

Úvod do výpočetní geometrie. Základní vztahy.

Tomáš Bayer | bayertom@natur.cuni.cz

Katedra aplikované geoinformatiky a kartografie. Přírodovědecká fakulta UK.

Obsah přednášky

- 1 Úvod do výpočetní geometrie
- 2 Problémy řešené v digitální kartografii/GIS
- 3 Základní vztahy a testy
- 4 Datové modely
 - Špagetový model
 - Doubly Connected Edge List

1. Úvod

Druhá polovina 20. stol, rozvoj informačních technologií.

Pronikání do řady vědních oborů včetně kartografie \Rightarrow vznik digitální kartografie (Digital Cartography).

Přechod od subjektivních postupů (umělecké, výtvarné, estetické) na exaktní (tj. objektivně měřitelná).

70. léta 20. stol: vznik výpočetní geometrie (Computational Geometry).

Zabývá se problematikou algoritmů geometrické povahy souvisejících s pořizováním, ukládáním, editací, analýzou či vizualizací prostorových dat (vektorových i rastrových).

Algoritmy poskytují výsledky řešení problémů, v této souvislosti vyvstává řada otázek:

- je nalezené řešení optimální či dále optimalizovatelné?
- v jakém čase a s jakými hardwarovými požadavky lze nalézt řešení problému pro vstupní množinu?
- jsou algoritmy citlivé na konfiguraci prvků vstupní množiny?

Znalost odpovědí na tyto otázky umožní navrhnout co nejvýhodnější postup řešení zadaného problému.

Vývoj nových algoritmů je stále aktuální, nové techniky měření (např. laserové skenování) a zpracování dat.

2. Nejčastější problémy řešené s využitím výpočetní geometrie

Výpočetní geometrie: poskytuje nástroje a postupy pro řešení problémů ve 2D/3D s využitím algoritmů geometrické povahy.

Nejčastěji řešené problémy:

- Konvexní obálky (2D/3D): hledání nejmenšího polygonu nad množinou zadaných bodů.
- Voronoi diagramy: rozdělení prostorou do podoblastí se stejnou vzdáleností od centra (bodové i plošné).
- Triangulace (2D/3D): tvorba sítě trojúhelníků/čtyřstěnů na oblasti dané konvexní obálkou či nekonvexní oblasti.
- Průniky a průsečíky: hledání průniků a průsečíků bodů, linií, polygonů či složitějších entit ve 3D.
- Geometrické vyhledávání: Nalezení bodu/bodů ležících uvnitř konvexních/nekonvexních mnohoúhelníků.
- Plánování pohybu: plánování optimální cesty robota v prostředí tvořeném překážkami.

3. Aplikace v digitální kartografii/GIS

Příklady použití:

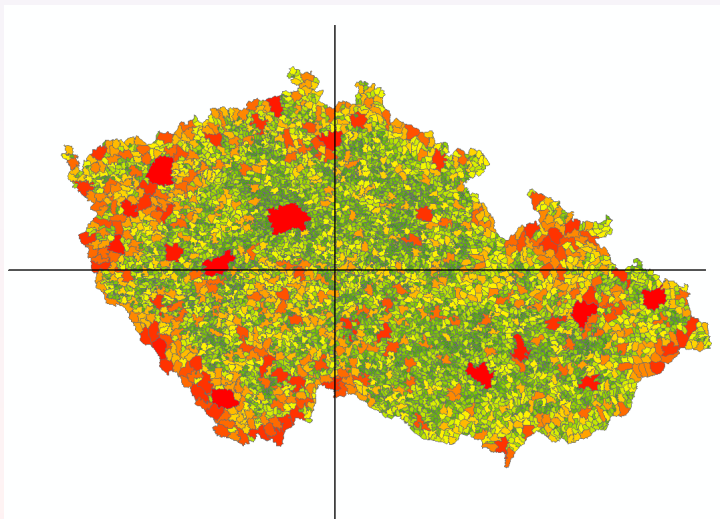
- Optimalizace tvaru regionů, hledání nejbližšího správního centra: Voronoi diagram, zobecněný Voronoi diagram.
- Digitální modely terénu a jejich analýzy: Delaunay triangulace, datově závislé triangulace.
- Kartografická generalizace: skeletony, konvexní obálky, Voronoi diagram, vyhlazování.
- Operace v GIS: booleovské operace s oblastmi, offset oblasti, geometrické vyhledávání prvků.

Samotná výpočetní geometrie nestačí, nutno kombinovat s robustním matematickým / statistickým aparátem.

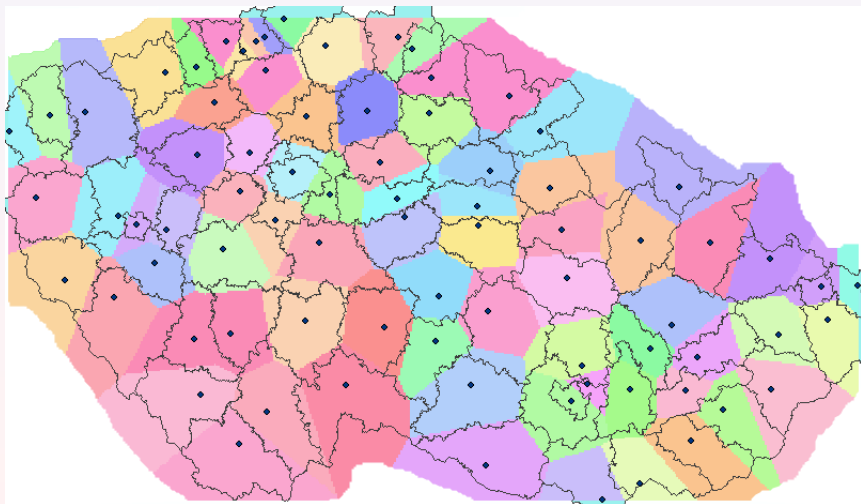
Příklady použití:

- Vektorizace, rasterizace dat: Voronoi diagramy, teorie grafů.
- Automatické rozpoznávání tvarů: Voronoi diagramy, skeletony, geostatistika, teorie grafů.
- Klasifikační algoritmy, řízená/neřízená klasifikace rastrových dat: geostatistika.
- Hledání dopravního spojení, logistické úlohy: teorie grafů.
- Komprese dat vektorových/rastrových: Diskrétní Fourierova transformace, Huffmanovo kódování, binární stromy.
- Interpolace, extrapolace dat: geostatistika

