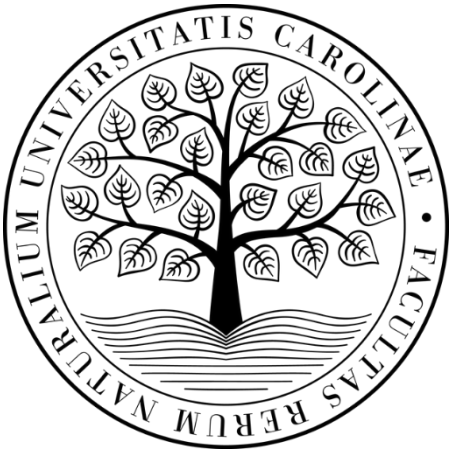
Univerzita Karlova

Přírodovědecká fakulta



Úvod do programování

Výpočet četností znaků v textu

Jan Prýmek

1. ročník N-GKDPZ

Příbram 2024

Zadání

Pro vstupní text zahrnující písmena “A-Ž”, “a-ž”, číslovky “0-9”, speciální znaky “,.?!;” oddělené mezerami vypočítejte absolutní a relativní četnost jednotlivých znaků. Výsledek znak-absolutní četnost-relativní četnost vypište sestupně dle hodnot absolutních četností. Pokud budou v textu nepodporované znaky, ignorujete je. Příklad bude řešen s využitím objektově orientovaného programování.

Problém

U každého znaku v textu je třeba zkontrolovat, jestli splňuje zadanou specifikaci, nebo zda je třeba ho ignorovat. Dále je potřeba určit, jestli se jedná o první výskyt znaku. Pokud ano, je potřeba přičíst k celkovému počtu výskytů tohoto znaku jedničku. Tuto informaci o počtu četností určitého znaku je nutné uložit, seřadit znaky od nejčetnějších a dopočítat relativní četnosti.

Algoritmus

Byla definována třída *CharCounter*, která obsahuje metody pro práci s textem a analýzu četností znaků. V konstruktoru této třídy je inicializován prázdný slovník *char_freq* pro ukládání četností znaků. První metoda *count_chars* přijímá text jako vstup a iterativně ho prochází znak po znaku pomocí cyklu for. Pro každý znak zjišťuje, zda je platný (podle definovaného souboru platných znaků) a poté zvyšuje jeho četnost ve slovníku *char_freq* o jedničku nebo je do slovníku přidán s hodnotou jedna.

Metoda *valid_chars* zjišťuje, zda je zadaný znak platný. Platné znaky jsou definovány v metodě pomocí množiny znaků obsahujících písmena A-Ž, a-ž, číslice 0-9 a interpunkční znaménka „ ,.?!; ". Nespecifikované znaky jsou ignorovány. Pseudokód:

```
cetnosti_znaku = {}
zadany_text = „Python je vysokoúrovňový programovací jazyk, který v roce 1991 navrhl Guido van Rossum.“

for znak in text:
        if znak in (A-Ž, a-ž, 0-9, ,.?!;):
                if znak již v textu byl:
                        přidej k četnosti 1
                else (jde o první výskyt znaku)
                        vytvoř klíč pro znak a dej mu hodnotu 1
```

Metoda *sort_chars_freq* dále řadí slovník *char_freq* podle četnosti jednotlivých znaků sestupně. Volá metodu *sort_chars*, která provádí řazení dvojic (znak, četnost) podle druhé složky (četnost) metodou tzv. bublinkového řazení. Tento algoritmus opakovaně prochází seznam, přičemž porovnává každé dva sousedící prvky, a pokud nejsou ve správném pořadí, prohodí je. Prvky s vyšší hodnotou tak „probublávají“ na konec seznamu (Szturcová 2024). Pseudokód:

```
Procedure BubbleSort(char_freq_pairs: seznam dvojic)
    n = délka(char_freq_pairs)
    for i = 0 do n-1 udělej:
        for j = 0 do n-i+1 udělej:
            if char_freq_pairs[j][1] < char_freq_pairs[j+1][1] pak:
                prohoď (char_freq_pairs[j], char_freq_pairs[j+1])
```

Další metoda *calculate_total_chars* vrací celkový počet znaků v textu spočítaný z hodnot ve slovníku *char_freq*. A metoda *print_chars_freq* vypisuje absolutní a relativní četnost znaků. Absolutní četnost je počet výskytů znaku v textu a relativní četnost je podíl četnosti znaku na celkovém počtu znaků v textu.

