. Kontrola celistvosti probíhá průběžně pomocí rádiových prostředků, již není nutné použití pevných zařízení. [18], [19] a [20]



Obrázek 18: ETCS Level 3. Zdroj: [20]

3.1.2 Lokalizace s použitím GNSS

Jedním z nejpoužívanějších GNSS systémů je systém GPS, který je pod správou Ministerstva obrany Spojených států amerických. Existují však další systémy, jakým je třeba evropský Galileo, ruský GLONASS nebo čínský BeiDou. U těchto systémů, kde figuruje satelitní zařízení, je potřeba počítat s odchylkou přesnosti polohy. V zabezpečovací technice je však nutné tuto chybu eliminovat, tím je použití pouze systému GNSS vyloučeno. Existují různé systémy, které dokáží tuto odchylku z části nebo úplně odstranit. Jeden ze způsobů, jak toho dosáhnout, je diferenciální GPS. [18]

3.1.3 Lokalizace s použitím GNSS a podpůrných systémů

Podpůrných systémů v Evropě a celém světě existuje řada. V Rakousku je to *Train Control System*, v Itálii systém *3InSat*, ve Francii *LOCOPROL* a bylo by možno pokračovat dále. Tyto systémy jsou navrženy různým způsobem, ale mají jedno společné. Pro zpřesnění GPS signálu používají diferenciální GPS. [18]

Diferenciální GPS

Diferenciální GPS je varianta pro zpřesnění výsledků měření GPS. Pro fungování je vyžadován minimálně jeden přijímač GPS signálu o známých souřadnicích, tzv. referenční stanice. Tato stanice vyhodnocuje ve stejných intervalech odchylku měření v systému GPS od stavu své polohy a vypočítává korekce. Tyto korekce jsou za pomoci mobilní sítě a internetu posílány uživatelům. [21]

U kolejových vozidel je zpřesnění pomocí diferenciální GPS doplněno o tzv. navigační výpočet, který probíhá pomocí algoritmů CDGPS. Tato kombinace umožňuje pracovat s přesností polohy v rozmezí 10 až 30 centimetrů. Data o poloze vozidla jsou posílána na dispečerskou stanici, takže umožňují řídit dopravu v rámci udržovaných tratí. [18] a [22]

3.2 Reprezentace železniční sítě v paměti počítače

Železniční síť si lze představit jako množinu stanic, které jsou spolu jistým způsobem propojeny. Z toho lze vyvodit, že v paměti počítače je možno železniční síť reprezentovat jako datovou strukturu graf. Jelikož není určen směr propojení mezi stanicemi (úseky jsou obousměrné) hovoříme o tzv. **neorientovaném grafu**. [18]

3.2.1 Abstraktní datový typ graf

ADT graf pracuje se dvěma odlišnými třídami prvků – **vrcholy** a **hranami**. Dle teorie grafů existují dva základní typy. A to buď **orientovaný**, nebo **neorientovaný** graf. U neorientovaného grafu hrany nemají danou orientaci. Z hlediska přístupu k datům lze kategorizovat na vrcholově a hranově orientovaný přístup. [23]

Vrcholově orientovaný přístup

Vstupem struktury je vrchol, který si nese informace o jeho datech a jeho předchůdcích / následnících. Mezi tímto vrcholem a následníkem / předchůdcem existuje hrana, která může být orientovaná. Při vyhledávání vrcholů jsou k dispozici informace o sousedních vrcholech, tedy i hranách vedoucích z / do vrcholu. [23]

Hranově orientovaný přístup

Nejprve je realizováno vyhledávání hran, až poté je přistupováno k vrcholům. Hrany mohou být vyhledávány dvěma způsoby. Prvním způsobem je vyhledání hrany na základě dvou vrcholů, kterým hrana přísluší (případně klíčů těchto vrcholů). Druhým způsobem je přiřadit hraně klíč, pomocí kterého bude vyhledána. [23]

V případě multidimenzionálních dat je vhodné ADT graf zkombinovat se strukturami. Například s těmi, které byly představeny v první kapitole této práce.

3.3 Topologie podle standardu M12

Standard M12 je předpis pro jednotné označování tratí a kolejíšť v informačním systému ČD. Kapitola čerpá ze zdroje [24].

3.3.1 Železniční dopravní cesta

Železniční dopravní cesta je souhrn hmotného majetku a zařízení umožňující jízdu kolejových vozidel. Dále jde o zabezpečení provozu a technického stavu trasy.

3.3.2 Železniční těleso

Železniční těleso je těleso určené bezprostředně k železničnímu provozu.

3.3.3 Trať

Trať se rozumí část železniční dopravní cesty, kterou tvoří železniční těleso a drážní zařízení mezi dvěma určenými místy. Železniční síť je tvořena souhrnem tratí na daném území.

3.3.4 Staničení

Staničení je lokalizační systém odvozený z výsledků geodetických měření a projektu stavby kolejí. Umožňuje určení skutečné vzdálenosti kolmého průmětu charakteristického bodu daného objektu železniční dopravní cesty do osy koleje od předpisem stanoveného počátku.

3.3.5 Staničnick

Staničnick je zobecněním pro pojmy **kilometrovník** a **hektometrovník**. Jsou to traťové značky, které udávají staničení tratě. Jak již může vyplynout z názvu, hektometrovníky jsou umístěny po sto a kilometrovníky tisíci metrech.

3.3.6 Definiční úsek (DÚ)

Definiční úsek je základní evidenční a identifikační jednotka určená číselníkem definičních úseků. Slouží k rozčlenění železniční sítě na menší části. Z evidenčního hlediska v sobě zahrnuje kromě zemského povrchu i všechny nemovité objekty a zařízení, které do něj územně spadají.

3.3.7 Traťový definiční úsek (TUDU)

Traťový definiční úsek je datový obraz definičního úseku a je reprezentován tabulkou dat. Rozšiřuje číselník definičních úseků a další jednotky, kterého ho charakterizují.

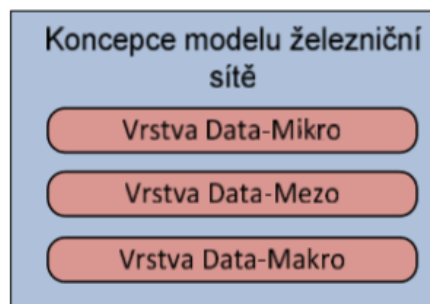
4 IMPLEMENTACE EDITORU ŽELEZNIČNÍ SÍTĚ

Čtvrtá kapitola je stěžejní kapitolou celé práce. Je zde popsáno, jakým způsobem bylo při tvorbě praktické části postupováno. Kapitola začíná představením toho, co má diplomová práce vlastně přinést. Jsou zde představena data, ze kterých práce vychází. Dále pak postup, jak s daty pracovat a jak je uchovávat. Tím se dostáváme k samotné tvorbě editoru. Tvorba je spojena s návrhem tříd, jakým způsobem je editor propojen s databází. Jak je realizována vizualizace a v neposlední řadě také s jakými problémy se bylo třeba zaobírat a jak je vyřešit.

4.1 Výchozí bod a představení dat

Diplomová práce si klade za cíl vytvořit editor železniční sítě. Železniční síť je reprezentována daty, která jsou uložena v databázi Oracle. Jediná možnost, jak data do této doby spravovat, byla psát si skripty na případnou editaci a jiné DML operace. A to je uživatelsky velice nepohodlné. Cílem práce je tedy vytvořit aplikaci, která zvládne načíst data z databáze a umožní jejich uživatelskou modifikaci.

Data jsou k dispozici v *.sql* souborech, které slouží pro jejich import do databázových tabulek. Každý soubor reprezentuje určitou logickou vrstvu. Tato data představují koncepci třívrstvého modelu železniční sítě. Každá vrstva modelu zachycuje jistou míru abstrakce železniční sítě (Obrázek 19).



Obrázek 19: Koncepcie třívrstvého modelu železniční sítě. Zdroj: [18]

V první, nejnižší *mikro* vrstvě je nejmenší míra abstrakce. Vrcholy grafu sítě odpovídají hektometrovníkům a jsou propojeny hranami. V druhé, prostřední *mezo* vrstvě je vyšší míra abstrakce. Jsou zde vrcholy z nižší (*mikro*) vrstvy, které jsou v incidenci s více než dvěma hranami tzv. **super-vrcholy**. Dále jsou v *mezo* vrstvě vrcholy, které reprezentují stanice. Ve třetí vrstvě, tedy *makro*, je nejvyšší míra abstrakce. Jsou zde pouze super-vrcholy.

Výchozím bodem práce jsou tedy *.sql* soubory, které jsou pojmenovány:

- Mikro.sql
- Mezo.sql
- Makro.sql

Tyto soubory obsahují skripty pro vytvoření databázových tabulek, prostorových indexů a následný import dat.

Dále je potřeba zmínit, že *mezo* vrstva obsahuje kromě super-vrcholů také vrcholy reprezentující stanice (resp. by měla, ale o tom níže). Tyto vrcholy je třeba napojit na tabulku, kde jsou stanice uchovávány. To je zajištěno bindovací tabulkou, ve které jsou na každém řádku pouze *id* z obou tabulek *station* a *mezo*.

Skripty pro tyto dvě tabulky, jejich vytvoření a naplnění jsou v souborech:

- Station.sql
- StationBinding.sql

Z důvodu snadné identifikace super-vrcholů i v *mikro* vrstvě je zde ještě druhá bindovací tabulka, která obsahuje na každém řádku *id* prvku *mezo* vrstvy a *id* prvku *makro* nebo *mikro* vrstvy. Je tedy důležité zdůraznit, že každý prvek v *makro* vrstvě má stejné *id* jako příslušný prvek z *mikro* vrstvy.

Skripty pro vytvoření bindovací tabulky a její naplnění jsou v souboru:

- SupernodeBinding.sql

Než budou podrobně představeny jednotlivé tabulky a význam jejich dat, je třeba provést import dat do databáze. Databázi je třeba nejprve správně nainstalovat a nastavit pro příslušného uživatele.

4.2 Instalace databáze

Instalovaná verze je *Oracle Database 12c Release 2*, která je dostupná na oficiálních webových stránkách⁵. Při instalaci Oracle nejprve žádá zadat email pro informování o možných bezpečnostních problémech, není však nutné ho vyplňovat.

V dalším kroku je možnost volby. Zprvč jenom nainstalovat databázi, zadruhé nainstalovat a rovnou nakonfigurovat, nebo zatřetí provést upgrade na novější verzi. Je určena první možnost – volba instalace, protože konfiguraci lze provést později.

V třetí části instalace jsou tři možnosti volby databáze:

- Single instance database installation
- Oracle Real Application Clusters database installation
- Oracle RAC One Node database installation

Je zvolena první volba, která nám dovolí nainstalovat jednu instanci databáze. Druhá volba umožní spojit dva a více počítačů, aby vytvořili databázový cluster, který se chová jako jediný systém a rozkládá zátěž na více serverů. Třetí možností je odlehčená verze Oracle RAC, která lze snadno na druhou možnost upgradovat. Rozdíl spočívá v tom, že běží pouze jedna instance na jednom serveru v clusteru. Při chybě je namísto zastavení a opětovného spuštění využito přemístění instance na jiný server v clusteru.

Ve čtvrté části instalace je volena verze databáze. Existují zde dvě možnosti:

- Standard Edition
- Enterprise Edition

Jelikož je potřeba využívat volbu Oracle Spatial, která je dostupná pouze ve verzi *Enterprise*, je zvolena tato možnost.

Pátým krokem je volba účtu z důvodu zabezpečení. Tam je doporučováno použít virtuální účet. Další možností je použít již existujícího Windows uživatele, nebo vytvořit úplně nového. Případně zbývá volba *build-in*, což je administrátorský účet.

⁵ <http://www.oracle.com/technetwork/database/enterprise-edition/downloads/oracle12c-windows-3633015.html>

Posledním krokem je umístění instalačních souborů. Tato volba má dvě cesty. Zaprvé, kde bude uložena složka *Oracle Base* obsahující konfigurační soubory. Zadruhé, kde bude uložen databázový software. Defaultní volba je uvnitř složky *Oracle Base*.

Před spuštěním instalace je provedena kontrola, zda lze produkt nainstalovat. Je ještě poslední možnost vše zkontrolovat. Pro instalaci databáze byl zvolen prázdný SSD disk kvůli rychlosti. Po úspěšném dokončení instalace se zobrazí hláška o bezchybném nainstalování databáze.

4.3 Konfigurace databáze

Ke spuštění manageru pro konfiguraci databáze je třeba spustit dávkový soubor *dbca.bat* v adresáři *bin*. Po spuštění manageru je zvolena možnost pro vytvoření databáze a následně pokročilý mód pro konfiguraci, kde je vybrána možnost *Custom Database*.

První částí je konfigurace globálního názvu databáze a SID, což je systémový identifikátor. V obou případech bylo použito označení *dp*, značící diplomovou práci. Na stejné stránce manageru je ještě možnost vytvoření kontejneru, která není použita.

Další možnosti jsou ponechány jako defaultní, až v záložce *Network Configuration* je třeba pro správnou funkci vytvořit listener, který je nazván jednoduše *LISTENER* a naslouchá na portu *1521*.

V záložce *Database Options* jsou ponechány všechny defaultní volby, je potřeba zkontrolovat zaškrtnutí volby *Oracle Spatial*.

Na následující záložce *Configuration Options* je nejprve nastavena paměť SGA na 900 MB a PGA na 300 MB. Celková paměť dostupná pro Oracle bude tedy 1200 MB. Ještě je třeba zkontrolovat, aby znaková sada byla nastavena na češtinu. Ostatní volby jsou ponechány defaultní. Poté je možno pokračovat na další záložku.

