

ČESKÉ VYSOKÉ UČENÍ TECHNICKÉ V PRAZE
FAKULTA STAVEBNÍ, OBOR GEODÉZIE A KARTOGRAFIE
KATEDRA GEOMATIKY

Název předmětu

ALGORITMY DIGITÁLNÍ KARTOGRAFIE A GIS

Úloha

3

Název úlohy

Digitální model terénu

Akademický rok

2018/2019

Semestr

3.

Studijní
skupina

60

Vypracoval

Janovský Michal
karving47@gmail.com

Datum

22. 11. 2018

Klasifikace

1 Obsah

1	Zadání úlohy	3
2	Údaje o bonusových úlohách	3
3	Popis a rozbor problému + vzorce	4
4	Popisy algoritmů formálním jazykem	4
4.1	Delaunay triangulace	4
4.2	Tvorba vrstevnic	4
4.3	Výpočet sklonu	4
4.4	Výpočet orientace	4
5	Problematické situace a jejich rozbor + ošetření těchto situací v kódu	5
6	Vstupní data, formát vstupních dat, popis	5
7	Výstupní data, formát výstupních dat, popis	5
8	Printscreen vytvořené aplikace	5
9	Dokumentaci: popis tříd, datových položek a jednotlivých metod	7
9.1	Třída Algorithms	7
9.1.1	Metody třídy Algorithms	7
9.2	Třída Draw	7
9.2.1	Datové položky třídy Draw	7
9.2.2	Metody třídy Draw	7
9.3	Třída Widget	8
9.3.1	Sloty třídy Widget	8
9.4	Třída GeneratePoints	8
9.4.1	Metody třídy GeneratePoints	8
9.5	Třída QPoint3D	8
9.5.1	Datové položky třídy QPoint3D	8
9.5.2	Metody třídy QPoint3D	9
9.6	Třída Edge	9
9.6.1	Datové položky třídy Edge	9
9.6.2	Metody třídy Edge	9
9.7	Třída Triangle	9
9.7.1	Datové položky třídy Triangle	9
9.7.2	Metody třídy Triangle	9
9.8	Třída SortByXAsc	9
10	Závěr, možné či neřešené problémy, náměty na vylepšení	10
10.1	Závěr	10
10.2	Náměty na vylepšení	10
10.3	Neřešené problémy	10

1 Zadání úlohy

Úloha č. 3: Digitální model terénu

Vstup: $P = \{p_1, \dots, p_n\}, p_i = [x, y, z_i]$

Výstup: Polyedrický DMT nad množinou P představovaný vrstevnicemi doplněný vizualizací sklonu a expozice trojúhelníků

Metodou inkrementální konstrukce vytvořte nad množinou P vstupních bodů 2D Delaunay triangulaci. Jako vstupní data použijte existující geodetická data (alespoň 300 bodů) popř. navrhnete algoritmus pro generování syntetických vstupních dat představujících významné terénní tvary (kupa, údolí, spočinek, hřbet,...).

Vstupní množiny bodů včetně níže uvedených výstupů vhodně vizualizujte. Grafické rozhraní realizujte s využitím frameworku QT. Dynamické datové struktury implementujte s využitím STI.

Nad takto vzniklou triangulací vygenerujte polyedrický digitální model terénu, dále proveďte tyto analýzy:

- S využitím lineární interpolace vygenerujte vrstevnice se zadaným krokem a v zadaném intervalu, proveďte jejich vizualizaci s rozlišením zvýrazněných vrstevnic.
- Analyzujte sklon digitálního modelu terénu, jednotlivé trojúhelníky vizualizujte v závislosti na jejich sklonu.
- Analyzujte expozici digitálního modelu terénu, jednotlivé trojúhelníky vizualizujte v závislosti na jejich expozici ke světové straně.

Zhodnoťte výsledný digitální model terénu z kartografického hlediska, zamyslete se nad slabinami algoritmu založeného na 2D Delaunay triangulaci. Ve kterých situacích (různé terénní tvary) nebude dávat vhodné výsledky? Tyto situace graficky znázorněte.

Zhodnocení činnosti algoritmu včetně ukázek proveďte alespoň na tři strany formátu A4.