Byla vytvořena jedna instance *counter* třídy *ChartCounter*. Metoda *count_chars* je volána na vytvořené instanci *counter* pro výpočet četnosti znaků v zadaném textu. Dále se volá metoda *calculate_total_chars*, aby se získal celkový počet znaků v textu a metoda *print_chars_freq*, která vypíše absolutní a relativní četnosti znaků v textu.

Vstupní data

Vstupem do algoritmu je řetězec znaků napsaný nebo vložený uživatelem programu.

Výstup

Výstupem algoritmu je seznam znaků spolu s jejich absolutní četností a relativní četností v zadaném textu.

Výstup bude ve formátu: Character: X, Absolute Frequency: Y, Relative Frequency: Z %,

kde X je znak, Y je absolutní četnost znaku v textu a Z je relativní četnost znaku v textu vyjádřená v procentech.

Dokumentace

Po spuštění programu je uživatel vyzván k napsání nebo zkopírování textu. Program následně vypíše tuple, který obsahuje znaky a jejich absolutní a relativní četnosti v zadaném textu seřazené podle absolutní četnosti.

Příklad vstupu a výstupu

Python je vysokoúrovňový programovací jazyk, který v roce 1991 navrhl Guido van Rossum.

Character: o, Absolute Frequency: 10, Relative Frequency: 13.3%
Character: v, Absolute Frequency: 7, Relative Frequency: 9.3%
Character: r, Absolute Frequency: 6, Relative Frequency: 8.0%
Character: a, Absolute Frequency: 5, Relative Frequency: 6.7%
Character: y, Absolute Frequency: 3, Relative Frequency: 4.0%
Character: n, Absolute Frequency: 3, Relative Frequency: 4.0%
Character: e, Absolute Frequency: 3, Relative Frequency: 4.0%
Character: s, Absolute Frequency: 3, Relative Frequency: 4.0%
Character: k, Absolute Frequency: 3, Relative Frequency: 4.0%
Character: t, Absolute Frequency: 2, Relative Frequency: 2.7%
Character: h, Absolute Frequency: 2, Relative Frequency: 2.7%
Character: j, Absolute Frequency: 2, Relative Frequency: 2.7%

Character: ý, Absolute Frequency: 2, Relative Frequency: 2.7%
Character: m, Absolute Frequency: 2, Relative Frequency: 2.7%
Character: c, Absolute Frequency: 2, Relative Frequency: 2.7%
Character: 1, Absolute Frequency: 2, Relative Frequency: 2.7%
Character: 9, Absolute Frequency: 2, Relative Frequency: 2.7%
Character: u, Absolute Frequency: 2, Relative Frequency: 2.7%
Character: P, Absolute Frequency: 1, Relative Frequency: 1.3%
Character: ú, Absolute Frequency: 1, Relative Frequency: 1.3%
Character: ň, Absolute Frequency: 1, Relative Frequency: 1.3%
Character: p, Absolute Frequency: 1, Relative Frequency: 1.3%
Character: g, Absolute Frequency: 1, Relative Frequency: 1.3%
Character: í, Absolute Frequency: 1, Relative Frequency: 1.3%
Character: z, Absolute Frequency: 1, Relative Frequency: 1.3%
Character: ,, Absolute Frequency: 1, Relative Frequency: 1.3%
Character: l, Absolute Frequency: 1, Relative Frequency: 1.3%
Character: G, Absolute Frequency: 1, Relative Frequency: 1.3%
Character: i, Absolute Frequency: 1, Relative Frequency: 1.3%
Character: d, Absolute Frequency: 1, Relative Frequency: 1.3%
Character: R, Absolute Frequency: 1, Relative Frequency: 1.3%
Character: ., Absolute Frequency: 1, Relative Frequency: 1.3%

Vylepšení

Rozšíření specifikace znaků, aby se zahrnovaly znaky z dalších jazyků.

Znaky s diakritikou by mohly být připočítány ke znakům bez ní, velká písmena by mohla být převedena na malá.

Seřazení znaků podle zvoleného pravidla, ne pouze podle absolutní četnosti.

Umožnění uživateli načíst text ze souboru.

Zdroj:

Szturcová, D. (2024): Algoritmizace prostorových úloh. Třídění, vyhledávání. Dostupné z: https://gisak.vsb.cz/wikivyuka/images/9/98/APU_Sorting15.pdf (cit. 15. 2. 2024)