

Projektové zadání: check.py

Cíl projektu

Vyvinout v Pythonu nástroj `check.py` pro sledování změn v souborech pomocí kontrolních součtů.

Požadavky

- **Počet členů týmu:** 2 nebo 3 studenti.
- **Repozitář:**
 - Název: `2425_INF_3DE_PRIJMENI1_PRIJMENI2_PRIJMENI3`
 - Umístění: Tým `2425_inf_3de` v organizaci `sps-cl-cz`
 - **Deadline odevzdání: 13.12. 23:59**
 - Obsahuje v kořenové složce soubor `rozdeleni` (bez koncovky).
 - Obsahuje v kořenové složce `README.MD`, ve kterém je popsáno, jak projekt nainstalovat (pokud je potřeba), spustit, a co jsou jeho **známé nedostatky**.
 - Obsahuje v kořenové složce soubor `check.py`, který bude spustitelný pomocí interpretu **Python 3.13.0**.
- Soubor `rozdeleni` musí obsahovat rozdělení bodů mezi členy týmu. Pokud se tedy někdo na projekt vykašle, toto je právě možnost pro zbytek týmu, aby nebyl poškozen. **Jakékoliv rozdělení jiné než 80-120% však budete muset písemně obhájit.**

Formát souboru `rozdeleni`

```
# Rozdělení
- Příjmení1   body1
- Příjmení2   body2
- Příjmení3   body3   (pokud je)
```

Celkový počet bodů:

- 200 bodů pro dvojici.
- 300 bodů pro trojici.

Příklad:

```
# Rozdělení
- Kříž 110
- Kocich 90
```

Popis programu check.py

Funkcionalita:

- **Sledování souborů** a jejich změn pomocí SHA-1 hashů.

Příkazy:

- `check.py -h`: Zobrazí nápovědu.
- `check.py init`: Inicializuje sledování (vytvoří `.check` soubor).
- `check.py add <pathspec>`: Přidá (aktualizuje) soubory ke sledování.
- `check.py remove <pathspec>`: Odebere soubory ze sledování.
- `check.py status`: Zobrazí stav sledovaných souborů.

Obecná doporučení

V projektu se mohou hodit například knihovny:

- <https://docs.python.org/3/library/argparse.html>
- <https://docs.python.org/3/library/hashlib.html>
- <https://docs.python.org/3/library/glob.html>

Formát obsahu souboru `.check` je na vás, ale měly by v něm být všechny potřebné informace:

- **relativní** cesta k souboru
- časová značka vytvoření (či aktualizace) záznamu
- SHA-1 hash

Detailní specifikace příkazů

`check.py -h` (i při samotném `check.py`)

- Zobrazí nápovědu k použití programu.
- Doporučuji využít [toto](#)

`check.py init`

- Vytvoří prázdný soubor `.check` v aktuálním adresáři.
- Obsahuje seznam sledovaných souborů, časové značky a SHA-1 hashe.

`check.py add <pathspec>`

- Přidá specifikované soubory k sledování.
- Pro každý soubor vypočítá SHA-1 hash a uloží informace do `.check`.
- Pokud se soubor aktualizuje, vypočítá se nový nový hash a zapíše se s novou časovou značkou.

check.py remove <pathspec>

- Odebere specifikované soubory ze sledování.

check.py status

- Zobrazí stav všech sledovaných souborů.
- 1 soubor = 1 nebo více řádků výstupu v závislosti na stavu.
- Formát výstupu:
 - [OK] SHA-1 timestamp cesta
 - [ERROR] Soubor nebyl nalezen: cesta
 - [CHANGE] SHA-1 timestamp cesta
 - NEW HASH: SHA-1 nového obsahu
- Na konci se zobrazí shrnutí:
 - Celkový počet souborů s různými stavy, např.:
12 OK, 3 CHANGE, 5 ERROR

Ukázka výpisu (velké mezery jsou znak tabulátoru \t):

```
[OK]      395199e90631aea24aeeddc57cc2f7f637d0b707      2024-11-25 15:08:24      check.py
[OK]      2fd4e1c67a2d28fced849ee1bb76e7391b93eb12      2024-11-25 15:08:24      a/s_1.txt
[CHANGE]   55d0b376e2c31822957c22f39e5db666a2808caa      2024-11-25 15:08:24      a/s_2.txt
NEW HASH:  650154435ebb8a9dfed78cc83b58a1efddb703e
[ERROR] Soubor nebyl nalezen: neexistujici.txt
[OK]      1878ace0cb4661e658c55ef9198730fb312ce090      2024-11-25 15:08:24      a/s_3.txt

3 OK, 1 CHANGE, 1 ERROR
```

Přílohy

Ukázka práce s knihovnou hashlib

```
import hashlib

def vypocti_sha1(soubor_cesta):
    sha1_hash = hashlib.sha1()
    with open(soubor_cesta, "rb") as f:
        for bajtovy_blok in iter(lambda: f.read(4096), b''):
            sha1_hash.update(bajtovy_blok)
    return sha1_hash.hexdigest()

# Vypočítání hashe souboru
hash_souboru = vypocti_sha1("soubor.txt")
print(f"SHA-1 hash souboru: {hash_souboru}")
```

Hodnocení

Maximálně lze získat 100 bodů (body navíc budou oříznuty).

- 20b Klíčová funkcionalita projektu, soubor `.check`
- 5b Správné náležitosti projektu na GitHubu (licence, `.gitignore`)
- 5b `check.py -h`
- 10b `check.py init`
- 10b `check.py add <pathspec>`
- 10b `check.py remove <pathspec>`
- 10b `check.py status`
- 20b Ošetření edge-cases, chyb (bez pádu programu)
- 10b Dodržení zásad Clean Code

Známkování

Počet bodů	Známka
> 88b	1
> 82b	1–
> 73b	2
> 67b	2–
> 53b	3
> 47b	3–
> 31b	4
> 25b	4–