Nyní následuje záložka *Management Options*, kde jsou opět ponechány defaultní volby. Případně lze změnit port pro Enterprise Managera, nicméně je ponechána hodnota *5500*.

Na záložce *User Credentials* se nastavují administrátorská hesla. Je zde možnost zvolit jiné heslo pro účty SYS a SYSTEM. Pro naše účely bylo zvoleno heslo stejné. Oracle doporučuje použití kombinaci malých a velkých písmen s číslicemi o minimálně délce 8 znaků, není to však podmínkou.

Na předposlední záložce za *Creation Option* je zaškrtnuta volba *Create database*. Možností je nechat si vygenerovat skript pro založení instance této databáze při případném výpadku nebo jiné poruše.

Na záložce *Summary* lze provést poslední kontrolu před instalací. Také je zde možnost pro uložení do souboru. Tento soubor *dbca.rsp* je dostupný jako příloha této práce.

Pokud vše proběhne v pořádku, je databáze úspěšně nainstalovaná. Přístup do Enterprise Manageru je dostupný přes adresu: *https://hostname:portnumber/em*. Konkrétně v našem případě se jedná o *https://localhost:5500/em*. Pro prvotní konfiguraci použijeme uživatele *sys* a administrátorské heslo, které bylo zadáno při instalaci.

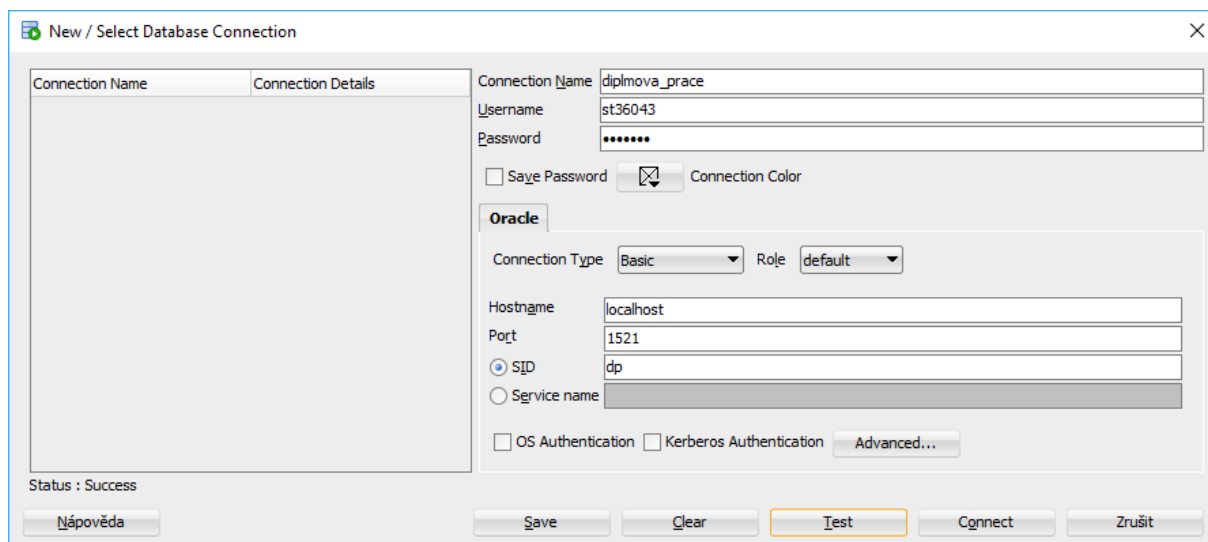
Po úspěšném přihlášení je potřeba vytvořit testovacího uživatele, který je dále používán v diplomové práci. Uživatel se vytváří v záložce *security*, sekce *users*. Je zvoleno přihlašovací jméno uživatele a heslo, dále je třeba přiřadit tabulkový prostor, defaultně *USERS*. Poslední záložkou je možnost uživateli přiřadit práva. Pro potřeby diplomové práce jsou uživateli nastavena práva, která jsou neomezená v rámci tabulkového prostoru *USERS*. Pomocí příkazu:

```
GRANT UNLIMITED TABLESPACE TO ST36043;
```

V defaultním nastavení tohoto prostoru není vytváření tabulek, které je potřeba. Je tedy nutné ho nastavit.

Nyní lze vyzkoušet, jestli se daný uživatel může k databázi připojit. To je možné provést například přes nástroj *sqldeveloper*, který je dostupný z oficiálních stránek. Nicméně v našem případě, kdy byla databáze instalována, je *sqldeveloper* dostupný ve složce s nainstalovanou databází, přesněji *//product/12.2.0/dbhome_1/sqldeveloper*.

V roli testovacího uživatele, který od databázového administrátora (uživatele *sys*) obdržel přihlašovací údaje, je realizováno připojení k databázi přes program *sqldeveloper*. Ukázka vyplněného okna (Obrázek 20).



Obrázek 20: Sql Developer – databázové připojení. Zdroj: vlastní

Do políčka *Connection Name* lze vyplnit jakýkoliv název připojení. Políčka *Username* a *Password* obsahují údaje testovacího uživatele. V políčku *Hostname* je hodnota *localhost* zachována. Portem se rozumí port listeneru, který nebyl při instalaci pozměněn a zůstává *1521*. Jako poslední nastavovaná hodnota je *SID*, ta byla určena při konfiguraci databáze jako *dp*. Pro ověření je stisknuto tlačítko *Test*. Jak lze z obrázku dále vyzorovat, test proběhl úspěšně *Status: Success*. Po takto vyplněných údajích je již možné navázat připojení k databázi.

4.4 Import dat do databáze

Při importu je jako vzor představen soubor *Makro.sql*, s dalšími soubory se postupuje analogicky. První částí souboru je skript pro vytvoření tabulek *RAILWAY_ABS_LINK\$* a *RAILWAY_ABS_NODE\$*. Ve skriptu je třeba upravit tabulkový prostor na námi definovaný *USERS* a také jméno testovacího uživatele. Tyto dvě úpravy je třeba provést ve všech představených souborech. Lze to provést jednoduše pomocí volby *Nahradit vše*, kterou disponuje téměř každý textový editor.

Po této opravě by již bylo možné tabulky vytvořit a naplnit. Nicméně součástí souboru je i skript pro vytvoření prostorového indexu.

```
CREATE INDEX "ST36043"."RAILWAY_ABS_NODES_IDX" ON
"ST36043"."RAILWAY_ABS_NODE$" ("GEOMETRY") INDEXTYPE IS
"MDSYS"."SPATIAL_INDEX";
```

Jak bylo zmíněno v teoretické části práce, prostorový index není schopen pracovat bez metadat.

Ta je nutné vytvořit pomocí následujícího skriptu. Analogicky i pro další tabulky, kde je třeba vytvořit prostorový index.

```
INSERT INTO USER_SDO_GEOM_METADATA (TABLE_NAME, COLUMN_NAME, DIMINFO, SRID)
VALUES ('RAILWAY_ABS_NODE$', 'GEOMETRY',
MDSYS.SDO_DIM_ARRAY
(MDSYS.SDO_DIM_ELEMENT('LONG', 48.5525, 51.055556, 0.005)
,MDSYS.SDO_DIM_ELEMENT('LAT', 12.091389, 18.858889, 0.005))
,8307);
```

Metadata platí pro tabulku *RAILWAY_ABS_NODE\$*, pro její sloupec *GEOMETRY*. Hodnota *SRID* je 8307 a značí, že se jedná o souřadnicový systém zeměpisné šířky a délky. Číslo 0.005 značí toleranci a zbylé hodnoty jsou krajní body mapy ČR. [25]

Poslední věc, která je třeba udělat před spuštěním celého skriptu, je nastavit testovacímu uživateli právo na vytvoření sekvence *CREATE SEQUENCE*.

Tímto způsobem jsou importována data do databáze ze všech souborů. Je tedy velice vhodné představit, co je vlastně importováno. Zaměříme se na geometrii.

```
INSERT
INTO ST36043.RAILWAY_ABS_LINK$
( LINK_ID, LINK_NAME, START_NODE_ID, END_NODE_ID, LINK_TYPE, ACTIVE,
LINK_LEVEL, GEOMETRY, COST)
VALUES ('118', 'L118', '936', '682', NULL, 'Y', '2',
MDSYS.SDO_GEOMETRY(2002, 8307, NULL, MDSYS.SDO_ELEM_INFO_ARRAY(1, 2, 1),
MDSYS.SDO_ORDINATE_ARRAY(
15.7862640277544, 49.9488477775786, 15.7869061112404, 49.9569811111026)), '1');
```

První hodnota 2002 říká, že se jedná o dvoudimenzionální geometrii. Druhá hodnota 8307 již byla představena výše. Třetí hodnota *NULL* značí, že *SDO_POINT* není vyplněn, jelikož se jedná o čáru. *SDO_ELEM_INFO* obsahuje předpis, kde první číslo znamená, že se začíná pracovat s hodnotami od prvního čísla. Druhé číslo značí, že se jedná o čáru a třetí hodnota značí přímku. V *SDO_ORDINATE_ARRAY* jsou uloženy souřadnice čáry.

4.5 Databázové tabulky

Tato podkapitola představuje všechny vytvořené tabulky v databázi.

4.5.1 Tabulka *RAILWAY_ABS_NODE\$*

ST36043.RAILWAY_ABS_NODE\$		
P *	NODE_ID	NUMBER
	NODE_NAME	VARCHAR2 (200 BYTE)
	NODE_TYPE	VARCHAR2 (200 BYTE)
	ACTIVE	VARCHAR2 (1 BYTE)
	PARTITION_ID	NUMBER
	GEOMETRY	SDO_GEOMETRY
	ROUTE	VARCHAR2 (11 CHAR)
	TUDU	VARCHAR2 (6 CHAR)
	KM	NUMBER (5,1)
	KV	VARCHAR2 (1 CHAR)
RAILWAY_ABS_NODE\$_PK (NODE_ID)		
RAILWAY_ABS_NODES_IDX (GEOMETRY)		
SYS_C00283346 (NODE_ID)		

Obrázek 21: Tabulka *RAILWAY_ABS_NODE\$*. Zdroj: vlastní

- *NODE_ID* – jednoznačný identifikátor prvku.
- *NODE_NAME* – jméno prvku, složen z prefixu *N* a jednoznačného identifikátoru.
- *NODE_TYPE* – typ prvku, není vyplněno.
- *ACTIVE* – značí vypnutí / zapnutí prvku, má pouze dvě hodnoty *Y* a *N*.
- *PARTITION_ID* – identifikační číslo oddílu.
- *GEOMETRY* – obsahuje geometrii prvku.
- *ROUTE* – sloupec, který byl vytvořen pro potřeby importu. [18]
- *TUDU* – označení TUDU, kterému prvek přísluší.
- *KV* – označení, zda je hektometrovník umístěn na koleji – *K*, nebo výhybce – *V*.

Tabulka prvků *mikro* vrstvy *RAILWAY_NODE\$* obsahuje stejné parametry, z tohoto důvodu ji není nutné popisovat.

4.5.2 Tabulka *RAILWAY_MEZO_NODE\$*

ST36043.RAILWAY_MEZO_NODE\$		
P *	NODE_ID	NUMBER
	NODE_NAME	VARCHAR2 (200 BYTE)
	NODE_TYPE	VARCHAR2 (200 BYTE)
	ACTIVE	VARCHAR2 (1 BYTE)
	PARTITION_ID	NUMBER
	GEOMETRY	SDO_GEOMETRY
	COST	NUMBER
	SUPERNODE_SIGN	VARCHAR2 (1 CHAR)
RAILWAY_MEZO_NODE\$ PK (NODE_ID)		
RAILWAY_MEZO_NODES_IDX (GEOMETRY)		
SYS_C00346892 (NODE_ID)		

Obrázek 22: Tabulka *RAILWAY_MEZO_NODE\$*. Zdroj: vlastní

- *COST* – cena prvku, není vyplněno.
- *SUPERNODE_SIGN* – parametr značí, zda jde o super-vrchol – N, či stanici – S.

4.5.3 Tabulka *RAILWAY_ABS_LINK\$*

ST36043.RAILWAY_ABS_LINK\$		
P *	LINK_ID	NUMBER
	LINK_NAME	VARCHAR2 (200 BYTE)
*	START_NODE_ID	NUMBER
*	END_NODE_ID	NUMBER
	LINK_TYPE	VARCHAR2 (200 BYTE)
	ACTIVE	VARCHAR2 (1 BYTE)
	LINK_LEVEL	NUMBER
	GEOMETRY	SDO_GEOMETRY
	COST	NUMBER
RAILWAY_ABS_LINK\$ PK (LINK_ID)		
RAILWAY_ABS_LINKS_IDX (GEOMETRY)		
SYS_C00283349 (LINK_ID)		

Obrázek 23: Tabulka *RAILWAY_ABS_LINK\$*. Zdroj: vlastní

- *LINK_ID* – jednoznačný identifikátor hrany.
- *LINK_NAME* – název hrany, složen z prefixu *L* a jednoznačného identifikátoru.
- *START_NODE_ID* – jednoznačný identifikátor prvku, ze kterého vychází hrana.
- *END_NODE_ID* – jednoznačný identifikátor prvku, kam hrana směřuje.
- *LINK_TYPE* – typ hrany, není vyplněno.
- *ACTIVE* – značí vypnutí / zapnutí hrany, má pouze dvě hodnoty *Y* a *N*.
- *LINK_LEVEL* – prioritní úroveň hrany, použití např. pro síťové analýzy.
- *GEOMETRY* – obsahuje geometrii prvku.
- *COST* – značí hodnotu (cenu) hrany, např. pro algoritmus nejkratší cesty.