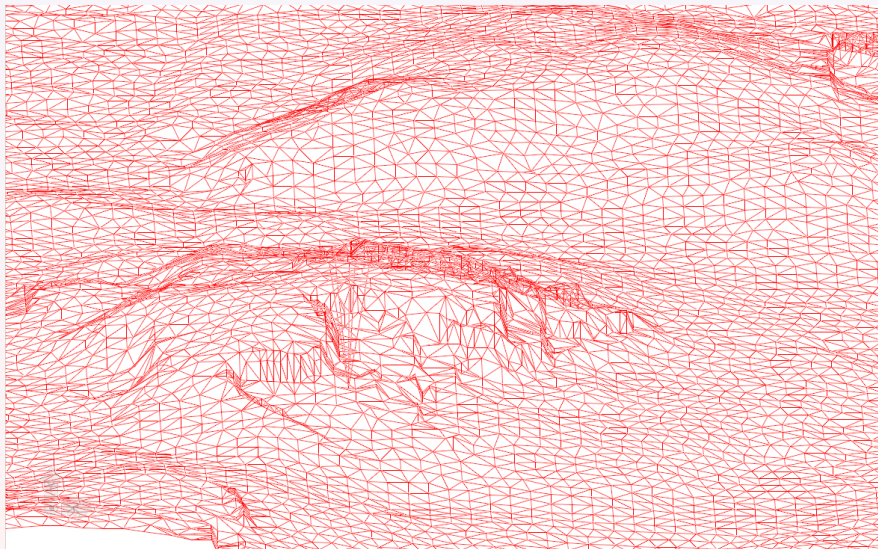
4. V kterém prvku leží kurzor (Geometrické vyhledávání)



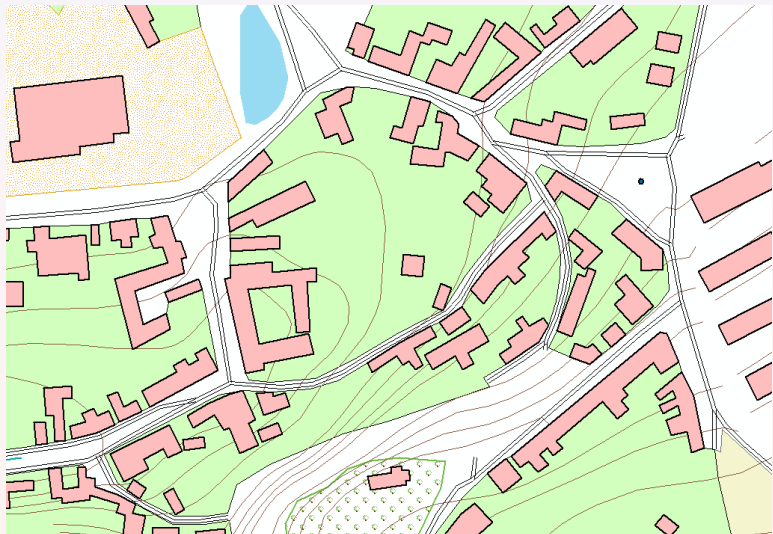
5. Optimalizace tvaru regionů vzhledem k centru (Voronoi diagram)



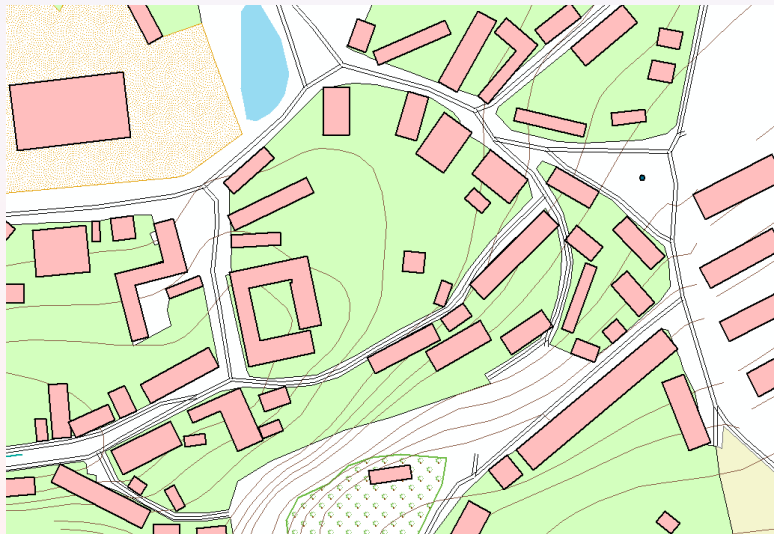
6. Tvorba DMT (Delaunay triangulace)



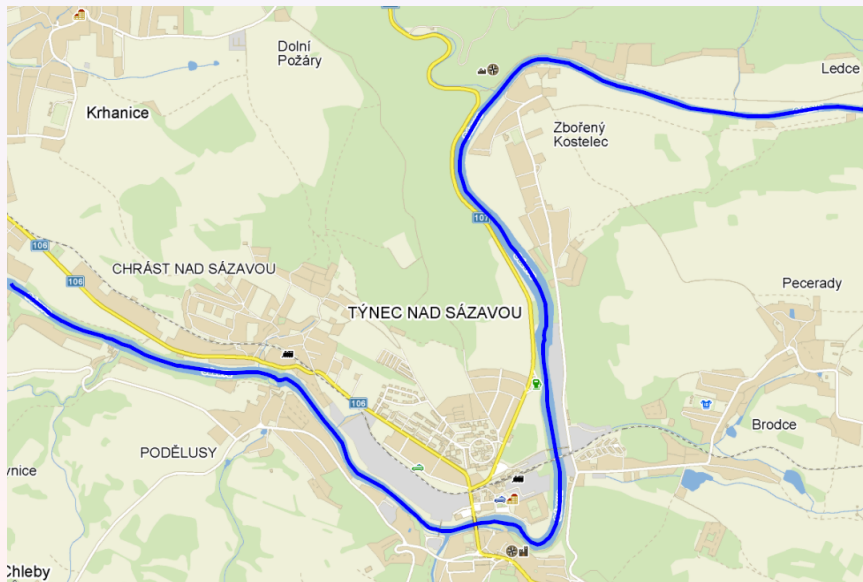
7. Kartografická generalizace (konvexní obálky, lineární regrese)



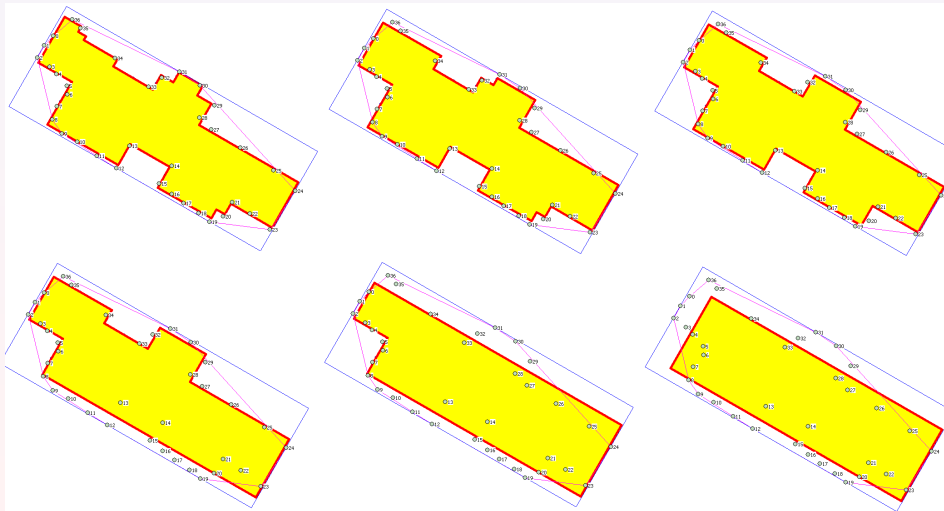
8. Kartografická generalizace (konvexní obálky, lineární regrese)



