Univerzita Hradec Králové

Fakulta informatiky a managementu

Katedra informatiky a kvantitativních metod

Teorie grafů - Skóre grafu

Bakalářská práce

Autor: Radim Krátký
Studijní obor: AI-3

Vedoucí práce: RNDr. Andrea Ševčíková

Odborný konzultant: RNDr. Andrea Ševčíková
Budova J, místnost 93200

Hradec Králové Duben 2018

Prohlášení:

Prohlašuji, že jsem bakalářskou práci zpracoval samostatně a s použitím uvedené literatury. Souhlasím se zapůjčováním práce.

V Hradci Králové dne 27.4.2018 Radim Krátký

Poděkování:

Děkuji všem, kteří se podíleli na vzniku této práce. Především děkuji RNDr. Andree Ševčíkové za metodické vedení práce, cenné rady a připomínky při zpracování bakalářské práce.

Anotace

Bakalářská práce se zabývá matematickou disciplínou teorie grafů, konkrétně skóre grafu. Teorie grafů patří do odvětví diskrétní matematiky. Součástí bakalářské práce je aplikace GraphScore verze 2.0, která slouží uživatelům k snadné práci s grafy a lepšímu porozumění dané problematiky. Bakalářská práce je rozdělena na dvě části. Část teoretickou, která se věnuje základním pojmům teorie grafů, a na část praktickou, jejímž cílem bylo vytvoření programu pro zjištění, zda daná posloupnost je skóre grafu, a nakreslení příslušného grafu. Důraz byl kladen především na vytvoření jednoduché a funkční aplikace s danými funkcemi. Z tohoto důvodu práce obsahuje pouze základní a stručnou teorii, která je potřebná k pochopení konkrétní oblasti teorie grafů – skóre grafů.

Annotation

Title: Graph theory – Degree sequence

This bachelor thesis deals with the mathematical discipline of the graph theory, specifically the degree sequence. The graph theory belongs to the discrete mathematics sector. GraphScore version 2.0 is a part of the bachelor's thesis. It enables users to easily work with graphs and to better understand the given issues. The bachelor thesis is divided into two parts. The theoretical part, which deals with the basic concepts of graph theory, and the practical part, which has the purpose to create a program to determine whether a given sequence is a sequence of a graph, and to draw a relevant graph. The emphasis was placed on the creation of a simple and functional application with the given functions. For this reason, the thesis contains only a basic and brief theory that is needed to understand a particular area of the graph theory – the degree sequence.

Obsah

1 Úvod ... 1
2 Cíl práce ... 4
3 Metodika zpracování ... 5
4 Přehled použitého značení ... 6
5 Historie ... 7
6 Teoretická východiska ... 8
7 Implementace GUI ... 24
7.1 Okno ... 24
7.2 Plátno ... 24
7.3 Objekty plátna ... 25
7.4 Menu ... 26
8 Algoritmy ... 29
8.1 Izomorfismus ... 29
8.2 Generování grafu ... 33
8.2.1 Generování grafu prohledáváním do hloubky ... 34
8.2.2 Generování grafu prohledáváním do šířky ... 35
9 Popis aplikace ... 37
9.1 Plátno aplikace ... 38
9.2 Navigace ... 38
9.3 Uložení grafu ... 38
9.4 Režim „Kreslení grafu“ ... 39
9.4.1 Plátno režimu „Kreslení grafu“ ... 40
9.4.2 Vlastnosti režimu „Kreslení grafu“ ... 40
9.5 Režim „Izomorfismus“ ... 43
9.5.1 Plátna režimu „Izomorfismus“ ... 43
9.5.2 Vlastnosti režimu „Izomorfismus“ ... 44
9.6 Režim „Kreslit podle skóre“ ... 46

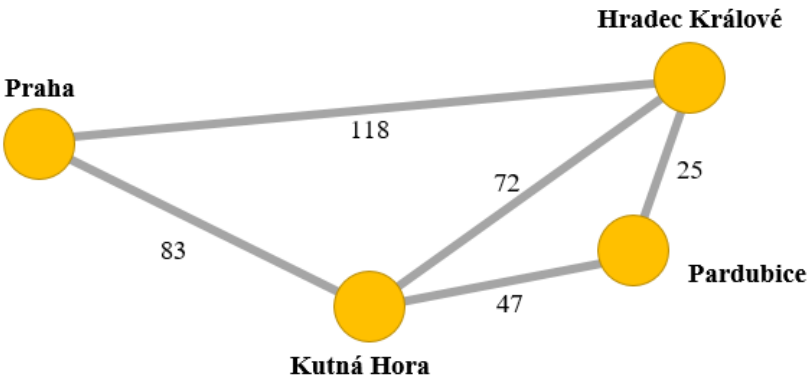
9.6.1 Plátno režimu „Kreslit podle skóre“ 46
9.6.2 Vlastnosti režimu „Kreslit podle skóre“ 47
10 Shrnutí výsledků 49
11 Závěry a doporučení 50
12 Seznam použité literatury 51

1 Úvod

Pod pojmem graf si většina lidí představí sloupcové grafy používané u statistik, nebo grafy znázorňující průběh funkcí.

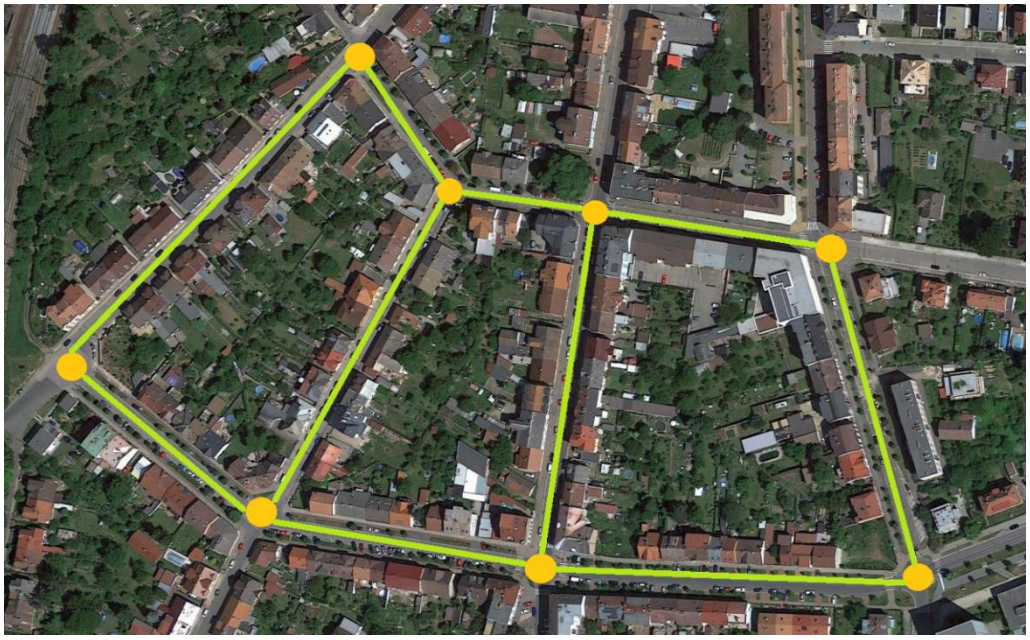
Grafy z oblasti teorie grafů si lze představit jako zjednodušení reálného světa, kde problematická oblast je znázorněna pomocí bodů a čar, které je spojují a tím popisují vlastnosti daného problému. Body se nazývají vrcholy grafů a čáry hrany grafu.

Aniž by si to člověk uvědomoval, setkává se s tímto tématem velmi často. Například při hledání nejkratší cesty do cíle mezi vybranými městy, kde města jsou reprezentovány vrcholy a silnice hranami.



Obrázek 1 - Příklad grafu v praxi

Nejedná se o jediný způsob využití, grafy mohou reprezentovat prakticky cokoli. Typickým příkladem využití teorie grafů je řešení infrastruktury měst například pro řízení dopravy a její optimalizaci. [6] Vrcholy grafů reprezentující křižovatky a hrany silnice.



Obrázek 2 - Využití grafů v městské infrastruktuře[1]

S grafy je dále možno se setkat v programování, kde algoritmy jsou často zapsány pomocí vývojových diagramů, nebo v časopisech k řešení různých rébusů. [6]

Každý takový graf je reprezentován tzv. skórem grafu. Jedná se o psaný popis grafu, ve formě číselné posloupnosti, z kterého lze vyčíst veškeré informace, avšak neurčuje jednoznačně graf. V některých případech mají různé grafy stejné skóre. [1] Problémem je nalezení některých nebo všech grafů daného skóre. Samozřejmě ne každá číselná posloupnost může být skórem grafu. Pro skóre platí několik pravidel a je využíváno především pro reprezentaci zakresleného grafu.

Výuka se v současné době neobejde bez počítačů, interaktivních tabulí a přístupu k internetu. Pedagogové i žáci využívají všech dostupných moderních technologií zlepšujících kvalitu a srozumitelnost výuky. Vytvářejí svoje vlastní prezentace, dokážou třídit informace a používat je při svoji práci. Školství je obor, který dokáže využívat pokrok velmi účinně.

Předmět Diskrétní matematika (DIMA) vyučující na FIM UHK zahrnuje především oblast teorie grafů. Teorie grafů poskytuje elegantní nástroj k popisu

[1] Obrázek 2 je vytvořen z mapy Hradce Králové – Pražské Předměstí https://goo.gl/maps/2KWEejHVywD2

různých situací nebo problému ze skutečného života. Také často poskytuje i návod na jejich řešení. Předmět DIMA je pro studenty přínosná do následujícího života, a proto je důležité, aby chápali témata a příslušné procesy vyučující v daném předmětu. [39]

S cílem podpořit efektivitu výuky předmětu DIMA byly vyvinuté specifické podpůrné multimediální nástroje, např. GrAlg [40] a ADIMA [41], které byly vyvinuté pro podporu výuky grafových algoritmů a GraPro [42] pro podporu výuky důkazů v teorii grafů. Tyto aplikace jsou používané jako doplněk k existujícímu výkladu na přednáškách, k zlepšení představivosti a celkovému pochopení náročných témat. Vizualizace pomocí multimediálních nástrojů může zlepšit schopnost soustředit se na konkrétní cíl a na jeho dosažení, a proto jsou navrhované a vyvíjené nové aplikace pro různé oblasti teorie grafů vyučované v předmětu DIMA.

2 Cíl práce

Práce je zaměřena na teorii grafů, konkrétně se zaměřením na skóre grafu. Cílem bylo vytvoření programu pro určení, zda posloupnost čísel je skóre grafu a následné nakreslení grafu s daným skóre, jeho popis a návod k použití.

Pro aplikaci, která byla pojmenována GraphScore 2.0, byly stanoveny funkce, které mají uživateli nabídnout základní operace nad grafy. Kromě volného kreslení grafů na plátno byly pro aplikaci vyžadovány následující funkce:

- Porovnávání izomorfismu zadaných grafů – zda jsou dva zakreslené grafy izomorfní či nikoli.
- Vygenerování grafu na plátno dle zadaného skóre grafu.
- Možnost nastavit vzhled grafu – změna barvy hran, vrcholů a jejich velikosti.
- Možnost psát si poznámky – bez možnosti uložení.

Kromě těchto hlavních požadavků byly stanoveny vlastní cíle, které nějakým způsobem vylepší aplikaci. Tyto cíle zvyšují jak nabízené funkce, tak i uživatelskou přívětivost programu. Pro příklad je uvedena funkce vypsání přehledu vlastností o kresleném grafu. Tato funkce uvádí, zda se jedná o rovinný, eulerovský, souvislý graf, zda je to strom či kolik obsahuje komponent.

Nedílnou součástí je i dokumentace, která slouží k získání základních vědomostí týkajících se skóre grafu, k popisu aplikace a jako návod k obsluze, umístěný jako příloha č. 3. Většina použitých pojmů je v práci vysvětlena, z tohoto důvodu pro studium této práce nejsou vyžadované hluboké znalosti matematiky. Je však očekávána znalost množinových pojmů, viz [7].

3 Metodika zpracování

Cílem práce je sestavení vlastní aplikace s použitím pouze vlastních znalostí. Veškerá teoretická východiska byla získána studiem zdrojů uvedených v referencích na konci této práce. Bylo dbáno především na sestavení vlastního algoritmu v jazyce JAVA pouze se znalostmi z uvedených zdrojů. Nebylo využito žádného kopírování kódů z internetu.

Na základě znalostí programátora byl pro tento úkol vybrán jazyk JAVA (více v [12]) a jeho BufferedImage (více v [13]) pro kreslení. Technologie byly zvoleny na základě zkušeností autora - programátora. Programátor v minulosti vypracoval několik projektů s využitím stejných technologií, v rámci předmětu Počítačová grafika na Univerzitě Hradec Králové.

Veškeré použité obrázky nejsou převzaty z jiných zdrojů. Jedná se o vlastnoručně vytvořená znázornění pro účely této práce a k snazšímu pochopení problematiky.

4 Přehled použitého značení

$A, B, C, \ldots$	prvky množiny vrcholů
$a, b, c, \ldots$	prvky množiny hran
$\{A, B\}$	neorientovaná hrana mezi vrcholy A a B
$A - B$	jiné zadání neorientované hrany
(A, B)	orientovaná hrana z vrcholu A do vrcholu B
$X \subseteq Y$	množina X je podmnožinou množiny Y
$X \cup Y$	sjednocení množin X a Y
$x \in X$	prvek x je prvkem množiny X
$deg(X)$	počet hran obsahujících vrchol X
K_n	úplný graf na n vrcholech

5 Historie

Teorie grafů je velmi mladá matematická disciplína. Za její počátek se považuje rok 1736, kdy švýcarský matematik a fyzik Leonhard Euler řešil úlohu **Sedmi mostů města Königsbergu**. [8]

Zadání znělo, zda je možné přejít přes všechny mosty právě jednou a vrátit se zpět do výchozího místa. Euler si úlohu představil tak, že vrcholy označovaly břehy a hrany grafu jako mosty, které břehy spojují. Nicméně nakonec dokázal, že úloha nemá řešení. [9]

Významné postavení v teorii grafů, v oblasti skóre grafů, má český matematik, **Václav Jaromír Havel,** narozen 17. prosince 1927. V. J. Havel, doktor věd, působil jako vysokoškolský pedagog. [28] Je autorem tzv. Havlova algoritmu. Jedná se o algoritmus, který řeší problém realizace grafu, tj. jestli lze rozhodnout, zdali danou přirozenou posloupnost čísel $a_1, a_2, \dots, a_n$ lze pokládat za stupně jednotlivých vrcholů nějakého konečného grafu, viz následující kapitola, Věta 1.12. [26] Algoritmus vytváří speciální řešení, pokud existuje, nebo dokazuje, že nelze najít pozitivní odpověď, tj. graf s daným skóre. Jedná se o rekurzivní algoritmus. Poprvé byl zveřejněn roku 1955 v článku „Poznámka o existenci konečných grafů“ v Časopisu pro pěstování matematiky. [26] V roce 1962 stejný algoritmus zveřejnil i americký matematik Seifollah Louis Hakimi ve své práci [27] (proto se algoritmus také nazýván jako Havel-Hakimi algoritmus) a v roce 1973 byl zobecněn Kleitmanem a Wangem [30]. Dané problematice se věnovali a navrhovali metody pro rozhodnutí, zda posloupnost nezáporných celých čísel může být skóre grafu, také další matematici, např. v roce 1960 Erdös a Gallai [31], Ruskez, Cohen, Eades a Scott v roce 1994 [32], Barnes a Savage v roce 1997 [33], Kohnert v roce 2004 [34], Tripathi, Venugopalan a West v roce 2010 [35]. Složitost všech těchto algoritmů je $\Omega(n^2)$ (viz [43]) V roce 2011 uvedli Iványi, Lucz, Móri a Sótér v článku [36] nový algoritmus nazvaný EGL (Erdös-Gallai Linear algorithm), kterého složitost byla $\Theta(n)$ (viz. [43])

6 Teoretická východiska

K pochopení problematiky je v této části vysvětlena veškerá potřebná teorie využívaná v aplikaci GraphScore. Jedná se o souhrn nejzákladnějších znalostí týkajících se skóre grafu. Veškeré definice, věty, tvrzení a důkazy z této kapitoly jsou převzaty ze zdroje [1].

Existuje mnoho druhů grafů (obyčejný, orientovaný, ohodnocený, konečný, nekonečný atd., více v [14]). V této práci pod pojmem graf bude myšlen obyčejný graf, viz následující definice.

Definice 1.1 **Obyčejný graf** G je uspořádaná dvojice (V, E), kde V je nějaká neprázdná množina a E je množina dvoubodových podmnožin množiny V. Prvky množiny V se jmenují vrcholy grafu G a prvky množiny E hrany grafu G.

Hrana mezi vrcholy u a v se zapisuje jako $\{u, v\}$, nebo také zkráceně uv. Vrcholy spojené hranou jsou nazývány sousedními vrcholy. [1]

Při popisování pomocí grafů může dojít k situaci, kdy dva na pohled odlišné grafy popisují stejnou situaci. Mohou se lišit označením svých vrcholů a hran či jiným uspořádáním, ale pořád se může jednat o stejné grafy. Takové dva grafy se nazývají izomorfní. [6]

Definice 1.2 Dva grafy $G = (V, E)$ a $G' = (V', E')$ nazveme **izomorfní**, jestliže existuje vzájemně jednoznačné zobrazení $f: V \to V'$ takové, že platí:

$$\{x, y\} \in E <=> \{ f(x), f(y) \} \in E'.$$

Zobrazení f nazýváme izomorfismus mezi grafy G a G'. Fakt, že grafy G a G' jsou izomorfní, zapisujeme $G \cong G'$.

Podgraf grafu G' je graf G, který vznikne vybráním některých vrcholů a hran původního grafu. Podgraf je, jednoduše řečeno, část grafu. [5]

Definice 1.3 Graf $G = (V, E)$ je **podgraf** grafu $G' = (V', E')$, jestliže $V \subseteq V'$ a $E \subseteq E'$, přičemž pro hranu e mezi vrcholy v, w platí:

$$e = \{v, w\} \in E' \text{ jenom když } v, w \in V'.$$

Definice 1.4 Vrchol v (respektivě vrchol w) a hranu e grafu G nazýváme **incidentní** právě tehdy, když $e = \{v, w\} \in E$.

Definice 1.5 Vrchol v grafu G nazýváme **izolovaný** právě tehdy, když není incidentní s žádnou hranou grafu G.