Parametry tabulky hran *mikro* vrstvy *RAILWAY_LINK\$* jsou stejně jako v případě prvků totožné s touto tabulkou.

4.5.4 Tabulka *RAILWAY_MEZO_LINK\$*

ST36043.RAILWAY_MEZO_LINK\$		
P *	LINK_ID	NUMBER
	LINK_NAME	VARCHAR2 (200 BYTE)
*	START_NODE_ID	NUMBER
*	END_NODE_ID	NUMBER
	LINK_TYPE	VARCHAR2 (200 BYTE)
	ACTIVE	VARCHAR2 (1 BYTE)
	LINK_LEVEL	NUMBER
	GEOMETRY	SDO_GEOMETRY
	COST	NUMBER
	BIDIRECTED	VARCHAR2 (1 BYTE)
RAILWAY_MEZO_LINK\$_PK (LINK_ID)		
RAILWAY_MEZO_LINKS_IDX (GEOMETRY)		
SYS_C00346895 (LINK_ID)		

Obrázek 24: Tabulka *RAILWAY_MEZO_LINK\$*. Zdroj: vlastní

- *COST* – cena hrany, není vyplněno.
- *BIDIRECTED* – značí, zda má hrana nějakou orientaci, není vyplněno.

4.5.5 Tabulka *STATION_N*

ST36043.STATION_N		
P *	ID_STATION	NUMBER (8,2)
	STATION	VARCHAR2 (42 BYTE)
	TUDU	VARCHAR2 (20 BYTE)
	KM	NUMBER (5,1)
	GPS	SDO_GEOMETRY
	V_SEGMENTU	VARCHAR2 (1 CHAR)
	STATION20	VARCHAR2 (26 BYTE)
	ODSTAV_KOL	VARCHAR2 (1 BYTE)
	MAKRO	VARCHAR2 (20 BYTE)
STATION_N_PK (ID_STATION)		
STATION_N_INDEX1 (STATION20)		
STATION_N_INDEX2 (STATION)		
STATION_N_INDEX3 (TUDU)		
STATION_N_PK (ID_STATION)		
ST_N_SPATIAL_IDX (GPS)		
TRIM_STATION_N_IDX (TRIM("STATION"))		
TRIM_STATION20_IDX (TRIM("STATION20"))		

Obrázek 25: Tabulka *STATION_N*. Zdroj: vlastní

- *ID_STATION* – jednoznačný identifikátor stanice.
- *STATION* – název železniční stanice.
- *TUDU* – TUDU na kterém se stanice nachází.
- *KM* – kilometr, na kterém se stanice nachází.
- *GPS* – bod středu stanice.
- *V_SEGMENTU* – má pouze dvě hodnoty *A* – ano a *N* – ne. Značí, zda je stanice ve vymezeném segmentu.

- *STATION20* – název stanice je upraven tak, aby se vešel do velikosti dvaceti znaků.
- *ODSTAV_KOL* – má pouze dvě hodnoty *A* – ano a *N* – ne. Značí, zda má stanice odstavnou kolej.
- *MAKRO* – má pouze dvě hodnoty *A* – ano a *N* – ne.

4.5.6 Bindovací tabulky

Každá z tabulek slouží k propojení hodnot tabulek na základě jednoznačných identifikátorů.

ST36043.SUPERNODE_BINDING	
* RAILWAY_MIDDLE_NODE_ID	NUMBER
* RAILWAY_NODE_ID	NUMBER

Obrázek 26: Tabulka *SUPERNODE_BINDING*. Zdroj: vlastní

Tabulka *SUPERNODE_BINDING* propojuje hodnoty tabulky *RAILWAY_MEZO_NODE\$* s tabulkou *RAILWAY_ABS_NODE\$*, případně *RAILWAY_NODE\$*. V této tabulce jsou tedy všechny prvky, které jsou zároveň super-vrcholy.

ST36043.STATION_BINDING	
RAILWAY_MIDDLE_NODE_ID	NUMBER
RAILWAY_STATION_ID	NUMBER

Obrázek 27: Tabulka *STATION_BINDING*. Zdroj: vlastní

Tabulka *STATION_BINDING* slouží také k propojení hodnot dvou tabulek, v tomto případě jde o tabulky *STATION_N* a *RAILWAY_MEZO_NODE\$*. Jak již bylo zmíněno výše, v této tabulce by měly být všechny prvky *mezo* vrstvy, které jsou zároveň stanicemi.

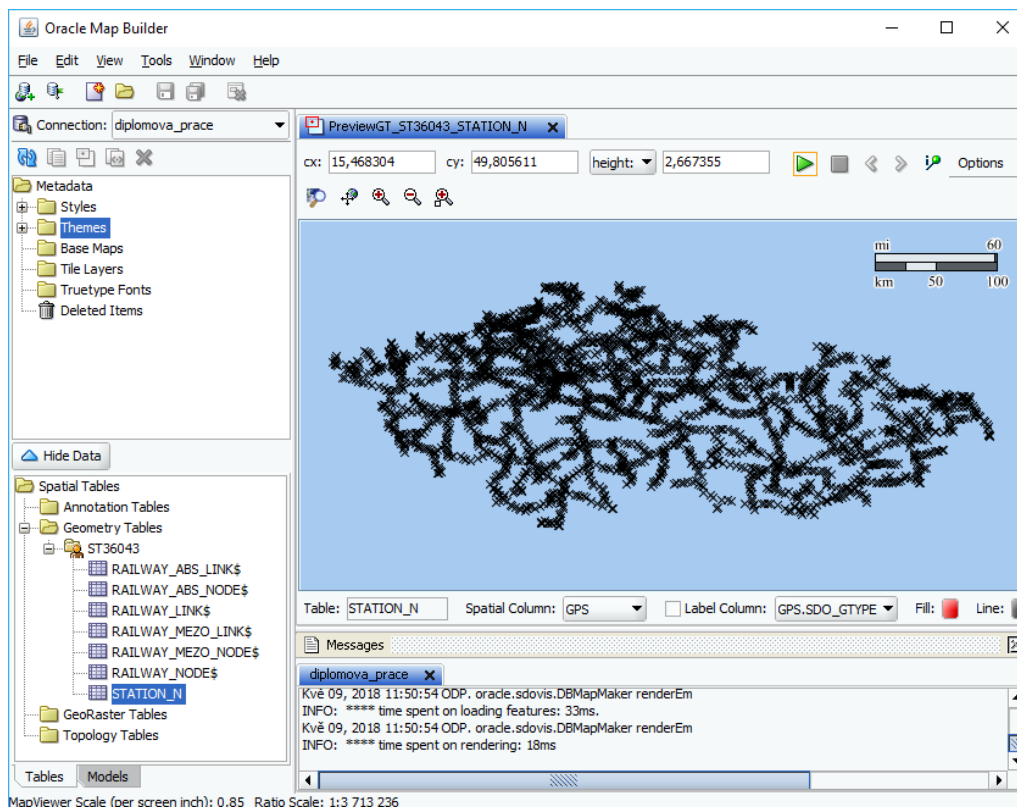
4.6 MapBuilder

Jednou z cest, jak ověřit, zda správně pracují prostorové indexy, je použít nástroj MapBuilder. Jak bylo zmíněno v teoretické části, MapBuilder slouží k vytváření schémat. Lze si v něm tedy vytvořit schéma nad příslušnými daty v databázi. Je k dispozici ke stažení z oficiálních stránek⁶. Nástroj se neinstaluje, lze po stažení jednoduše spustit pomocí příkazové řádky příkazem:

```
java -jar map_builder-12.2.1.3.0.jar
```

⁶ <http://www.oracle.com/technetwork/middleware/mapviewer/downloads/index-100641.html>

Po spuštění se lze připojit k databázi stejným způsobem jako u sqldeveloperu. Jakmile je realizováno připojení, jsou již k dispozici tabulky z databáze.



Obrázek 28: MapBuilder. Zdroj: vlastní

Případné kliknutí pravým tlačítkem myši na tabulku a následně volba *Preview* umožní zobrazení dat. Na obrázku jsou vidět vizualizovaná data z tabulky *STATION_N* (Obrázek 28). V tomto nástroji lze definovat styly pro data. Například různé barvy a tvary, zda budou zobrazeny popisky a z jakého sloupce tabulky. Případně za jakých podmínek tomu tak bude apod. Již zde fungují možnosti zoomu a posunu mapy, z toho vyplývá, že prostorové indexy pracují správně.

4.7 Knihovny potřebné pro implementaci

OJDBC

Pro propojení programu s databází Oracle je zapotřebí knihovny, která to umožní. Knihovna je dostupná ve složce `//product/12.2.0/dbhome_1/jdbc/lib` a nese název `ojdbc.jar`. Knihovna by měla být dodávána jako součást softwaru.

Než je realizováno připojení k databázi, je potřeba načíst její ovladač. To lze zajistit jednoduše pomocí jednoho řádku.

```
Class.forName("oracle.jdbc.OracleDriver").newInstance();
```

Oracle Spatial API

Knihovna nesoucí název *sdoapi.jar* umožňuje načíst datový typ *SDO_GEOMETRY* jako celek a dále s ním pracovat v jazyce Java. Je k dispozici ke stažení z webových stránek⁷. Další velmi důležitou vlastností této knihovny je, že umožňuje definovat vlastní datový typ, pomocí kterého lze kreslit na plátno MapVieweru, ale o tom až dále.

MapView JavaBean-Based API

Knihovna nese název *mv_client.jar* a slouží pro k použití MapVieweru v aplikaci. Knihovna umožňuje volat funkce MapVieweru. Jelikož tento nástroj ještě nebyl instalován, tak podrobnější informace jsou uvedeny až níže.

4.8 Propojení aplikace s databází

Propojení aplikace s databází je realizováno za pomoci návrhového vzoru Singleton. Ten zaručí, že bude vždy použito pouze jediné připojení. Po načtení ovladače se lze připojit k databázi pomocí následujícího příkazu.

```
DriverManager.getConnection(URL, USR, PSW);
```

Parametr *URL* je řetězec (connection string) pro připojení k databázi. Obecný tvar řetězce je *jdbc:oracle:<ovladač>:@<databáze>*, konkrétně *jdbc:oracle:thin:@localhost:1521:orcl*. Další dva parametry značí uživatelské jméno a heslo testovacího uživatele.

4.9 Datová struktura graf

Po realizaci připojení k databázi je třeba data načíst do paměti počítače. K jejich uchování slouží datová struktura graf. Ta je realizována ve třídě *Graph* a implementuje rozhraní *IGraph*. Pro uchování vrcholů v grafu byla nejprve zvolena kolekce *HashMap*. Jako klíč pro vkládání byl použit jednoznačný identifikátor, který je spolu s vrcholem načítán z databáze. Později tato kolekce byla nahrazena z důvodu efektivního vyhledávání dle souřadnic, jak je v práci dále uváděno. Každý vrchol, má svoji kolekci hran.

7 <http://www.oracle.com/technetwork/database/options/spatialandgraph/downloads/software/sp-download-distlic-522138.html>

Modelové třídy programu odrážejí tabulky databáze. Třída *Station* odráží databázovou tabulku *STATION_N* a uchovává informace o stanici. Od abstraktní třídy *Node*, ze které se nedají vytvářet instance, dědí třídy *NodeMicroMakro* a *NodeMezo*. Tyto třídy uchovávají informace o prvku z dané vrstvy, který je uložen v grafu jako jeho vrchol.

Analogicky existuje třída *Link*, od které dědí třídy *LinkMicroMakro* a *LinkMezo*. Třídy uchovávají informace o dané hraně příslušící dvěma vrcholům. Tato hrana je podobně jako vrchol uchovávána v grafu.

Třídy používají datový typ *JGeometry* z knihovny *sdoapi.jar* k uchování a následné práci s objektem *SDO_GEOMETRY*.

Dále je třeba zmínit třídu *Map* a její rozhraní *IMap*. Třída reprezentuje mapu. Stará se o načtení dat z databáze. Dále si drží instanci grafu, volá jeho metody pro přidání načtených dat. Dále metody upravení nebo smazání vrcholů, případně hran. Jelikož třída udržuje aktuálnost dat v grafu, musí také udržovat aktuální data v databázi. To ji zajišťují příslušné databázové objekty, které implementují rozhraní *IDAO*. Třída také poskytuje své rozhraní *IMap* uživatelskému rozhraní.

4.10 MapViewer

Po načtení dat z databáze do paměti počítače je třeba data umět vizualizovat, aby šla následně editovat. Pro vizualizaci byl zvolen nástroj MapViewer, který je propojen s aplikací pomocí knihovny. Na oficiálních stránkách je k dispozici několik balíčků pro různé způsoby použití v produktech.⁸ V práci je použit balíček *QuickStart kit* v aktuální verzi *12.2.1.3* z důvodu velmi snadné instalace.

Pro instalaci stačí spustit dávkový soubor *runMeFirst.bat*. Jakmile je instalace dokončena, MapViewer je dostupný z adresy <http://localhost:8080/mapviewer>. Defaultní přihlašovací údaje jsou *admin* a heslo *welcome1*. Server MapVieweru se nepouští automaticky, ale je třeba ho nastartovat ručně pomocí souboru *startServer.bat*, případně si automatické spuštění nastavit.