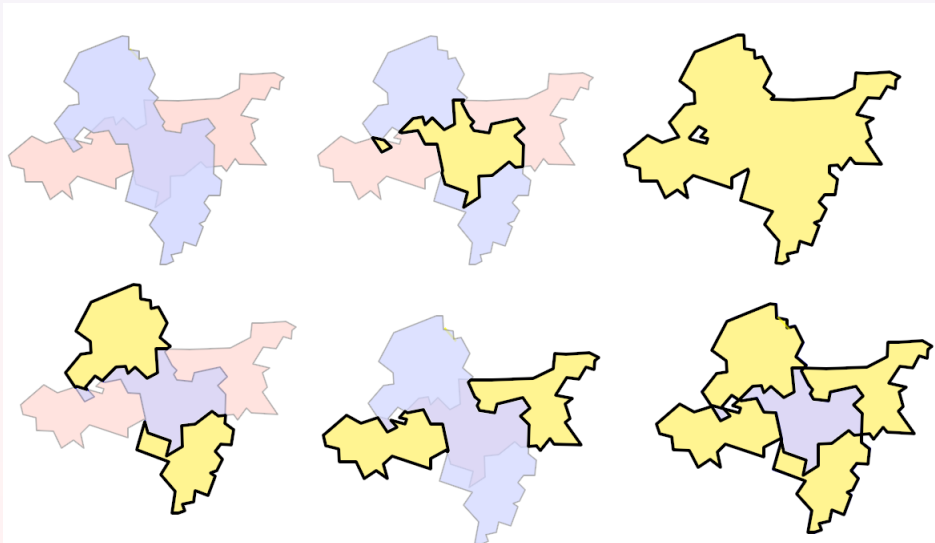
9. Kartografická generalizace (topologická kostra)



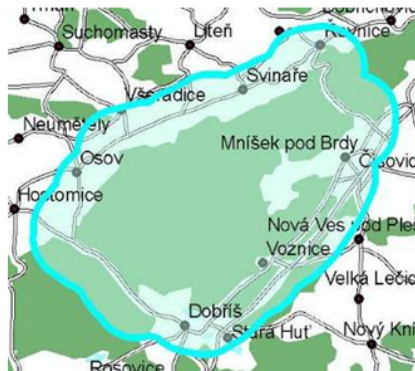
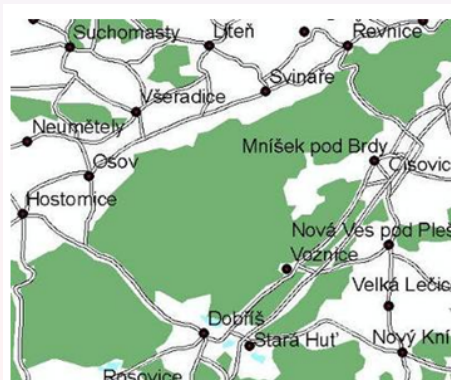
10. Kartografická generalizace (konvexní obálka, MBR)



11. Množinové operace s oblastmi v GIS



12. Offset polygonu (Minkowského suma)



13. Mnohoúhelníky a jejich vlastnosti

V1:

Mnohoúhelník P je část roviny ohraničená uzavřenou lomenou čarou tvořenou konečným počtem n vrcholů p_i . Nechť $e_i = \overline{p_i p_{i+1}}$ představuje stranu mnohoúhelníku.

V2:

Konvexní mnohoúhelník P je takový mnohoúhelník, jehož všechny vnitřní úhly ω_i jsou konvexní.

V3:

V konvexním mnohoúhelníku P neleží žádný bod spojnice dvou libovolných vrcholů $p_i, p_j, i \neq j$, vně P .

V4:

Přímka q , která není rovnoběžná se žádnou hranou konvexního mnohoúhelníku P , protíná P nejvýše ve dvou bodech.

V5:

Všechny úhlopříčky konvexního mnohoúhelníku P leží uvnitř P .

14. Mnohoúhelníky a jejich vlastnosti

V6:

Mnohoúhelník je monotónní vzhledem k přímce q , právě když každá přímka $t \perp q$ protne P nejvýše dvakrát.

V7:

Konvexní mnohoúhelník je monotónní vzhledem k libovolné přímce q .

V8:

Monotónní mnohoúhelník může být triangularizován v lineárním čase.

V9:

Pokud je alespoň jeden z vnitřních úhlů ω_i mnohoúhelníku P větší než π , mnohoúhelník je nekonvexní.

V10:

V nekonvexním mnohoúhelníku existuje alespoň jedna spojnice dvou vrcholů $p_i, p_j, i \neq j$, jejíž body leží vně mnohoúhelníku.

15. Mnohoúhelníky a jejich vlastnosti

V11:

Jednoduchý mnohoúhelník (Simple Polygon) je takový mnohoúhelník, pro jehož strany platí:

- Průsečíkem páru sousedních hran e_i, e_{i+1} je bod p_{i+1} : $e_i \cap e_{i+1} = p_{i+1}$.
- Hrany, které spolu nesousedí, nemají žádný průsečík: $e_i \cap e_j = \emptyset$ pro všechny $j \neq i + 1$.

V12:

Každý konvexní mnohoúhelník je jednoduchý.

V13:

Nejednoduchý mnohoúhelník (Not Simple Polygon): existuje průsečík hran q , které spolu nesousedí, tj. $e_i \cap e_j = q$ pro $j \neq i + 1$. Topologicky nekorektní, nebude uvažován.

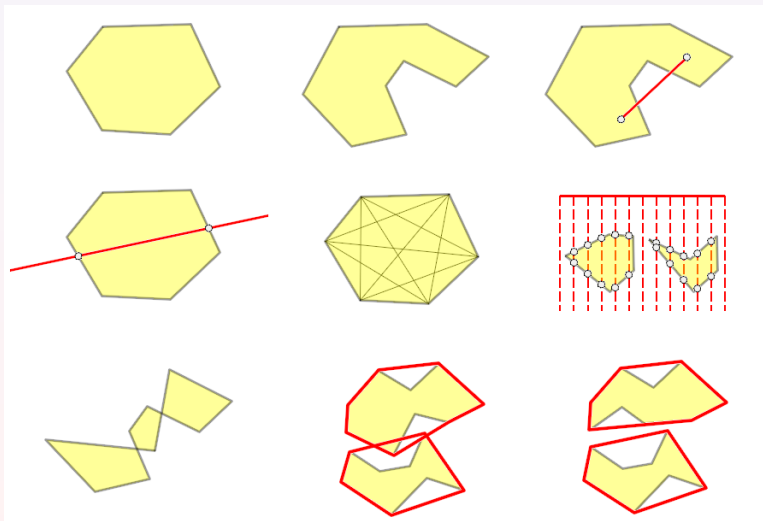
V14:

Lineárně separovatelné mnohoúhelníky P_1, P_2 jsou takové mnohoúhelníky, jejichž konvexní obálky $H(P_1), H(P_2)$ se neprotínají.

V15:

Lineárně neseparovatelné mnohoúhelníky P_1, P_2 jsou takové mnohoúhelníky, jejichž konvexní obálky $H(P_1), H(P_2)$ se protínají.

16. Ukázky mnohoúhelníků



17. Mnohoúhelníky a jejich vlastnosti

V16 (Jordanova věta):

Každá jednoduchá uzavřená křivka může být rozdělena na dvě části (týká se i mnohoúhelníků).