Dále bude definován tzv. indukovaný podgraf, který vznikne vybráním některých vrcholů původního grafu a všech jejich incidentních hran. [5]

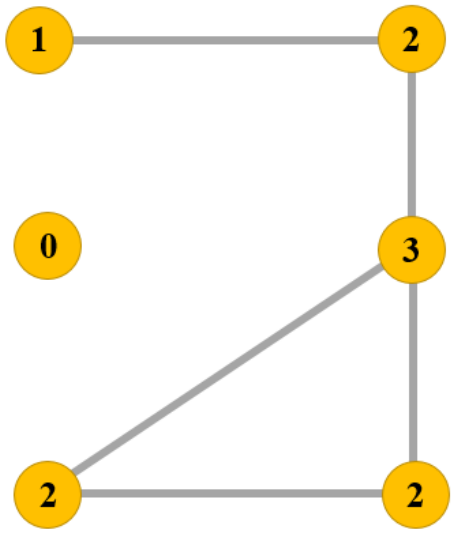
Definice 1.6 Řekněme, že graf H je **indukovaným podgrafem** grafu G, jestliže $V(H) \subseteq V(G)$ a $E(H) = E(G) \cap \binom{V(H)}{2}$.

Obrázek 3 – Indukovaný podgraf G' grafu G

Před definicí skóre grafu bude definován tzv. stupeň vrcholu grafu, který udává počet, kolik z daného vrcholu vychází hran. Právě tuto vlastnost vrcholu značí pojem stupeň vrcholu.

Definice 1.7 **Stupeň vrcholu v** v grafu G je číslo rovnající se počtu hran incidentních s vrcholem v. Značíme jej $deg_G(v)$ nebo krátce $d_G(v)$.

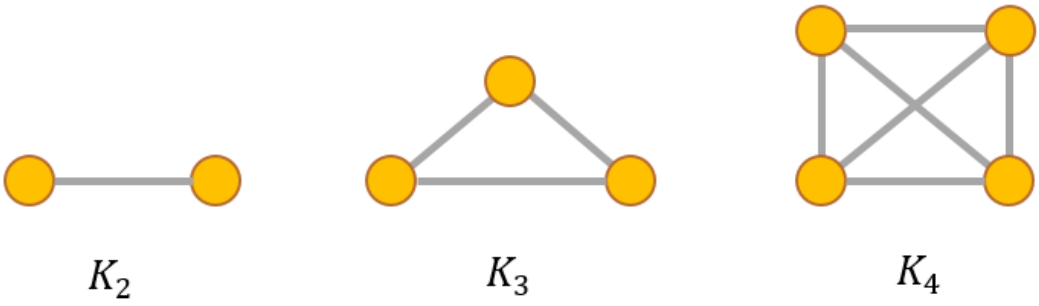
Na Obrázku 4 je zakreslen graf, kde každý vrchol má v kroužku uveden svůj stupeň $deg_G(v)$, s kolika jinými vrcholy je spojen hranou.



Obrázek 4 – Ukázka grafu s označeným stupněm jednotlivých vrcholů grafu

Definice 1.8 Úplný graf G je takový graf, jehož každé dva různé vrcholy jsou spojené hranou. Má-li úplný graf n vrcholů, značíme jej K_n.

Na Obrázku 5 jsou znázorněny grafy K_2, K_3, K_4, úplné grafy o dvou, třech a čtyřech vrcholech.



Obrázek 5 – Úplné grafy K_2, K_3, K_4

Je zřejmé, že v libovolném grafu o n vrcholech je stupeň každého vrcholu nejvýše $n - 1$. [1]

Lze přistoupit k definici samotného pojmu skóre grafu neboli grafové posloupnosti.

Definice 1.9 Nechť G je graf, v jeho vrchol. Symbolem $deg_G(v)$ označme počet hran grafu G obsahujících vrchol v. Číslo $deg_G(v)$ nazveme stupněm vrcholu v grafu G.

Označme vrcholy grafu G postupně $v_1, v_2, \dots, v_n$. Posloupnost

$$(deg_G(v_1), deg_G(v_2), \dots, deg_G(v_n))$$

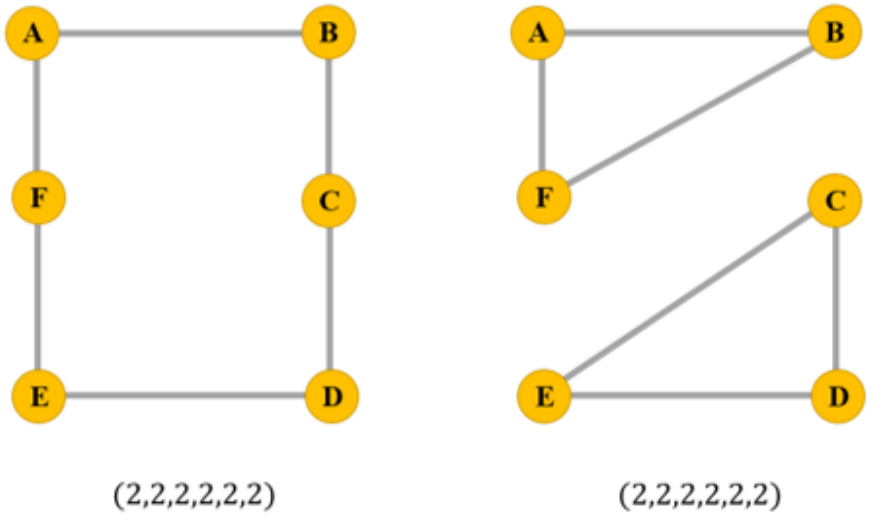
nazýváme **grafovou posloupnost stupňů grafu** G, nebo-li **skóre grafu** G.

(1,2,1,0)

Obrázek 6 - Příklad skóre grafu

Přičemž dvě skóre se považují za stejná, pokud jedno se získá z druhého přerovnáním čísel (permutací – viz [37]). To znamená, že na pořadí nezáleží, i když je zaužívané, že se většinou zapisuje od největšího stupně k nejmenšímu nebo naopak. Z toho platí, že pokud mají dva grafy různé skóre, pak nejsou izomorfní. [6, 10]

Opak tohoto tvrzení ovšem **neplatí**: Pokud mají dva grafy stejné skóre, nemusí být izomorfní, viz Obrázek 7. [1]



Obrázek 7 - Příklad dvou neizomorfních grafů se stejným skóre

Aby posloupnost čísel byla skórem grafu, musí splnit určité vlastnosti. Jedním z nich je následující věta o sudosti, kde stupeň vrcholu v udává počet hran grafu G incidentních s vrcholem v. [1]

Věta 1.10 (Princip sudosti) Pro každý graf $G = (V, E)$ platí:

$$\sum_{v \in V} deg_G(v) = 2|E|$$

Důkaz předešlého tvrzení je zřejmý:

Každá hrana je započítána dvakrát. Jednou do stupně vrcholu, ve kterém hrana začíná, podruhé do stupně vrcholu, ve kterém končí. Sečteme-li tedy stupně všech vrcholů grafu, dostaneme dvojnásobek počtu hran. [1] ■

Důsledek 1.11 Počet vrcholů lichého stupně je v každém grafu sudý.

Důsledek 1.11 nestačí pro charakterizaci skóre grafu. Aby daná posloupnost byla skórem nějakého grafu, musí pro ni platit následující věta:

Věta 1.12 (Havel-Hakimi Věta o skóre) [27] Nechť $D = (d_1, d_2, \dots, d_n)$ je posloupnost přirozených čísel. Předpokládejme, že $d_1 \leq d_2 \leq \dots \leq d_n$, a označme symbolem D' posloupnost $(d'_1,\ d'_2,\ \dots, d'_{n-1})$, kde

$$d = \begin{cases} d_i & \text{pro } i < n - d_n \\ d_i - 1 & \text{pro } i \geq n - d_n. \end{cases}$$

Potom D je skóre grafu právě když D' je skóre grafu.

Důkaz věty 1.12 [1]

Havel-Hakimi věta je ekvivalence, proto je nutné dokázat dvě implikace:

$\Leftarrow$: Nechť D' je skóre grafu $G' = (V', E')$, kde $V' = \{v_1, \dots, v_{n-1}\}$ a $deg_G(v_i) = d'_i$. Pak bude zvolen nový vrchol v_n, různý od $v_1, \dots, v_{n-1}$, a definován nový graf $G = (V, E)$ následovně:

$$V = V' \cup \{v_n\}$$

$$E = E' \cup \{\{v_i, v_n\};\quad i = n - d_n, n - d_n + 1, \dots, n-1\}.$$

Jinak řečeno: nový vrchol v_n bude připojen k posledním d_n vrcholům grafu G'. Je zřejmé, že skóre grafu G je právě $D = (d_1, d_2, \dots,\ d_{n-1}, d_n) =$

$= \left(d'_1, d'_2, \dots,\ d'_{n-d_n} + 1, d'_{n-d_n+1} + 1, \dots., d'_{n-1} + 1, d_n\right)$.

Tím je dokázaná implikace $\Leftarrow$.

$\Rightarrow$: Nechť D je skóre nějakého grafu. Je uvažována množina $\mathcal{G}$ všech grafů na množině vrcholů $\{v_1, \dots, v_n\}$, kde $\deg v_i = d_i$ (tj. mají skóre $D = (d_1, d_2, \dots, d_n)$). Je potřeba dokázat následující pomocné tvrzení:

Pomocné tvrzení: V množině $\mathcal{G}$ existuje graf G_0, v němž je vrchol v_n spojen právě s vrcholy $v_{n-d_n}, v_{n-d_n+1}, \dots, v_{n-1}$ (s posledními d_n vrcholy).

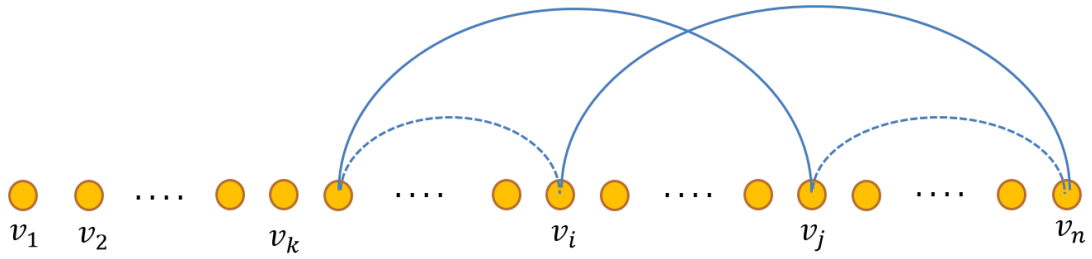
Bude-li graf G_0 jako v pomocném tvrzení, je zřejmé, že graf $G' = (\{v_1, \dots, v_{n-1},\}, E')$, kde $E' = \{e \in E(G_0);\ v_n \notin e\}$, má skóre D', a tím bude věta o skóre dokázána. Zbývá dokázat pomocné tvrzení.

Pokud $d_n = n-1$, tj. vrchol v_n je spojen se všemi vrcholy $v_1, \dots, v_{n-1}$, pak pomocnému tvrzení vyhovuje kterýkoli graf z $\mathcal{G}$ a je hotovo. Když to neplatí, v_n není spojen se všemi ostatními vrcholy, bude definované pro každý graf $G \in \mathcal{G}$ číslo $j(G)$, což bude největší index $j \in \{1, 2, \dots, n-1\}$ takový, že $\{v_j, v_n\} \notin E(G)$. Nechť G_0 je graf, pro nějž je $j(G)$ nejmenší možné; je potřeba dokázat, že $j(G_0) = n - d_n - 1$, z čehož je už patrné, že G_0 vyhovuje pomocnému tvrzení.

Pro spor bude přepokládano, že $j = j(G_0) > n - d_n - 1$. Vrchol v_n musí být spojen s d_n vrcholy a z nich nejvýš $d_n - 1$ může následovat po vrcholu v_j. Proto existuje nějaké $i < j$ takové, že v_i je spojen s vrcholem v_n. Pak $\{v_j, v_n\} \notin E(G_0)$, $\{v_i, v_n\} \in E(G_0)$. Vzhledem k tomu, že $deg_{G_0}(v_i) \leq deg_{G_0}(v_j)$, existuje nějaký vrchol v_k, který je spojený hranou s v_j, ale nikoli s v_i. V této situaci je uvažován nový graf $G = (V(G_0), E')$, kde

$$E' = \left(E(G_0) \setminus \left\{\{v_i, v_n\}, \{v_j, v_k\}\right\}\right) \cup \left\{\{v_j, v_n\}, \{v_i, v_k\}\right\}.$$

Viz následující Obrázek 8:



Obrázek 8 – Hrany mezi vrcholy k důkazu věty 1.12

Je snadno vidět, že graf G' má rovněž skóre D, a přitom $j(G') \leq j(G_0) - 1$, což je spor s volbou grafu G_0. Tím je dokázáno pomocné tvrzení a také implikace $\Rightarrow$, a tudíž i věta 1.12. ■

V praxi se věta o skóre aplikuje na posloupnost tak, že se ze seřazené posloupnosti odebere poslední (nejvyšší) stupeň d_n a od tolika d_n bezprostředně předchozích stupňů se odečtou jedničky. Zbývající stupně se nezmění. Následně je zapotřebí posloupnost opět seřadit dle velikosti. [11] Postup je předveden v následujícím příkladu:

Příklad k Větě 1.12:

Určete, zda posloupnost $(1, 2, 2, 3, 3, 5)$ je skóre grafu.

Řešení:

Nejprve zjistíme, zda počet lichých čísel je sudý.

$1 + 2 + 2 + 3 + 3 + 5 = 16$ **Ano**

Následně se podíváme, zda poslední číslo je menší než počet čísel v posloupnosti.

$5 < 16$ **Ano**

Můžeme tedy aplikovat Větu 1.12 na posloupnost, o které zjistíme, zda je nebo není skóre nějakého grafu. V každém kroku bude seřazena posloupnost čísel, následně odebráno nejvyšší číslo, jehož hodnota udává kolika po něm následujícím číslům bude odečtena hodnota 1.

$$D = (\mathbf{1}, \mathbf{2}, \mathbf{2}, \mathbf{3}, \mathbf{3}, \mathbf{5})$$

$$D = (\mathbf{1} - 1, \mathbf{2} - 1, \mathbf{2} - 1, \mathbf{3} - 1, \mathbf{3} - 1, \cancel{\mathbf{5}})$$

V prvním kroku bylo odebráno nejvyšší číslo hodnoty 5. To znamená, že pěti následujícím číslům v posloupnosti bude hodnota zmenšena o 1.

$$D = (\mathbf{0}, \mathbf{1}, \mathbf{1}, \mathbf{2}, \mathbf{2})$$

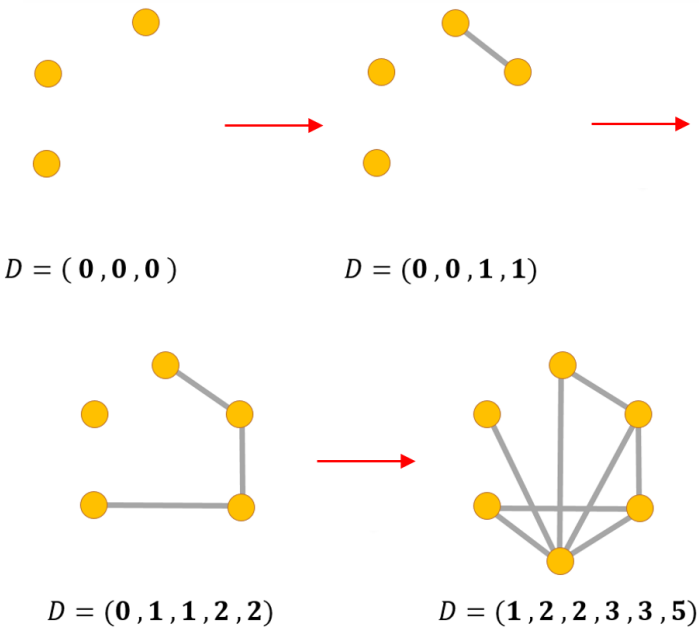
$$D = (\mathbf{0}, \mathbf{1}, \mathbf{1} - \mathbf{1}, \mathbf{2} - 1, \cancel{\mathbf{2}})$$

Po seřazení v druhém kroku bylo odebráno číslo 2. Budou tedy zmenšeny pouze dvě následující čísla, ostatním číslům se hodnota nezmění.

$$D = (\mathbf{0}, \mathbf{0}, \mathbf{1}, \mathbf{1})$$

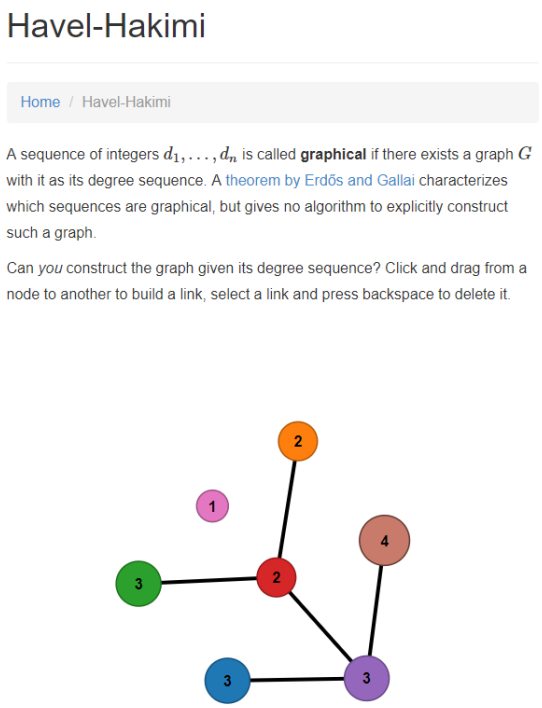
$$D = (\mathbf{0}, \mathbf{0}, \mathbf{0})$$

Lze vidět, že posloupnost $(0, 0, 0)$ je skóre grafu, jedná se totiž o graf skládající se ze tří izolovaných vrcholů, pak i původní posloupnost je skóre grafu. Kontrolu můžeme provést opačným postupem, kdy do grafu $(0, 0, 0)$ postupně přidáváme jednotlivé vrcholy a podle jejich stupně ho spojujeme s ostatními vrcholy grafu, viz Obrázek 9.



Obrázek 9 - Opačný postup rozboru skóre z příkladu k Větě 1.12

Zajímavost: Algoritmus Havel-Hakimi byl využit také v on-line „hře“, na stránkách http://jacquerie.github.io/hh/, kde se hráč snaží spojit vrcholy hranami na základě jejich stupně, viz Obrázek 10. [29]



Obrázek 10 – On-line hra Havel-Hakimi[29]

V následující části této kapitoly budou vysvětleny jednotlivé typy grafu, které jsou využívány v aplikaci GraphScore. Jedná se o grafy rovinné, souvislé, eulerovské či stromy.

Nejdříve budou definovány pojmy sled, tah, cesta a kružnice, které jsou používány v popisu aplikace.