MapViewer server tedy nyní běží a je třeba provést konfiguraci. Ve webovém rozhraní Mapvieweru v sekci *Configuration* a záložce MapViewer je konfigurační *xml* soubor, kde je třeba nakonfigurovat datový zdroj. V konfiguračním souboru je již zakomentováno předdefinované nastavení, které je upraveno. Pro šifrované heslo je třeba uvést znak vykřičník

⁸ <http://www.oracle.com/technetwork/middleware/mapviewer/downloads/index.html>

jako první znak řetězce. Po opětovném restartu bude heslo zašifrováno. Nastavení je následující:

```
<map_data_source name="diplomova_prace"
    jdbc_host="localhost"
    jdbc_sid="dp"
    jdbc_port="1521"
    jdbc_user="st36043"
    jdbc_password="p8/0bVbFtpcXh04Q6On6gdppRhbpsXHk"
    jdbc_mode="thin"
/>
```

Změny je třeba uložit a provést restart serveru. Po restartu již je v sekci *Admin* na záložce *Data sources* dostupný nový datový zdroj.

Výše byla představena knihovna *mv_client.jar*, ta je dostupná ve složce MapVieweru, přesněji *mapviewer12c_qs/apps/mvdemo.war/WEB-INF/lib*. Dále jsou představeny její metody, které jsou v práci použity.

4.10.1 Inicializace MapVieweru

MapVieweru je třeba nastavit adresu URL služby. V našem případě se jedná o *http://localhost:8080/mapviewer/omserver*. To lze zařídit zaprvé v konstruktoru.

```
MapView mv = new MapViewer("http://localhost:8080/mapviewer/omserver");
```

V druhém případě pomocí metody *setServiceURL*.

```
MapView mv = new MapViewer();
mv.setServiceURL("http://localhost:8080/mapviewer/omserver");
```

Následující metody jsou nezbytné pro správné fungování.

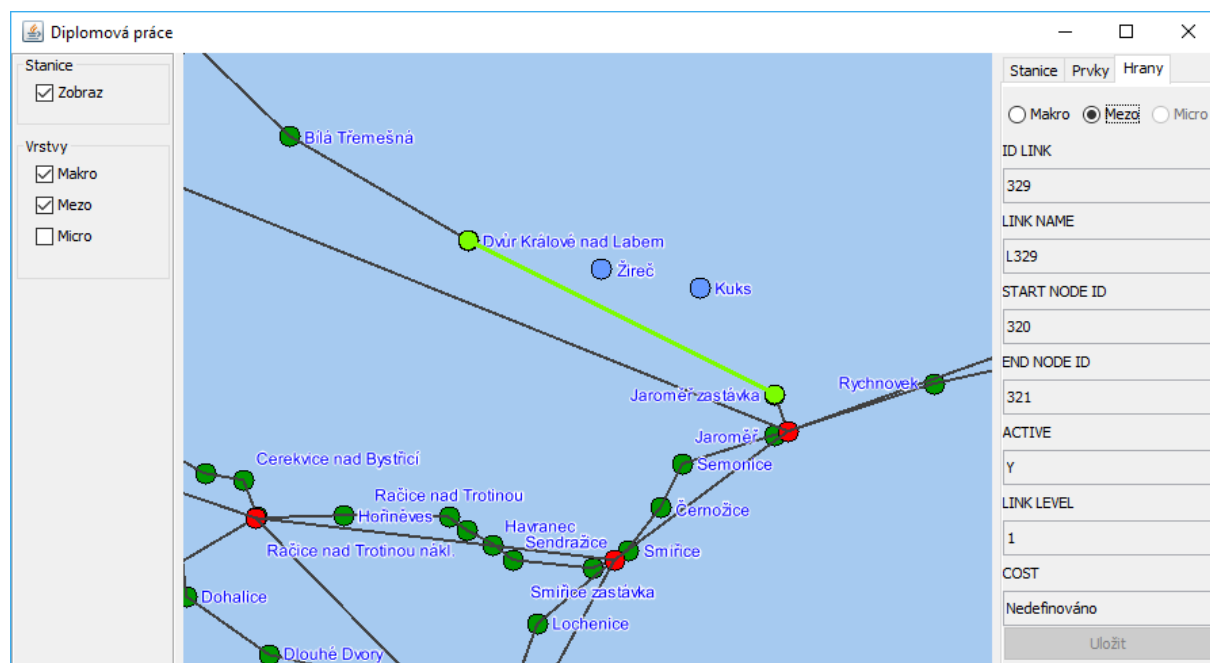
- `setDataSourceName("diplomova_prace");` - nastavení zdroje,
- `setMapRequestSRID(8307);` - nastavení souřadnicového systému.

Jelikož byla schémata pro data definována za pomoci nástroje MapBuilder, je vhodné je také nastavit při inicializaci a případně povolit, nebo zakázat.

- `addPredefinedTheme("THEME_STATION");` - přidání předdefinovaného schématu,
- `setThemeEnabled(false, "THEME_STATION");` - povolení / zakázání schématu.

Definice schémat je dostupná ve třídě *EnumTheme*. Nyní je již možné pomocí metody *run* vytvořit požadavek, který je předán serveru MapVieweru. Na serveru dojde k vyhodnocení,

následnému vykonání operací a odeslání odpovědi klientovi. Instance MapVieweru je ve třídě MapView. Tato třída dědí od třídy *JPanel* a umožňuje vizualizaci a kreslení. Jestliže schémata byla správně načtena, je již možné data v aplikaci vizualizovat (Obrázek 29).



Obrázek 29: Ukázka vizualizovaných dat. Zdroj: vlastní

4.10.2 Nastavení parametrů žádosti MapVieweru

V této podkapitole jsou představeny některé *set* metody, které jsou nastaveny před odesláním požadavku na server.

- `setAntiAliasing()` – nastavení vykreslení pomocí antialiasingu,
- `setBackgroundColor()` – nastavení barvy pozadí,
- `setCenterAndScale()` – vycentrování a nastavení požadovaného zoomu,
- `setDeviceSize()` – nastavení rozměrů mapy,
- `setMapTitle()` – nastavení titulku mapy,
- `setMapLegend()` – nastavení legendy, má několik variant,
- `setThemeEnabled()` – nastavení zapnutí / vypnutí schématu.

Případně pomocí objektu *ScaleBar* a jeho metody `setScaleBar` lze přidat měřítko mapy.

4.10.3 Vkládání vlastností

Kromě základního nastavení lze také přidávat nové body, úsečky a třeba dynamicky načítat schémata. Také lze vytvářet a přidávat nové styly. Tato část kapitoly představuje výčet metod, pomocí kterých toho lze dosáhnout.

- `addPointFeature()` – vloží bod na mapu,
- `addLinearFeature()` – vloží úsečku na mapu,
- `addColorSchemeStyle()` – přidá barevný styl,
- `addLineStyle()` – přidá styl pro úsečku,
- `addTextStyle()` – přidá styl pro text,
- `addMarkerStyle()` – přidá styl pro vektorovou značku.

4.10.4 Odeslání žádosti na server

Kromě metody *run*, která odesílá žádost na server, existují i další.

- `ZoomIn()` – přiblížení mapy,
- `ZoomOut()` – oddálení mapy,
- `Pan()` – posun mapy.

Metody existují v několika variantách, nemusí se po nich volat metoda *run*.

4.10.5 Získání odpovědi ze serveru

Po odeslání žádosti server poskytne odpověď.

- `getGeneratedMapImage()` – vrátí vygenerovaný obrázek,
- `getMapMBR()` – vrací souřadnice hranic mapy,
- `getDataSources()` – vrací seznam aktuálně dostupných zdrojů na serveru,
- `getUserPoint()` – vrací bod mapy, např. při kliknutí myši.

4.10.6 Kreslení na mapu

Knihovna dále podporuje kreslení na mapu, aniž by musela být poslána žádost na server. To je pro aplikaci velice důležité. V případě, že by tato možnost chyběla a mělo dojít k přesunu hrany z jednoho vrcholu na druhý, tak by se musela neustále volat metoda *run* a nebylo by možné plynule provádět kreslení.

`MapView` realizuje datový typ *LiveFeature*. Po vytvoření instance tohoto typu lze pomocí jeho metody *setShape(Shape shape)* nastavit tvar. Příklad vytvoření tvarů:

```
Shape elipsa = new Ellipse2D.Double(bod.x - 5, bod.y - 5, 10, 10);
Shape usecka = new Line2D.Double(Point start, Point cil);
```

Jakmile je nastaven tvar, je možné ho přidat pomocí metody *drawLiveFeature()* do instance `MapView` a vykreslit, aniž by musel být odeslán požadavek na server. Zdroj, ze kterého je čerpáno v této kapitole, je oficiální dokumentace [15]. Bohužel v ní není uvedeno, že tato

metoda existuje. Žádný jiný zdroj o použití či existenci se nepodařilo vyhledat. Je vhodné tuto metodu představit podrobněji, k čemu je v práci využívána.

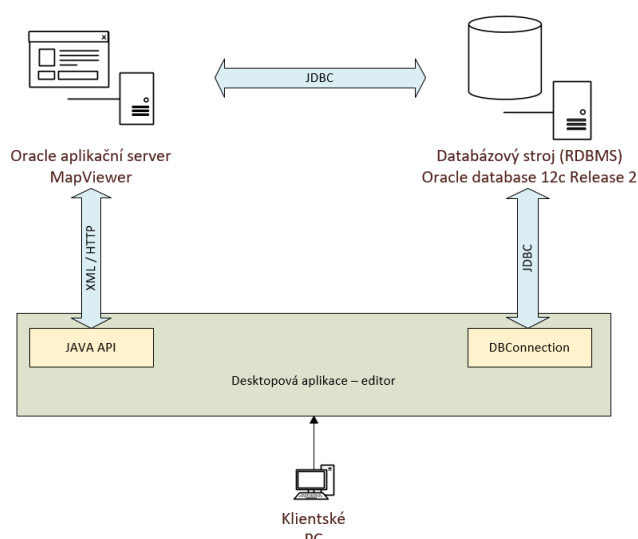
```
drawLiveFeature(LiveFeature lf, Graphics2D g2, Color stroke, Color fill,  
double pointRadius, double strokeWidth)
```

Prvním parametrem je *LiveFeature*, kterému je přiřazen tvar. Druhým parametrem je *Graphics2D*. Jedná se o rozšíření třídy *Graphics*, která reprezentuje grafické objekty. Dalším parametrem je barva ohraničení tvaru. Čtvrtý parametr je barva výplně. Pátý parametr je rádius tvaru, který v práci není použit a je nastaven na hodnotu 0. Posledním parametrem je šířka ohraničení.

Metoda je volána v metodě *paintComponent* a stará se o vykreslení na plochu *MapVieweru*.

4.11 Celková koncepce řešení

Celová koncepce se skládá z databázového stroje, aplikačního serveru *MapVieweru* a desktopové aplikace – editoru (Obrázek 30).



Obrázek 30: Celková koncepce řešení. Zdroj: vlastní

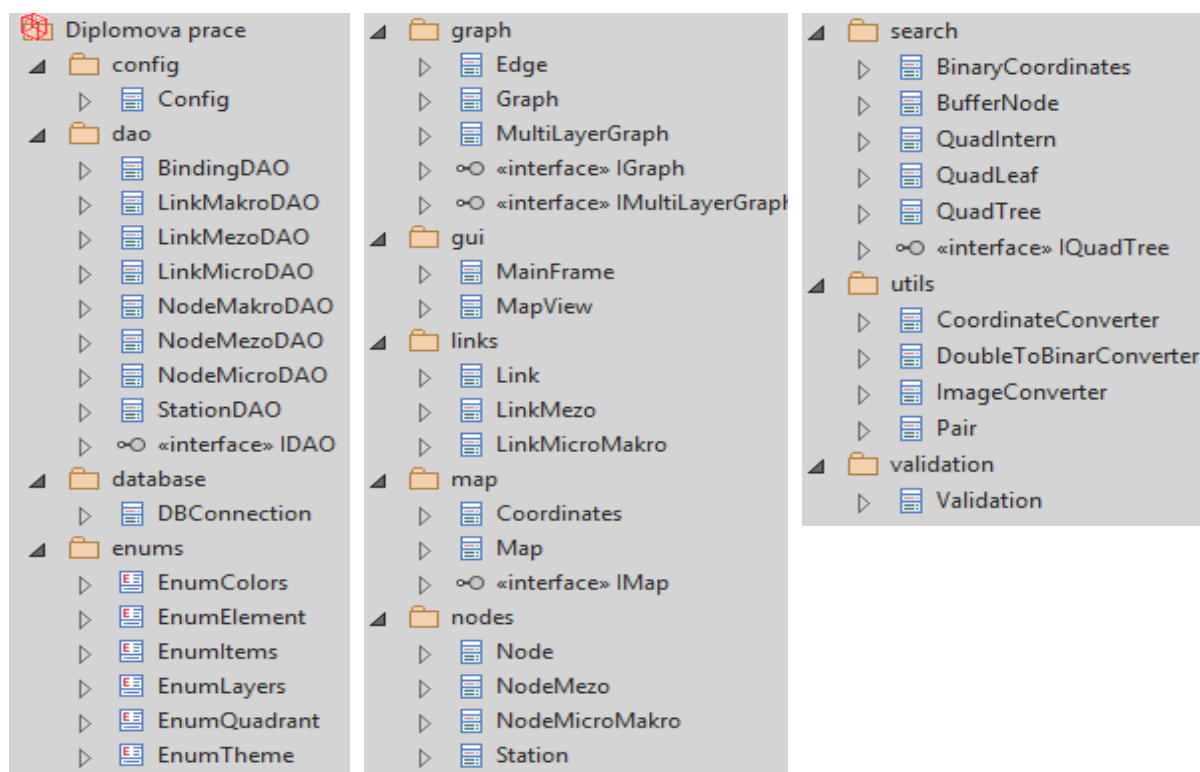
Na databázovém stroji je nainstalován systém *Oracle database 12c Release 2* ve variantě Enterprise pro podporu tabulek s prostorovými (spatial) daty. Klientská aplikace naváže za pomoci třídy *DBConnection* a knihovny *JDBC* spojení. Třída *DBConnection* je implementována jako návrhový vzor Singleton a zaručí použití jediného připojení.