V17:

Suma vnitřních úhlů mnohoúhelník s n vrcholy je $(n - 2)\pi$.

V18:

Každý mnohoúhelník musí mít alespoň jeden striktně konvexní vrchol (tj. vrchol, jehož přiléhající hrany svírají úhel $< \pi$, nikoliv π).

V19 (Meistersova věta):

Každý mnohoúhelník s $n \geq 4$ vrcholy má diagonálu.

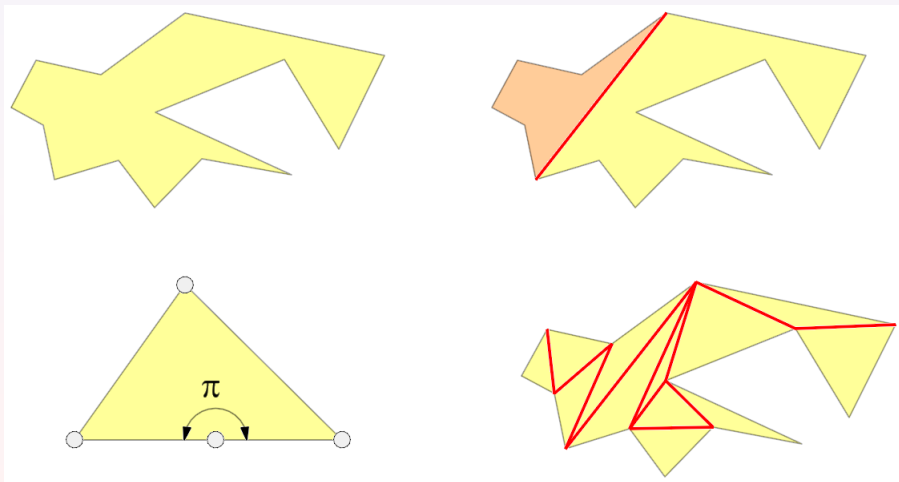
V20:

Každý mnohoúhelník s n vrcholy může být přidáním úhlopříček dekomponován na množinu trojúhelníků.

V21:

Každá triangulace mnohoúhelníku s n vrcholy používá $n - 3$ diagonál a $n - 2$ trojúhelníků.

18. Vlastnosti mnohoúhelníků



19. Orientace vektorů

Dáno:	vektory $\vec{u} = (u_x, u_y)$, $\vec{v} = (v_x, v_y)$
Hledáme:	Orientace vektorů \vec{u} , \vec{v} .

Vektory \vec{u} , \vec{v} jsou orientovány pozitivně, pokud úhel $\omega \in (0, \pi)$.

Vektory \vec{u} , \vec{v} jsou orientovány negativně, pokud úhel $\omega \in (\pi, 2\pi)$.

Pokud

$$t = \begin{vmatrix} u_x & u_y \\ v_x & v_y \end{vmatrix} \Rightarrow \begin{cases} t < 0 & \vec{u}, \vec{v} \text{ negativně orientovány,} \\ t > 0 & \vec{u}, \vec{v} \text{ pozitivně orientovány.} \end{cases}$$

Test bývá nazýván HalfPlane testem.

20. Poloha bodu a přímky

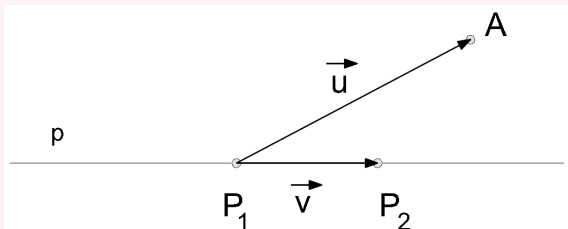
Dáno:	Bod $A = [x_A, y_A]$, přímka p , $p \approx \overleftrightarrow{P_1, P_2}$, $P_1 = [x_1, y_1]$, $P_2 = [x_2, y_2]$.
Hledáme:	Poloha A k p .

Používán test orientace vektorů, leží bod A v levé či v pravé polovině vzhledem k p ?

$$\vec{u} = (x_2 - x_1, y_2 - y_1)$$

$$\vec{v} = (x_A - x_1, y_A - y_1)$$

$$t = \begin{vmatrix} u_x & u_y \\ v_x & v_y \end{vmatrix} = \begin{vmatrix} x_1 & y_1 & 1 \\ x_2 & y_2 & 1 \\ x_3 & y_3 & 1 \end{vmatrix} \Rightarrow \begin{cases} t < 0 & A \text{ vlevo od } p \\ t = 0 & A \in p \\ t > 0 & A \text{ vpravo od } p \end{cases} = 2S\Delta(P_1, P_2, A)$$



21. Rovnice přímky dané dvěma body

Dáno:	přímka p , $p \approx \overleftrightarrow{P_1, P_2}$, $P_1 = [x_1, y_1]$, $P_2 = [x_2, y_2]$.
Hledáme:	rovnice p .

Rovnice přímky dané dvěma body

$$y - y_1 = \frac{y_2 - y_1}{x_2 - x_1}(x - x_1).$$

Vyjádření determinantem

$$\begin{vmatrix} x & y & 1 \\ x_1 & y_1 & 1 \\ x_2 & y_2 & 1 \end{vmatrix} = 0.$$

Obecná rovnice přímky

$$ax + by + c = 0,$$

kde

$$a = y_1 - y_2$$

$$b = x_2 - x_1$$

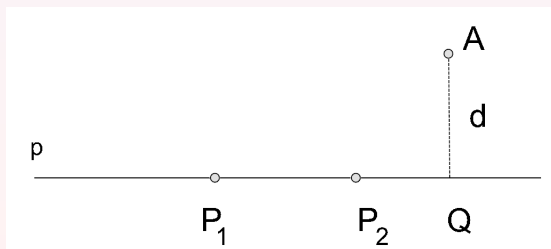
$$c = x_1 y_2 - y_1 x_2$$

22. Vzdálenost bodu od přímky $d(A, p)$

Dáno:	Bod $A = [x_A, y_A]$, přímka p , $p(P_1, P_2)$, $P_1 = [x_1, y_1]$, $P_2 = [x_2, y_2]$.
Hledáme:	$d(A, p)$.

$$d(A, p) = \frac{|a \cdot x_A + b \cdot y_A + c|}{\sqrt{a^2 + b^2}}$$

$$d(A, p) = \frac{x_A(y_1 - y_2) + x_1(y_2 - y_A) + x_2(y_A - y_1)}{\sqrt{(x_2 - x_1)^2 + (y_2 - y_1)^2}}$$



23. Test existence průsečíku dvou úseček

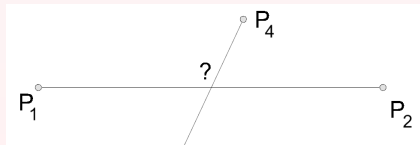
Dáno:	Úsečka $\overline{P_1P_2}$, kde $P_1 = [x_1, y_1]$, $P_2 = [x_2, y_2]$ a úsečka $\overline{P_3P_4}$, kde $P_3 = [x_3, y_3]$, $P_4 = [x_4, y_4]$.
Hledáme:	$\overline{P_1P_2} \cup \overline{P_3P_4}$.