Definice 1.13

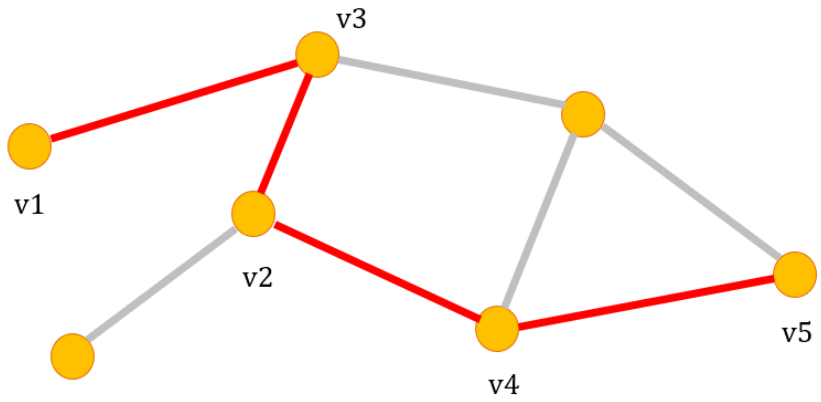
Sled délky k, $k \geq 0$, v grafu G je posloupnost $(v_0, e_1, v_1, \ldots, e_k, v_k)$, kde $e_i = \{v_{i-1}, v_i\}$, $i = 1, \ldots, k$.

Tah délky k, $k \geq 0$, v grafu G je sled délky k s navzájem různými hranami, tj. pro $i \neq j$ platí $e_i \neq e_j$.

Uzavřený sled délky k, $k \geq 0$, v grafu G je sled délky k, ve kterém $v_0 = v_k$.

Uzavřený tah délky k, $k \geq 0$, v grafu G je tah délky k, ve kterém $v_0 = v_k$.

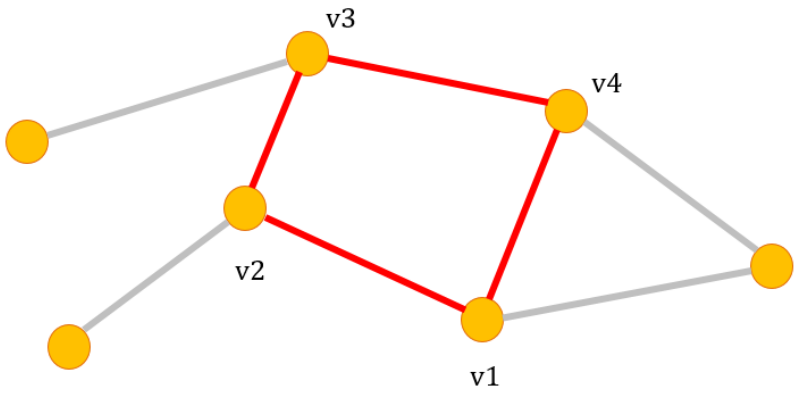
Cesta délky k, $k \geq 0$, v grafu G je sled délky k s navzájem různými vrcholy, tj. pro $i \neq j$ platí $v_i \neq v_j$.



Obrázek 11 - Příklad cesty v grafu (v1, v2, v3, v4, v5)

Cesta vede mezi dvěma vrcholy právě tehdy, když je mezi nimi hrana, nebo pokud je možné se po hranách grafu přes jiné vrcholy dostat z jednoho do druhého. Takto definovaná cesta se nazývá cestou z v_0 do v_t délky t. [11]

Definice 1.14 **Kružnice délky k**, $k \geq 3$, v grafu G je posloupnost $(v_0, e_1, v_1, \ldots, e_k, v_0)$, kde $e_i = \{v_{i-1}, v_i\}$, }, $i = 1, \ldots, k-1$, $e_k = \{v_{k-1}, v_0\}$ a pro $i \neq j$ platí $v_i \neq v_j$.



Obrázek 12 - Příklad kružnice velikosti 4

Jestliže graf obsahuje alespoň dva vrcholy, mezi kterými vedou dvě různé cesty, pak obsahuje kružnici. Délka kružnice je rovna celkovému počtu hran, které ji tvoří. [1, 6]

Jak je zřejmé, definice cesty a kružnice jsou si velice podobné, jedná se vždy o posloupnosti hran a vrcholů. Avšak u kružnice je první a poslední vrchol posloupnosti stejný. [11]

Jakmile byly zavedeny pojmy sled, tah, cesta a kružnice, lze přistoupit k samotným vlastnostem grafu.

Definice 1.15 Řekneme, že graf G je **souvislý**, jestliže pro každé dva jeho vrcholy x a y v něm existuje cesta z x do y.

Vrcholy x a y souvisí v grafu G, když existuje cesta mezi x a y v G.

Definice 1.16 Komponenta grafu G je maximální souvislý podgraf grafu G a značíme ji písmenem k. [1]

Graf G je vlastně sjednocením všech jeho podgrafů. Graf G je souvislý, právě tehdy, když počet jeho komponent je roven jedné ($k = 1$). [1]

Všichni asi znají úlohu, jak nakreslit domeček jedním tahem. Právě tuto schopnost řeší další vlastnost grafu, zda je graf eulerovský. Označují se tak grafy, které lze „kreslit“ jedním tahem, kdy se začíná a končí ve stejném vrcholu.

Definice 1.17

Eulerovský tah v grafu G je uzavřený nebo otevřený tah, který obsahuje všechny hrany grafu G.

Eulerovský graf je souvislý graf G, ve kterém existuje eulerovský tah.

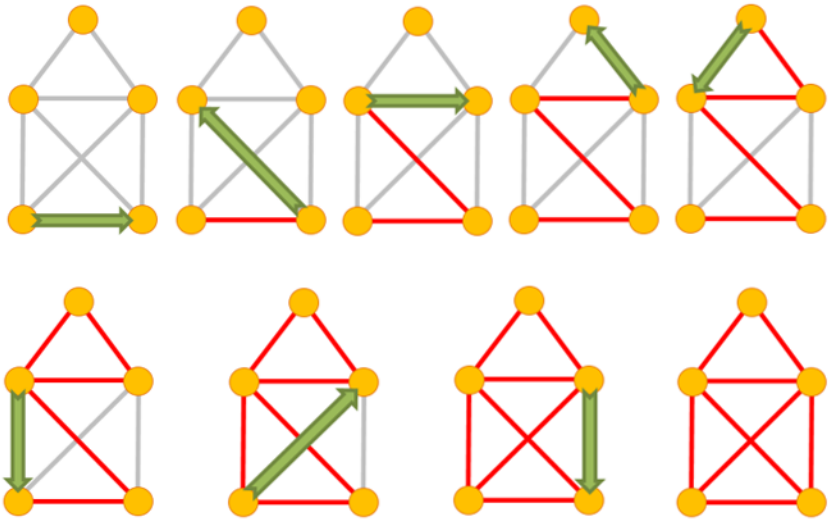
Tento typ grafu nese pojmenování podle slavného švýcarského matematika Leonharda Eulera z roku 1736, kdy řešil problém sedmi mostů v Královci. [8]

Tehdy městští radní potřebovali vědět, jestli mohou přejít po každém ze sedmi mostů právě jednou. Problém byl vyřešen právě tímto definovaným způsobem. [9]

Věta 1.18 (Charakterizace eulerovských grafů)

Pro každý graf $G = (V, E)$ platí:

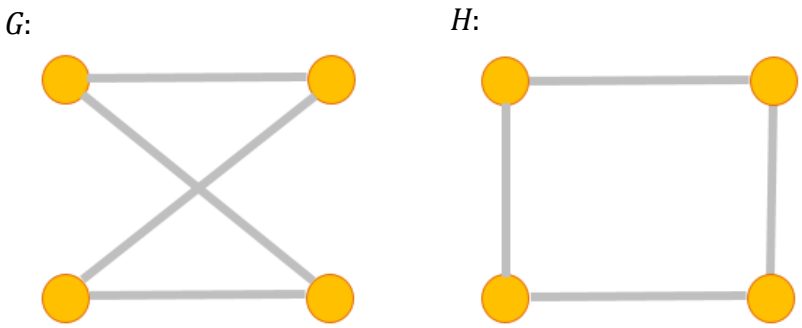
Graf G je eulerovský právě tehdy, když graf G je souvislý a má buď všechny vrcholy sudého stupně, nebo právě dva vrcholy lichého stupně.



Obrázek 13 - Eulerovský tah

Definice 1.19 **Rovinné nakreslení** grafu G je zobrazení, ve kterém jsou vrcholy znázorněny jako různé body v rovině a hrany jako oblouky (jednoduché křivky) spojující body svých koncových vrcholů. Přitom hrany se nesmí nikde křížit ani procházet jinými vrcholy než svými koncovými body.

Definice 1.20 Graf G je **rovinný** právě tehdy, pokud má rovinné nakreslení.



Obrázek 14 – *H* je rovinné nakreslení grafu *G*.

Dalším pojmem, který bude nyní vysvětlen, je strom. Jedná se o speciální typ grafu.

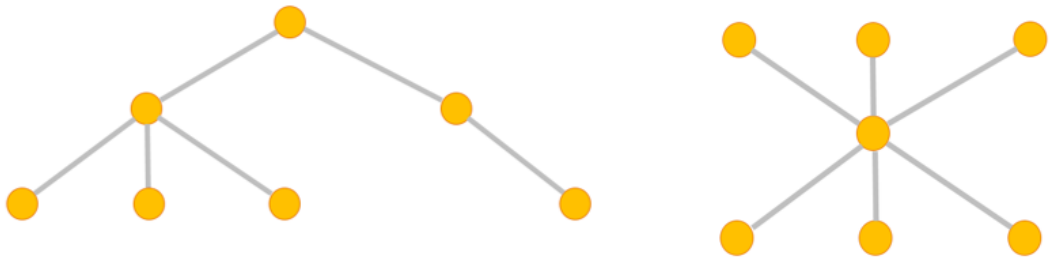
Definice 1.21 **Strom** je souvislý graf neobsahující kružnici.

Pro stromy platí následující vlastnosti:

- Mezi každými dvěma vrcholy vede právě jedna cesta.
- Každý strom má $n - 1$ hran, kde n je počet vrcholů grafu. [6]

Strom obsahující pouze jeden vrchol nazýváme triviálním stromem. Stromy se často využívají pro hierarchické uspořádání či ukládání dat v informatice v podobě datové struktury (například binární vyhledávací strom – viz [38]) [1, 6]

Definice 1.22 Nechť $T = (V, E)$ je strom. Vrchol $v \in V$ stupně jedna nazýváme **listem** stromu T, vrchol $v \in V$ stupně většího než jedna nazýváme **vnitřním vrcholem** stromu T.



Obrázek 15 - Příklady stromů

Při generování grafu podle zadaného skóre v aplikaci GraphScore je využíván princip prohledávání grafu. Spočívá v provedení sémantického průchodu všemi vrcholy grafu. Práce se zabývá dvěma druhy prohledávání grafu, do hloubky a do šířky. [3]

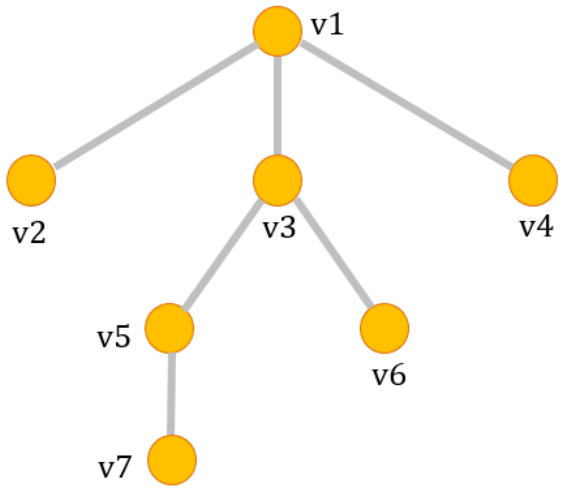
Procházení „do hloubky“ – úschovna U, nazývaná **LIFO**, je implementována jako **zásobník**, tj. dále se prohledává od posledních nalezených vrcholů. Úschovna U je posloupnost navštívených vrcholů grafu. [3]

Procházení „do šířky“ – úschovna U, označována jako **FIFO**, je implementována jako **fronta**, tj. dále se prohledává od prvních nalezených vrcholů. Úschovna U je posloupnost navštívených vrcholů grafu. [3]

Datové struktury LIFO a FIFO jsou využívány k určení, z kterého vrcholu bude provedeno rozšíření. [3] Rozdíl mezi těmito principy lze jednoduše pochopit z následujícího příkladu:

Příklad k definici č. 1.24:

Mějme graf, který je stromem (Obrázek 16).



Obrázek 16 – Graf příkladu k definici 1.24 – Prohledávání stromu

Je zřejmé, že při použití pravidla procházení do šířky, se do úschovny U zařazují postupně vrcholy ve stejné vrstvě (úrovni) stromu. Pohybuje se tak do šířky a do úschovny U budou přidány nejprve vrcholy z 0. vrstvy, tedy vrchol v_1, a následně vrcholy z dalších vrstev stromu: $(v_1, v_2, v_3, v_4, v_5, v_6, v_7)$.

Naopak u pravidla „do hloubky“ je snaha se co nejvíce se zanořit dolů do stromu. Pohybuje se tak do hloubky a do úschovny U budou postupně zařazeny vrcholy: $(v_1, v_2, v_3, v_5, v_7, v_6, v_4)$.

7 Implementace GUI[2]

Program GraphScore 2.0 je zaměřen na matematické téma teorie grafů, přesněji na skóre grafu. Umožňuje zjištění, zda posloupnost je skóre grafu a následně znázornit graf, získat jeho vlastnosti a usnadňuje pochopit danou problematiku uživateli. Aplikace je naprogramována v jazyce JAVA s použitím potřebných knihoven.

V následujících částech budou popsány implementace použitých prvků uživatelského rozhraní.

7.1 Okno

Okno programu má pevně definovanou výšku a šířku. Nelze tedy měnit jeho velikost, která činí 1280x856 pixelů. Fixní velikost okna není nejlepší řešení, ale ulehčuje to práci s plátnem. Bylo určeno právě toto rozlišení, protože je vhodné pro 95% všech počítačů podle světové statistiky displejových rozlišení (viz [44]). Na celém okně jsou aplikovány rozhraní ComponentListener [16] a WindowListener [17], které umožňují zachytávat pohyby okna. S jejich pomocí je okno vykresleno i v případě, že se s oknem hýbe.

7.2 Plátno

Pro reprezentaci plátna, na kterém je samotný graf vykreslován, byl použit BufferedImage. Jeho hlavní výhodou je, že se plátno dokola nepřekresluje. Tento způsob by byl zbytečný a náročný, z tohoto důvodu je plátno překreslováno pouze

[2] GUI znamená grafické uživatelské rozhraní, více na [15]

po určitém uživatelském úkonu. Aplikace nevyužívá žádných vláken. Velikost plátna je 1080x856 pixelů (v režimu „Izomorfismus“ – Kapitola 9.5 – jsou to dvě okna velikosti 585x866). Na celém plátně jsou aplikovány rozhraní MouseListener [18] a MouseMotionListener [19], které umožňují snímat pohyby myši v plátně, díky kterým je možné v plátně kreslit.

7.3 Objekty plátna

Plátno slouží ke grafické reprezentaci dvou odlišných tříd objektů.

Třída „Vrchol“ slouží k reprezentaci bodů neboli vrcholů grafu. Mezi hlavní atributy vrcholu patří pozice na plátně (souřadnice X a Y), název, popis, barva, šířka a plátno (BufferedImage), viz Obrázek 21, bod 1. Vlastnost popis je jediná, která může nabývat hodnoty „null“. Barva a šířka vrcholu jsou po spuštění aplikace nastaveny na defaultní hodnoty, za běhu aplikace ale mohou být změněny. Příklad vytvořené instance třídy „Vrchol“ lze nalézt na Obrázku 21, bod 2.

Třída „Hrana“ slouží k reprezentaci hran mezi vrcholy. Hrany jsou definovány dvěma vrcholy (počáteční a koncový vrchol), barvou a tloušťkou. Počáteční a koncový vrchol nemohou být jeden a ten samý vrchol. Barva a tloušťka hrany jsou po spuštění aplikace nastaveny na defaultní hodnoty, za běhu aplikace ale mohou být změněny. Příklad vytvořené instance třídy „Hrana“ lze nalézt na Obrázku 21, bod 3.

Každé plátno má jiné vrcholy a jiné hrany, které si uchovává v kolekcích. Díky tomu je možné přepínat mezi jednotlivými plátny s uchováním vrcholů a hran.

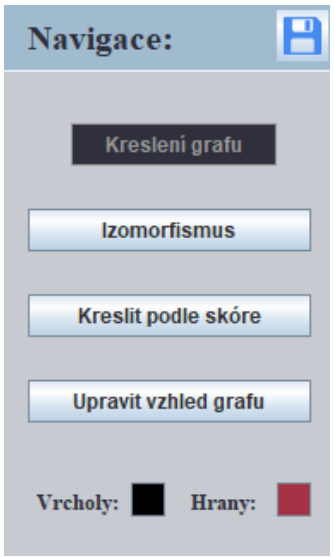
Po kliknutí pravým tlačítkem myši nad některý z vrcholů, se zobrazí jeho detailní popis v novém JFrame okně [20], obsahující jeho název, popis, id a stupeň. Dále je zde možnost smazat daný vrchol. Výsledkem je odstranění vrcholu z kolekce vrcholů a aktualizace plátna.

7.4 Menu

Nabídka menu programu GraphScore v2.0 je implementován jako JPanel, který se skládá ze dvou dalších JPanelů (panel1, viz Obrázek 17, a panel2, viz Obrázek 18).

Panel1 (Obrázek 17) slouží k hlavní navigaci. Zde je možné najít tlačítka v podobě JButtonů [21]. Pomocí tlačítek „Kreslení grafu“, „Izomorfismus“ a „Kreslit podle skóre“ lze přepínat mezi jednotlivými plátny, kde každé plátno je určeno k vyřešení jiného problému.

- Tlačítko „Upravit vzhled grafu“ umožňuje změnit vzhledovou stránku grafu. Po kliknutí se zobrazí nový JFrame s nabídkou úprav, jako například změnit barvy vrcholů a hran, jejich šířky nebo šířky budoucích vrcholů a hran.
- Tlačítko pro uložení v pravém horním rohu využívá třídy Saver, která pracuje s JFileChooserem [22] a umožňuje tak uložit graf do uživatelem zvoleného cíle, typu .JPG, .PNG a .GIF.



Obrázek 17 – Panel1 – Hlavní navigace

Panel2 (Obrázek 18) se také skládá z několika JButtonů, které nabízejí určité akce s daným plátnem. Každé plátno má však odlišné akce.

Tlačítka uvnitř panelu „Vlastnosti“ vždy upravují nebo pracují s kolekcí vrcholů či hran. Nabízejí přidávání nových vrcholů, vyčištění celého plátna (vyčištění kolekcí), odstranění některých nebo všech hran.

- „Přidej vrchol“ otevře nový JFrame s kolonkami pro vyplnění názvu, volitelného popisu vrcholu a pro zvolení jeho výchozí polohy na plátně. Po kliknutí na výběr polohy bodu bude uživatel přesměrován na hlavní okno s plátnem, které dostane „focus“. Pomocí MouseListeneru a levého tlačítka myši je následně vybrán bod z plátna a „focus“ se předá zpět do formuláře přidání vrcholu. Pro vytvoření vrcholu nesmí být název a poloha rovna hodnotě „null“. Nový vrchol bude přidán do listu vrcholů.
- Odstranit hranu lze dvěma způsoby, jednu po druhé nebo všechny najednou. Při výběru „Odstranit hranu“ se zobrazí nový JFrame se seznamem hran (JTextField, [23]) a tlačítky, v podobě JButtonů, vedle nich sloužící k jejich odstranění. V obou případech dojde k smazání příslušných hran z kolekce a k aktualizaci plátna.
- „Poznámka“ se nachází úplně dole a nabízí uživateli nový JFrame tvořený pouze z JTextArea [24] pro napsání rychlé poznámky.
- Posledním tlačítkem v panelu je „Návod“, které po kliknutí otevře uživatelský návod typu .docx výchozí aplikací v počítači.