Komunikace klientské aplikace s aplikačním serverem probíhá následovně:

- Aplikace vytvoří žádost pro získání mapy, zavolá přes protokol *http* servlet MapVieweru, kterému pošle XML požadavek.
- MapViewer požadavek analyzuje a z tabulek s prostorovými daty vygeneruje mapu, která je uložena jako obrázek v cache paměti aplikačního serveru.
- Aplikační server pošle klientovi odpověď ve formě XML, který obsahuje *url* adresu obrázku.
- Klient parsuje XML ze kterého získá *url* adresu obrázku mapy, ta je následně zobrazena na obrazovku.

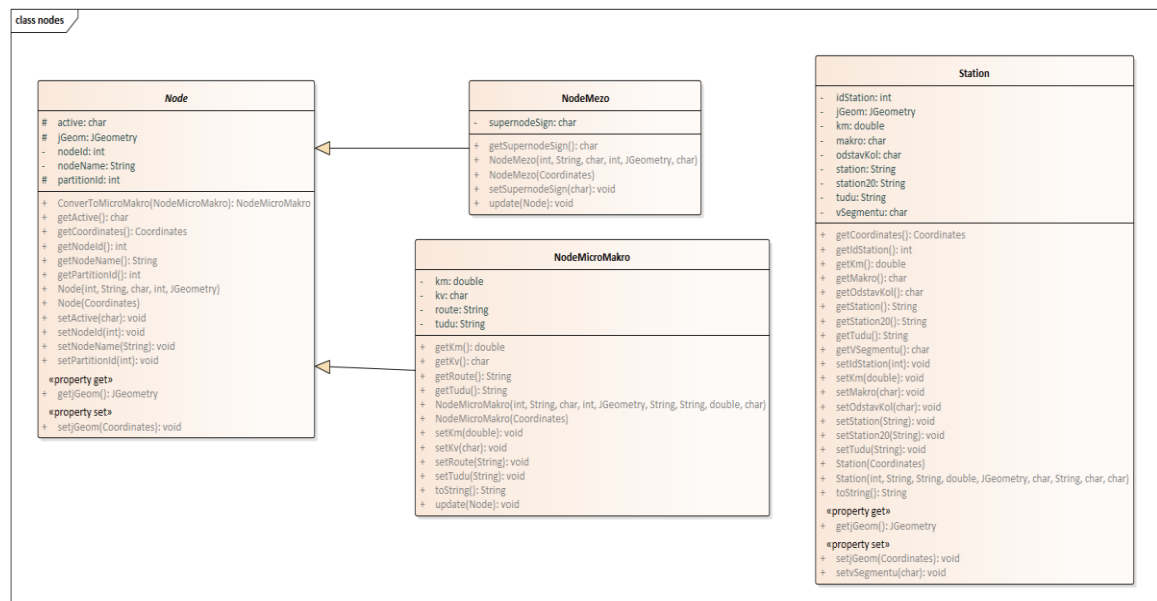
4.12 Struktura projektu aplikace

Projekt aplikace je složen z dvanácti balíčků (Obrázek 31)



Obrázek 31: Struktura projektu aplikace. Zdroj: vlastní

Nodes

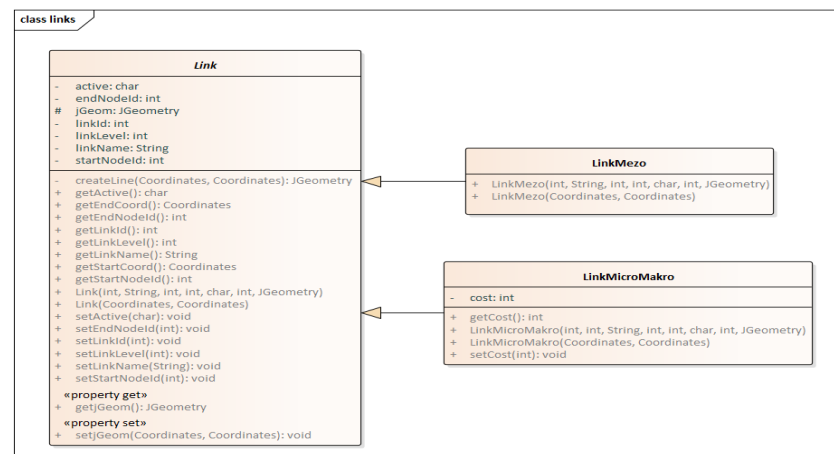


Obrázek 32: Balíček nodes. Zdroj: vlastní

Balíček *nodes* obsahuje třídy, které reprezentují vrcholy grafu a odráží databázové tabulky *nodes*. Třída *Node* je abstraktní, nelze z ní tedy vytvářet instance.

Links

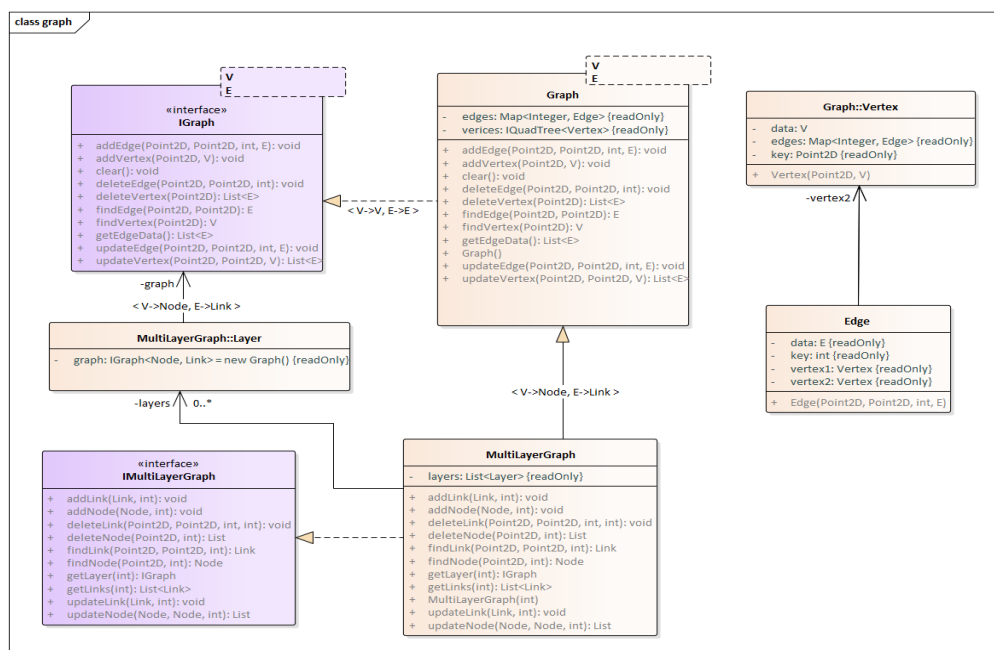
Balíček *links* je obdobou balíčku *nodes*, třídy zde však reprezentují hrany grafu a odráží databázové tabulky *links*. Třída *Link* je opět abstraktní.



Obrázek 33: Balíček links. Zdroj: vlastní

Graph

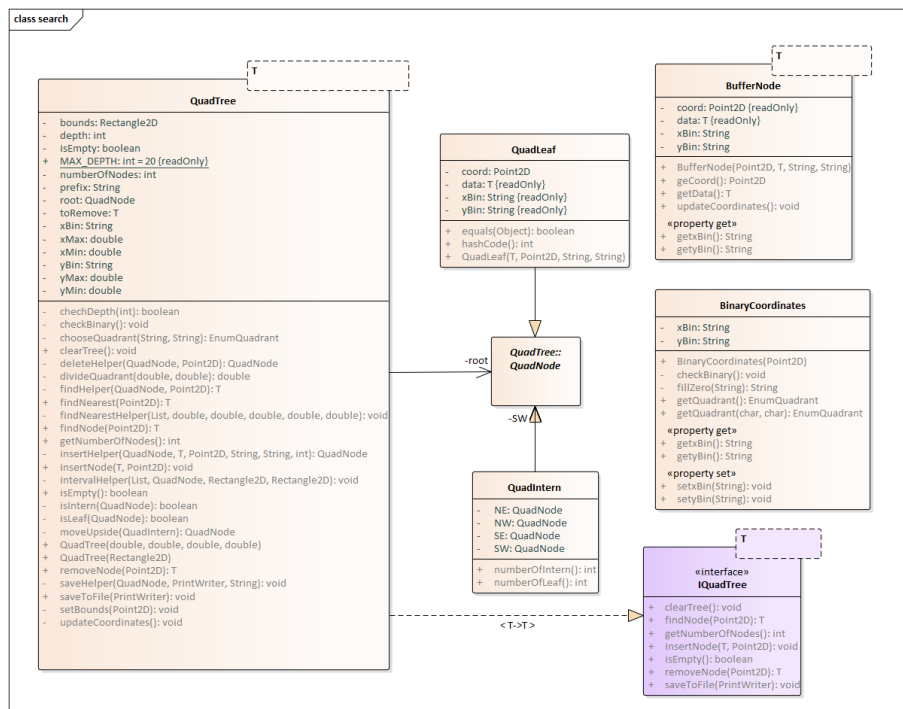
V balíčku *graph* existuje třída *Graph* a její vnitřní třídy *Vertex* a *Edge*. Třída reprezentuje datovou strukturu neorientovaný graf a je generická. Generický parameter *V* představuje data pro vrcholy, obdobně parameter *E* data hran. Třída má vlastní rozhraní *IGraph*. Dále je v balíčku třída *MultiLayerGraph* s rozhraním *IMultiLayerGraph*, ta spravuje záznamy v příslušné vrstvě grafu.



Obrázek 34: Baliček graph. Zdroj: vlastní

Search

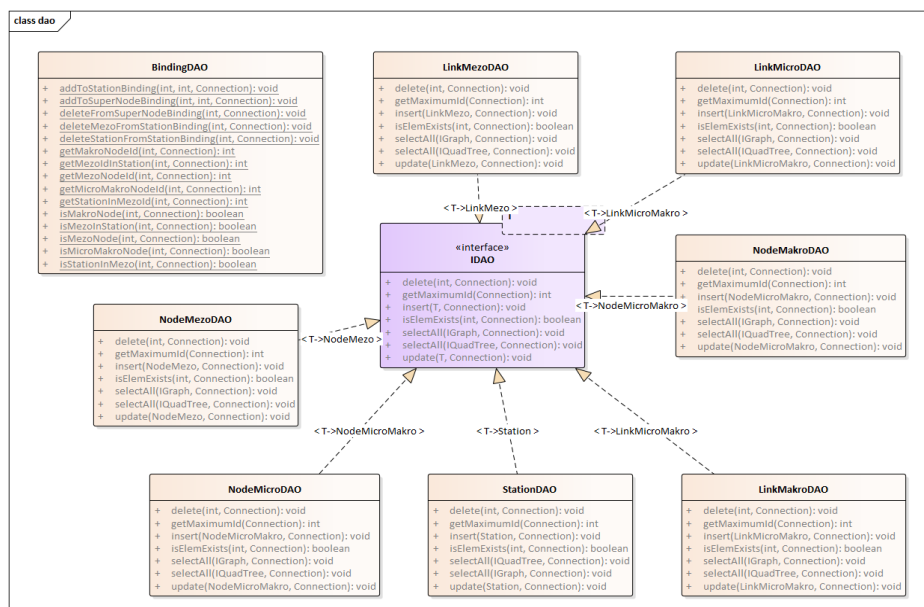
Balíček *search* uchovává třídy quadstromu. Třída *QuadTree* má vlastní rozhraní *IQuadTree*, přes které jsou její metody, obdobně jako u třídy *Graph*, přístupné. Třída *QuadTree* obsahuje abstraktní vnitřní třídu *QuadNode*, ta představuje prvky stromu. Tyto prvky, jak vychází z principu tohoto typu quadstromu, jsou dvojího typu. A to interní – třída *QuadIntern* a listy – třída *QuadLeaf*. O navigaci ve stromu se stará třída *BinaryCoordinates*, princip navigace byl popsán v teoretické části. Třída *BufferNode* je pouze třídou pomocnou pro rekurzivní vyhledávání. Generický parametr *T* reprezentuje data listů stromu.



Obrázek 35: Balíček search. Zdroj: vlastní

Database a dao

Balíček *database* obsahuje třídu *DBConnection*, která se stará o propojení s databází. Balíček *dao* obsahuje třídy reprezentující databázové objekty. Tyto třídy implementují rozhraní *IDAO* a spravují záznamy v databázi.



Obrázek 36: Balíček dao. Zdroj: vlastní

Map

Balíček *map* uchovává třídu *Map* a její rozhraní *IMap*. Třída *Map* si drží instanci grafu a udržuje aktuálnost dat v databázi. Třída poskytuje své metody za pomoci rozhraní *IMap* grafickému uživatelskému rozhraní. V balíčku je dále obsažena třída *Coordinates* reprezentující zeměpisné souřadnice.

Gui a Validation

Balíček *gui* obsahuje třídy reprezentující grafické rozhraní aplikace. Jedná se o *MapView*, což je plátno, které slouží k vykreslení MapVieweru. Dále třídu *MainFrame*, která představuje hlavní okno aplikace s ovládacími prvky. Třída *Validation* je umístěna ve stejnojmenném balíčku a stará se o to, aby nedošlo k uložení chybných dat do databáze.