Bonusové úlohy:

- Triangulace nekonvexní oblasti zadané polygonem
- Výběr barevných stupnic při vizualizaci sklonu a expozice
- Automatický popis vrstevnic
- Automatický popis vrstevnic respektujících kartografické zásady
- Algoritmus pro automatické generování terénních útvarů
- 3D vizualizace terénu s využitím promítání
- Barevná hypsometrie

2 Údaje o bonusových úlohách

V rámci úlohy byly implementovány bonusové úlohy:

- Automatický popis vrstevnic
- Algoritmus pro automatické generování terénních útvarů

3 Popis a rozbor problému

Úkolem je vytvoření programu, který nad množinou bodů vytvoří trojúhelníkovou síť, vytvoří vrstevnice a pro jednotlivé trojúhelníky sítě vypočte jejich sklon a expozici. Pro tvorbu trojúhelníkové sítě byl použit algoritmus inkrementální konstrukce Delaunay triangulace ve 2D.

4 Popisy algoritmů formálním jazykem

Metody použité v této úloze, byly implementovány v programovacím jazyce C++ v prostředí Qt Creator.

4.1 Delaunay triangulace

Vlastnosti:

- V opsané kružnici trojúhelníka neleží žádný jiný bod
- Maximalizuje min. úhel
- Vůči kritériu minimálního úhlu je optimální jak lokálně, tak globálně
- Jednoznačná triangulace pokud jsou právě 3 body na kružnici

Postup tvorby trojúhelníkové sítě:

1. Nalezení pivota
2. Nalezení nejbližšího bodu od pivota
3. Vytvoření hrany
4. Hledání Delaunayova bodu
5. Vytvoření zbývajících hran trojúhelníka
6. Vložení hran do trojúhelníka

4.2 Tvorba vrstevnic

Vrstevnice byly vypočteny pomocí lineární interpolace, a to na hranách trojúhelníků vzniklých předešlou triangulací. Z výšek koncových bodů hran bylo nejprve určeno, zda tato hrana obsahuje průsečík s vrstevnicí, a to ze vztahu: $(z - z_i)(z - z_{i+1})$. Pokud je tento vztah roven 0, celá hrana náleží vrstevnici, pokud >0 neobsahuje průsečík a pokud <0 počítáme průsečík.

Průsečík:

$$x = \frac{(x_2 - x_1)}{(z_2 - z_1)}(z - z_1) + x_1$$
$$y = \frac{(y_2 - y_1)}{(z_2 - z_1)}(z - z_1) + y_1$$

4.3 Výpočet sklonu

Sklon je úhel mezi normálou trojúhelníka a svislicí.

V programu je sklon vizualizován odstíny šedi.

Svislice:

$$n = (0,0,1)$$

Normála:

$$n_t = \vec{u} \times \vec{v}$$

Sklon:

$$\varphi = \arccos\left(\frac{n_t n}{|n_t||n|}\right)$$

4.4 Výpočet orientace

Orientace (expozice) je brána jako orientace jednotlivých trojúhelníků vůči světovým stranám.

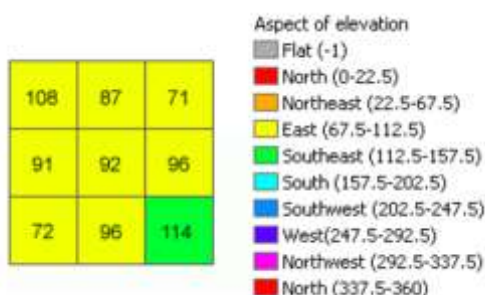
V programu je sklon vizualizován 9 barvami znázorňující směr světových stran.