Dále se v panelu „Vlastnosti“ nacházejí speciální tlačítka, která jsou charakteristická pro dané plátno.

- „Zobrazit vlastnosti“ (pouze v režimu „Kreslení grafu“) nereaguje na kliknutí, nýbrž na najetí myši pomocí rozhraní MouseListener. Rychle tak lze získat důležité informace o zakresleném grafu.

- „Jsou izomorfní?“ (pouze v režimu „Izomorfismus“) výsledkem je informační okno JOptionPane [25] s výsledkem, zda dva zadané grafy jsou izomorfní.
- „Zadat skóre“ (pouze v režimu „Izomorfismus“) po kliknutí se zobrazí nový JFrame s kolonkou pro vyplnění skóre v definovaném tvaru. Uživatel následně si může nechat vygenerovat graf na plátno podle skóre, které zadal do pole. Graf se vygeneruje podle algoritmu kreslení „do hloubky“.
- Pokud existuje i jiný způsob, jak graf vykreslit, vyskočí na uživatele v panelu1 nové tlačítko „Alternativní graf“ (pouze v režimu „Izomorfismus“), které vykreslí skóre grafu druhým algoritmem tzv. „do šířky“. Samotná implementace obou postupů je vysvětlena v následující kapitole.
- Po vygenerování grafu do plátna, uživatel dostane novou možnost v panelu1 a to „Rozbor skóre“ (pouze v režimu „Izomorfismus“), jehož výsledkem je nový JFrame obsahující rozbor skóre. Jeho cílem je, aby uživatel porozuměl postupu rozboru skóre.

Obrázek 18 – Panel2 – Akce a vlastnosti grafu

8 Algoritmy

Aplikace GraphScore v2.0 nabízí několik funkcí, které vyžadují implementaci. V dnešní době lze tyto implementace snadno vyhledat na internetu. Tyto webové zdroje v práci nebyly použity. Veškeré algoritmy, týkající se teorie grafů, byly navrženy a naprogramovány samotným autorem aplikace. Důraz byl kladen především na vlastnoručně napsaný kód programu.

V následujících částech budou popsány stěžejní algoritmy aplikace GraphScore.

8.1 Izomorfismus

Postup rozhodnutí, zda jsou dva zadané grafy izomorfní, se skládá ze tří částí. Nejprve se porovnávají jejich skóre, a pokud se rovnají, posuzuje se, jestli každý vrchol prvního grafu má stejné hrany s ostatními vrcholy jako druhý graf a nakonec se zjišťuje, zda oba grafy obsahují stejný počet kružnic velikosti tři.

Srovnání skóre

Na základě zakreslených grafů do obou pláten aplikace získá jejich skóre. Pod pojmem skóre grafu si můžeme představit posloupnost čísel, kde každé číslo reprezentuje stupeň jednoho vrcholu grafu, neboli s kolika dalšími vrcholy je vrchol spojen hranou. Aplikace si uchovává informace o zakreslených hranách a vrcholech v podobě listů objektů. Na základě těchto znalostí je vytvořen číselný řetězec reprezentující skóre grafu. Posloupnost stupňů je následně uspořádána sestupně a porovnána s posloupností druhého grafu. Aby dva grafy byly izomorfní, shoda jejich skóre je nutnou podmínkou.

Porovnání stupňů sousedních vrcholů

Při shodě skóre aplikace pokračuje ve vyhodnocení izomorfismu tak, že projde každý vrchol grafu, zjistí jaké má hrany, respektive s jakým jiným vrcholem je spojen. Každá hrana je reprezentována jako spojnice vrcholů o nějakém stupni. Aplikace testuje má-li hrana v druhém grafu stejné informace.

Algoritmus prochází všechny vrcholy obou pláten a u každého vrcholu z prvního plátna zjišťujeme, jaký stupeň mají jeho sousední vrcholy. Následně hledáme takový vrchol z druhého plátna, jehož sousedé budou mít stejné stupně jako vrchol z prvního plátna. Počet takových shod sousedních vrcholů musí být roven počtu sousedů vrcholu prvního plátna. V případě, že mají oba vrcholy stejné sousední vrcholy, označíme vrchol z prvního plátna jako navštívený a už se tedy nebude pokoušet ho porovnávat s jiným vrcholem z druhého plátna. Ukázka algoritmu viz Algoritmus 1.

Krokový postup:

1. Z grafu 1 se vezme nenavštívený vrchol v_1
2. Z grafu 2 se vyhledá nenavštívený vrchol v_2, jehož stupeň je shodný s vrcholem v_1
    a. Následně se porovnává stupeň sousedního vrcholu k vrcholu v_1 se sousedním vrcholem vrcholu v_2
        i. Počet shod je ukládán
        ii. Při neshodě se opakuje bod a
    b. Celkový počet shod musí být roven stupni vrcholu v_1
        i. V případě, že se počet neshoduje, opakuje se celý proces od bodu 2
        ii. V případě, že se shoduje, pak je vrchol v_1 shodný s vrcholem v_2. Počet takových shod je ukládán a vrcholy jsou označeny jako navštívené.
    c. Následně je opakován bod 1 pro nalezení dalšího shodného vrcholu
3. Celkový počet shodných vrcholů v_1 s vrcholy v_2 musí být roven celkovému počtu vrcholů grafu 1.
    a. Pokud není roven, pak grafy nejsou izomorfní
    b. Pokud se počet shod rovná počtu vrcholů, pak lze přistoupit k dalšímu testování.

Algoritmus 1 - Porovnávání stupňů sousedních vrcholů

```
// porovnávání stupňů sousedních vrcholů
for (int i = 0; i < vrch1.size(); i++) {
for (int k = 0; k < vrch2.size(); k++) {
   if (vrch1.get(i).navstiveno == false && vrch2.get(k).navstiveno == false) {
      for (Vrchol v1 : vrch1.get(i).getSousedi()) {
         v1.setProzkoumano(false);
      }
      for (Vrchol v1 : vrch2.get(k).getSousedi()) {
         v1.setProzkoumano(false);
      }
      int uspechu = 0;
      if(vrch2.get(k).getStupen() == vrch1.get(i).getStupen()){
         for(int j=0; j<vrch1.get(i).getSousedi().size(); j++){
            int count = 0;
            int soucet = 0;
            for (int l=0; l<vrch2.get(k).getSousedi().size(); l++) {
               if ((vrch1.get(i).getSousedi().get(j).getStupen() ==
               vrch2.get(k).getSousedi().get(l).getStupen()) &&
               (vrch2.get(k).getSousedi().get(l).getProzkoumano() == false)) {
                  count++;
                  vrch2.get(k).navstiveno = true;
                  vrch2.get(k).getSousedi().get(l).setProzkoumano(true);
               }
            }
            int chyby = 0;
            for (int h = 0; h < vrch2.get(k).getSousedi().size(); h++) {
               if (vrch2.get(k).getSousedi().get(h).getProzkoumano() == true) {
                  chyby++;
               }
            }
            if ((chyby == vrch2.get(k).getSousedi().size() && count == 0)
                     || (count == 0 && soucet == 0)) {
               JOptionPane.showMessageDialog(null, "Grafy nejsou izomorfní.",
                     "Izomorfismus", 1);
               return;
            }
            soucet++;
            if (count != 0) {
               uspechu++;
               for (int l = 0; l < vrch2.get(k).getSousedi().size(); l++) {
                  vrch2.get(k).getSousedi().get(l).setProzkoumano(false);
               }
            } else {
                j--;
            }
            }
         }
         if (uspechu == vrch1.get(i).getSousedi().size()) {
            vrcholy++;
            vrch1.get(i).navstiveno = true;
         } else
            continue;
      }
   }
}
```

Porovnání počtu kružnic

Následuje poslední testovací algoritmus. Aby byly grafy izomorfní, oba musí mít stejný počet kružnic stejné délky, platí pro libovolné n, označující délku kružnice C_n. [11] Nicméně algoritmus, který by porovnával veškeré kružnice v grafech, by byl velice implementačně náročný. Z tohoto důvodu program GraphScore zjišťuje a porovnává počet kružnic pouze velikosti tři v obou grafech. Přestože se jedná o zjednodušený algoritmus, který je nejefektivnější pro grafy obsahující kružnice o velikosti nejvýše tři, neměl v testování žádný negativní vliv na výsledky.

Algoritmus 2 - Algoritmus porovnávání počtu kružnic

```
if (vrcholy == vrch1.size()) {
   // počítání kružnic v plátně 1
   int kruznice1 = 0;
   for (int i = 0; i < vrch1.size(); i++) {
      for (int j = 0; j < vrch1.get(i).getSousedi().size() - 1; j++) {
         for (int k = j; k < vrch1.get(i).getSousedi().size(); k++) {
            for (Vrchol v : vrch1.get(i).getSousedi().get(j).getSousedi()) {
              if(v.getNazev() == vrch1.get(i).getSousedi().get(k).getNazev()){
                 kruznice1++;
              }
             }
          }
       }
   }
   // počítání kružnic v plátně 2
   int kruznice2 = 0;
   for (int i = 0; i < vrch2.size(); i++) {
      for (int j = 0; j < vrch2.get(i).getSousedi().size() - 1; j++) {
         for (int k = j; k < vrch2.get(i).getSousedi().size(); k++) {
            for (Vrchol v : vrch2.get(i).getSousedi().get(j).getSousedi()) {
             if (v.getNazev()==vrch2.get(i).getSousedi().get(k).getNazev()) {
                 kruznice2++;
              }
             }
          }
   }}
   // porovnání počtu kružnic velikosti 3
   if (kruznice1 == kruznice2) {
       JOptionPane.showMessageDialog(null, "Grafy jsou izomorfní.", "Izomorfismus",
       1);
      return;
   } else {
       JOptionPane.showMessageDialog(null, "Grafy nejsou izomorfní.", "Izomorfismus",
       1);
      return;
   }
} else {
   JOptionPane.showMessageDialog(null, "Grafy nejsou izomorfní.", "Izomorfismus", 1);
   return;
}
```

Pokud kterýkoli z předešlých testů vyhodnotí, že daný graf se neshoduje s druhým grafem, aplikace informuje uživatele, že zadané grafy nejsou izomorfní.

Jedná se o vlastní algoritmus autora, který byl testován na 70 dvojicích grafů a ve všech se dostavil správný výsledek. Program bude dále testován samotnými uživateli a v případě objevení chyby bude průběžně laděn. Každý program se vyladí až průběžným používáním.

8.2 Generování grafu

Aplikace GraphScore v2.0 nabízí uživateli možnost vykreslení grafu aplikací na základě zadaného skóre. Princip spočívá v odděleném generování vrcholů a hran grafu. Vytváření grafu dle skóre je možné dvěma způsoby - „do hloubky“ a „do šířky“, přičemž vykreslení samotných vrcholů není nikterak vázáno na zvoleném způsobu. Novým vrcholům se přidělují názvy podle abecedy a na plátně jsou vykreslovány na kružnici ve středu plátna. V závislosti na jejich počtu jsou vrcholy rozprostřeny na kružnici, aby vzdálenost rozestupu mezi nimi byla vždy stejná. V případě, že uživatel zadá skóre jednoho vrcholu, vykreslí se jeho poloha přímo doprostřed plátna. Algoritmus generování vrcholů:

Algoritmus 3 - Generování vrcholů

```
// generování bodů do plátna do kruhu
public void generatePoints(int vrcholy) {
   clear();
   if (vrcholy > 0) {
      int x;
      int y;
      char[] alphabet = "ABCDEFGHIJKLMNOPQRSTUVWXYZ".toCharArray();
      if (vrcholy > 1) {
         for (int i = 0; i < vrcholy; i++) {
            double angle = 2 * Math.PI * i / vrcholy;
            x = (int) Math.round((sirka / 2) + 300 * Math.cos(angle));
            y = (int) Math.round((vyska / 2) + 300 * Math.sin(angle));
               mapaservice.pridejVrchol(new Vrchol(x, y, String.valueOf(alphabet[i]),
               " ", null, new Color(0, 0, 0)));
         }
      } else if (vrcholy == 1) {
               mapaservice.pridejVrchol(new Vrchol(sirka / 2, vyska / 2, "A", " ",
               null, new Color(0, 0, 0)));
      }
   }
}
```

Jakmile aplikace vygeneruje vrcholy grafu, pokračuje dalším krokem, jehož cílem je vygenerování hran mezi vrcholy.

8.2.1 Generování grafu prohledáváním do hloubky

Aplikace má posloupnost vrcholů seřazenou podle výše jejich stupně. Postup metody generování grafu do hloubky je ukázán na následujícím příkladu:

Vrcholy:	(A, B, C, D, E, F)		
Stupně:	(2, 2, 2, 2, 2, 2)		

Metoda generování hran do hloubky spočívá v postupném procházení vrcholů od největšího stupně po nejmenší. Vezme se tedy bod A se stupněm 2 a spojí se s dalším bodem v pořadí (B), což způsobí snížení jejich stupně. Vrcholy se přeuspořádají podle stupně a pokračuje se dál, ale tentokrát je počátečním bodem vrchol B. Postupuje se čím dál více do hloubky a cílem je dostat se co nejdříve zpátky do bodu A. Ve výsledku se tento způsob snaží vytvořit grafu co nejvíce komponent.

Vrcholy:	(A, B, C, D, E, F)	->	(C, D, E, F, A, B)
Stupně:	(1, 1, 2, 2, 2, 2)	->	(2, 2, 2, 2, 1, 1)

Vrchol B nelze spojit s vrcholem A, protože tato hrana již existuje, proto se nyní vrchol B spojí hranou s následujícím vrcholem C.

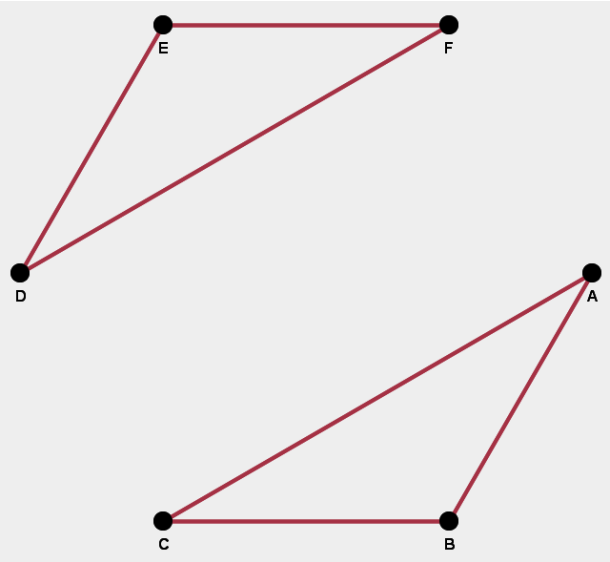
Vrcholy:	(C, D, E, F, A, B)	->	(D, E, F, A, C, B)
Stupně:	(1, 2, 2, 2, 1, 0)	->	(2, 2, 2, 1, 1, 0)

Nyní je již možné jít zpátky z vrcholu C do A.

Vrcholy:	(D, E, F, A, C, B)	->	(D, E, F, A, C, B)
Stupně:	(2, 2, 2, 1, 1, 0)	->	(2, 2, 2, 0, 0, 0)

V tuto chvíli již dál z bodů A, B a C nelze pokračovat. Vezme se tedy bod D a pokračuje se stejným způsobem.

Jak je vidět z posledního skóre, tato metoda generování nám vytvoří graf složený ze dvou komponent, viz Obrázek 19.



Obrázek 19 - Metoda „do hloubky“ skóre (2,2,2,2,2,2)

8.2.2 Generování grafu prohledáváním do šířky

Metoda prohledávání grafu do šířky je v mnoha ohledech podobná. Liší se především postupem vytváření hran. Na rozdíl od metody „do hloubky“ se tento způsob snaží spojit co nejvíce vrcholů do jedné komponenty. Metoda bude poukázána na stejném příkladu:

Vrcholy: (A, B, C, D, E, F)
Stupně: (2, 2, 2, 2, 2, 2)

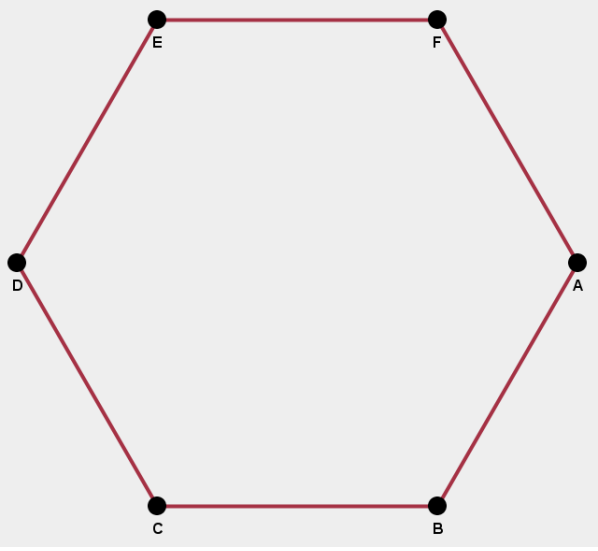
Postup je přesně daný. Vezme se první vrchol v pořadí (A) a spojí se s následujícím (B).

Vrcholy: (A, B, C, D, E, F)

Stupně: (1, 1, 2, 2, 2, 2)

Při použití této metody již není potřeba řadit vrcholy podle jejich stupně. Pokračuje se dál z vrcholu B, který se spojí hranou s vrcholem C. Stejným způsobem se vygenerují i další hrany (z C do D, z D do E, z E do F), až nakonec se kruh uzavře hranou z F do A. Pokud žádný následující vrchol nemá stupeň větší než 0, hledá se cílový vrchol mezi předcházejícími zpětným vynořováním. Pokud by nebylo možné spojit bod E s F, testuje se, zda lze vytvořit hranu mezi E a C (mezi E a D již hrana existuje) a při dalším neúspěchu mezi E a B, dokud nedojde k úspěšnému vytvoření hrany. Výsledek metody na Obrázku 20.