Utils

Balíček *utils* obsahuje různé pomocné třídy aplikace. Za zmínku stojí třída *ImageConverter*, která se stará o konverzi *Image* na *BufferedImage*, který je vyžadován pro plynulost MapVieweru. Dále například třída *CoordinateConverter* pro převod zeměpisných souřadnic na souřadnice obrazové.

Enums a config

Balíček *enums* obsahuje výčty prvků, které jsou použity napříč aplikací, balíček *config* uchovává základní nastavení aplikace a MapVieweru.

4.13 Funkce editoru

V práci již bylo představeno, jakým způsobem jsou data načtena, uchovávána a také způsob jejich vizualizace. Nyní je potřeba představit, jak s daty pracovat. Jde o editaci, přidávání nových záznamů a odstranění současných. V té nejvyšší abstrakci editor pracuje s dvěma typy dat napříč vrstvami. Jedná se o vrcholy a hrany. Editor také musí umět přepínat mezi vrstvami. To je jednoduché, jelikož stačí pro daný moment pouze zakázat příslušné schéma.

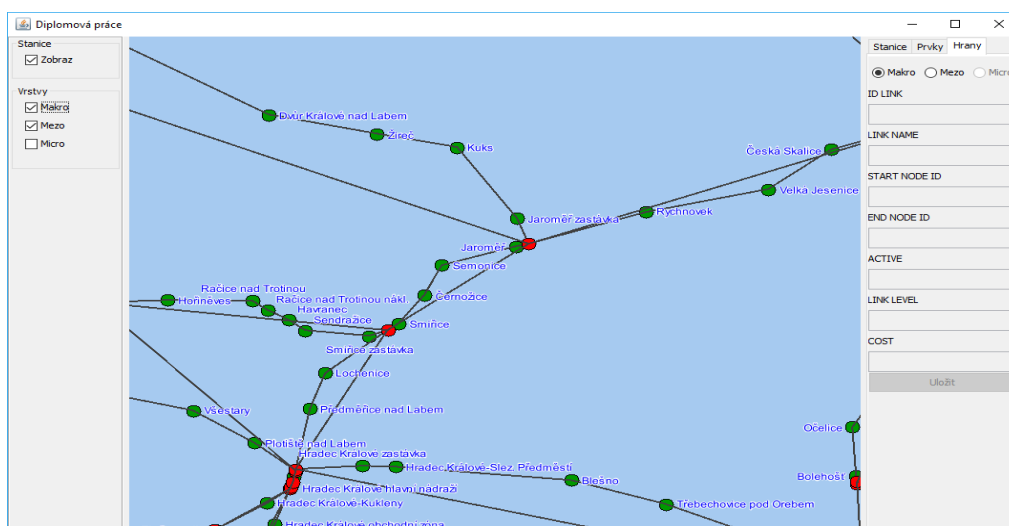
V některých datech je chyba, jelikož každá stanice z daného kraje by měla mít odpovídající prvek v *mezo* vrstvě, a tedy i v tabulce *STATION_BINDING*. Barvy, které reprezentují dané vrstvy, jsou zobrazeny níže (Tabulka 7).

Tabulka 7: Barevné zobrazení vrstev.

Vrstva	Barva
Stanice	Modrá
Makro	Červená
Mezo	Zelená
Mikro	Žlutá

Zdroj: vlastní

Jak lze vypožiorovat z přechozího obrázku, některé stanice nejsou v *mezo* vrstvě (Obrázek 29). Tedy další důležitou funkcí editoru je realizace přesunu stanice do *mezo* vrstvy. S tím je spojená i editace hran (Obrázek 37).



Obrázek 37: Přesun do mezo vrstvy. *Zdroj: vlastní*

Jak lze vyvodit, elementy reprezentující vrcholy a hrany je třeba označovat kliknutím myši. K jejich uchování slouží graf, přesněji struktura *HashMap*, kam jsou prvky vkládány dle *Id*. Je tedy třeba změnit klíč, aby prvky byly vkládány dle jiného jednoznačného identifikátoru – souřadnic. Při kliknutí myši na vrchol jsme schopni zjistit pouze přibližné souřadnice vrcholu. Je tedy nutné projít všechny vrcholy grafu a vrátit prvek o souřadnicích, který je nejbližší souřadnicím myši. Tato varianta by se dala akceptovat v případě *makro* nebo *mezo* vrstvy, kde není tolik dat. Nastane-li varianta, že v *mikro* vrstvě budou data z celé ČR, uživatel by již musel čekat, než se mu daný prvek po kliknutí myši zvýrazní. Tento problém řeší až následující kapitola. Nyní se spokojme s neefektivní variantou procházení všech prvků a vrácení toho nejbližšího.

Další nezbytnou funkcí editoru je zoom na pozici myši. Při spuštění je také signalizováno, pokud se nelze připojit k serveru MapVieweru.

Vkládání / editace / mazání hran je triviální záležitostí, jelikož pracujeme vždy pouze v rámci jedné vrstvy. Práce s vrcholy je oproti hranám podstatně složitější, protože je třeba pracovat napříč vrstvami.

4.13.1 Hrany

Při výběru hrany se není potřeba připojovat k databázi, protože data jsou načtena v paměti počítače. Vkládáme-li novou hranu, je pomocí myši zvolen počáteční vrchol a tažením je vytvořena hrana ke koncovému vrcholu. Poté je na uživateli, jak se zachová. Buď může nakreslenou hranu zrušit, nebo ji uložit. V případě uložení je otevřeno připojení k databázi a změny se do databáze uloží. U odebrání je potřeba data smazat jak z paměti počítače, tak i z databáze.

4.13.2 Makro vrstva

Na práci s vrcholy je nahlíženo z pohledu vrstev. Pokud nebudeme uvažovat selekci vrcholu, pak přidání nového je nejjednodušší variantou. Při přidání do *makro* vrstvy je třeba ten samý vrchol přidat i do *mezo* a *mikro* vrstev a také do tabulky *SUPERNODE_BINDING*. Pokud je editován prvek této vrstvy, je třeba podle toho upravit i příslušné prvky z ostatních vrstev. V případě, že jsou editovány souřadnice vrcholu, je krom toho také třeba upravit souřadnice incidenčních hran všech příslušných vrcholů v rámci vrstev. Při odebrání je odebrán prvek z *makro* vrstvy a jeho incidenční hrany. Dále příslušné prvky z *mezo* a *mikro* vrstvy a jejich incidenční hrany. Je nutné také odebrat záznam z bindovací tabulky *SUPERNODE_BINDING*.

4.13.3 Mezo vrstva

Vkládání prvku do *mezo* vrstvy není v této vrstvě třeba realizovat, jelikož vkládání super-vrcholů je realizováno v *makro* vrstvě. Přidání vrcholu, který reprezentuje stanici je prováděno přes vrstvu stanic. Editace a odebrání super-vrcholu je analogické s *makro* vrstvou. Jedná-li se o prvek reprezentující stanici, tak při editaci (změna souřadnic *mezo* prvku) je provedena i editace stanice (také změna souřadnic). Editor tuto možnost editace umožňuje, ale v praxi to nejspíš nebude příliš časté. Proto je uživatel informován, zde tuto změnu lze opravdu provést. Při odebrání vrcholu je však stanice zachována a odebrána pouze reference na stanici z tabulky *STATION_BINDING*.

4.13.4 Mikro vrstva

Přidáním prvku do *mikro* vrstvy se vkládá prvek, který není super-vrcholem. Při editaci a odebírání prvku, který je super-vrcholem se postupuje analogicky jako v případě *makro* vrstvy. Při editaci prvku, který není super-vrcholem, jsou upraveny též jeho hrany. Tyto hrany jsou v případě odebírání prvku odebrány spolu s ním.

4.13.5 Stanice

Vrstva stanic, kromě přidávání, editace a mazání, umožňuje také přesun stanice do *mezo* vrstvy v případě, že stanice v *mezo* vrstvě nemá odpovídající prvek. Při přidání nové stanice však není zároveň vytvořen i *mezo* prvek. Toho lze dosáhnout následným přesunem stanice do *mezo* vrstvy. Při editaci souřadnic stanice jsou spolu s nimi editovány i případné souřadnice *mezo* prvku, uživatel je opět o této případně změně informován. Při odebrání stanice je odebrán i příslušný prvek z *mezo* vrstvy a jeho incidenční hrany. Dále je odebrán příslušný záznam z tabulky *STATION_BINDING*.

4.13.6 Validace

Data každého elementu je třeba před úpravami validovat. O validaci se stará třída *Validation*, kde je případná pravidla možno upravovat.

- *Station* – první písmeno vždy upraveno na velké, maximální délka 50 znaků.
- *Station20* – první písmeno vždy upraveno na velké, maximální délka 20 znaků.
- *TUDU* – pevná délka šesti znaků, kombinace velkých písmen a čísel, písmena automaticky upravena na velká.
- *Km* – kladná hodnota double, ne větší než maximální povolená.
- *Latitude, Longitude* – kladná hodnota double, v rámci mezí souřadnic ČR.
- *V_SEGMENTU, ODSTAV_KOL, MAKRO* – kontrola, zda jde o znak, pouze hodnoty *A* a *N*.
- *ACTIVE* – kontrola, zda jde o znak, pouze hodnoty *Y* a *N*.
- *PARTION_ID, COST* – kontrola, zda jde o integer. Pouze hodnota rovna jedné.
- *ROUTE* – kombinace velkých písmen a čísel, písmena automaticky upravena na velká. Délka řetězce 10 nebo 11 znaků.
- *KV* – kontrola, zda jde o znak, pouze hodnoty *K* a *V*.
- *SUPERNODE_SIGN* – kontrola, zda jde o znak, pouze hodnoty *S* a *N*.

4.13.7 Cache paměť

Při přesunu vrcholu v jakékoliv vrstvě na novou pozici nejsou okamžitě provedeny změny na plátně MapVieweru, i když do databáze jsou tyto změny promítnuty. Je to způsobeno *cache* pamětí MapVieweru. V knihovně neexistuje žádná metoda, která by paměť umožnila vyčistit za běhu programu vždy před případnou změnou souřadnic. *Cache* je třeba vypnout v administraci MapVieweru, ve stejném *xml* souboru, jako byl nastavován datový zdroj.

```
<spatial_data_cache max_cache_size="0" report_stats="false" />
```

4.14 Zakomponování Quadstromu

Nyní by šlo s editorem z hlediska funkčnosti pracovat podle představ. Přidávat vrcholy v rámci vrstev, ty editovat a mazat. Dále mezi vrcholy vkládat hrany a také provádět jejich úpravy a případné odstranění. V neposlední řadě také přesunovat stanice do *mezo* vrstvy. Vrstvy přepínat a používat zoom mapy na pozici myši za použití kolečka.

Zbývá vyřešit, jakým způsobem efektivně vyhledávat data. Jako efektivní datová struktura byl zvolen regionální quadstrom. Díky quadstromu lze efektivně vyhledat nejbližší prvek při kliknutí myši na plátno MapVieweru. Quadstrom lze realizovat buď jako vyhledávací struktura nad grafem, nebo jako struktura, která nahradí již použitou strukturu *HashMap*. V prvním případě by musela být data při odebírání z grafu též odebrána z quadstromu. Byl tedy použit druhý případ a nahrazena struktura *HashMap*.

Při volbě struktury bylo rozhodováno mezi quadstromem a 2D stromem. Rozhodujícím faktorem byly výsledky testu z práce [26]. V té autor mimo jiné provádí porovnání právě těchto datových struktur. Vybudování struktury před spuštěním programu a následné vyhledávání nejbližšího prvku jsou tedy nejvíce využívanými metodami struktury. Z tohoto důvodu, na základě výsledků práce, byl zvolen právě quadstrom (Obrázek 38).

		2D strom	Range strom	Prioritní vyhledávací strom	Quad strom	Grid soubor
Předem známá data	Vybudování	8,98%	80,02%	17,70%	7,05%	100%
	Vložení	1,29%	22,07%	4,01%	2,95%	100%
	Odebrání	2,82%	18,34%	4,43%	2,22%	100%
	Hledání prvku	1,19%	2,85%	1,38%	1,60%	100%
	Hledání nejbližšího prvku	3,41%	4,05%	1,72%	1,43%	100%
	Intervalové hledání (obdélník)	19,57%	35,36%	20,33%	25,30%	100%
	Intervalové hledání (kruh)	9,06%	23,42%	12,95%	17,71%	100%
	Intervalové hledání (pás)	5,24%	4,59%	6,15%	5,90%	100%
Dynamický proud dat	Vybudování	0,04%	2,84%	0,16%	0,10%	100%
	Vložení	1,38%	24,84%	3,89%	2,85%	100%
	Odebrání	2,86%	28,88%	4,28%	2,18%	100%
	Hledání prvku	1,26%	3,52%	1,94%	1,69%	100%
	Hledání nejbližšího prvku	4,82%	5,03%	2,69%	1,44%	100%
	Intervalové hledání (obdélník)	21,02%	39,42%	23,75%	24,77%	100%
	Intervalové hledání (kruh)	12,36%	25,02%	15,40%	18,30%	100%
	Intervalové hledání (pás)	6,10%	4,91%	5,69%	5,84%	100%

Obrázek 38: Porovnání datových struktur. Zdroj: [26]

4.15 Problém duplicitních souřadnic

Před výměnou datové struktury, než byla nahrazena quadstromem, byl změněn současný klíč struktury *id* za souřadnice. Při pokusu o vybudování struktury program skončil s chybovou hláškou, že v dané vrstvě již existuje několik prvků, které mají stejné souřadnice. Tato situace je však chybná a nikdy by neměla nastat. Existuje několik prvků, které mají duplicitní souřadnice (Tabulka 8).