Normála:

$$n_x = (u_y v_z - u_z v_y)$$
$$n_y = -(u_x v_z - u_z v_x)$$

Expozice:

$$A = \arctan2\left(\frac{n_x}{n_y}\right)$$



5 Problematické situace a jejich rozbor + ošetření těchto situací v kódu

Jediným nalezeným problémem je nemožnost použití Delaunay triangulace na takové množiny bodů, kde jsou body pravidelně rozmístěny (např. grid). Tento problém je způsoben tím, jak tato metoda triangulace funguje. Hledáme totiž bod, který spolu s 2 body úsečky tvoří nejmenší opsanou kružnici. Problém nastává, když takových bodů je více a program se zasekne (a spadne). Jiné řešení, než je použití jiné metody konstrukce trojúhelníkové sítě není autorovi známo. Z tohoto důvodu při generování bodů byla do polohy bodů zanesena náhodná hodnota v souřadnicích, aby se možnost těchto případů minimalizovala.

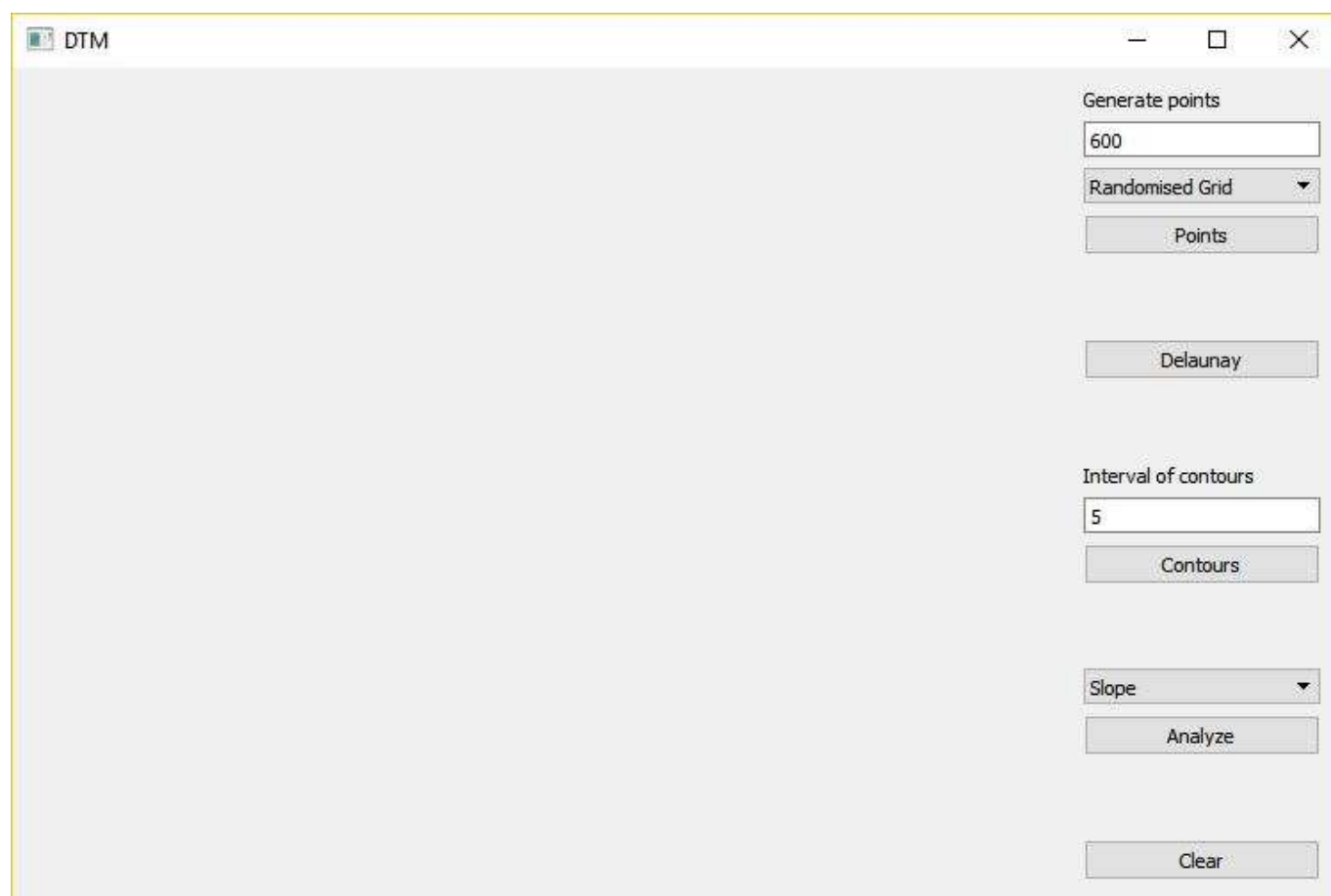
6 Vstupní data, formát vstupních dat, popis