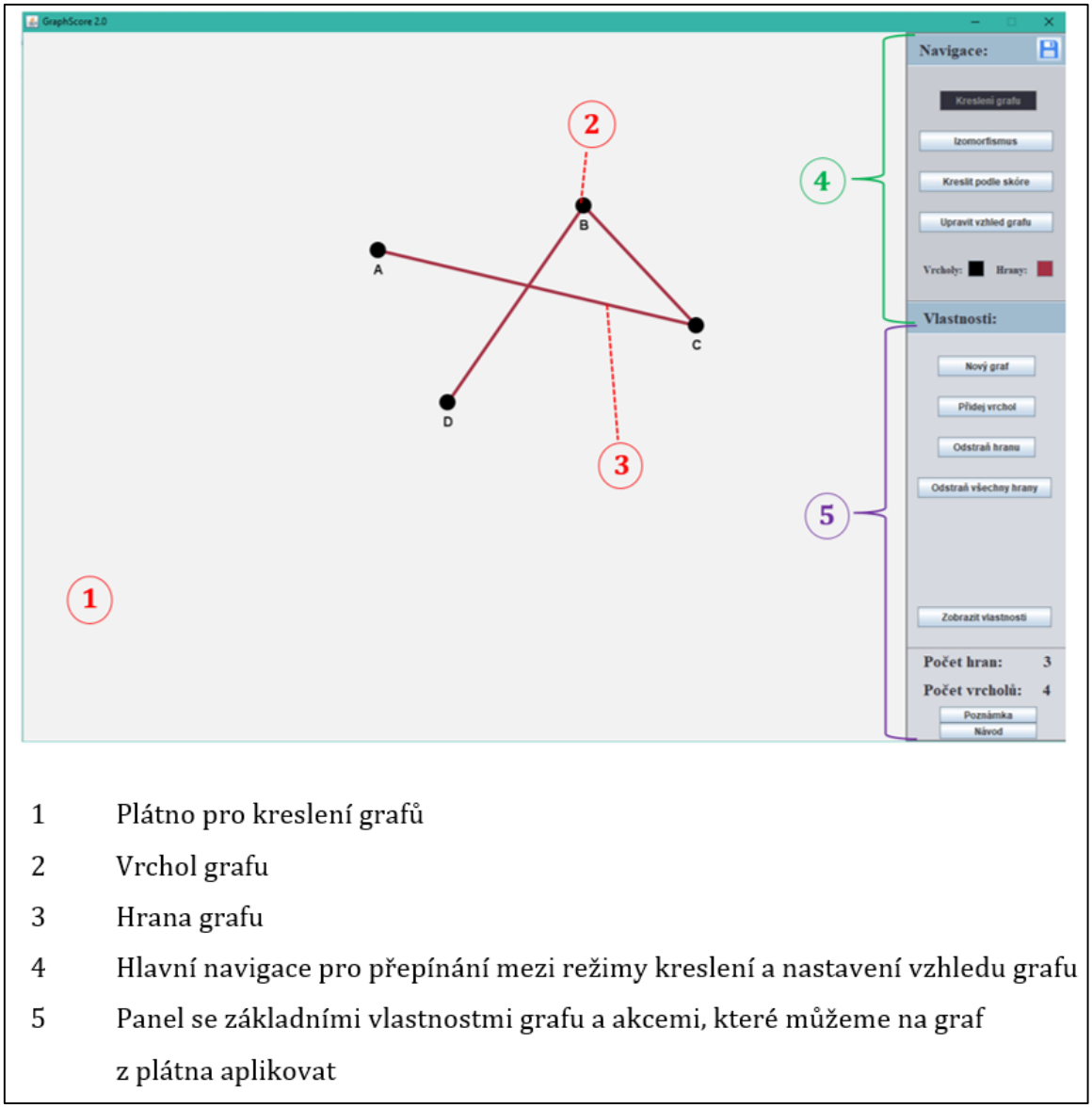
Vrcholy:	(A, B, C, D, E, F)	->	Vrcholy:	(A, B, C, D, E, F)
Stupně:	(1, 1, 2, 2, 2, 2)	->	Stupně:	(1, 0, 1, 2, 2, 2)
Vrcholy:	(A, B, C, D, E, F)	->	Vrcholy:	(A, B, C, D, E, F)
Stupně:	(1, 0, 0, 1, 2, 2)	->	Stupně:	(1, 0, 0, 0, 1, 2)
Vrcholy:	(A, B, C, D, E, F)	->	Vrcholy:	(A, B, C, D, E, F)
Stupně:	(1, 0, 0, 0, 0, 1)	->	Stupně:	(0, 0, 0, 0, 0, 0)



Obrázek 20 – Metoda „do šířky“ skóre (2,2,2,2,2,2)

9 Popis aplikace

Celá aplikace se dělí na dvě hlavní části, na plátno a na boční panel vlastností (Obrázek 21). V průběhu popisu aplikace budou části plátna a panelu vlastností samostatně popsány, jak se mění dle zvoleného režimu.



1	Plátno pro kreslení grafů
2	Vrchol grafu
3	Hrana grafu
4	Hlavní navigace pro přepínání mezi režimy kreslení a nastavení vzhledu grafu
5	Panel se základními vlastnostmi grafu a akcemi, které můžeme na graf z plátna aplikovat

Obrázek 21 - Struktura aplikace GraphScore v2.0

V dalších částech budou popsány jednotlivé prvky aplikace a nabízené funkce v závislosti na zvoleném režimu kreslení.

9.1 Plátno aplikace

Plátno slouží k zakreslení grafu do aplikace a pro jeho vykreslení po provedení úprav nad tímto grafem. Kreslení na plátno není nijak omezeno maximálním počtem vrcholů nebo hran. Ovládání plátna jeho velikost se liší na základě zvoleného kreslícího režimu, ale základní ovládání objektů na plátně mají režimy společné.

Vytvoření hrany probíhá pomocí prostředního tlačítka myši, táhnutím z jednoho vrcholu na druhý. Hrana se vytvoří pouze mezi dvěma navzájem odlišnými vrcholy, které ještě nejsou spojeny hranou. Nemůže tak dojít k duplicitním výskytům, které by mohly způsobit chyby v aplikaci. Ve výsledku bude nová hrana přidána do listu hran a plátno se aktualizuje.

Stejným způsobem lze prostředním tlačítkem myši přesouvat vrcholy grafu libovolně po plátně a umístit je na preferované místo.

9.2 Navigace

Hlavní navigace je umístěna v pravém horním rohu aplikace. Slouží k přepínání mezi jednotlivými režimy kreslení grafu, pro jednoduchou orientaci. Aplikace nabízí tři kreslící režimy „Kreslení grafu“, „Izomorfismus“ a „Kreslit podle skóre“. Každý režim nabízí jinou funkcionalitu odpovídající problematice, kterou řeší.

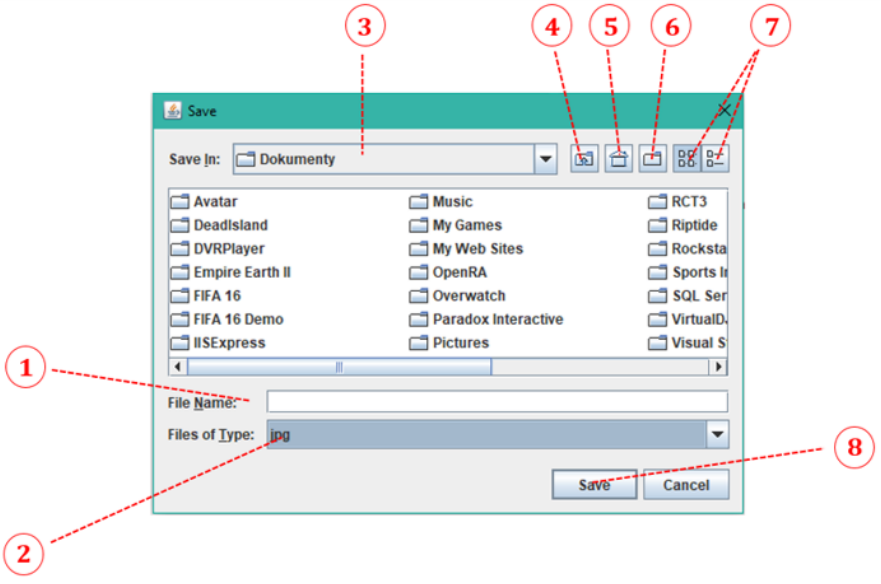
Mimo jiné jsou zde umístěna tlačítka pro vzhledové úpravy. V dolní části navigace jsou umístěna tlačítka „Vrcholy“ a „Hrany“, díky kterým lze nastavit barvy pro veškeré budoucí vrcholy a hrany, které budou zakresleny do plátna.

V neposlední řadě se v pravém horním rohu nachází tlačítko pro uložení kreslícího plátna ve formě obrázku.

9.3 Uložení grafu

Po kliknutí na tlačítko uložení, umístěné v pravém horním rohu aplikace, se otevře souborový prohlížeč pro uložení grafu. Na uživatele vyskočí okno

z Obrázku 22. Po úspěšném zvolení cesty a kliknutí na „Save“, bude obrázek uložen do cíle ve vybraném formátu.



1	Jméno obrázku
2	Formát obrázku
3	Hierarchie složek
4	Vylézt o složku výš
5	Domů
6	Nová složka
7	Režimy zobrazení
8	Tlačítko pro uložení obrázku

Obrázek 22 - Souborový prohlížeč

9.4 Režim „Kreslení grafu“

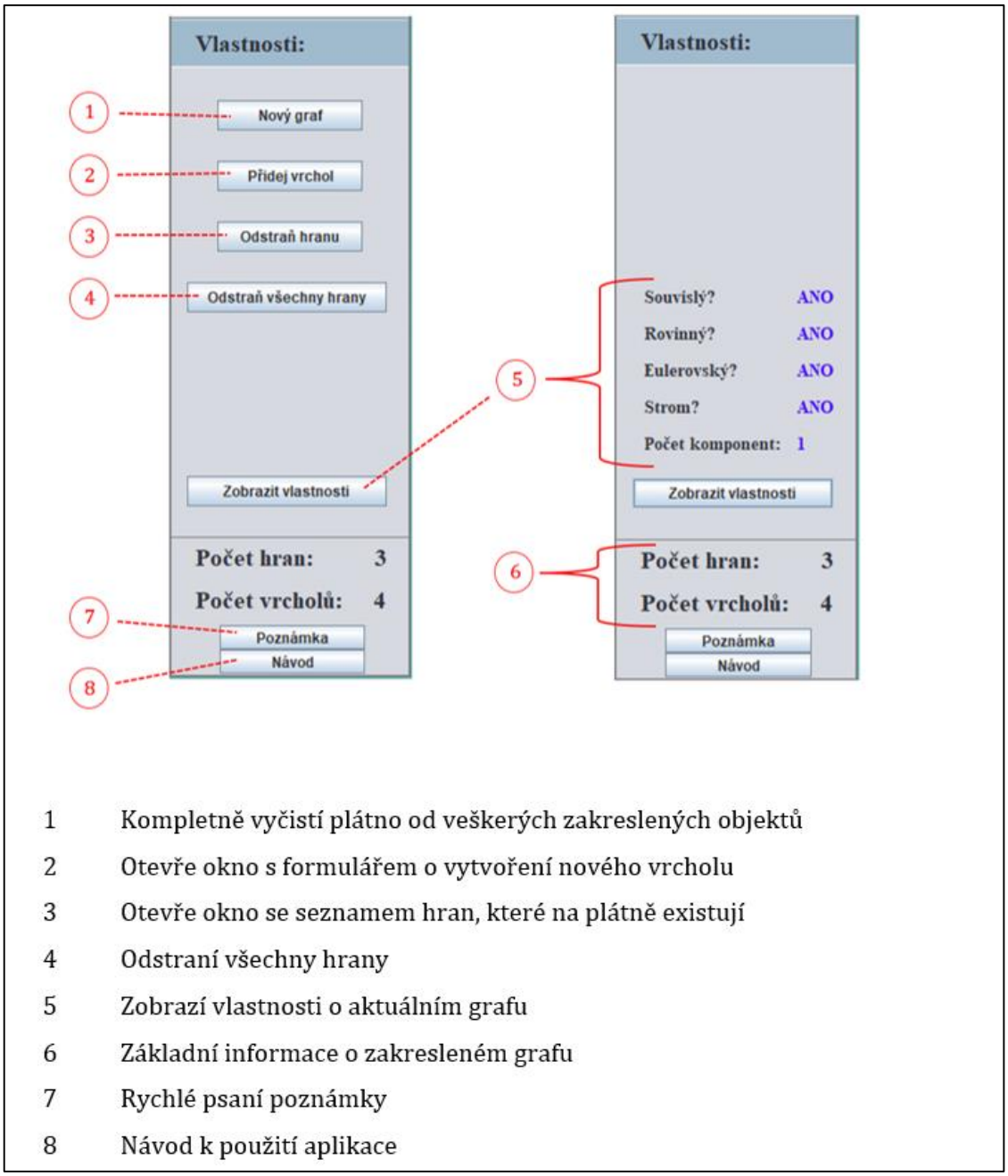
Ve výchozím stavu se uživatel nachází v režimu „Kreslení grafu“ nebo se do něj může přemístit pomocí příslušného tlačítka v navigaci, Obrázek 17. Výsledkem tohoto základního režimu je graf, který si uživatel sám podle sebe zakreslí do plátna a získá jeho obecné informace o jeho vlastnostech, vrcholech a hranách.

9.4.1 Plátno režimu „Kreslení grafu“

Plátno je uzpůsobeno uživateli pro přehledné ovládání a pro jednoduché kreslení grafu. Plocha pro samotné kreslení je dostatečně velká, tak aby bylo stále možné zakreslit složitější grafy přehledně a smysluplně. Tento režim podporuje změnu vzhledu zakresleného grafu. Uživatel tak může měnit barvy a tloušťky vrcholům a hranám grafu. Veškeré akce a informace v panelu „Vlastnosti“ se týkají grafu právě z tohoto plátna. Plátno naleznete na Obrázku 21, bod 1.

9.4.2 Vlastnosti režimu „Kreslení grafu“

Panel „Vlastnosti“ obsahuje veškeré potřebné nástroje, co se týče kreslení grafu. Naleznete zde také základní informace o zakresleném grafu a jeho vlastnosti, viz Obrázek 23.

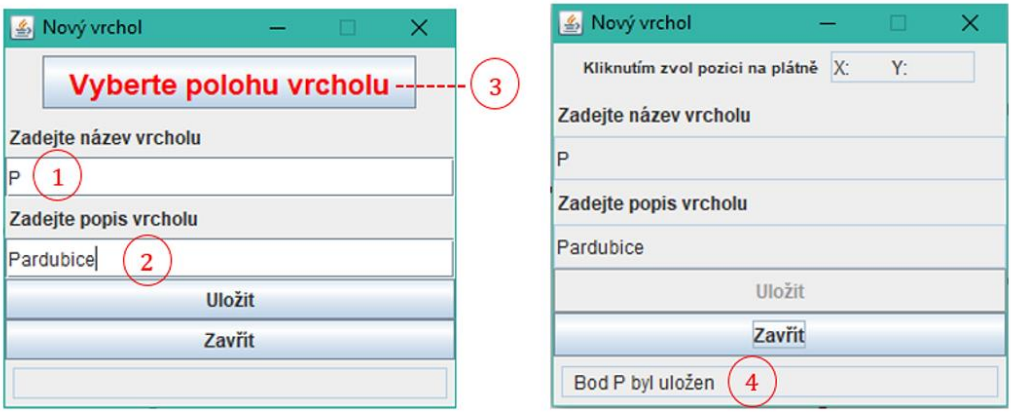


1	Kompletně vyčistí plátno od veškerých zakreslených objektů
2	Otevře okno s formulářem o vytvoření nového vrcholu
3	Otevře okno se seznamem hran, které na plátně existují
4	Odstraní všechny hrany
5	Zobrazí vlastnosti o aktuálním grafu
6	Základní informace o zakresleném grafu
7	Rychlé psaní poznámky
8	Návod k použití aplikace

Obrázek 23 – „Kreslení grafu“ – Vlastnosti

- Tlačítko „Nový graf“ vymaže zakreslený graf z plátna a vyčistí jeho skóre uložené v aplikaci, aby mohl uživatel začít znovu.
- Tlačítko „Přidej vrchol“ otevře formulář pro vytvoření nového vrcholu v grafu, viz Obrázek 24. Uživatel musí vyplnit název vrcholu a vybrat polohu vrcholu z plátna, jinak nebude vrchol vytvořen. Po kliknutí na „Vyberte polohu vrcholu“ je uživatel přepnut do aplikace, kde musí

vybrat pozici na plátně pro nový vrchol kliknutím levého tlačítka myši. Popis vrcholu je volitelný, lze tak určit vrcholu vlastnosti. Při úspěšném přidání vrcholu se vypíše hlášení v dolní části formuláře, viz obrázek 24 bod 4, a nový vrchol bude přidán do listu vrcholů.



1 Krátké označení vrcholu
2 Bližší popis vrcholu
3 Vybrání polohy bodu z plátna
4 Úspěšné přidání vrcholu

Obrázek 24 – Vytvoření nového vrcholu

- Tlačítkem „Odstraň hranu“ se zobrazí nové okno se seznamem všech hran v grafu, z kterého může uživatel vybrat hranu, kterou chce odstranit (Obrázek 33). Po každém smazání hrany se seznam aktualizuje, lze ho tedy využívat k opětovnému mazání.
- Volba „Odstraň všechny hrany“, jak napovídá název, odstraní všechny hrany ze zakresleného grafu.
- Po najetí myší nad tlačítko „Zobrazit vlastnosti“ se schovají některé ovládací prvky a zobrazí se uživateli seznam vlastností grafu z plátna, jako na Obrázku 23, bod 5. Jakmile uživatel myší sjede z tlačítka, ovládací prvky se opět vrátí na svá místa.

- V dolní části se nacházejí tlačítka pro rychlé psaní poznámky a návod. „Poznámka“ otevře nové okno pro zapisování libovolného textu. Uživatel těchto poznámek může otevřít libovolný počet, není tedy problém si dělat poznámky ke každému grafu, s kterým pracuje. Tlačítko „Návod“ otevře uživatelský návod aplikace GraphScore verze 2.0, ve kterém lze nalézt postupy k veškerým nabízeným akcím.

