UNIVERZITA KARLOVA

Přírodovědecká fakulta

Studijní obor: Sociální geografie a geoinformatika

Ester Kaliská

Úvod do programování

Skúškové zadanie 1

Praha 2026

1. Zadanie

Úlohou programu je spracovať zadaný textový reťazec a vyhľadať v ňom všetky samohlásky. Každá nájdená samohláska má byť zvýraznená tým, že sa uzavrie do okrúhlych zátvoriek. Vstupný text môže obsahovať písmená latinskej abecedy (vrátane znakov s diakritikou), číslice, vybrané interpunkčné znamienka a medzery. Ak sa v texte objavia znaky, ktoré nie sú v zadaní podporované, program ich ignoruje a nezobrazí vo výsledku.

Cieľom riešenia je ukázať schopnosť pracovať s textovými reťazcami, podmienkami a cyklami, ako aj správne ošetriť rôzne typy vstupov od používateľa.

2. Popis a rozbor problému

Spracovanie textu patrí medzi časté úlohy v programovaní, či už ide o jednoduché filtre alebo zložitejšie jazykové analýzy. V tomto prípade ide o pomerne jednoduchú, no dobre definovanú transformáciu vstupu, ktorá pozostáva z niekoľkých krokov.

Základným problémom je správne rozlíšenie jednotlivých znakov vstupného reťazca. Program musí vedieť identifikovať:

- samohlásky
- ostatné povolené znaky
- nepovolené znaky, ktoré sa nemajú vo výstupe objaviť

Ďalším krokom je samotná transformácia – teda zmena pôvodného reťazca tak, aby každá samohláska bola jednoznačne zvýraznená pomocou zátvoriek, pričom ostatné znaky ostávajú nezmenené. Výsledkom má byť nový textový reťazec, ktorý zachová pôvodnú štruktúru textu, medzery aj interpunkciu.

Z formálneho hľadiska môžeme vstup chápať ako reťazec S, ktorý je tvorený znakmi z určitej abecedy. Výstupom je nový reťazec *S*, ktorý vznikne aplikáciou transformačných pravidiel na každý znak vstupu. Tieto pravidlá zabezpečujú konzistentné správanie programu pre všetky možné prípady.

3. Použitý algoritmus

Riešenie je založené na postupnom prechádzaní vstupného reťazca znak po znaku. Tento prístup je pre daný problém prirodzený a prehľadný.

Slovný opis algoritmu:

- inicializuje sa prázdny reťazec, do ktorého sa bude ukladať výsledok
- program načíta vstupný text od používateľa
- pomocou cyklu prechádza každý znak vstupného reťazca
- ak je znak samohláska, pridá sa do výsledku v zátvorkách
- ak ide o iný povolený znak, pridá sa bez úprav

- nepovolené znaky sa preskočia
- po spracovaní celého vstupu sa výsledný reťazec vypíše na obrazovku

Takto navrhnutý algoritmus je jednoduchý, čitateľný a zároveň ľahko rozšíriteľný, napríklad o ďalšie typy zvýrazňovania.

4. Ošetrenie problematických situácií

Pri implementácii bolo potrebné myslieť aj na situácie, ktoré by mohli viesť k nesprávnemu správaniu programu.

Jedným z problémov je rozlišovanie veľkých a malých písmen. Keďže samohlásky existujú v oboch tvaroch, program ich musí spracovať rovnako. Tento problém je vyriešený tým, že zoznam samohlások obsahuje malé aj veľké písmená.

Ďalším špecifickým aspektom je diakritika, ktorá je bežná v slovenskom a českom jazyku. Znaky ako á, é, í, ó, ú, ô alebo ů musia byť považované za samohlásky, inak by program neposkytoval korektné výsledky pre bežné texty.

Program je taktiež odolný voči prázdnemu vstupu. Ak používateľ nezadá žiadny text, program jednoducho vráti prázdny výstup bez toho, aby došlo k chybe alebo pádu aplikácie.

Nepovolené znaky, napríklad špeciálne symboly alebo ovládacie znaky, sú automaticky ignorované, čo zabezpečuje, že výsledok vždy spĺňa požiadavky zadania.

5. Vstupné dáta

Vstupom programu je textový reťazec zadaný používateľom prostredníctvom štandardného vstupu. Podporované sú:

- písmená latinskej abecedy vrátane diakritiky
- číslice od 0 do 9
- vybrané interpunkčné znamienka
- medzery

Príklad vstupu:

Ahoj, ako sa máš? 123

6. Výstupné dáta

Výstupom programu je nový textový reťazec, v ktorom sú všetky samohlásky zvýraznené pomocou okrúhlych zátvoriek. Ostatné povolené znaky zostávajú nezmenené a nepovolené znaky sa vo výstupe nenachádzajú.

Príklad výstupu:

(A)h(o)j, (a)k(o) s(a) m(á)š? 123

Takýto výstup umožňuje jednoduchú vizuálnu kontrolu samohlások v texte.

7. Programová dokumentácia

Program je navrhnutý s využitím objektovo orientovaného prístupu, ktorý zvyšuje prehľadnosť a uľahčuje ďalšie rozširovanie. Hlavnou triedou je trieda Highlighter, ktorá obsahuje všetku potrebnú logiku.

Konštruktor triedy slúži na definovanie vnútorných množín znakov, čím sa oddelia konštantné údaje od samotného algoritmu. Hlavná metóda zabezpečuje spracovanie vstupu a vytvorenie výsledného reťazca. Tento prístup umožňuje jednoduché opätovné použitie kódu v iných úlohách. S použitím OOP je jednoduchšie doplniť alebo zmeniť formu vyznačovania samohlások alebo rýchlo a jednoducho pridať iné funkcie, ktoré sa nad textom dajú vykonať.

8. Možnosti rozšírenia riešenia

Program v súčasnej podobe spĺňa zadanie, no existuje viacero spôsobov, ako by ho bolo možné vylepšiť. Pri spracovaní veľmi dlhých textov by bolo efektívnejšie využívať zoznamy znakov a metódu join(). Ďalším rozšírením by mohla byť automatická detekcia samohlások pomocou štandardných knižníc pre prácu s Unicode.

Zaujímavým rozšírením by bolo aj vytvorenie jednoduchého grafického rozhrania, v ktorom by používateľ mohol text zadávať a upravovať bez nutnosti práce s konzolou.

Zoznam literatúry:

Python documentation: *Lexical analysis,* https://docs.python.org/3/reference/lexical_analysis.html (3.2.2026)

W3Schools: *Python Classes,* https://www.w3schools.com/python/python_oop.asp (3.2..2026)

Vlastné poznámky a prezentácie Úvod do programovaní