Vstupními daty je vygenerovaná množina bodů, která představuje významné terénní útvary. Množina je tvořena body o souřadnicích x,y,z. Druhým možným vstupem je

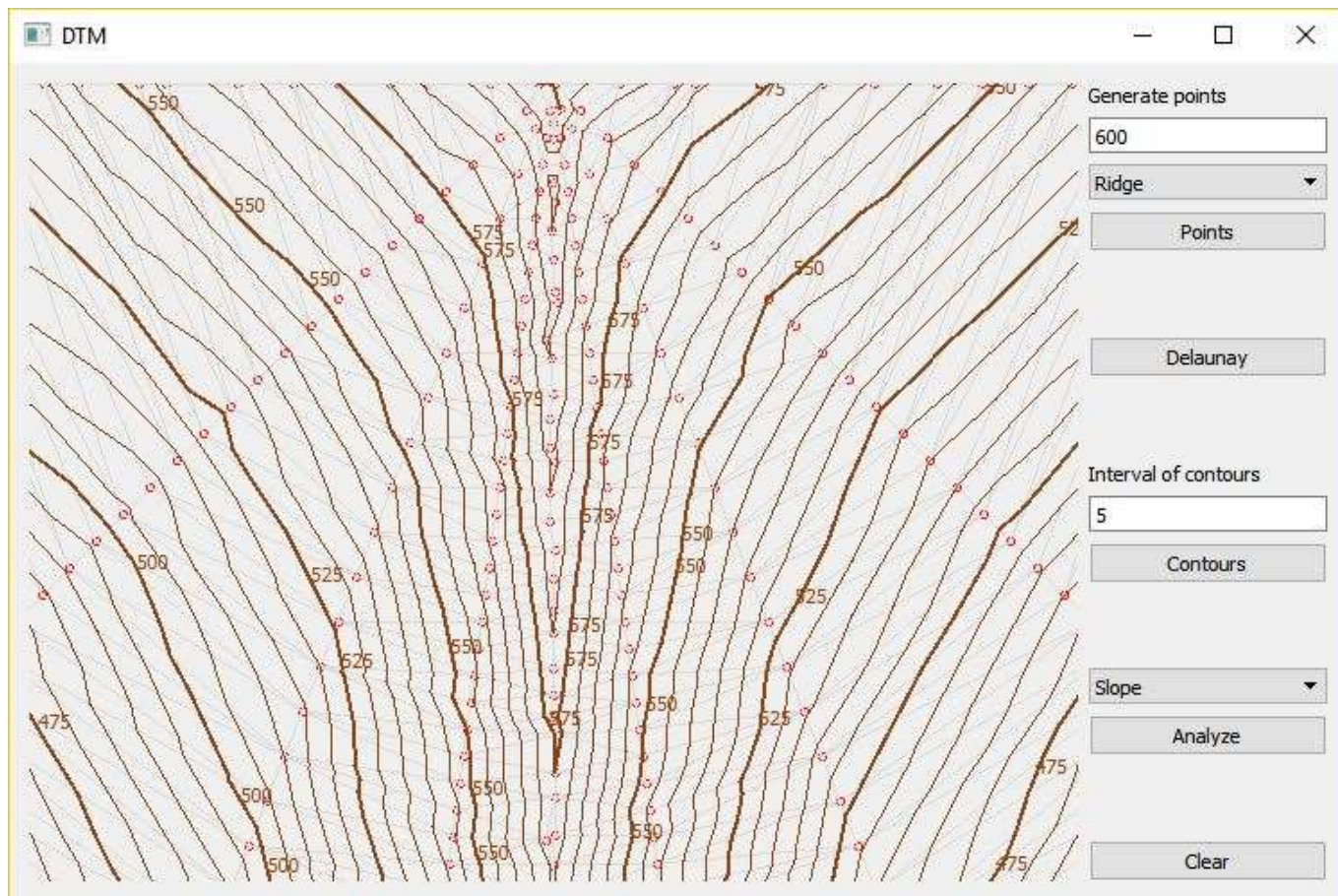
7 Výstupní data, formát výstupních dat, popis

Výstupem z programu je trojúhelníková síť z množiny vygenerovaných bodů, nad kterou jsou zkonstruovány a vizualizovány vrstevnice a sklonu a orientace jednotlivých trojúhelníků.

