

DOKUMENTACE PROGRAMU

Seznam všech řádků, na kterých se vyskytují jednotlivá slova v zadaném textu

Autor: Jan Šindelář

Předmět: Úvod do programování

Ročník: 2. B-FGG

Datum odevzdání: 12.2.2026

Obsah

1. Zadání
2. Úvod
3. Specifikace úlohy
4. Návrh řešení
5. Popis implementace
6. Uživatelská dokumentace
7. Omezení a možná rozšíření
8. Závěr
9. Seznam použité literatury

1. Zadání

Ve vstupním textu zahrnujícím písmena “A-Ž”, “a-ž”, číslovky “0-9”, speciální znaky “, . ? ! ;” oddělené mezerami, naleznete zadaná slova. Výstupem pro bude pro každé slovo uspořádaná dvojice “slovo:[řádky]”. Pokud se v textu slovo nevyskytuje, uvedeme jako výstup pomlčku. Pokud budou v textu nepodporované znaky, ignorujte je.

2. Úvod

Tento dokument popisuje návrh, tvorbu a použití programu v jazyce Python, jehož účelem je vyhledávání zadaných slov v textovém souboru. Dokumentace je určena jak pro běžného uživatele programu, tak pro případného vývojáře, který by chtěl program dále rozšiřovat.

Program je koncipován jako jednoduchý konzolový nástroj. Jeho hlavní funkcí je vyhledat výskyt jednoho nebo více slov ve vstupním textu a zobrazit čísla řádků, na kterých se tato slova nacházejí. Vyhledávání je realizováno bez rozlišení velikosti písmen.

Cílem programu je demonstrovat práci se soubory, řetězci, slovníky, cykly a ošetřením výjimek v jazyce Python.

3. Specifikace úlohy

3.1 Zadání

Vytvořit program, který:

- načte od uživatele název textového souboru,
- načte seznam hledaných slov oddělených mezerou,
- vyhledá všechna zadaná slova v souboru,
- pro každé slovo vypíše seznam čísel řádků, kde se dané slovo nachází,
- ošetří situaci, kdy zadaný soubor neexistuje.

3.2 Vstupy

Program vyžaduje dva vstupy:

1. Název vstupního souboru

- Textový soubor musí existovat.
- Musí být dostupný pro čtení.
- Předpokládá se kódování UTF-8.

2. Hledaná slova

- Zadávají se jako řetězec.
- Jednotlivá slova jsou oddělena mezerou.

Příklad vstupu:

Enter the name of the input text: text.txt

Enter the searched words (separated by space): lorem, dolor, ahoj

3.3 Výstupy

Pro každé hledané slovo program vypíše:

- seznam čísel řádků, kde se slovo nachází,
- znak „-“ pokud se slovo v textu nevyskytuje.

Příklad výstupu:

lorem [1, 3, 5, 9]

dolor [1, 3, 5, 7]

ahoj [-]

3.4 Chybové stavy

Program ošetřuje následující chybový stav:

- **Soubor nebyl nalezen** – vypíše chybovou hlášku ("Error: The specified file was not found.") a ukončí běh programu.

4. Návrh řešení

4.1 Zvolený přístup

Program je realizován jako lineární skript bez použití funkcí nebo tříd. Tento přístup byl zvolen s ohledem na jednoduchost zadání a přehlednost kódu.

Hlavní kroky algoritmu:

1. Načtení vstupních údajů.
2. Inicializace datové struktury pro ukládání výsledků.
3. Otevření zadaného souboru.
4. Čtení souboru po řádcích.
5. Kontrola výskytu každého slova na každém řádku.
6. Uložení čísla řádku do příslušného seznamu.
7. Vypsání výsledků.

4.2 Datové struktury

Použit je slovník (dictionary):

```
found = {word: [] for word in words}
```

- Klíč: hledané slovo
- Hodnota: seznam čísel řádků

Tato struktura umožňuje efektivní organizaci výsledků a jejich přehledný výpis.