Opakovaná aplikace HalfPlane testu, celkem 4x: analýza polohy koncových bodů jedné úsečky vzhledem k druhé úsečce a naopak.

$$t_1 = \begin{vmatrix} x_2 - x_1 & y_2 - y_1 \\ x_4 - x_1 & y_4 - y_1 \end{vmatrix} \quad t_2 = \begin{vmatrix} x_2 - x_1 & y_2 - y_1 \\ x_3 - x_1 & y_3 - y_1 \end{vmatrix}$$

$$t_3 = \begin{vmatrix} x_4 - x_3 & y_4 - y_3 \\ x_1 - x_3 & y_1 - y_3 \end{vmatrix} \quad t_4 = \begin{vmatrix} x_4 - x_3 & y_4 - y_3 \\ x_2 - x_3 & y_2 - y_3 \end{vmatrix}$$

Průsečík neexistuje, pokud t_1 a t_2 mají stejné znaménko nebo t_3 a t_4 mají stejné znaménko, tj. koncové body jedné úsečky vzhledem k druhé úsečce leží ve stejných polorovinách.



24. Test existence průsečíku dvou přímek

Dáno:	Přímka p_1 , $p_1 \approx \overleftrightarrow{P_1, P_2}$, $P_1 = [x_1, y_1]$, $P_2 = [x_2, y_2]$ a přímka p_2 , $p_2 \approx \overleftrightarrow{P_3, P_4}$, $P_3 = [x_3, y_3]$, $P_4 = [x_4, y_4]$.
Hledáme:	$p_1 \cup p_2$?

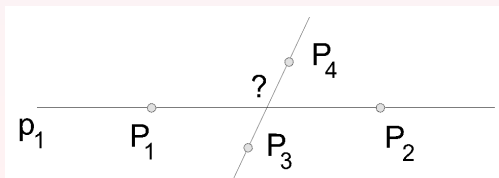
Směrové vektory:

$$\vec{u} = (x_2 - x_1, y_2 - y_1)$$

$$\vec{v} = (x_4 - x_3, y_4 - y_3)$$

Determinant test:

$$t = \begin{vmatrix} u_x & u_y \\ v_x & v_y \end{vmatrix} \Rightarrow \begin{cases} t = 0 & \text{jsou } \parallel \\ t \neq 0 & \text{nejsou } \parallel \end{cases}$$



25. Průsečík 2 úseček

Dáno:	Úsečka $\overline{P_1P_2}$, kde $P_1 = [x_1, y_1]$, $P_2 = [x_2, y_2]$ a úsečka $\overline{P_3P_4}$, kde $P_3 = [x_3, y_3]$, $P_4 = [x_4, y_4]$.
Hledáme:	$p_1 \cap p_2 = Q$.

$$\vec{u} = (x_2 - x_1, y_2 - y_1)$$

$$k_1 = v_x w_y - v_y w_x$$

$$\vec{v} = (x_4 - x_3, y_4 - y_3)$$

$$k_2 = u_x w_y - u_y w_x$$

$$\vec{w} = (x_1 - x_3, y_1 - y_3)$$

$$k_3 = v_y u_x - v_x u_y$$

$$\alpha = \frac{k_1}{k_3} \quad \beta = \frac{k_2}{k_3}$$

Pokud $k_1 = 0$ a $k_2 = 0$, úsečky jsou rovnoběžné.

Pokud $k_1 \neq 0$, $k_2 \neq 0$ a $k_3 = 0$ úsečky jsou kolinéární.

Pokud $\alpha \in \langle 0, 1 \rangle$ a $\beta \in \langle 0, 1 \rangle$, průsečík úseček Q existuje,

$$x_Q = x_1 + \alpha u_x = x_3 + \beta v_x$$

$$y_Q = y_1 + \alpha u_y = y_3 + \beta v_y$$

U testu průsečíku přímek $\alpha \in \langle -\infty, \infty \rangle$ a $\beta \in \langle -\infty, \infty \rangle$.

26. Průsečík 3 přímk v bodě

Dáno:	přímka $p_1(a_1, b_1, c_1)$, $p_2(a_2, b_2, c_2)$, $p_3(a_3, b_3, c_3)$,
Hledáme:	Protínají se v jednom bodě?

Používán determinant test

$$t = \begin{vmatrix} a_1 & b_1 & c_1 \\ a_1 & b_2 & c_2 \\ a_3 & b_3 & c_3 \end{vmatrix} \Rightarrow \begin{cases} t = 0 & \text{protínají se v jednom bodě,} \\ t \neq 0 & \text{neprotínají se v jednom bodě.} \end{cases}$$

27. Odchylka dvou přímek

Dáno:	Přímka $p_1, p_1 \approx \overleftrightarrow{P_1, P_2}, P_1 = [x_1, y_1], P_2 = [x_2, y_2]$ a přímka $p_2, p_2 \approx \overleftrightarrow{P_3, P_4}, P_3 = [x_3, y_3], P_4 = [x_4, y_4]$.
Hledáme:	odchylku φ přímek p_1, p_2 .

Směrové vektory

$$\vec{u} = (x_2 - x_1, y_2 - y_1)$$

$$\vec{v} = (x_4 - x_3, y_4 - y_3)$$

Odchylka φ přímek $p_1, p_2, \varphi \in \langle -\frac{\pi}{2}, \frac{\pi}{2} \rangle$

$$\cos \varphi = \frac{\vec{u} \cdot \vec{v}}{|\vec{u}| \cdot |\vec{v}|}$$

28. Kružnice daná trojicí bodů

Dáno:	Kružnice k , $k(P_1, P_2, P_3)$ kde $P_1 = [x_1, y_1]$, $P_2 = [x_2, y_2]$, $P_3 = [x_3, y_3]$.
Hledáme:	$k(S, r)$, kde $S = [m, n]$.

$$m = 0.5 \frac{k_{12}(-k_4) + k_{11}k_5 - (k_{10} + k_4k_5)k_6}{x_3(-k_4) + x_2k_5 + x_1(-k_6)}$$

$$n = 0.5 \frac{k_1(-k_9) + k_2k_8 + k_3(-k_7)}{y_1(-k_9) + y_2k_8 + y_3(-k_7)}$$

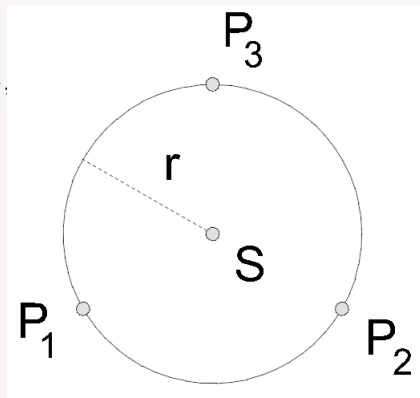
$$r = \sqrt{(x_1 - m)^2 + (y_1 - n)^2}$$

$$k_1 = x_1^2 + y_1^2 \quad k_5 = y_1 - y_3 \quad k_9 = x_2 - x_3$$

$$k_2 = x_2^2 + y_2^2 \quad k_6 = y_2 - y_3 \quad k_{10} = x_1^2$$

$$k_3 = x_3^2 + y_3^2 \quad k_7 = x_1 - x_2 \quad k_{11} = x_2^2$$

$$k_4 = y_1 - y_2 \quad k_8 = x_1 - x_3 \quad k_{12} = x_3^2$$



29. Poloha bodu a kružnice I.

Dáno:	Kružnice k , $k(P_1, P_2, P_3)$, kde $P_1 = [x_1, y_1]$, $P_2 = [x_2, y_2]$, $P_3 = [x_3, y_3]$, bod $P = [x, y]$.
Hledáme:	Poloha bodu P vzhledem ke k .