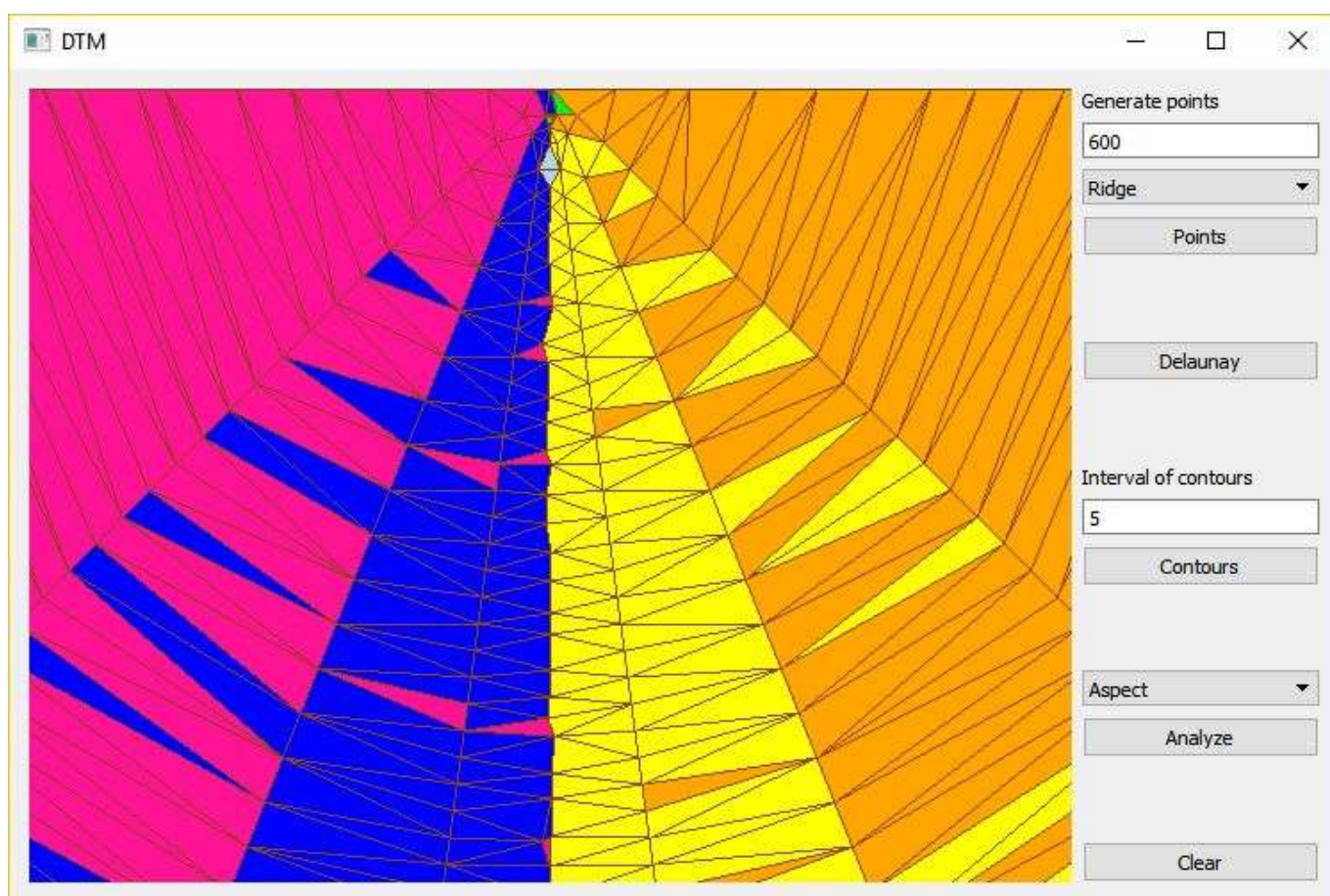
8 Printscreen vytvořené aplikace



Obrázek 1 - program po spuštění



Obrázek 2 - vygenerované body a nad nimi vytvořené vrstevnice s kótami



Obrázek 3 - expozice

9 Dokumentaci: popis tříd, datových položek a jednotlivých metod

9.1 Třída Algorithms

Třída obsahující veškeré početní výkony pro vygenerování trojúhelníků, vytváření vrstevnic a výpočtu sklonů a orientací trojúhelníků

9.1.1 Metody třídy Algorithms

```
static TPosition getPointLinePosition(QPoint3D &q, QPoint3D &a, QPoint3D &b);
```

metoda určující vztah bodu q k úsečce tvořené body a,b

```
static double getCircleRadius(QPoint3D &p1, QPoint3D &p2, QPoint3D &p3, QPoint3D &c);
```

vypočte střed kružnice c, která prochází body p1, p2, p3

```
static int getNearestPoint(QPoint3D &p, std::vector<QPoint3D> &points);
```

vrací index nejbližšího z bodů ve vektoru points k bodu p

```
static double distance(QPoint3D &p1, QPoint3D &p2);
```

vrací hodnotu vzdálenosti mezi body p1 a p2

```
static int getDelaunayPoint(QPoint3D &s, QPoint3D &e, std::vector<QPoint3D> &points);
```

vrací index bodu kde má spojnice a bod minimální poloměr opsané kružnice

```
static std::vector<Edge> DT(std::vector<QPoint3D> &points);
```

provádí triangulaci a vrací seznam hran

```
static QPoint3D getContourPoint(QPoint3D &p1, QPoint3D &p2, double z);
```

nalezne bod o zadané výšce z mezi body p1 a p2

```
static std::vector<Edge> createContours(std::vector<Edge> &dt, double z_min, double z_max, double dz);