4.3 Způsob vyhledávání

Vyhledávání je realizováno pomocí operátoru:

```
if word in line_lower:
```

Jedná se o vyhledávání podřetězce. Nejsou kontrolovány hranice slov.

5. Popis implementace

5.1 Zdrojový kód

```
# Enter of the input from the user
```

```
text = input("Enter the name of the input text: ")
```

```
words = input("Enter the searched words (separated by spaces): ").lower().split()
```

```

# Dictionary for storing found lines for each word
found = {word: [] for word in words}

# Search for entered words in the text
try:
    with open(text, "r", encoding="utf-8") as file:
        for line_number, line in enumerate(file, start=1):
            line_lower = line.lower()

            for slovo in words:
                if slovo in line_lower:
                    found[slovo].append(line_number)
except FileNotFoundError:
    print("Error: The specified file was not found.")
    exit()

# Listing of searched words
for word, lines in found.items():
    if lines:
        print(f"{word} [{', '.join(map(str, lines))}]")
        found_any = True
    else:
        print(f"{word} [-]")

```

5.2 Práce se souborem

Použit je kontextový manažer `with`, který zajišťuje automatické uzavření souboru po skončení práce.

5.3 Ošetření výjimek

Použit blok `try – except` pro zachycení výjimky `FileNotFoundError`.

5.4 Práce s řetězcí

- Metoda `.lower()` zajišťuje nerozlišování velikosti písmen.
- Metoda `.split()` rozdělí vstupní řetězec na seznam slov.

- Funkce `join()` slouží ke spojení seznamu čísel řádků do řetězce.

6. Uživatelská dokumentace

6.1 Požadavky na prostředí

- Python 3.x
- Textový soubor v kódování UTF-8

6.2 Spuštění programu

Program se spouští z příkazové řádky: `python vyhledani_slov.py`

6.3 Postup použití

1. Spusťte program.
2. Zadejte název vstupního souboru.
3. Zadejte hledaná slova oddělená mezerou.
4. Program zobrazí výsledky na obrazovce.

7. Omezení a možná rozšíření

7.1 Omezení

- Vyhledávání je založeno na podřetězci.
- Není řešena diakritika.
- Nejsou kontrolovány hranice slov.
- Program nemá grafické rozhraní.

7.2 Možná rozšíření

- Použití regulárních výrazů pro přesné vyhledávání slov.
- Ignorování diakritiky.
- Počítání počtu výskytů.
- Uložení výsledků do souboru.
- Vytvoření grafického rozhraní.

8. Závěr

Program splňuje zadání – umožňuje vyhledávání více slov v textovém souboru a zobrazuje čísla řádků jejich výskytu. Implementace je jednoduchá, přehledná a vhodná pro výukové účely.

Řešení demonstruje práci se základními konstrukcemi jazyka Python, zejména se soubory, slovníky, cykly a ošetřením výjimek.

Program lze dále rozšířit o pokročilejší textovou analýzu nebo uživatelské rozhraní.

9. Seznam použité literatury

KRYL, Tomáš. *Dokumentace programů* [online]. KSVI MFF UK. Dostupné z: <https://ksvi.mff.cuni.cz/~kryl/dokumentace.htm> [cit. 12. 2. 2026].

PYTHON SOFTWARE FOUNDATION. *Python 3 Documentation* [online]. Python documentation verze 3.14.3. © 2001–2026 Python Software Foundation. Dostupné z: <https://docs.python.org/3/> [cit. 12. 2. 2026].

LUTZ, Mark. *Learning Python*. 5th ed. Sebastopol: O'Reilly Media, 2013. ISBN 978-1-4493-5573-9. [online]. Dostupné z: https://cfm.ehu.es/ricardo/docs/python/Learning_Python.pdf [cit. 12. 2. 2026].