Testovací kritérium t :

$$t = (x - m)^2 + (y - n)^2 - r^2 \Rightarrow \begin{cases} t > 0 & P \text{ leží vně } k \\ t = 0 & P \in k \\ t < 0 & P \text{ leží uvnitř } k \end{cases}$$

$$m = 0.5 \frac{k_{12}(-k_4) + k_{11}k_5 - (k_{10} + k_4k_5)k_6}{x_3(-k_4) + x_2k_5 + x_1(-k_6)},$$

$$n = 0.5 \frac{k_1(-k_9) + k_2k_8 + k_3(-k_7)}{y_1(-k_9) + y_2k_8 + y_3(-k_7)}.$$

$$k_1 = x_1^2 + y_1^2$$

$$k_2 = x_2^2 + y_2^2$$

$$k_3 = x_3^2 + y_3^2$$

$$k_4 = y_1 - y_2$$

$$k_5 = y_1 - y_3$$

$$k_6 = y_2 - y_3$$

$$k_7 = x_1 - x_2$$

$$k_8 = x_1 - x_3$$

$$k_9 = x_2 - x_3$$

$$k_{10} = x_1^2$$

$$k_{11} = x_2^2$$

$$k_{12} = x_3^2$$

30. Poloha bodu a kružnice II.

Dáno:	Kružnice k , $k(P_1, P_2, P_3)$, kde $P_1 = [x_1, y_1]$, $P_2 = [x_2, y_2]$, $P_3 = [x_3, y_3]$, bod $P = [x, y]$.
Hledáme:	Poloha bodu P vzhledem ke k .

Varianta bez dělení s použitím determinantu, často používána.

Předpoklad: body P_1, P_2, P_3 uspořádány po směru hodinových ručiček.

Testovací kritérium t :

$$t = \begin{vmatrix} x & y & x^2 + y^2 \\ x_1 & y_1 & x_1^2 + y_1^2 \\ x_2 & y_2 & x_2^2 + y_2^2 \\ x_3 & y_3 & x_3^2 + y_3^2 \end{vmatrix} \Rightarrow \begin{cases} t > 0 & P \text{ leží vně } k \\ t = 0 & P \in k \\ t < 0 & P \text{ leží uvnitř } k \end{cases}$$

31. Poloha úsečky a kružnice

Dáno:	Kružnice $k : (S, r)$, kde $S = [m, n]$ a úsečka $\overline{P_1P_2}$, kde $P_1 = [x_1, y_1]$, $P_2 = [x_2, y_2]$
Hledáme:	Vzájemná poloha k a $\overleftrightarrow{P_1P_2}$:

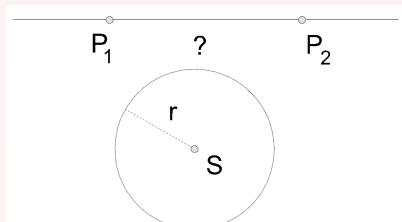
$$\vec{u} = (x_2 - x_1, y_2 - y_1)$$

$$d = \sqrt{u_x^2 + u_y^2}$$

$$D = \begin{vmatrix} x_1 - m & x_2 - m \\ y_1 - n & y_2 - n \end{vmatrix}$$

testovací kritérium t :

$$t = r^2 d^2 - D^2 \Rightarrow \begin{cases} t > 0 & P_1P_2 \text{ protíná } k \\ t = 0 & P_1P_2 \text{ je tečna } k \\ t < 0 & P_1P_2 \text{ neprotíná } k \end{cases}$$



32. Průsečík úsečky a kružnice

Dáno:	Kružnice $k : (S, r)$, kde $S = [m, n]$, úsečka $\overline{P_1P_2}$, kde $P_1 = [x_1, y_1]$, $P_2 = [x_2, y_2]$.
Hledáme:	Kružnice $k(S, r) \cup \overline{P_1P_2} = Q_1, Q_2$

$$\vec{u} = (x_2 - x_1, y_2 - y_1)$$

$$d = \sqrt{u_x^2 + u_y^2}$$

$$D = \begin{vmatrix} x_1 - m & x_2 - m \\ y_1 - n & y_2 - n \end{vmatrix}$$

$$t = r^2 d^2 - D^2$$

$$\text{sgn}(u_y) = \begin{cases} -1 & u_y < 0 \\ 1 & u_y \geq 0 \end{cases}$$

Souřadnice průsečíku:

$$x_{Q_{1,2}} = \frac{u_y D \pm \text{sgn}(u_y) u_x \sqrt{t}}{d^2}$$

$$y_{Q_{1,2}} = \frac{-u_x D \pm |u_y| \sqrt{t}}{d^2}$$

Pokud

$$\begin{cases} t > 0 & 2 \text{ průsečíky} \\ t = 0 & 1 \text{ průsečík} \\ t < 0 & 0 \text{ průsečíků} \end{cases}$$

33. Analýza polohy dvou kružnic

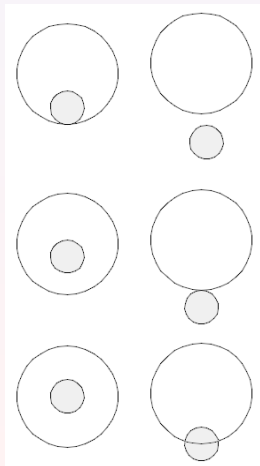
Dáno:	Kružnice $k_1 : (S_1, r_1)$ a $k_2 : (S_2, r_2)$, kde $S_1 = [m_1, n_1]$ a $S_2 = [m_2, n_2]$.
Hledáme:	Vzájemná poloha k_1, k_2 .

Vzdálenost středů d :

$$d = \sqrt{(m_2 - m_1)^2 + (n_2 - n_1)^2}$$

Pokud:

$$\begin{cases} d = 0 & k_1, k_2 \text{ soustředné} \\ 0 < d < |r_2 - r_1| & \text{jedna kružnice uvnitř druhé} \\ d = |r_2 - r_1| & k_1, k_2 \text{ vnitřní dotyk} \\ |r_2 - r_1| < d < r_1 + r_2 & k_1, k_2 \text{ mají 2 průsečíky} \\ d = r_1 + r_2 & k_1, k_2 \text{ vnější dotyk} \\ d > r_1 + r_2 & k_1 \text{ vně } k_2 \end{cases}$$



34. Průsečík dvojice kružnic

Dáno:	Kružnice $k_1 : (S_1, r_1)$ a $k_2 : (S_2, r_2)$, kde $S_1 = [m_1, n_1]$ a $S_2 = [m_2, n_2]$.
Hledáme:	Kružnice $k_1 \cup k_2 = Q_1, Q_2$.