9.5 Režim „Izomorfismus“

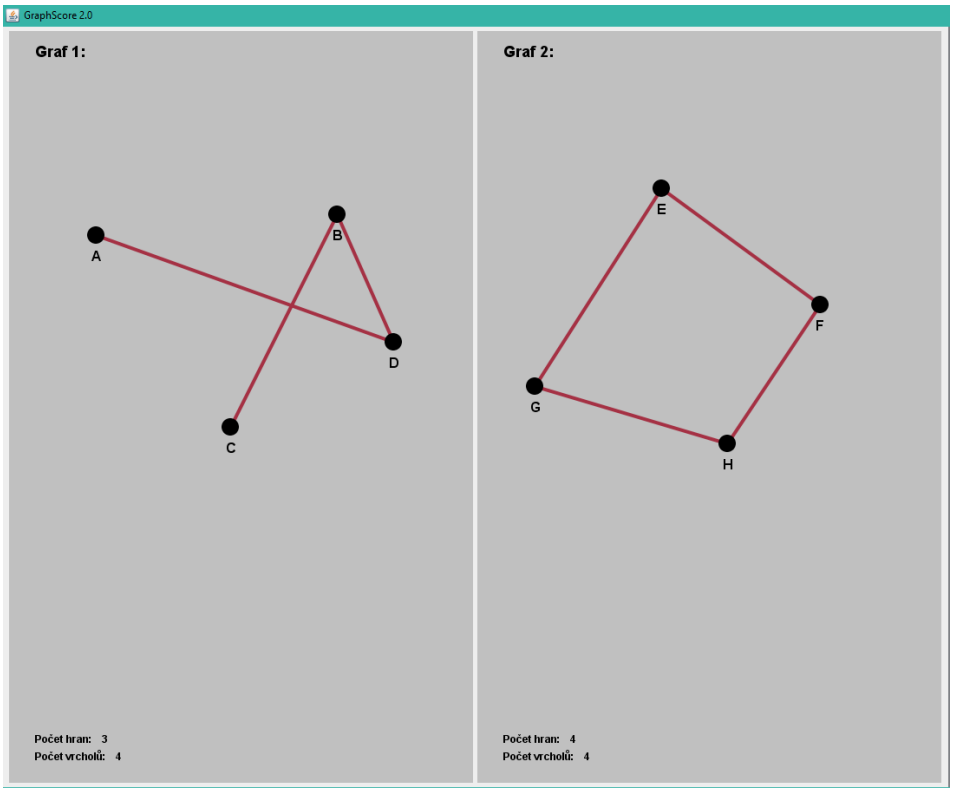
Tento režim sice také nabízí kreslení grafů, ale už nevyužívá možností vzhledových úprav. Není tedy tolik zaměřen na přehlednost a intuitivnost grafů, ale spíše je zaměřen na rychlé zakreslení grafů a následné získání informace, zda jsou tyto grafy vzájemně izomorfní. Rozhodnutí je založeno na algoritmu izomorfismu, kterým se tato práce zabývala v Kapitole 8. Kreslení grafů do plátna probíhá odděleně v závislosti na přepínači umístěném v horní části panelu „Vlastnosti“. Uživatel se do toho režimu přesune kliknutím na tlačítko „Izomorfismus“ umístěné v navigaci, Obrázek 17.

Několik funkcí, které režim nabízí, má stejný postup a formu s režimem „Kreslení grafu“. Návod jak použít tyto akce byly probírány v Kapitole 9.4.2. Jedná se o akce přidání nového vrcholu a odstranění hran.

9.5.1 Plátna režimu „Izomorfismus“

V režimu „Izomorfismus“ lze nalézt v hlavním okně hned dvě plátna pro kreslení grafů (Obrázek 25). Je tomu tak z jednoduchého důvodu. K porovnání, zda jsou dva grafy izomorfní, je zapotřebí každý graf zakreslit do odlišných pláten. Z důvodu fixní velikosti hlavního okna mají obě plátna menší rozměry, než tomu tak bylo u režimu „Kreslení grafu“. U izomorfismu v této aplikaci nezáleží na přehlednosti grafů, spíše na funkčnosti samotného algoritmu. Obě plátna mají v levém horním rohu označení a v levém dolním rohu základní informace o zakresleném grafu. Mezi plátny nelze přenášet objekty. To znamená, že uživatel

nemůže levým tlačítkem přemístit vrchol z grafu 1 do grafu 2. Kreslení grafů tedy probíhá odděleně.



Obrázek 25 – Plátna v režimu „Izomorfismus“

9.5.2 Vlastnosti režimu „Izomorfismus“

Panel „Vlastnosti“ obsahuje veškeré potřebné nástroje, co se týče kreslení grafů a jejich izomorfismu.

1	Přepínač určující, pro který graf budou prováděny následující úpravy
2	Kompletně vyčistí plátno od veškerých zakreslených objektů
3	Otevře okno s formulářem o vytvoření nového vrcholu
4	Otevře okno se seznamem hran, které na plátně existují
5	Odstraní všechny hrany
6	Vyhodnocení zda jsou grafy vzájemně izomorfní
7	Rychlé psaní poznámky
8	Návod k použití aplikace

Obrázek 26 - „Izomorfismus“ – Vlastnosti

Novinkou v tomto režimu je přepínač (Obrázek 26, bod 1). Aniž by se to zdálo, jedná se o nejdůležitější ovládací prvek. Slouží totiž k přepínání mezi plátny, na které budou akce aplikovány. Pokud tedy uživatel chce odstranit všechny hrany v grafu 2, musí nejprve přepnout přepínač do druhé polohy, a až poté kliknout na tlačítko „Odstraň všechny hrany“. Pokud by se tak nestalo, mohlo by dojít k situaci, kdy uživatel nechtěně odstraní všechny hrany z grafu 1. Tento způsob je nutný pro všechny ovládací prvky vyjma tlačítek: „Jsou izomorfní?“, „Poznámka“ a „Návod“.

Ovládání samotného plátna zůstává stejné, uživatel tak nemusí přepínat kvůli přemístění vrcholu na jinou polohu.

- Tlačítko „Jsou izomorfní?“ spouští algoritmus z Kapitoly 8.1 a zjišťuje, zda dva zakreslené grafy jsou izomorfní. Po vyhodnocení se uživateli zobrazí nové informační okno se samotným výsledkem. Pokud jsou obě plátna prázdná, algoritmus se nespustí a zobrazí se informační okno.

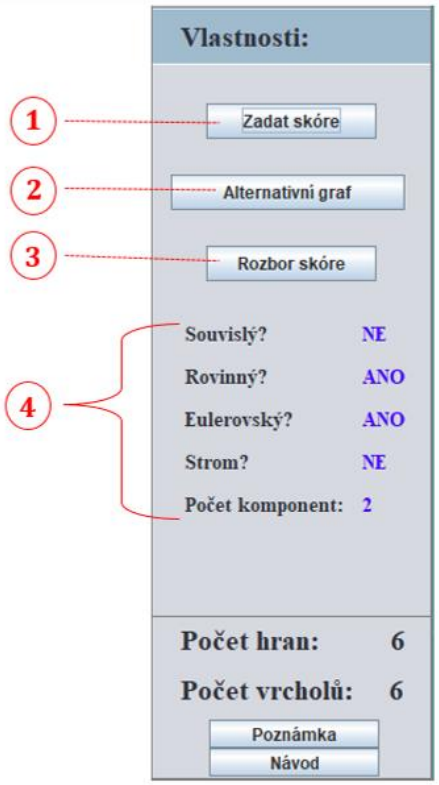
9.6 Režim „Kreslit podle skóre“

Tento režim spočívá v takovém postupu, kde uživatel zadá skóre grafu a aplikace mu sama vygeneruje na plátno graf odpovídající zadanému skóre. Lze ho také využít k určení, zda zadané skóre je opravdu skórem grafu. Do režimu je možné se přepnout pomocí tlačítka v navigaci, Obrázek 17.

9.6.1 Plátno režimu „Kreslit podle skóre“

Plátno v režimu „Kreslit podle skóre“ má stejné vlastnosti jako plátno v režimu „Kreslení grafu“, viz Obrázek 21, bod 1. Rozdílem je, že uživatel do tohoto plátna nekreslí graf, vygeneruje ho za něj sama aplikace ze zadaného skóre. Uživatel samozřejmě může pohybovat samotnými objekty, ale nemůže vytvářet nové.

9.6.2 Vlastnosti režimu „Kreslit podle skóre“



1 Zadání skóre pro vygenerování grafu

2 Vykreslení jiného grafu se stejným skóre

3 Zobrazení rozboru skóre, jak se potupovalo při generování grafu

4 Přehled vlastností o vygenerovaném grafu

Obrázek 27 - „Kreslení podle skóre“ – Vlastnosti

Ovládací prvky tohoto režimu jsou na první pohled odlišné. Uživatel už nemá zapotřebí kreslit do plátna graf, z tohoto důvodu jsou kreslící akce nahrazeny akcemi pro generování a zjištění vlastností na základě zadaného skóre uživatelem.

Zadat skóre může uživatel pomocí tlačítka „Zadat skóre“, které nabídne formulář pro zadávání. Skóre musí mít stejný formát jako je tomu v nápovědě, tj. $(A, B, C, \dots)$. Ostatní formáty nejsou podporovány, pokud tedy bude skóre obsahovat znaky, které nejsou žádoucí, zobrazí se chybové hlášení. Za chybu se považují i bílé znaky (mezery, odstavce). Při správném zadání skóre se na plátno vygeneruje odpovídající graf metodou „do hloubky“ (Kapitola 8.2.1), aktualizují se

vlastnosti o grafu v bočním panelu a zobrazí se nová tlačítka „Alternativní graf“ a „Rozbor skóre“.

- „Alternativní graf“ nabízí vygenerování grafu ze zadaného skóre druhým způsobem, „do šířky“ (Kapitola 8.2.2). Následně se opět aktualizují vlastnosti grafu v bočním panelu.
- Tlačítko „Rozbor skóre“ nabízí uživateli možnost zjistit, jakým postupem se skóre grafu rozebírá pro lepší pochopení problematiky. Po kliknutí se otevře nové okno obsahující rozbor skóre.

10 Shrnutí výsledků

Autor si při psaní bakalářské práce osvojil znalosti z oblasti matematické disciplíny teorie grafů a programování v jazyce JAVA. Největší časovou náročnost si vyžádala praktická část, kdy se bylo potřeba vypořádat s několika implementačními problémy.

Byla vytvořená aplikace, která je intuitivní a nabízí snadné kreslení grafů a získání jejich vlastností. Program tak splňuje veškeré cíle, které byly na začátku práce určeny.

Samotná aplikace GraphScore je odrazem programátorských znalostí autora, který se sám za programátora nepovažuje. Zdatnější programátoři, kteří se v jazyce JAVA pohybují, by jistě přišli na jiná, mnohdy i lepší řešení tohoto úkolu.

Veškeré funkcionality programu byly testovány autorem. V režimu „Izomorfismus“ bylo otestováno celkem 70 dvojic grafů, které skončily správným výsledkem. Každý program se vyladí až průběžným používáním.

Aplikaci lze nalézt na přiloženém DVD, jejíž výhodou je malá náročnost na vybavení počítače. Stačí mít na počítači nainstalovaný software JAVA minimální verze 8.

11 Závěry a doporučení

Hlavním cílem práce bylo vytvořit aplikaci pro určení, zda posloupnost je skóre grafu, pro kreslení grafů s možnými funkcemi zjistit, zda jsou grafy izomorfní a pro vygenerování grafu na základě zadaného skóre. Aplikace bude sloužit uživatelům, kteří si budou moci osvojit znalosti z oblasti teorie grafů a především skóre grafu. Často studentům dělá problémy představit si graf ze zadaného skóre. Příčin může být více, někdo se lépe učí z papírů, někdo z obrazovky a někdo si musí sám problematiku prakticky vyzkoušet, aby porozuměl. Právě pro tyto účely je aplikace určena. Kombinuje snadné a přehledné kreslení grafů na plátno a rychlé získání vlastností o zakresleném grafu. Aplikace také nabízí rozbor skóre, který je stěžejním k pochopení, jakým postupem se provádí rozbor skóre.

Nedílnou součástí bakalářské práce je dokument, který má posloužit uživatelům jako souhrn základních teoretických pojmů a také jako popis a návod k použití aplikace GraphScore verze 2.0.

Práce popisuje jak použité algoritmy, tak i rozvržení tlačítek a knoflíků v aplikaci pro přehledné ovládání. V návodu uživatel najde postup provedení veškerých funkcí, které nejsou na první pohled zřetelné nebo vyžadují složitější zadání.

Cíl, který byl stanoven pro bakalářskou práci, byl splněn. Teorie grafů, konkrétně skóre grafu, jako téma bakalářské práce si autor vybral z důvodu předchozích zkušeností. Autor v něm našel zalíbení již při studiu předmětu Diskrétní matematika na Univerzitě Hradec Králové, oboru aplikovaná informatika.

12 Seznam použité literatury

[1] MATOUŠEK, Jiří; NEŠETŘIL, Jaroslav. *Kapitoly z diskrétní matematiky*. Vyd. 2., opr., (V nakl. Karolinum 1.). Praha: Karolinum, 2000. ISBN 80-246-0084-6.

[2] L. Kučera, Kombinatorické Algoritmy, SNTL, Praha, 1983.

[3] Jiří Demel: Grafy a jejich aplikace, nakladatelství Academia, Praha 2002, ISBN 80-200-0990-6.

[4] GROSS, Jonathan L. a Jay YELLEN, ed. *Handbook of graph theory*. Boca Raton, Fla.: CRC Press, c2004. Discrete mathematics and its applications. ISBN 1-58488-090-2.

[5] R. Diestel, *Graph Theory* (third edition), Springer, 2005. Dostupné z: http://www.math.uni-hamburg.de/home/diestel/books/graph.theory/.

[6] *Teorie grafů* [online]. Praha: Univerzita Karlova, 2008 [cit. 2017-08-22]. Dostupné z: http://teorie-grafu.cz/.

[7] Množiny. *Matematika.cz* [online]. Brno: Nová média, 2005 [cit. 2018-04-21]. Dostupné z: https://matematika.cz/mnoziny

[8] Leonhard Euler. In: *Wikipedia: the free encyclopedia* [online]. San Francisco (CA): Wikimedia Foundation, 2001 [cit. 2018-04-21]. Dostupné z: https://cs.wikipedia.org/wiki/Leonhard_Euler

[9] Seven Bridges of Königsberg. In: *Wikipedia: the free encyclopedia* [online]. San Francisco (CA): Wikimedia Foundation, 2001 [cit. 2018-04-21]. Dostupné z: https://en.wikipedia.org/wiki/Seven_Bridges_of_K%C3%B6nigsberg

[10] MAŠTEROVÁ, Ludmila. *Teorie grafů* [online]. Kotlářská 2, 611 37 Brno, 2016 [cit. 2018-04-21]. Dostupné z: https://is.muni.cz/th/409120/prif_b/BP_L._Masterova.pdf. Bakalářská práce. Přírodovědecká fakulta, Masarykova univerzita Ústav matematiky a statistiky. Vedoucí práce RNDr. Pavel Šišma, Dr.

[11] SEDLÁČEK, Jiří. *Úvod do teorie grafů*. 3. vyd. Praha: Academia, 1981. Cesta k vědění (Academia).

[12] JAVA. In: *Wikipedia: the free encyclopedia* [online]. San Francisco (CA): Wikimedia Foundation, 2001 [cit. 2018-04-26]. Dostupné z: https://cs.wikipedia.org/wiki/Java_(programovac%C3%AD_jazyk)

[13] BufferedImage. *Oracle* [online]. Redwood City, CA 94065, U.S.A: Oracle America, 2015 [cit. 2018-04-26]. Dostupné z: https://docs.oracle.com/javase/7/docs/api/java/awt/image/BufferedImage.html

[14] Graf. In: *Wikipedia: the free encyclopedia* [online]. San Francisco (CA): Wikimedia Foundation, 2001 [cit. 2018-04-26]. Dostupné z: https://cs.wikipedia.org/wiki/Graf_(teorie_graf%C5%AF)#D%C5%AFle%C5%BEit%C3%A9_typy_graf%C5%AF

[15] GUI. In: *Wikipedia: the free encyclopedia* [online]. San Francisco (CA): Wikimedia Foundation, 2001 [cit. 2018-04-26]. Dostupné z: https://cs.wikipedia.org/wiki/Grafick%C3%A9_u%C5%BEivatelsk%C3%A9_rozhran%C3%AD

[16] ComponentListener. *Oracle* [online]. Redwood City, CA 94065, U.S.A: Oracle America, 2015 [cit. 2018-04-26]. Dostupné z: https://docs.oracle.com/javase/7/docs/api/java/awt/event/ComponentListener.html

[17] WindowListener. *Oracle* [online]. Redwood City, CA 94065, U.S.A: Oracle America, 2015 [cit. 2018-04-26]. Dostupné z: https://docs.oracle.com/javase/7/docs/api/java/awt/event/WindowListener.html

[18] MouseListener. *Oracle* [online]. Redwood City, CA 94065, U.S.A: Oracle America, 2015 [cit. 2018-04-26]. Dostupné z: https://docs.oracle.com/javase/7/docs/api/java/awt/event/MouseListener.html

[19] MouseMotionListener. *Oracle* [online]. Redwood City, CA 94065, U.S.A: Oracle America, 2015 [cit. 2018-04-26]. Dostupné z: https://docs.oracle.com/javase/7/docs/api/java/awt/event/MouseMotionListener.html

[20] JFrame. *Oracle* [online]. Redwood City, CA 94065, U.S.A: Oracle America, 2015 [cit. 2018-04-26]. Dostupné z: https://docs.oracle.com/javase/7/docs/api/javax/swing/JFrame.html

[21] JButton. *Oracle* [online]. Redwood City, CA 94065, U.S.A: Oracle America, 2015 [cit. 2018-04-26]. Dostupné z: https://docs.oracle.com/javase/7/docs/api/javax/swing/JButton.html

[22] JFileChooser. *Oracle* [online]. Redwood City, CA 94065, U.S.A: Oracle America, 2015 [cit. 2018-04-26]. Dostupné z: https://docs.oracle.com/javase/7/docs/api/javax/swing/JFileChooser.html

[23] JTextField. *Oracle* [online]. Redwood City, CA 94065, U.S.A: Oracle America, 2015 [cit. 2018-04-26]. Dostupné z: https://docs.oracle.com/javase/7/docs/api/javax/swing/JTextField.html

[24] JTextArea. *Oracle* [online]. Redwood City, CA 94065, U.S.A: Oracle America, 2015 [cit. 2018-04-26]. Dostupné z: https://docs.oracle.com/javase/7/docs/api/javax/swing/JTextArea.html

[25] JOptionPane. *Oracle* [online]. Redwood City, CA 94065, U.S.A: Oracle America, 2015 [cit. 2018-04-26]. Dostupné z: https://docs.oracle.com/javase/7/docs/api/javax/swing/JOptionPane.html

[26] HAVEL, Václav. Poznámka o existenci konečných grafů. *Časopis pro pěstování matematiky*. 1955, roč. 080, čís. 4, s. 477-480. [cit. 2018-04-26]. ISSN 0528-2195. Dostupné z: https://dml.cz/bitstream/handle/10338.dmlcz/108220/CasPestMat_080-1955-4_8.pdf

[27] HAKIMI, S. L. On realizability of a set of integers as degrees of the vertices of a linear graph. *Journal of the Society for Industrial and Applied Mathematics* [online]. 1962 [cit. 2017-04-27]. Čís. 10, s. 496-506. Dostupné z: http://compalg.inf.elte.hu/~tony/Oktatas/Algoritmusok-hatekonysaga/Hakimi-SIAM62.pdf

[28] Václav Havel. In: *Wikipedia: the free encyclopedia* [online]. San Francisco (CA): Wikimedia Foundation, 2001- [cit. 2018-04-27]. Dostupné z: https://cs.wikipedia.org/wiki/V%C3%A1clav_Havel_(matematik)

[29] Havel-Hakimi. *Jacopo Notarstefano* [online]. Ženeva, Švýcarsko: Jacopo Notarstefano, 2014 [cit. 2018-04-27]. Dostupné z: http://jacquerie.github.io/hh/

[30] Kleitman D.J. and Wang D.L. *Algorithms for Constructing Graphs and Digraphs with Given Valences and Factors Discrete Mathematics*, 6(1), pp. 79-88 (1973)

[31] P. Erd ös , T. Gallai, *Graphs with vertices having prescribed degrees* (Hungarian), Mat. Lapok 11, (1960) 264–274.

