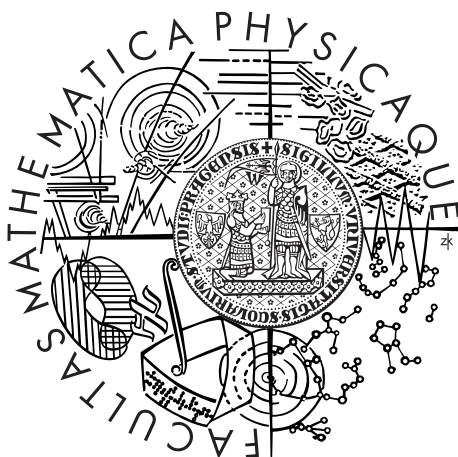


Univerzita Karlova v Praze
Matematicko-fyzikální fakulta

ZÁPOČTOVÝ PROGRAM



Hoang Anh Tuan

Flashcards

Vedoucí zápočtového programu: Vancák Vladislav, Bc.

Studijní obor: Datové a softwarové inženýrství

Semestr: Zimní semestr 2018/2019

Praha 2019

Předmluva

Flashcards jsem se rozhodl udělat kvůli tomu, že na trhu aplikací nebyl žádný program, který by mi vyhovoval, tak jsem se rozhodl udělat vlastní.

Flashcards slouží jako pomůcka k učení, nejčastěji to bývají slovíčka. Na jedné straně je napsané dané slovo a na druhé straně jeho překlad. Využívá se také například u chemického názvosloví. Na kartě nemusí být pouhé slovíčko, ale celé téma či jiná problematika.

Obsah

1	Zadání	2
2	Algoritmus	2
2.1	Práce se soubory	2
2.2	Hra	2
2.3	Úprava sekcí a karet	2
3	Program	3
3.1	Hlavní menu a výběr operace	3
3.2	Hra - <i>procedure Game()</i>	3
3.2.1	<i>function PrepareGame(section: string): PFlash</i>	3
3.2.2	<i>function Gameplay(SecondTry: boolean): boolean</i>	3
3.3	Úprava sekcí a karet	3
3.3.1	<i>procedure InsertWordToSection()</i> <i>procedure EditWordInSection()</i> <i>procedure DeleteWordInSection()</i>	3
3.3.2	<i>procedure InsertSection()</i> <i>procedure RenameSection()</i> <i>procedure DeleteSection(SectionName: string)</i>	4
3.3.3	<i>procedure LoadFile()</i>	4
3.4	Pomocné procedury a funkce	4
3.4.1	<i>function FindArrow(line: string): integer</i>	4
3.4.2	<i>procedure SwitchNames()</i>	4
3.4.3	<i>procedure PrintSection(section: string)</i>	4
3.4.4	<i>function ChooseSection(): string</i>	4
4	Reprezentace dat	5
4.1	V souboru	5
4.2	Vstupní data	5
4.3	Výstupní data	5
5	Co nebylo doděláno	5
6	Závěr	5

1 Zadání

Program je schopen načítání slovíček do sekcí. Slovíčka se načítají buď manuálně v programu slovíčko po slovíčku nebo jej dokáže načíst ze souboru s danou strukturou. Sekce mohou být upravovány - přejmenování a mazání nejen sekcí ale i jednotlivých slovíček. Nejdůležitější je možnost zkoušky ze slovíček. Zkouška vybírá slova náhodně a ověřuje správnost odpovědi. Na konci hry lze hru opakovat s kartami, které byly zodpovězeny špatně.

2 Algoritmus

2.1 Práce se soubory

Potřebná data jsou uložena v souboru *main.txt*, ze kterého se čtou informace. Jelikož je nutno data souběžně číst, upravovat a psát, tak se zapisují do souboru *temp.txt*. Na konci tedy *temp.txt* obsahuje aktuálnější informace, proto je přejmenován na *main.txt* a původní *main.txt* na *temp.txt* a tento proces se opakuje.