```

vytvoří vektor hran vrstevnic s rozestupem dz

```
static double getSlope(QPoint3D &p1, QPoint3D &p2, QPoint3D &p3);
```

vypočítá sklon zadaného trojúhelníka

```
static double getAspect(QPoint3D &p1, QPoint3D &p2, QPoint3D &p3);
```

vypočítá expozici zadaného trojúhelníka

```
static std::vector<Triangle> analyzeDTM(std::vector<Edge> &dt);
```

vypočítá sklon a expozici pro všechny trojúhelníky

9.2 Třída Draw

Slouží k vykreslení bodů, trojúhelníků, vrstevnic, a vybarvení trojúhelníků.

9.2.1 Datové položky třídy Draw

```
std::vector<QPoint3D> points;
```

vektor bodů

```
std::vector<Edge> dt;
```

vektor hran

```
std::vector<Edge> contours;
```

vektor vrstevnic

```
std::vector<Triangle> dtm;
```

vektor trojúhelníků

9.2.2 Metody třídy Draw

```
void paintEvent(QPaintEvent *e);
```

metoda volaná při každém kreslení

```
void mousePressEvent(QMouseEvent *e);
```

slouží k vykreslení manuálně „naklikaných“ bodů

```

void clearPoints()
smazání množiny bodů

void clearDT();
smazání trojúhelníků

std::vector<QPoint3D> & getPoints()
slouží k předání hodnoty points

std::vector<Edge> & getDT()
slouží k předání trojúhelníků

void setDT(std::vector<Edge> &dt_)
„setr“ nastavení hran triangulace

void setContours(std::vector<Edge> &contours_)
„setr“ nastavení vrstevnic

void setDTM(std::vector<Triangle> &dtm_)
„setr“ nastavení trojúhelníků

```

9.3 Třída Widget

Třída pro práci s grafickým prostředím.