[32] F. Ruskey , R. Cohen, P. Eades, A. Scott, *Alley CAT's in search of good homes*, Congressus Numerantium, 102 (1994) 97–110.

[33] T. M. Barnes, C. D. Savage, *Efficient generation of graphical partitions*, Discrete Applied Mathematics 78, (1–3) (1997) 17–26.

[34] A. Kohnert, *Dominance order and graphical partitions*, Elec. J. Comb. 11,(1) (2004) No. 4. 17 pp.

[35] A. Tripathi , S. Venugopalanb, D. B. West, *A short constructive proof of the Erd˝os-Gallai characterization of graphic lists*, Discrete Mathematics 310, (4) (2010) 833–834.

[36] A. Iványi, L. Lucz, T. F. Móri, P. Sótér, *On Erdös-Gallai and Havel-Hakimi algorithms*, Acta Univ. Sapientiae, Inform. 3 (2011) 230–268.

[37] Permutace. In: *Wikipedia: the free encyclopedia* [online]. San Francisco (CA): Wikimedia Foundation, 2001- [cit. 2018-05-02]. Dostupné z: https://cs.wikipedia.org/wiki/Permutace

[38] Binární vyhledávací strom. In: *Wikipedia: the free encyclopedia* [online]. San Francisco (CA): Wikimedia Foundation, 2001- [cit. 2018-05-02]. Dostupné z: https://cs.wikipedia.org/wiki/Bin%C3%A1rn%C3%AD_vyhled%C3%A1vac%C3%AD_strom

[39] Ševčíková, A., Milková, *E. Multimedia Applications: Graph Algorithms visualization*, PROCEEDINGS, Budapešť: IEEE, 2016. 6s. ISBN: 978-1-5090-3909-8

[40] J. Šitina, *Grafové algoritmy a jejich využití*, Diplomová práce, Univerzita Hradec Králové, červen 2010

[41] A. Hübner, *Aplikace pro podporu výuky předmětu Disktrétní matematika*, Bakalářská práce, Univerzita Hradec Králové, Červenec 2014

[42] M. Růtová-Šťastná, *Vizuální podpora výuky předmětů zabývajících se teorií grafů a grafovými algoritmy*, Diplomová práce, Univerzita Hradec Králové, Srpen 2017

[43] Algorithmic Complexity. *School of Computer Science* [online]. 5000 Forbes Avenue Pittsburgh: Carnegie Mellon University, 2016 [cit. 2018-05-05]. Dostupné z: https://www.cs.cmu.edu/~adamchik/15-121/lectures/Algorithmic%20Complexity/complexity.html

[44] Screen Resolution Stats Worldwide. *StatCounter GlobalStats* [online]. Dublin 8, Ireland: StatCounter, 1999 [cit. 2018-05-05]. Dostupné z: http://gs.statcounter.com/screen-resolution-stats

Seznam obrázků

Následující obrázky pro bakalářskou práci vytvořil Radim Krátký.

Obrázek 1 - Příklad grafu v praxi 1
Obrázek 2 - Využití grafů v městské infrastruktuře 2
Obrázek 3 – Indukovaný podgraf G' grafu G 9
Obrázek 4 – Ukázka grafu s označeným stupněm jednotlivých vrcholů grafu 10
Obrázek 5 – Úplné grafy $K2, K3, K4$ 11
Obrázek 6 - Příklad skóre grafu 11
Obrázek 7 - Příklad dvou neizomorfních grafů se stejným skóre 12
Obrázek 8 – Hrany mezi vrcholy k důkazu věty 1.12 15
Obrázek 9 - Opačný postup rozboru skóre z příkladu k Větě 1.12 17
Obrázek 10 – On-line hra Havel-Hakimi[29] 17
Obrázek 11 - Příklad cesty v grafu (v1, v2, v3, v4, v5) 18
Obrázek 12 - Příklad kružnice velikosti 4 19
Obrázek 13 - Eulerovský tah 21
Obrázek 14 – H je rovinné nakreslení grafu G. 21
Obrázek 15 - Příklady stromů 22
Obrázek 16 – Graf příkladu k definici 1.24 – Prohledávání stromu 23
Obrázek 17 – Panel1 – Hlavní navigace 26
Obrázek 18 – Panel2 – Akce a vlastnosti grafu 28
Obrázek 19 - Metoda „do hloubky“ skóre (2,2,2,2,2,2) 35
Obrázek 20 – Metoda „do šířky“ skóre (2,2,2,2,2,2) 36
Obrázek 21 - Struktura aplikace GraphScore v2.0 37
Obrázek 22 - Souborový prohlížeč 39
Obrázek 23 – „Kreslení grafu“ – Vlastnosti 41
Obrázek 24 – Vytvoření nového vrcholu 42
Obrázek 25 – Plátna v režimu „Izomorfismus“ 44
Obrázek 26 - „Izomorfismus“ – Vlastnosti 45
Obrázek 27 - „Kreslení podle skóre“ – Vlastnosti 47

Seznam algoritmů

Algoritmus 1 - Porovnávání stupňů sousedních vrcholů 31
Algoritmus 2 - Algoritmus porovnávání počtu kružnic 32
Algoritmus 3 - Generování vrcholů 33

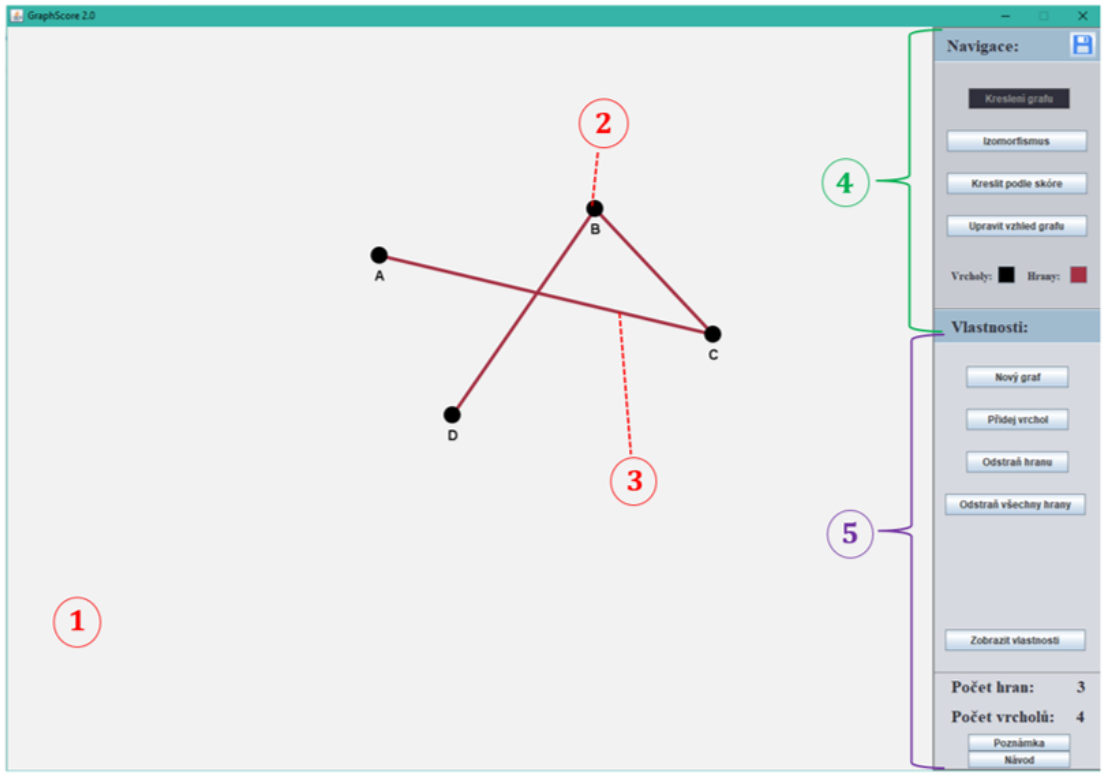
Návod aplikace GraphScore

V této části čtenář nalezne podrobný návod k použití aplikace GraphScore verze 2.0. V návodu jsou popsány veškeré funkce, které může uživatel využívat při obsluze programu.

Obsah návodu

1. Popis aplikace 2
2. Ovládání plátna 3
   2.1. Vytvoření hrany 3
   2.2. Přesouvání vrcholů 4
   2.3. Detail vrcholu 4
3. Režim „Kreslení grafu“ 5
   3.1. Nový graf 5
   3.2. Přidání nového vrcholu 5
   3.3. Odstranění vrcholu 6
   3.4. Odstranění hran 6
   3.5. Zobrazení vlastností grafu 7
   3.6. Změna vzhledu grafu 8
4. Režim „Izomorfismus“ 10
   4.1. Přepínání mezi plátny 10
   4.2. Nový graf 10
   4.3. Přidání nového vrcholu 10
   4.4. Odstranění vrcholu 11
   4.5. Odstranění hran 11
   4.6. Jsou grafy izomorfní? 11
5. Režim „Kreslit podle skóre“ 12
   5.1. Vykreslení grafu podle skóre 12
   5.2. Zjednodušení skóre 13
6. Uložení grafu 13
7. Psaní poznámky 14
8. Otevřít návod k programu 14

1. Popis aplikace



1	Plátno pro kreslení grafů
2	Vrchol grafu
3	Hrana grafu
4	Hlavní navigace pro přepínání mezi režimy kreslení a nastavení vzhledu grafu
5	Panel se základními vlastnostmi grafu a akcemi, které můžeme na graf z plátna aplikovat

Obrázek-návod 1 – Struktura aplikace GraphScore v2.0

2. Ovládání plátna

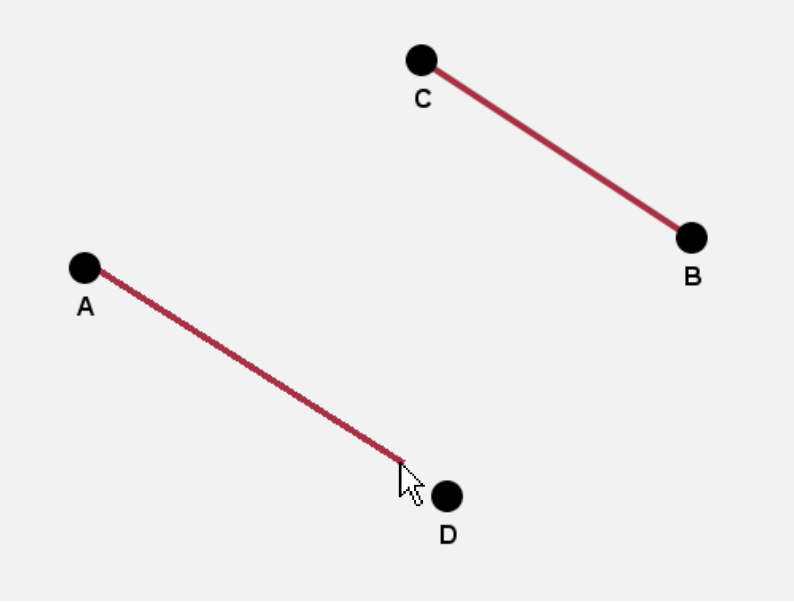
Pro práci s grafy zakreslenými na plátně se využívá pouze myši a samotných ovládacích prvků na bočním panelu aplikace (Obrázek-návod 1, body 4 a 5), které po výběru provedou zvolenou akci nebo případně otevřou podokno s upřesňujícími možnostmi. Nicméně tyto ovládací prvky se na bočním panelu mění v závislosti na zvoleném režimu kreslení. Následující ovládání plátna je stejné pro všechny režimy kreslení.

Pro rychlejší a snazší práci s programem jsou na plátnech vytvořeny čtyři výchozí vrcholy A, B, C a D.

2.1. Vytvoření hrany

Stisknutím a podržením **levého** tlačítka myši můžete spojit dva vrcholy a tím vytvořit novou hranu (přidání nového vrcholu viz bod 3.2.). Kliknutí na první vrchol a následné podržení a přetažení na druhý vrchol, vytvoří hranu mezi těmito vrcholy.

Hrana se vytvoří pouze mezi dvěma navzájem odlišnými vrcholy, které ještě nejsou spojeny hranou.



Obrázek-návod 2 - Vytvoření nové hrany

2.2. Přesouvání vrcholů

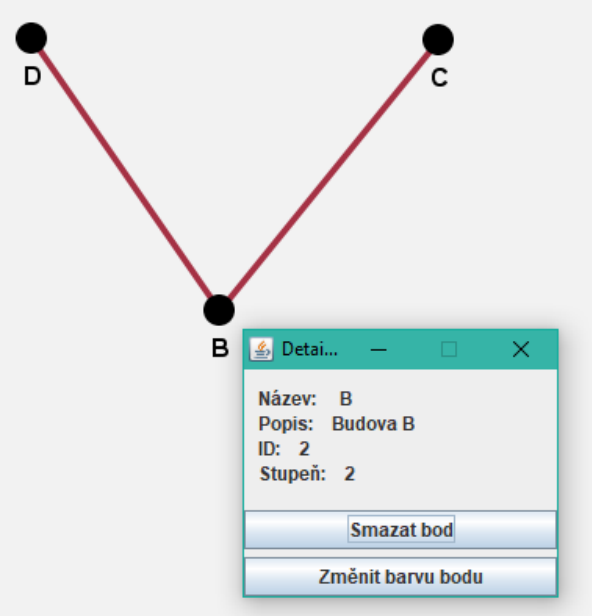
Stejným způsobem lze **prostředním** tlačítkem myši přesouvat vrcholy grafu libovolně po plátně a umístit je na místo, které vám vyhovuje. Opět musíte prvním kliknutím zamířit na daný vrchol a poté podržením ho přesouvat po ploše plátna.

Vrcholy nelze přesunout mimo rozměry kreslícího plátna.

2.3. Detail vrcholu

Kliknutím **pravého** tlačítka myši na některým z vrcholů se zobrazí nové detailní okno, které vám nabídne rychlé úpravy nad vrcholem jako změnit jeho barvu nebo možnost jeho smazání.

Dále se zde dozvíte užitečné informace o daném vrcholu. Jeho název, popis (pokud byl zadán), jeho ID nebo také stupeň vrcholu, udávající počet, s kolika dalšími vrcholy je vrchol spojen hranou.



Obrázek-návod 3 - Detail vrcholu

3. Režim „Kreslení grafu“

Jedná se o výchozí režim programu. Slouží ke kreslení libovolného grafu. Do režimu se přepnete pomocí tlačítka „Kreslení grafu“ v panelu navigace.

3.1. Nový graf

Tlačítko „Nový graf“ vymaže celé plátno. Odstraní jak zakreslené vrcholy, tak i všechny hrany. Tento krok je **nevratný**, proto ho používejte se zvýšenou opatrností.

3.2. Přidání nového vrcholu

Přidání vrcholu do plátna se provádí přes formulář vytvoření vrcholu, pomocí příslušného tlačítka umístěného v bočním panelu „Vlastnosti“ (Obrázek-návod 1, bod 5).

Ve formuláři musíte zadat povinný **název** vrcholu a vybrat povinnou **polohu** pomocí tlačítka „Vyberte polohu vrcholu“. Můžete mu také zadat popis, který je nepovinný, viz Obrázek-návod 5.

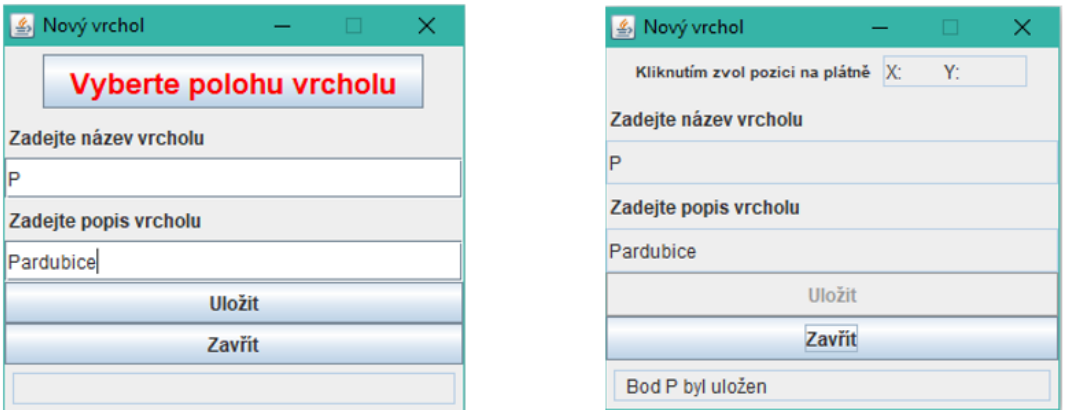
Pro příklad se v následujícím obrázku zadává nový vrchol s názvem „P“, který bude reprezentovat město Pardubice. Názvy vrcholů volte co nejkratší pro co největší přehlednost.

Dále je zapotřebí vybrat polohu vrcholu z plátna. Po kliknutí na tlačítko „Vyberte polohu vrcholu“ vás aplikace přesměruje na hlavní okno. Levým tlačítkem myši vyberte, kam chcete nový bod umístit. Aplikace vás následně přesměruje zpátky do formuláře vytvoření vrcholu.

Pokud byla vybrána poloha z plátna, objeví se v pravém horním rohu formuláře jeho pozice X a Y. V případě, že jste vyplnili všechny potřebné informace, potvrďte vytvoření vrcholu tlačítkem „Uložit“.

O úspěšném přidání vrcholu do plátna vás aplikace informuje v dolní části formuláře. Nový vrchol byl vytvořen, zavřete formulářové okno.

Počet celkových vrcholů na plátně není nijak omezen.



Obrázek-návod 4 - Vytvoření nového vrcholu

3.3. Odstranění vrcholu

Vrchol z grafu můžete odstranit dvěma způsoby. Pokud chcete vymazat pouze jeden vrchol, postačí kliknutí pravého tlačítka myši na daný vrchol a v detailu vrchol zvolit možnost „Smazat bod“.

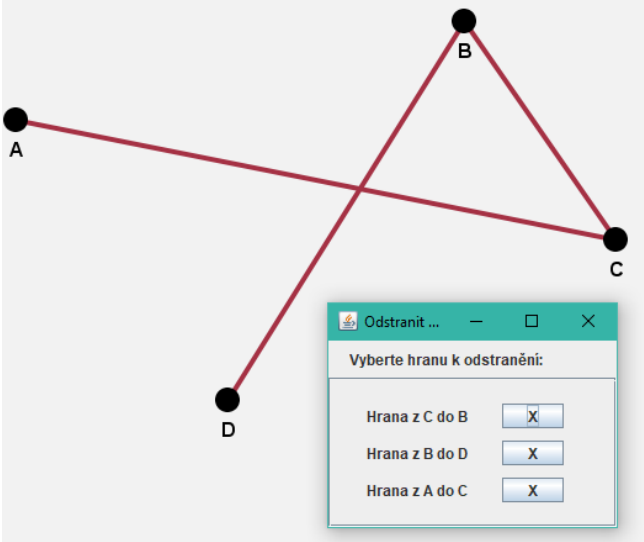
Druhým způsobem je možnost vymazat celé plátno, tedy smazat všechny **vrcholy i hrany** z plátna a začít kreslit od začátku. Tuto možnost provedete klinutím na tlačítko „Nový graf“.