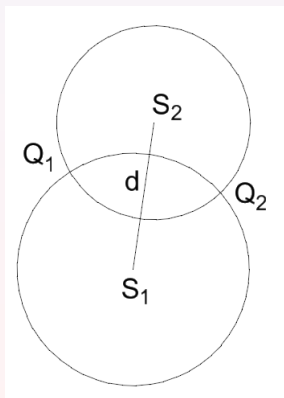
Vzdálenost středů d :

$$d = \sqrt{(m_2 - m_1)^2 + (n_2 - n_1)^2}$$

Souřadnice průsečíků:

$$x_{Q_{1,2}} = \frac{m_1 + m_2}{2} + \frac{(m_2 - m_1)(r_1^2 - r_2^2)}{2d^2} \pm \frac{n_2 - n_1}{2d^2} \sqrt{((r_1 + r_2)^2 - d^2)(d^2 - (r_2 - r_1)^2)}$$

$$y_{Q_{1,2}} = \frac{n_1 + n_2}{2} + \frac{(n_2 - n_1)(r_1^2 - r_2^2)}{2d^2} \pm \frac{m_2 - m_1}{2d^2} \sqrt{((r_1 + r_2)^2 - d^2)(d^2 - (r_2 - r_1)^2)}$$



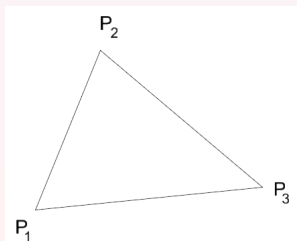
35. Plocha trojúhelníku

Dáno:	Vrcholy $\triangle P_1, P_2, P_3$, kde $P_1 = [x_1, y_1]$, $P_2 = [x_2, y_2]$, $P_3 = [x_3, y_3]$.
Hledáme:	Plocha trojúhelníku S .

$$S = \frac{1}{2} \begin{vmatrix} x_1 & y_1 & 1 \\ x_2 & y_2 & 1 \\ x_3 & y_3 & 1 \end{vmatrix}$$

Pokud body P_1, P_2, P_3 orientovány po směru hod. ručiček, $S > 0$, v opačném případě $S < 0$.

Vzorec představuje Half Plane test, lze použít jako **test orientace** vrcholů v trojúhelníku.



36. Plocha obecného mnohoúhelníku

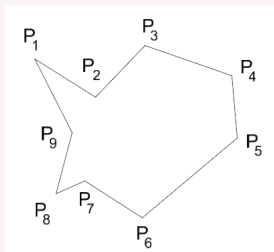
Dáno:	Vrcholy mnohoúhelníku P_1, \dots, P_n , kde $P_i = [x_i, y_i]$.
Hledáme:	Plocha mnohoúhelníku A .

LH vzorec:

$$A = \frac{1}{2} \sum_{i=1}^n x_i (y_{i+1} - y_{i-1})$$

Pokud body P_1, \dots, P_n , orientovány po směru hod. ručiček, pak $S > 0$, v opačném případě $S < 0$.

Vzorec lze použít jako **test orientace** vrcholů.



37. Centroid mnohoúhelníku

Dáno:	Vrcholy mnohoúhelníku P_1, \dots, P_n , kde $P_i = [x_i, y_i]$.
Hledáme:	centroid mnohoúhelníku x_c, y_c .

Výpočet centroidu (těžiště) mnohoúhelníku:

$$x_c = \frac{1}{6A} \sum_{i=1}^n (x_i + x_{i+1})(x_i y_{i+1} - x_{i+1} y_i),$$

$$y_c = \frac{1}{6A} \sum_{i=1}^n (y_i + y_{i+1})(x_i y_{i+1} - x_{i+1} y_i),$$

kde

$$A = \frac{1}{2} \sum_{i=1}^n x_i (y_{i+1} - y_{i-1}).$$

38. Poloha bodu a trojúhelníku

Dáno:	Vrcholy $\triangle P_1, P_2, P_3$ a testovaný bod P , kde $P_1 = [x_1, y_1]$, $P_2 = [x_2, y_2]$, $P_3 = [x_3, y_3]$, $P = [x, y]$.
Hledáme:	Poloha P vzhledem k \triangle .

Předpokládáme, že vrcholy trojúhelníku uspořádány po směru hodinových ručiček.
Používán opakovaný Hafsl Plane test (nad každou stranou \triangle).

Testovací kritérium:

$$t_1 = (x_2 - x_1)(y - y_1) - (y_2 - y_1)(x - x_1)$$

$$t_2 = (x_3 - x_2)(y - y_2) - (y_3 - y_2)(x - x_2)$$

$$t_3 = (x_1 - x_3)(y - y_3) - (y_1 - y_3)(x - x_3)$$

Pak

$$\begin{cases} t_1 > 0 \cap t_2 > 0 \cap t_3 > 0 & P \text{ uvnitř } \triangle. \\ t_1 < 0 \cup t_2 < 0 \cup t_3 < 0 & P \text{ vně } \triangle. \\ t_1 = 0 \cup t_2 = 0 \cup t_3 = 0 & P \text{ na hraně } \triangle. \end{cases}$$

39. Datové modely

V praxi používány dva základní datové modely:

- *Špagetový model:*
Neumožňuje ukládat topologické vazby, hodí se pouze pro jednoduché úlohy.
Nelze uchovávat vzájemnou polohu prvků, např. sousedství
- *Topologický model:*
Komplexní model umožňující ukládání topologických vazeb mezi prvky.
Používán Doubly Connected Half Edge List.

Při designu programů využity vlastnosti OOP, programy strukturovány do tříd

Při návrhu tříd a jejich vzájemných vztahů používány principy dědičnosti i kompozice (ISA-HASA test).

40. Špagetový model a jeho implementace

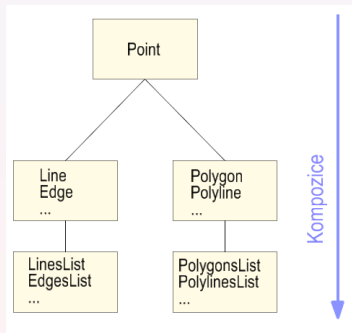
Komplexní návrh využívá společného předka, abstraktní třídu `GO` (Graphic Object), jejímiž potomky jsou třídy `Point`, `Line`, `Polygon`.

Zjednodušený návrh: pro návrh tříd `Line`, `Polygon` využít princip kompozice.

Využívání dynamických datových struktur:

- Sekvenční kontejnery (`list`, `vector`, ...).
- Kontejnerové adaptéry (`queue`, `stack`, `priority_queue`).

- Asociativní kontejnery (`set`).



41. Třída Point

Návrh třídy Point:

```
class Point2D
{
    private:
        double x, y;

    public:
        Point2D() : x(0), y(0) {}
        Point2D(double xx, double yy) : x(xx), y(yy) {}
        Point2D(const Point2D &p);
        bool operator < (const Point2D &p);
        bool operator ==(const Point2D &p);
        bool operator !=(const Point2D &p);
        Point2D operator =(const Point2D &p);
    ...
}
```

Vhodné definovat operátory: ==, !=, =, <.