Tabulka 8: Duplicitní prvky

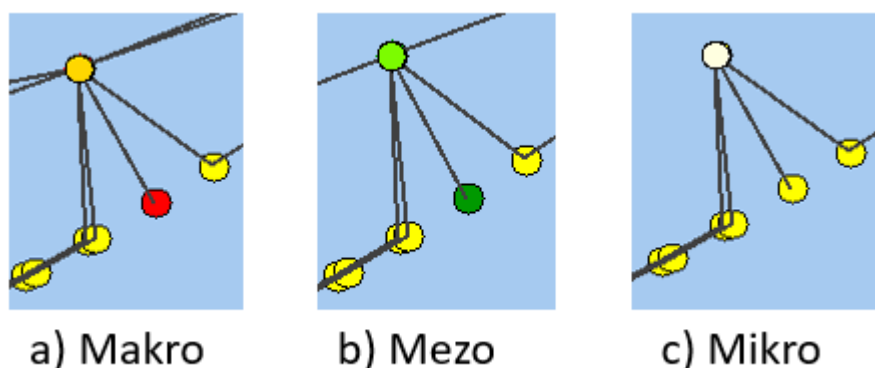
Duplicity				
Vrstva	Souřadnice	Prvek	Hrany(z)	Hrany (do)
MAKRO	15.3413030555513, 50.0258055559794	N678	L94 , L95, L96	L93
MAKRO	15.3413030555513, 50.0258055559794	N679	žádná	L94
MAKRO	15.633383055557,49.8936527782016	N716	žádná	L102
MAKRO	15.633383055557,49.8936527782016	N10703	žádná	L101
MEZO	15.3413030555513, 50.0258055559794	N502	žádná	L554
MEZO	15.3413030555513, 50.0258055559794	N501	L554 ,L555,L557	L553
MEZO	15.633383055557,49.8936527782016	N9	žádná	L8
MEZO	15.633383055557,49.8936527782016	N11	žádná	L9
MICRO	16.0651433330112, 50.2136858336131	N5473	žádná	L3391
MICRO	16.0651433330112, 50.2136858336131	N723	L55	L3390
MICRO	16.1523241665628, 50.2892397223579	N5246	L6413	žádná
MICRO	16.1523241665628, 50.2892397223579	N815	L149	L6412
MICRO	15.3413030555513, 50.0258055559794	N678	L9 , L10,L11	L8
MICRO	15.3413030555513, 50.0258055559794	N679	žádná	L9
MICRO	15.633383055557,49.8936527782016	N716	L49	žádná
MICRO	15.633383055557,49.8936527782016	N10703	žádná	L10337

Zdroj: vlastní

V tabulce jsou všechny prvky, které mají duplicitní souřadnice. Barevné odlišení značí duplicitu v rámci jedné nebo více vrstev. Editor lze použít k odstranění duplicit, není-li prozatím pozměněn klíč.

4.15.1 První duplicita

Nejjednodušší je odstranit duplicitu, která je v tabulce označena modrou barvou. Ke smazání je zvolen prvek s označením N679. Posuneme-li prvek, který je nad ním (N678) dostaneme následující zobrazení v rámci vrstev.



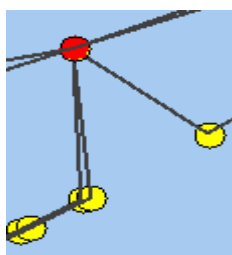
Obrázek 39: První duplicita. *Zdroj: vlastní*

Prvek *N679* je označen červenou barvou. Jeho duplicita je v rámci stejné vrstvy označena barvou oranžovou. Zelená barva značí prvek *N502*, který má být také odebrán. Světle zelená prvek, která má být zachována. Podobně je to i u *mikro* vrstvy. Chceme tedy odebrat prvek *N679*.

Jeho smazáním se stane následující:

- Odebrána hrana *L94* (z *makro* vrstvy),
- Odebrána hrana *L9* (z *mezo* vrstvy),
- Odebrána hrana *L554* (z *mikro* vrstvy),
- Odebrán vrchol *N679* (z *makro* vrstvy),
- Odebrán vrchol *N679* (z *mikro* vrstvy),
- Odebrán vrchol *N502* (z *mezo* vrstvy),
- Odebrán záznam z tabulky *SUPERNODE_BINDING*. (propojení *N502* a *N679*).

Po odebrání prvku vznikne následující stav:

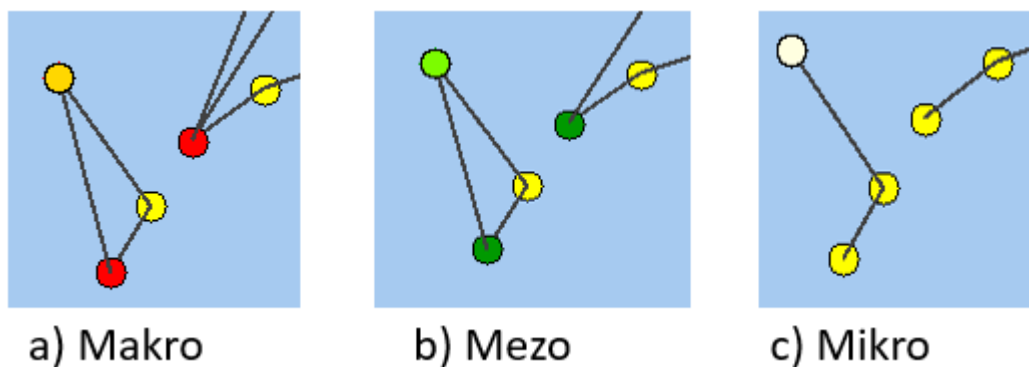


Obrázek 40: Stav po odebrání první duplicity. *Zdroj: vlastní*

Po odebrání je nutné vrátit posunutý prvek na původní místo, k tomu lze použít příslušné souřadnice z výše uvedené tabulky (Tabulka 8).

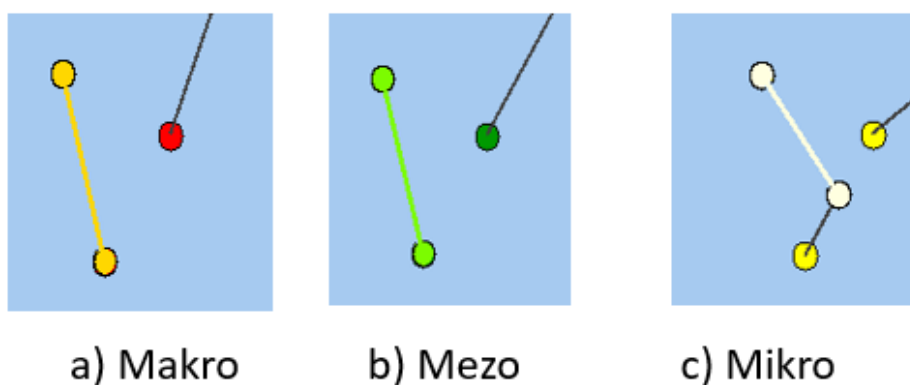
4.15.2 Druhá duplicita

Odstranění druhé duplicity, označené zelenou barvou v tabulce, je složitější. Oproti předchozímu případu nestačí totiž odebrat pouze jeden vybraný prvek. Stanovme případ, kdy je třeba odebrat prvek s označením *N716*. Byla by odebrána i hrana *L102*, kterou je třeba zachovat. Pokud by byl zvolen prvek *N10703* byla by odebrána hrana *L101*, která má také zůstat zachována. Je tedy nutné správně přemístit hrany před smazáním. Po posunutí vrchního prvku *N716*, který je označen oranžovou barvou, vznikne následující rozložení.



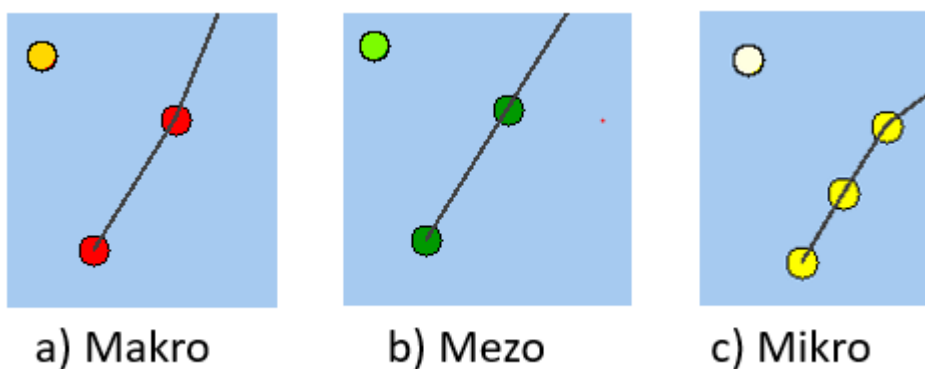
Obrázek 41: Druhá duplicita. Zdroj: vlastní

Z výše uvedeného rozložení (Obrázek 41) lze vypožorovat, že pro odstranění označeného prvku stačí přesunout příslušnou hranu v dané vrstvě (Obrázek 42).



Obrázek 42: Hrany k přesunu. Zdroj: vlastní

Situaci po přesunu znázorňuje následující obrázek (Obrázek 43).

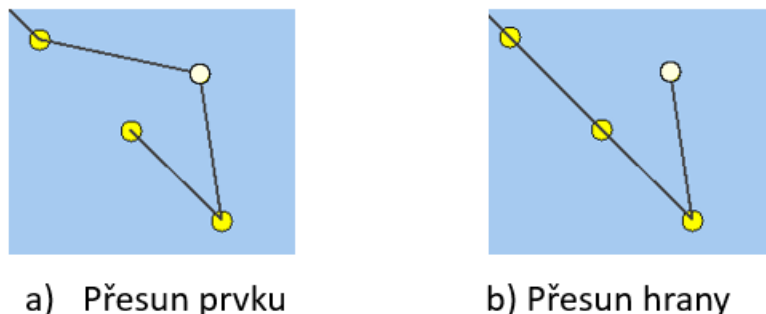


Obrázek 43: Situace po přesunu. Zdroj: vlastní

Nyní je již možné označený prvek odstranit, při jeho odstranění dojde k vymazání záznamu z tabulky *SUPERNODE_BINDING*.

4.15.3 Třetí a čtvrtá duplicita

Zbývající dvě duplicity se týkají prvků, které jsou pouze v rámci *mikro* vrstvy. U první z nich byl zvolen prvek *N815* a následně posunut (Obrázek 44 vlevo).



Obrázek 44: Třetí duplicita. Zdroj: vlastní

Vybraný prvek je možné odstranit až po posunu hrany. Poté, co byla hrana posunuta (Obrázek 44 vpravo), lze vybraný vrchol bezpečně odebrat.

Druhou duplicitou *mikro* vrstvy jsou prvky *N723* a *N5473*. Je zvolen prvek *N723*, ten je přesunut. Je upravena hrana a tento prvek je následně odebrán. Jde o analogický postup jako u prvku *N815*.

Poté, co jsou odstraněny duplicity a struktura změněn klíč na souřadnice, již struktura přijme všechny prvky. Nyní je již možné v grafu strukturu *HashMap* nahradit novou strukturou, quadstromem.

Závěr

Všechny cíle diplomové práce byly splněny. V teoretické části byla nejprve představena multidimenzionální data a možnosti jejich uchování, jednak v databázových systémech, ale také datové struktury pro jejich reprezentaci v paměti počítače.

Následně byl popsán databázový systém Oracle, přesněji jeho rozšíření Oracle Spatial, které se stará právě o uchování multidimenzionálních dat a práci s nimi. Dále byl představen Oracle MapViewer což je nástroj sloužící pro vizualizaci dat uložených v systému Oracle Spatial.

V další kapitole práce bylo třeba seznámit čtenáře s infrastrukturou železniční sítě a také předpisem ČD M12, jenž byl třeba pro pochopení některých parametrů třívrstvého datového modelu, ze kterého práce vychází. V rámci této kapitoly byly také popsány vybrané systémy pro lokalizaci kolejových vozidel, jednak bez použití satelitního navigačního systému, ale také s ním a dalšími podpůrnými systémy.

Stěžejní část práce byla zaměřena na implementaci editoru železniční sítě, pracující nad třívrstvým modelem železniční sítě. Dále byla představena reálná data, popis provedení instalace a konfigurace databázového systému Oracle s rozšířením Oracle Spatial a také import dat.

Pro ověření zda byla data správně naimportována, a také správně vytvořeny prostorové indexy, byl použit nástroj Oracle MapBuilder. V nástroji lze vytvořit schéma nad příslušnými daty uloženými v databázi a ověřit tím funkčnost prostorových indexů.

Další částí byl návrh tříd odrážející datový model a volba vhodné datové struktury. K načtení a vizualizaci dat je třeba knihoven, které jsou součástí programu. K vizualizaci je využíván Oracle MapViewer. Zde je vhodné zmínit, že knihovna, která slouží pro použití MapVieweru v aplikaci, obsahuje metodu pro kreslení na mapu. Tato metoda je v práci využívána, avšak její existence není zmíněna v oficiální dokumentaci.

Závěr praktické části je věnován problému s duplicitními souřadnicemi, na který se při tvorbě editoru přišlo.

Přínosem práce je softwarový editor, který umožňuje díky uživatelskému rozhraní pohodlně spravovat záznamy v databázi. Také byla objevena metoda, která je velice vhodná pro kreslení bez odeslání požadavku na server a není zmíněna v oficiální dokumentaci Oracle.