9.3.1 Sloty třídy Widget

```

void on_Points_clicked();
po kliknutí se vygeneruje množina bodů P

void on_Delaunay_clicked();
po kliknutí se nad množinou P sestaví trojúhelníková síť

void on_Contours_clicked();
po kliknutí se nad trojúhelníkovou sítí vygenerují vrstevnice

void on_Clear_clicked();
po kliknutí smaže vše a navrátí aplikaci do počátečního stavu

void on_Analyze_clicked();
po kliknutí se vypočte sklon a expozice a graficky se znázorní

```

9.4 Třída GeneratePoints

Slouží ke generování množin bodů P.

9.4.1 Metody třídy GeneratePoints

```

static std::vector<QPoint3D> generateGrid(int &n, QSize &size);
static std::vector<QPoint3D> generateKupa(int &n, QSize &size);
static std::vector<QPoint3D> generateUdoli(int &n, QSize &size);
static std::vector<QPoint3D> generateSpocinek(int &n, QSize &size);
static std::vector<QPoint3D> generateHrbet(int &n, QSize &size);

```

metody pro generování množin bodů P o n bodech představující zvolený terénní útvar

9.5 Třída QPoint3D

Třída odvozená od třídy QPoint rozšířená o souřadnici Z

9.5.1 Datové položky třídy QPoint3D

```

double z
hodnota výšky bodu

```


9.5.2 Metody třídy QPoint3D

```
double getZ()  
„getr“ vrátí výšku bodu
```

```
void setZ(double z_)  
„setr“ nastaví výšku bodu
```

9.6 Třída Edge

Třída Edge slouží k ukládání hran trojúhelníků

9.6.1 Datové položky třídy Edge

```
QPoint3D s, e;  
Počáteční a koncové body hrany
```

9.6.2 Metody třídy Edge

```
QPoint3D & getS()  
QPoint3D & getE()  
„getr“ navrátí počáteční/koncový bod  
  
void switchOr()  
obrácení orientace hrany  
  
bool operator == (const Edge &e_)  
přetížení operátoru == k porovnávání „hrana == hrana“
```

9.7 Třída Triangle

Třída složená pro ukládání a práci s trojúhelníky

9.7.1 Datové položky třídy Triangle

```
QPoint3D p1, p2, p3;  
Body definující trojúhelník
```

```
double slope, aspect;  
hodnoty sklonu a expozice trojúhelníků
```

9.7.2 Metody třídy Triangle

```
QPoint3D getP1()  
QPoint3D getP2()  
QPoint3D getP3()  
Vrací jednotlivé body trojúhelníka  
double getSlope()  
double getAspect()  
Vrací hodnoty sklonu a expozice trojúhelníka
```

9.8 Třída SortByXAsc

Slouží k seřazení bodů podle X souřadnice vzestupně

10 Závěr, možné či neřešené problémy, náměty na vylepšení

10.1 Závěr

Pomocí Delaunay triangulace byly sestrojeny trojúhelníky, z nichž se dále generovaly vrstevnice a byly vypočteny hodnoty sklonu a expozice. Jako data pro úlohu byla vytvořena třída `GeneratePoints`, ve které jsou metody pro generování umělých dat představujících terénní útvary kupa, spočinek, údolí a hřbet. Při vytváření této třídy byl zjištěn největší nedostatek Delaunay triangulace, a to že tato triangulace nefunguje, pokud mají body stejný rozestup. Toto bylo zjištěno při pokusu použít tuto metodu na gridu a dále na kružnici s body o stejném rozestupu.

10.2 Náměty na vylepšení

Náměty na vylepšení se týkají především estetické části programu. Při generování jsou mnohdy vygenerované body rozloženy nerovnoměrně, což ovlivňuje vzhled vrstevnic. Toto by se dalo vyřešit vyhlazením vrstevnic, nebo rovnoměrným rozmýstěním bodů. Dále u popisu vrstevnic je problém s překrýváním kót a vrstevnic, a kóty nejsou natočené ve směru svahu a natočené. Pro tyto problémy však zatím autor nemá řešení.

10.3 Neřešené problémy

Při popisu vrstevnic dochází k tomu, že popis je překrýván vrstevnicí. Dále u popisu vrstevnic se popis generuje u každé hrany, to bylo eliminováno popisem pouze každé X-té hrany, nicméně takto je popis nepravidelný a stává se, že popis je v některých částech plochy častý a na jiných částech chybí.