42. Předefinování operátorů v třídě Point

Operátor = zohledňuje práci s reálnými čísly, test na rovnost není bezpečný!

```
bool Point2D::operator ==(const Point2D &p) {  
    return (x-p.x)*(x-p.x)+(y-p.y)*(y-p.y) < MIN * MIN;  
}
```

Ostatní operátory:

```
Point2D::Point2D(const Point2D &p) {  
    if (this != &p){  
        this->x=p.x; this->y=p.y;  
    }  
}
```

```
bool Point2D::operator < (const Point2D &p) {  
    return (this->x<p.x) || (this->x==p.x)&&(this->y<p.y);  
}
```

```
Point2D Point2D:: operator =(const Point2D &p){  
    if (this != &p){  
        this->x=p.x; this->y=p.y;  
    }  
    return *this;  
}
```

43. Třída Line

Navržena za použití principu kompozice (IS/HAS test).
Linie MÁ dva koncové body.

Datové položky třídy Line tvoří instance třídy Point.

```
class Line
{
    private:
        Point2D p1;
        Point2D p2;

    public:
        Line();
        Line( Point2D &p1, Point2D &p2);
        Line(const Line &l);

    ...
};
```

44. Třída LinesList

Obsahuje seznam jednotlivých linií uložených v STL kontajneru vector, list.

Ukázka třídy LinesList:

```
class LinesList
{
    private:
        std::vector <Edge> edges;    //Seznam linií

    public:
        EdgeList();
        void add(Edge e){edges.push(e);};
        void clear(){edges.clear();};
        ...
};
```

45. Doubly Connected Edge List

Jedna z nejčastěji používaných datových struktur výpočetní geometrie. Popisuje dekompozici kartografického díla na OD, 1D a 2D entity, tj. na body, linie a polygony) se zachováním vzájemných topologických vazeb \Rightarrow vytvoření grafové struktury *DCEL*.

Obsahuje informace o každém bodu (Point), hraně (Edge) a polygonu (Face).

K prvkům mohou být také ukládány dodatečné informace atributového charakteru.

Umožňuje nalézt všechny hrany náležící určitému polygonu a procházet jimi v obou směrech, polygony sousedící s určitým polygonem.

Každá orientovaná hrana obvykle sousedí se dvěma polygony, je průsečnicí dvou polorovin.

Každou hranu, tzv. *Half Edge*, nutno popsat z "obou polorovin".

Obě *HalfEdge* vytvářejí zdvojenou hranu, tzv. *Twin Edge*.

46. Struktura DCEL

Každý bod nese pointer na hranu, jejíž je počátečním bodem.

Half Edge je orientována, nese pointer na počátečním bodě v.

Uchová také pointer na předchozí a následující Half Edge a Half Edge s protilehlou orientací sdílenou v rámci Twin Edge.

Dále obsahuje pointer na incidující polygon vlevo od Half Edge.

Hrany v polygonu uloženy ve směru hodinových ručiček.

Obsahuje-li nějaký polygon otvory, jsou popsány jako polygony s opačnou orientací, tj. po směru hodinových ručiček.

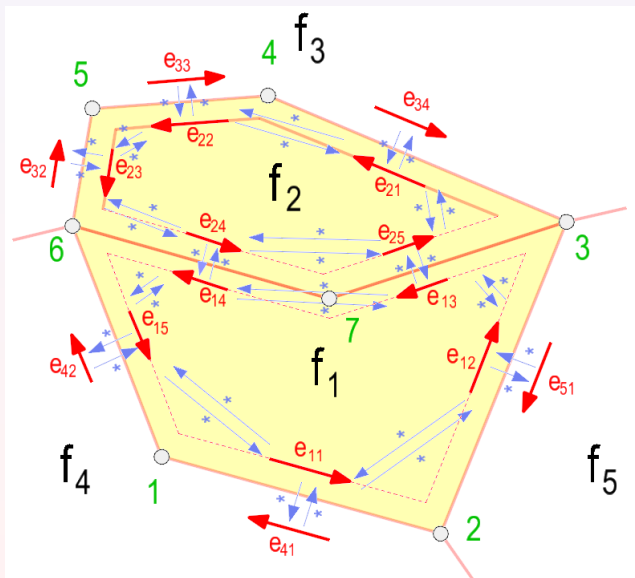
Všechny hrany tvořící otvor leží vlevo od hran nadřazeného polygonu

Každý polygon nese informaci o:

vnitřních komponentách: pointer na svou libovolnou hranu (NULL, pokud neuzavřen).

vnějších komponentách: pointer na libovolnou hranu otvoru (NULL, pokud žádný není).

47. Ilustrace Doubly Connected Edge List



48. Datový model DCEL

Tabulka vrcholů (Vertices):

Vertex	x	y	Inc. hrana
1	978	2110	e_{11}
2	867	2086	e_{12}
3	817	2213	e_{21}
...

Tabulka polygonů (Faces):

Face	Vnitřní hrana	Vnější hrana
f_1	NULL	e_{11}
f_2	NULL	e_{12}
f_3	NULL	NULL
...

49. Datový model DCEL

Tabulka Half Edges:

Half Edge	Start	Twin	Incid. polygon	Další hrana	Předchozí hrana
e_{11}	1	e_{41}	f_1	e_{12}	e_{15}
e_{12}	2	e_{51}	f_1	e_{13}	e_{11}
e_{13}	3	e_{25}	f_1	e_{14}	e_{12}
e_{14}	7	e_{24}	f_1	e_{15}	e_{13}
e_{15}	6	e_{42}	f_1	e_{11}	e_{14}
e_{21}	3	e_{34}	f_2	e_{22}	e_{25}
e_{22}	4	e_{33}	f_2	e_{23}	e_{21}
e_{23}	5	e_{32}	f_2	e_{24}	e_{22}
e_{24}	6	e_{14}	f_2	e_{25}	e_{23}
...

50. Práce s reálnými čísly

U $(m + n)$ reprezentace čísel v důsledku přeplnění mantisy při provádění aritmetických operací dochází ke ztrátám přesnosti \Rightarrow zaokrouhlením čísla.

$$\begin{array}{r}
 \begin{array}{cccccccc}
 +/ - & 2^2 & 2^1 & 2^0 & 2^{-1} & 2^{-2} & 2^{-3} & 2^{-4} \\
 (4.625)_{10} \rightarrow & \boxed{0} & \boxed{1} & \boxed{0} & \boxed{0} & \boxed{1} & \boxed{0} & \boxed{1} & \boxed{0} \\
 & & & & & & & & = (4.625)_{10}
 \end{array} \\
 \begin{array}{cccccccc}
 +/ - & 2^2 & 2^1 & 2^0 & 2^{-1} & 2^{-2} & 2^{-3} & 2^{-4} \\
 (4.600)_{10} \rightarrow & \boxed{0} & \boxed{1} & \boxed{0} & \boxed{0} & \boxed{1} & \boxed{0} & \boxed{0} & \boxed{1} \\
 & & & & & & & & = (4.5625)_{10}
 \end{array}
 \end{array}$$

Důsledky:

- *Porovnávání čísel*

Porovnáváme -li dvě reálné hodnoty a, b , pak podmínka

$$a = b.$$

V obecném případě nemusí být pravdivá. Testujeme absolutní hodnotu jejich rozdílu $|a - b|$ s hodnotou $\varepsilon \gg 0$:

$$|a - b| < \varepsilon.$$

- *Komutativní, asociativní a distributivní zákony*

Při práci s reálnými čísly tyto zákony obecně nemusejí platit.

Pozor při návrhu operátoru = pro jednotlivé třídy !!!