Použitá literatura

- [1] SAMET, Hanan. *Foundations of multidimensional and metric data structures*. Boston: Elsevier/Morgan Kaufmann, 2006. ISBN 978-0-12-369446-1.
- [2] MERTA, Jan. *Návrh a implementace systému pro lokalizaci kolejových vozidel*. Univerzita Pardubice, 2015. Diplomová práce. Vedoucí práce Jan Fikejz.
- [3] KIMPL, Libor. *Prostorové nadstavby nekomerčních databází* [online]. b.r. [cit. 2017-11-04]. Dostupné z: <http://www.geoinformatics.upol.cz/dprace/bakalarske/kimpl10/metody.html>
- [4] *Rozdíly - Výhody a nevýhody - MyISAM vs InnoDB v MySQL* [online]. b.r. [cit. 2017-12-07]. Dostupné z: <http://blog.pidisoft.cz/clanky/380-rozdily---vyhody-a-nevyhody---myisam-vs-innodb-v-mysql/>
- [5] *Using PostGIS: Data Management and Queries* [online]. b.r. [cit. 2017-12-09]. Dostupné z: <http://postgis.refrations.net/documentation/manual-1.4/ch04.html>
- [6] Microsoft SQL Server. In: *Wikipedia: the free encyclopedia* [online]. San Francisco (CA): Wikimedia Foundation, 2001-2017 [cit. 2017-09-12]. Dostupné z: https://en.wikipedia.org/wiki/Microsoft_SQL_Server
- [7] Spatial Data (SQL Server). In: *Microsoft* [online]. b.r. [cit. 2017-12-09]. Dostupné z: <https://docs.microsoft.com/en-us/sql/relational-databases/spatial/spatial-data-sql-server?view=sql-server-2017>
- [8] Oracle Database. In: *Wikipedia: the free encyclopedia* [online]. San Francisco (CA): Wikimedia Foundation, 2001-2018 [cit. 2018-01-25]. Dostupné z: https://en.wikipedia.org/wiki/Oracle_Database
- [9] *Spatial* [online]. In: . b.r. [cit. 2018-02-12]. Dostupné z: <http://www.orafaq.com/wiki/spatial>
- [10] Oracle Database 18c: Now available on the Oracle Cloud and Oracle Engineered Systems. In: *Oracle* [online]. b.r. [cit. 2018-03-01]. Dostupné z:

<https://blogs.oracle.com/database/oracle-database-18c-:-now-available-on-the-oracle-cloud-and-oracle-engineered-systems>

- [11] *Oracle Database: Developer's Guide* [online]. In: . b.r. [cit. 2018-03-01]. Dostupné z: <https://docs.oracle.com/en/database/oracle/oracle-database/18/spatl/spatial-concepts.html>
- [12] *Oracle® Spatial and Graph* [online]. 2018 [cit. 2018-03-01]. Dostupné z: <https://docs.oracle.com/en/database/oracle/oracle-database/18/topol/spatial-and-graph-topology-data-model-and-network-data-model-graph-developers-guide.pdf>
- [13] KOTHURI, Ravi, Albert. GODFRIND a Euro. BEINAT. *Pro Oracle Spatial for Oracle database 11g*. New York, NY: Distributed to the book trade worldwideby Springer-Verlag New York, 2007. ISBN 15-905-9899-7.
- [14] Cartesian coordinate system. In: *Wikipedia: the free encyclopedia* [online]. San Francisco (CA): Wikimedia Foundation, 2001-2018 [cit. 2018-03-01]. Dostupné z: https://en.wikipedia.org/wiki/Cartesian_coordinate_system
- [15] Fusion Middleware User's Guide for Oracle MapViewer. In: *Oracle* [online]. 2018 [cit. 2018-03-01]. Dostupné z: https://docs.oracle.com/middleware/12212/lcm/JIMPV/vis_start.htm
- [16] Transport Yearbook. In: *Sydos* [online]. 2016 [cit. 2018-03-01]. Dostupné z: https://www.sydos.cz/cs/rocenka-2016/rocenka/htm_cz/index.html
- [17] Základní charakteristika železniční sítě. In: *SŽDC* [online]. 2016 [cit. 2018-03-01]. Dostupné z: <http://www.szdc.cz/o-nas/zeleznice-cr/zeleznici-sit-v-cr.html>
- [18] FIKEJZ, Jan. *Systém na podporu dispečerského zařízení železniční dopravy analyzující data o poloze kolejových vozidel získaných ze systémů GNSS*. Univerzita Pardubice, 2016. Disertační práce.
- [19] FERFECKI, Lukáš. *Vlakový zabezpečovací systém ETCS*. Univerzita Pardubice, 2010. Bakalářská práce. Vedoucí práce Pavel Mazač.

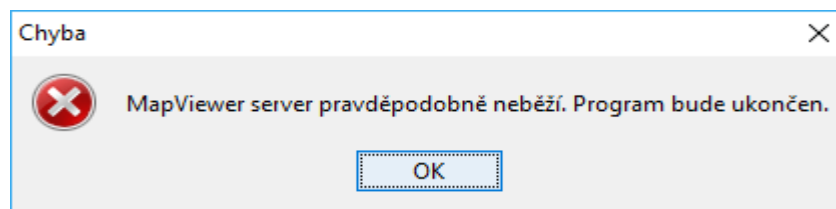
- [20] European Train Control System. In: *Wikipedia: the free encyclopedia* [online]. San Francisco (CA): Wikimedia Foundation, 2001-2018 [cit. 2018-05-12]. Dostupné z: https://en.wikipedia.org/wiki/European_Train_Control_System
- [21] Diferenciální GPS. In: *Wikipedia: the free encyclopedia* [online]. San Francisco (CA): Wikimedia Foundation, 2001-2016 [cit. 2018-05-12]. Dostupné z: https://cs.wikipedia.org/wiki/Diferenciální_GPS
- [22] BUSSE, Franz. Relative Navigation for Formation Flying Spacecraft Using Carrier-Phase Differential GPS. In: *MIT SSL* [online]. 2002 [cit. 2018-05-12]. Dostupné z: <http://www.mit.edu/~jhow/gps1.htm>
- [23] KAVIČKA, Antonín. *Elektronické sylaby přednášek k předmětu Datové struktury a algoritmy* [PDF]. Univerzita Pardubice. 2017 [cit. 2018-05-12].
- [24] ČD M12, . *Předpis pro jednotné označování tratí a kolejíšť v informačním systému ČD* [PDF]. Brno: České dráhy, 1996 [cit. 2017-11-16].
- [25] Krajní body Česka. In: *Wikipedia: the free encyclopedia* [online]. San Francisco (CA): Wikimedia Foundation, 2001-2018 [cit. 2018-05-12]. Dostupné z: https://cs.wikipedia.org/wiki/Krajn%C3%AD_body_%C4%8Ceska
- [26] MICHEK, Tomáš. *Datové struktury pro uchovávání geografických dat*. Univerzita Pardubice, 2011. Diplomová práce. Vedoucí práce Antonín Kavička.

Přílohy

Příloha A – <i>Uživatelská příručka</i>	82
---	----

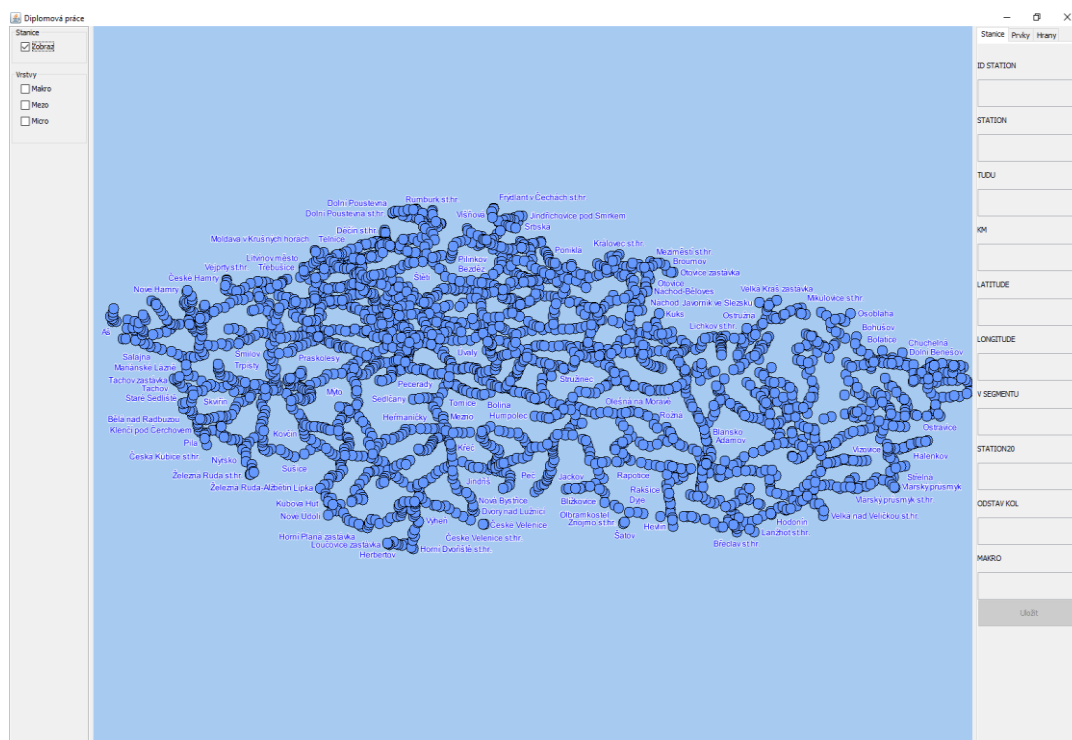
Příloha A – Uživatelská příručka

Příloha obsahuje základní popis uživatelského rozhraní aplikace. Potřebné knihovny jsou součástí aplikace. Před spuštěním programu je nutné, aby byl spuštěn MapViewer server. Pokud tomu tak není, zobrazí se následující chybová hláška (Obrázek 45).



Obrázek 45: MapViewer není spuštěn. Zdroj: vlastní

Pokud je MapViewer spuštěn, je zobrazeno uživatelské rozhraní aplikace (Obrázek 46).



Obrázek 46: Uživatelské rozhraní. Zdroj: vlastní

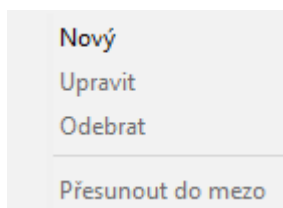
V levé části uživatelského rozhraní lze určovat, jaké vrstvy budou zobrazeny na mapě. Defaultní volba při startu je zobrazení všech stanic. V pravé části je konfigurační panel, kde lze přepínat mezi stanicemi, prvky nebo hranami. V prostřední části, kde je zobrazena mapa, lze ovládat přiblížení pomocí kolečka myši. Mapa se také přizpůsobuje aktuální velikosti okna.

Hlavní funkce pro stanice, prvky a hrany jsou:

- Selekce
- Přidávání
- Editace
- Odstranění

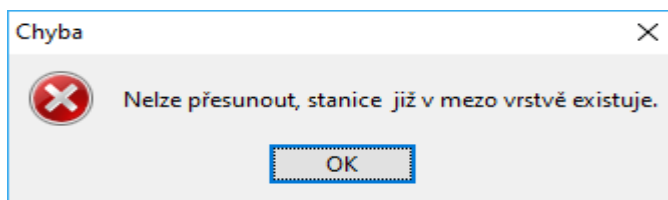
Selekce prvku je provedena dvojklikem LTM na daný prvek. V konfiguračním panelu je třeba mít zvoleno, v jaké vrstvě se uživatel pohybuje.

Při stisknutí PTM je zobrazeno kontextové menu (Obrázek 47).



Obrázek 47: Kontextové menu. Zdroj: vlastní

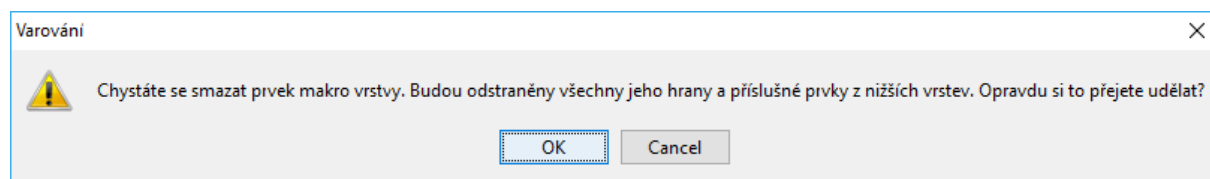
Volby kontextového menu jsou přístupné pro příslušnou situaci. Pokud není vybrán prvek, nelze ho tedy upravit, ani odebrat. Je přístupná pouze volba přidání nového. Volba přesunu do *mezo* vrstvy je dostupná pouze pro stanice, které v této vrstvě nemají příslušný prvek (Obrázek 48).



Obrázek 48: Přesun existující stanice do mezo vrstvy. Zdroj: vlastní

Přidávání hran je realizováno selekcí prvku, poté volbou *Nový* a následným držením PTM k cílovém prvku, kam chceme hranu umístit. U editace je tomu podobně s tím rozdílem, že je vybrána hrana a tažení probíhá od jednoho nebo druhého koncového bodu hrany.

Uživatel je také informován o majoritních zásazích do infrastruktury sítě. Je tomu tak například při odebrání prvku z *makro* vrstvy (Obrázek 49).



Obrázek 49: Odebrání prvku z makro vrstvy. Zdroj: vlastní