2.2 Hra

Uživatel si vybere sekci, ze které bude zkoušen. Program náhodně vybírá kartičky, ze kterých zkouší uživatele, pak na konci hry vypíše statistiku správných a špatných odpovědí. Pokud se v průběhu zkoušky vyskytly špatné odpovědi, je uživateli nabídnuto opakovat hru s těmito kartami.

2.3 Úprava sekcí a karet

Podle libosti uživatele lze jednotlivé sekce a karty opravovat, mazat či přidávat. Jednotlivé sekce/slovíčka se najdou v souboru a jsou podle uživatele opraveny. Názvy sekcí musejí být unikátní, pokud uživatel zadá duplikující se název, bude uživatel jednou vyzván k zadání unikátního názvu. Při dalším neúspěchu se daná operace ruší a vrátí uživatele na úvodní menu.

3 Program

3.1 Hlavní menu a výběr operace

Úvodní menu je založeno na cyklickém *repeat;until*, který je ukončen, když uživatel zadá 0. Uživateli je zobrazeno menu a jednotlivé výběry jsou očíslované. Uživatel tedy zadá číslo a podle *switch* se posune v programu dále.

3.2 Hra - *procedure Game()*

3.2.1 *function PrepareGame(section: string): PFlash*

Před samotnou hrou si uživatel vybere sekci. Ze souboru *main.txt* se naleznou slovíčka podle názvu sekce a jsou připraveny do kruhového spojového seznamu. Funkce *PrepareGame* vrátí ukazatel na poslední kartu.

3.2.2 *function Gameplay(SecondTry: boolean): boolean*

V průběhu hry se vybere náhodné číslo mezi 1 a počtem zbývajících karet v seznamu, který určuje kolikátá karta od současného ukazatele (na začátku hry ukazuje na poslední kartu) se vybere. Tato karta je odebrána ze seznamu a přední strana vybrané karty se vypíše uživateli a program čeká na odpověď, kterou ověří se zadní stranou karty. Pokud je odpověď správná, přidá uživateli bod, pokud je ale odpověď špatná, tak se karta přidá na kruhový spojový seznam špatných odpovědí. Odpoví-li uživatel na všechny karty správně, funkce vrátí hodnotu *true*. Tutéž hodnotu funkce vrátí, pokud se uživatel rozhodne hru se špatnými odpověďmi neopakovat. Pokud se ale rozhodne opakovat, vrátí funkce *false* a hra se opakuje. Hra se takto opakuje pouze jednou.

3.3 Úprava sekcí a karet

Před úpravou si uživatel vybere sekci, kterou chce upravit a program mu vypíše jednotlivé karty a operace, které může udělat. Karty a sekce mohou tedy být přidány, upraveny či smazány. Program čte jednotlivé řádky z *main.txt* a po příslušné úpravě je opět zapisuje do *temp.txt*. Algoritmy těchto operací jsou podobné, tudíž je popíšu současně.

3.3.1 *procedure InsertWordToSection()* *procedure EditWordInSection()* *procedure DeleteWordInSection()*

Program najde danou sekci, ve které mají operace proběhnout tak, že nalezne *#nazev_sekce*. Poté najde potřebný řádek s kartou. V případě mazání, celý řádek smaže. Pokud se jedná o úpravu slova, rozdělí řádek na tři části - na přední stranu, šipku a zadní stranu. Uživatel si pak může vybrat, zda-li danou stranu opraví nebo ne. Tyto tři části se nakonec sloučí a vloží se do souboru. V případě přidání nové karty, program čte řádky dokud opět nenarazí na *#*. Zeptá se uživatele jak má vypadat přední a zadní strana karty a tyto strany sloučí dohromady se šipkou uprostřed a vloží na konec sekce.