3.4. Odstranění hran

Odstranit hrany můžete jednu po druhé nebo všechny najednou. Pro odstranění určité hrany zvolte tlačítko „Odstranit hranu“. Otevře se vám nové okno se seznamem veškerých hran, které v plátně existují, viz Obrázek-návod 6.

Hrany jsou popsány jako hrana z bodu A do B. Vpravo od popisu hrany se nachází tlačítko se znakem „**X**“. Po jeho kliknutí se odstraní vámi zvolená hrana a okno se seznamem hran se obnoví.

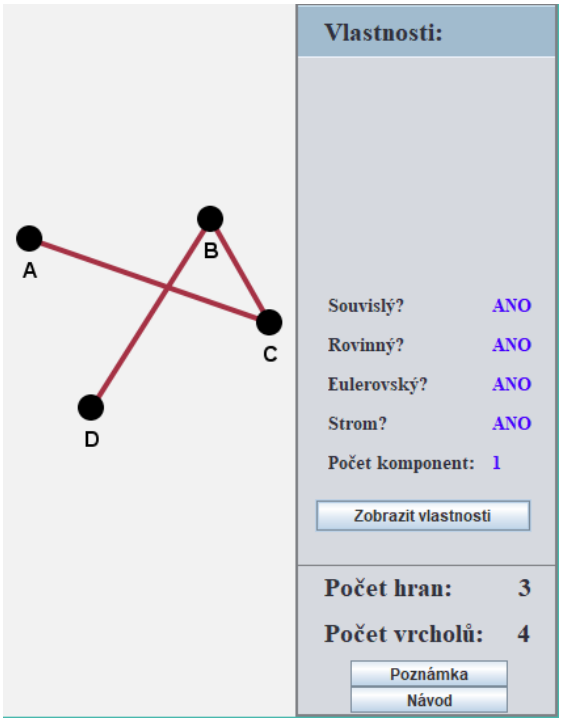
Smazat všechny hrany můžete pomocí tlačítka „Odstraň všechny hrany“ nebo již zmiňovaného tlačítka „Nový graf“.



Obrázek-návod 5 - Mazání hran grafu

3.5. Zobrazení vlastností grafu

Pro zobrazení vlastností grafu, zda je stromem, zda je graf souvislý, eulerovský, rovinný a kolik má komponent, **najeďte** na tlačítko „Zobrazit vlastnosti“. Jakmile myší tlačítko opustíte, vrátí se ovládací panely na boční panel.



Obrázek-návod 6 - Vlastnosti grafu

3.6. Změna vzhledu grafu

Program nabízí v režimu kreslení mnoho možností, jak si upravit zakreslený graf podle svého gusta, tak aby vám vzhledově vyhovoval. Díky těmto nástrojům můžete dosáhnout, aby i velmi složitý graf byl hezký, přehledný a intuitivní. V bočním panelu lze najít možnosti změnit nejen aktuální, ale i výchozí barvu vrcholů a hran.

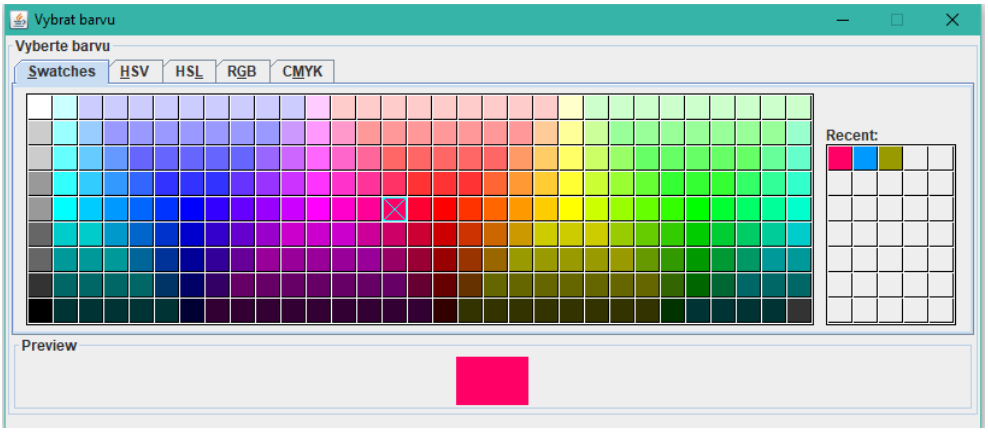
Co se týče vzhledu, máte možnost měnit velikost vrcholů a tloušťku hran, opět aktuální i výchozí.

Výchozí barva vrcholů a hran

Pro nastavení výchozích barev vrcholů a hran slouží tlačítka „Vrcholy“, „Hrany“ v dolní části panelu „Navigace“ (Obrázek-návod 1, bod 4). Tato tlačítka jsou zbarvena do aktuálně nastavených výchozích barev.

Po kliknutí se otevře paleta. Barvu si můžete zvolit z nabízených vzorů nebo máte možnost si vytvořit svoji vlastní. Pro míchání barev lze využít hned čtyř barevných modelů: HSV, HSL, RGB a CMYK. V dolní části palety je zobrazena aktuálně vybraná barva a v pravé části naposledy použité barvy.

Po vybrání vámi preferované barvy stačí okno palety zavřít, od této chvíle bude vámi zvolená barva nastavena jako výchozí. Nově vytvořené hrany a vrcholy budou mít zvolenou barvu.



Obrázek-návod 7 – Paleta pro výběr barvy

Nastavení velikostí vrcholů a hran

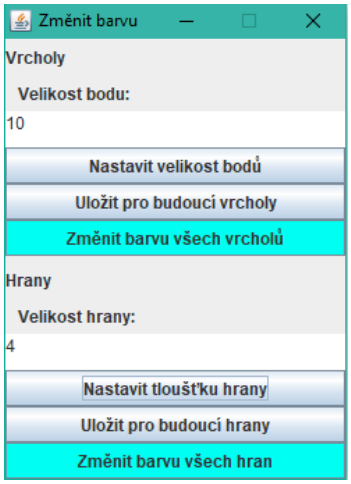
Tlačítko „Upravit vzhled grafu“ (Obrázek-návod 4) poskytuje další vzhledové úpravy grafu. „Změnit barvu všech vrcholů“ a „Změnit barvu všech hran“ umožňují, jak říkají jejich názvy, změnit barvu všech vrcholů nebo hran, které se aktuálně nacházejí na plátně. Pro volbu barvy je opět použita paleta, viz Obrázek-návod 3.

Dále je zde možnost nastavit **velikost** bodů a **tloušťku** hran a to aktuálních i následujících. Výchozí velikost vrcholů je nastavena na deset pixelů, tato hodnota označuje poloměr vrcholu na plátně. Tloušťka hran je defaultně nastavena na hodnotu čtyř pixelů.

Pro změnu hodnoty stačí jednoduše přepsat aktuální hodnotu v políčku na požadující hodnotu.

- Velikost bodu musí být v rozsahu 4-15
- Velikost hrany musí být v rozsahu 1-10

Celý proces následně stačí potvrdit jedním z níže položených tlačítek, které tuto hodnotu přiřadí aktuálním resp. budoucím vrcholům.



Obrázek-návod 8 – Nabídka vzhledových úprav grafu

4. Režim „Izomorfismus“

Slouží k vyhodnocení, zda jsou zadané grafy izomorfní. Do režimu se přepnete pomocí tlačítka „Izomorfismus“ v panelu navigace.

4.1. Přepínání mezi plátny

Přepínač má jednoduchou roli a to určit, pro který graf budou prováděny následující operace. Toto pravidlo se týká pouze ovládacích prvků, které nějakým způsobem upravují grafy. Jedná se o akce „**Nový graf**“, „**Přidej vrchol**“, „**Odstraň hranu**“ a „**Odstraň všechny hrany**“.

Pokud tedy chcete odstranit všechny hrany v grafu 2, musíte nejdříve přepnout do režimu úprav grafu 2, teprve poté můžete kliknout na tlačítko „Odstraň všechny hrany“. Pokud by se tak nestalo, mohlo by dojít k **nechtěnému odstranění** hran v grafu 1.

Ovládání plátna pomocí myši zůstává neměnné, nezávislé na poloze přepínače. Ovládací prvky z panelu „Vlastnosti“ mají totožné ovládání jako u předešlého režimu, viz bod 3.

Obrázek-návod 9 - Přepínač mezi plátny

4.2. Nový graf

Viz bod 3.1.

4.3. Přidání nového vrcholu

Viz bod 3.2.

4.4. Odstranění vrcholu

Viz bod 3.3.

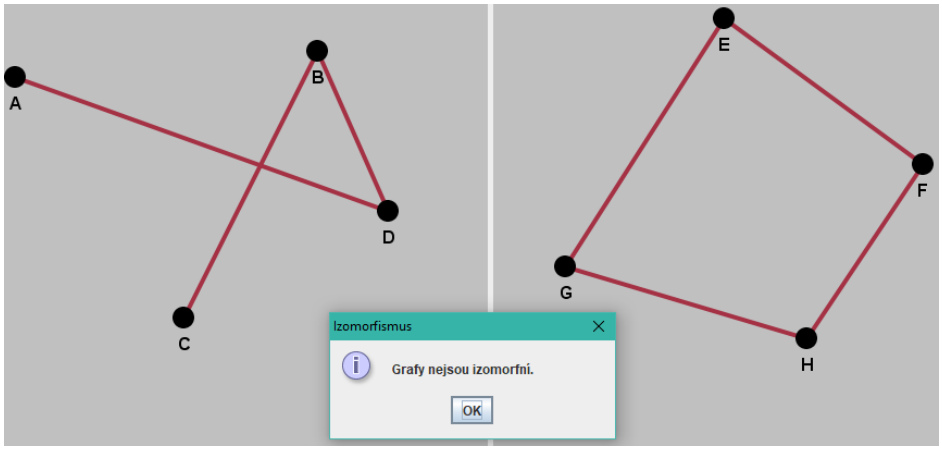
4.5. Odstranění hran

Viz bod 3.4.

4.6. Jsou grafy izomorfní?

Pro zjištění, zda jsou grafy izomorfní, klikněte na tlačítko „Jsou izomorfní?“. Výsledek se vám zobrazí v novém okně, viz Obrázek-návod 9.

V případě, že obě plátna neobsahují ani jednu hranu a ani jeden vrchol, zobrazí se pouze informační okno a porovnání grafů se nespustí.



Obrázek-návod 10 – Vyhodnocení izomorfismu

5. Režim „Kreslit podle skóre“

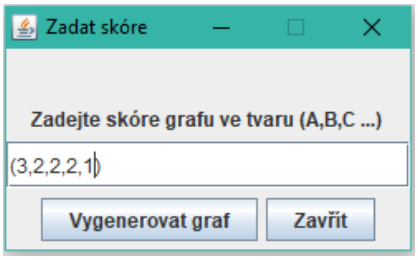
Ovládání plátna myší je stejné s předchozími režimy kreslení. Nicméně jsou dostupné akce pouze pravého a prostředního tlačítka myši nabízející posouvání a detail vrcholu. Do režimu se přepnete pomocí tlačítka „Kreslit podle skóre“ v panelu navigace.

5.1. Vykreslení grafu podle skóre

Po kliknutí na tlačítko „Zadat skóre“, zadejte skóre grafu do políčka v novém okně, které musí být zadáno ve tvaru (A,B,C…) pro příklad (3,2,2,2,1), jako na Obrázku-návod 10. Žádné jiné znaky nejsou podporovány, jinak zadávání bude ukončeno a zobrazí se vám chybové hlášení.

Pozor dejte především na bílé znaky ve formě mezer. V zadávaném skóre by se mezery neměly vyskytovat.

Pro vygenerování zadaného skóre zvolte „Vygenerovat graf“. Výsledkem je vygenerovaný graf metodou do hloubky.



Obrázek-návod 11 - Formulář pro zadání skóre

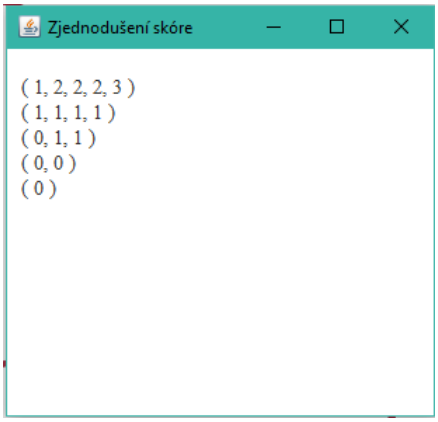
Po vygenerování grafu se vám nabídne možnost vygenerovat alternativní graf metodou do šířky v podobě tlačítka „Alternativní graf“ na bočním panelu „Vlastnosti“ (Obrázek-návod 1, bod 5).

Po kliknutí se vám vygeneruje alternativní graf metodou do šířky.

5.2. Zjednodušení skóre

Pro lepší pochopení práce se skórem, je zde tlačítko „Rozbor skóre“, které se vám zobrazí po vygenerování grafu na plátno v panelu „Vlastnosti“.

Po kliknutí se otevře nové textové okno obsahující úplný rozbor skóre, pomocí kterého můžete lépe porozumět práci se skórem, jak se postupovalo při vykreslování grafu na plátno.

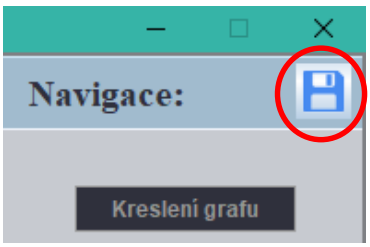


Obrázek-návod 12 - Rozbor zadaného skóre

6. Uložení grafu

Pomocí tlačítka pro uložení grafu, v pravém horním rohu (Obrázek-návod 13), můžete jednoduše uložit zakreslený graf jako obrázek ve formátu .JPG, .PNG nebo .GIF. Ostatní souborové formáty nejsou podporovány.

Po kliknutí na tlačítko uložení se otevře souborový prohlížeč, pomocí něhož si zvolte cestu, kam se má obrázek uložit. Následně stačí zadat do pole „File Name:“ jak se má obrázek jmenovat a vybrat jeden z preferovaných formátů (JPG, PNG, GIF).

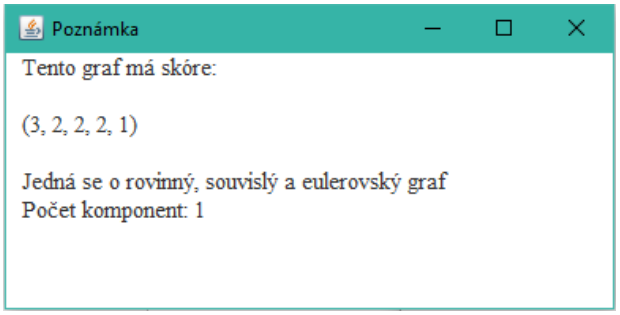


Obrázek-návod 13 – Tlačítko uložení

7. Psaní poznámky

Nezávisle na zvoleném režimu kreslení máte možnost psát rychlé poznámky. Naleznete je v bočním panelu úplně dole. Poznámky nejsou nijak limitovány, lze tedy do nich psát jakýkoli text dle uvážení. Jedná se prakticky o nové okno pro psaní, které je možné libovolně zvětšovat či zmenšovat.

Takových poznámkových oken můžete otevřít libovolný počet a sepsat si tak jednoduše poznámky ke každému ze zakreslených grafů.



Obrázek-návod 14 - Rychlá poznámka

8. Otevřít návod k programu

Stejné tlačítko jako mělo psaní poznámky, má také samotný návod. Jedná se o tlačítko „Návod“ umístěné opět úplně dole v bočním panelu „Vlastnosti“. Po kliknutí se otevře dokument, díky kterému lze rychle najít potřebné informace o aplikaci.

Seznam obrázků

Obrázek-návod 1 – Struktura aplikace GraphScore v2.0 2
Obrázek-návod 2 - Vytvoření nové hrany 3
Obrázek-návod 3 - Detail vrcholu 4
Obrázek-návod 6 - Vytvoření nového vrcholu 6
Obrázek-návod 7 - Mazání hran grafu 7
Obrázek-návod 8 - Vlastnosti grafu 7
Obrázek-návod 4 – Paleta pro výběr barvy 8
Obrázek-návod 5 – Nabídka vzhledových úprav grafu 9
Obrázek-návod 9 - Přepínač mezi plátny 10
Obrázek-návod 10 – Vyhodnocení izomorfismu 11
Obrázek-návod 11 - Formulář pro zadání skóre 12
Obrázek-návod 12 - Rozbor zadaného skóre 13
Obrázek-návod 13 – Tlačítko uložení 13
Obrázek-návod 14 - Rychlá poznámka 14

Univerzita Hradec Králové
Fakulta informatiky a managementu
Akademický rok: 2016/2017

Studijní program: Aplikovaná informatika
Forma: Prezenční
Obor/komb.: Aplikovaná informatika (ai3-p)

Podklad pro zadání BAKALÁŘSKÉ práce studenta

PŘEDKLÁDÁ:	ADRESA	OSOBNÍ ČÍSLO
Krátký Radim	Vilémov 13, Vilémov	I1500382

TÉMA ČESKY:

Teorie grafů

TÉMA ANGLICKY:

Graph Theory

VEDOUCÍ PRÁCE:

RNDr. Andrea Ševčíková - KIKM

ZÁSADY PRO VYPRACOVÁNÍ:

Cílem práce je podat přehledný výklad vybraného problému (vybraných problémů) z teorie grafů a jeho (jejich) řešení při konkrétních zadání z praxe. Součástí práce bude programová implementace metody vhodné pro řešení vybraného problému.

SEZNAM DOPORUČENÉ LITERATURY:

Demel Jiří: Grafy a jejich aplikace, nakladatelství Academia, Praha 2002, ISBN 80-200-0990-6.
Matoušek, Jiří; Nešetřil, Jaroslav. Kapitoly z diskrétní matematiky. Vyd. 2., opr., (V nakl. Karolinum 1.). Praha: Karolinum, 2000. ISBN 80-246-0084-6.
L. Kučera, Kombinatorické Algoritmy, SNTL, Praha, 1983.
GROSS, Jonathan L. a Jay YELLEN, ed. Handbook of graph theory. Boca Raton, Fla.: CRC Press, c2004. Discrete mathematics and its applications. ISBN 1-58488-090-2.

Podpis studenta: ..... **Datum:** 3. 4. 2017

Podpis vedoucího práce: ..... **Datum:** 3. 4. 2017