3.3.2 *procedure InsertSection()* *procedure RenameSection()* *procedure DeleteSection(SectionName: string)*

V souboru *main.txt* se nachází název sekce dvarát - v seznamu selekcí a ještě jednou ve tvaru *#nazev_sekce*, kde pod tím jsou vypsané jednotlivé karty. Při přidání a přejmenování sekce se kontroluje v seznamu sekcí, je-li nový název sekce unikátní, pokud ne, tak se uživatele ještě jednou požádá o název a upraví jej. Pokud uživatel vybere možnost přidání nové sekce, sekce se přidá na konec souboru s tím, že uživatel rovnou zadává karty. Je požádán o přední a zadní kartu, které jsou opět slévány do jednoho řádku se šipkou uprostřed. Pokud se uživatel rozhodne smazat celou sekci, smaže se i s ním spojené karty. Karty se mažou, dokud řádek nebude obsahovat název následující sekce nebo *#* označující konec souboru.

3.3.3 *procedure LoadFile()*

Uživatel při v hlavním menu má na výběr *Load Section*, ve kterém má na výběr manuálního vložení sekce popsané viz výše nebo má možnost vložit textový dokument do složky programu a podle daného systému mít již předem nadefinované karty. Uživatel zadá název souboru a následně je požádán o název sekce. Poté program přečte daný soubor a vloží jeho obsah do svého.

3.4 Pomocné procedury a funkce

3.4.1 *function FindArrow(line: string): integer*

Funkci je předán string, ve kterém nalezne šipku *"-->"* a číselnou pozici prvního charu šipky. Tato funkce pomáhá při rozdělování karty na přední a zadní část.

3.4.2 *procedure SwitchNames()*

Jelikož ze souboru *main.txt* se pouze čte a potřebné úpravy se zapisují do souboru *temp.txt*, tak se musejí na konci operace přejmenovat. *temp.txt* je přejmenován na *main1.txt*, *main.txt* na *temp.txt* a na závěr *main1.txt* na *main.txt*.

3.4.3 *procedure PrintSection(section: string)*

Procedura nalezne sekci, kterou dostane v podobě stringu. V souboru nalezne *#nazev_sekce* a vypíše jeho kartičky očíslované pro další operace.

3.4.4 *function ChooseSection(): string*

V průběhu programu často potřebujeme vybrat jednu ze sekcí. Funkce uspořádá sekce, které přečetl se seznamu sekcí, do lineárního spojového seznamu a číslovane jej vypíše. Počká na číslo od uživatele a o tolik míst se ve spojovém seznamu posune a vrátí název sekce.

4 Reprezentace dat

4.1 V souboru

Data se ukládají do textového souboru *main.txt*. Na začátku souboru je seznam sekcí, kde jsou na jednotlivých řádcích názvy sekcí. Seznam sekcí je ohraničen samostatných řádcích tečkami. Poté vždy následuje název jednotlivých sekcí ve tvaru *#nazev_sekce* a na následujících řádcích jednotlivé karty ve tvaru *predni_strana-->zadni_strana*. Na konci souboru je nakonec vypsán *#*, který značí konec souboru.

4.2 Vstupní data

Uživatel na žádost programu zadává čísla při výběru operace, sekce a karet. Může být ale i požádán o řetězec znaků reprezentující odpověď, názvu sekce, při vkládání nebo úpravy karet či názvu souboru při vkládání sekce.

4.3 Výstupní data

Program uživateli vypisuje celovětné instrukce, podle kterých by se měl řídit ke správné funkci programu.

5 Co nebylo doděláno

Program nebyl zajištěn proti špatným odpovědím, například, když má uživatel zadat číslo sekce, program tiše doufá, že uživatel skutečně napíše číslo existující sekce.

Dále by se program mohl vylepšit tak, že by byl ovládán šipkami za neustálého refreshování obrazovky po každém kliknutí. Program by byl intuitivnější a uživatelsky přívětivější.

Zkouška zatím funguje jenom jednosměrně, to znamená, že program ukáže uživateli přední stranu a kontroluje správnost odpovědi se zadní stranou. Dalo by se doimplementovat zkoušení zadních stran s ověřováním předních stran.

6 Závěr

Vytvořil jsem program Flashcards, který lze prakticky využívat. Program je inspirován dobrými idejemi jiných programů na stejném základě a je doplněn o různé možnosti, které mi připadaly užitečné. Díky tomuto projektu jsem se pořádně naučil ukazatele a také práci se soubory.