Pracovný list – Reťazcové metódy, zložené a vnorené podmienky

Skúmanie

|  |  |
| --- | --- |
| **Úloha 1** | Otvorte program **12\_01\_metody.py**. Program spusťte viackrát pre rôzne reťazce.vstup = 'vOLÁM SA jOŽKO mRKVIČKA, MÁM 15 ROKOV'vystup = vstup.swapcase()print(f'{vstup} => {vystup}')Čo je výsledkom metódy retazec.swapcase()? |

|  |  |
| --- | --- |
| **Úloha 2** | Otvorte program **12\_02\_metody.py**. Program spusťte viackrát pre rôzne reťazce.meno = input('Zadajte svoje meno: ')if meno.isalpha()and meno[0] == meno[0].upper(): print(f'Zadané meno {meno} je v poriadku.')else: print(f'Zadané meno {meno} nie je v poriadku.')Pre ktoré reťazce program vypíše vetu 'Zadané meno {meno} je v poriadku.'?Pre ktoré reťazce program vypíše vetu 'Zadané meno {meno} nie je v poriadku.'?Čo je výsledkom príkazu retazec.isalpha()?Čo je výsledkom príkazu retazec.upper()? |

Vysvetlenie

|  |  |
| --- | --- |
| **Úloha 3** | Otvorte program **12\_03\_pismena.py**. Doplňte tento program tak, aby ku každému znaku vstupného reťazca vypísal veľké písmeno ekvivalentné danému znaku. Vhodnú metódu použite namiesto zápisu ‘…’ v príkaze print. Môžete predpokladať, že na vstupe je reťazec pozostávajúci len z malých a veľkých písmen. |

|  |  |
| --- | --- |
| **Úloha 4** | S využitím skúseností z úlohy 2 riešte nasledujúcu úlohu:Vytvorte program **uprava.py**, ktorý dostane na vstupe meno používateľa. Podmienkou je, aby vstupný reťazec obsahoval len písmená. Program skontroluje, či zadaný vstup obsahuje len písmená – ak nie, vypíše oznam o chybnom vstupe. Následne upraví vstupný reťazec tak, aby prvé písmeno bolo veľké a všetky ostatné malé. |

Rozpracovanie

|  |  |
| --- | --- |
| **Úloha 5** | Škola v rámci svojho informačného systému ponúka žiakom vlastný komunikačný kanál. Žiak pri registrácii zadá svoju prezývku a vek. Ak ich zadal správne, získa prístup a môže komunikovať s ostatnými žiakmi školy. Vek je prirodzené číslo v rozsahu od 10 do 20, pre prezývku platia podmienky:* na začiatku prezývky je podreťazec ‘ZIAK’,
* za podreťazcom ‘ZIAK’ nasledujú už len číslice, pričom číselnú časť tvoria aspoň 3, najviac však 8 číslic.

Vytvorte program **registracia.py**, ktorý otestuje vstupné hodnoty podľa uvedených podmienok. |

Sebahodnotiaci test

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 1. | Pani učiteľka v rámci záverečného opakovania pripravuje na každú hodinu slovenského jazyka krátky diktát vo forme doplňovačky: namiesto i/y napíše znak \_ a žiaci vpisujú na tento znak správne písmeno.Vyberte správny kód, ktorý upraví vstupný text diktátu tak, že nahradí všetky výskyty písmen i, í, y, ý, I, Í, Y, Ý znakom \_, napr. pri vstupnom texte „V našej peci myši pištia. Asi nie sú sýte.“ získame výstupný text „V našej pec\_ m\_š\_ p\_št\_a. As\_ n\_e sú s\_te.“Pomôcka: V pomocnom materiáli sa dozviete viac o metóde *replace()*.

|  |  |
| --- | --- |
| **Kód č. 1** | **Kód č. 2** |
| text = **'V koryte spí milá myš.'**pismena = **'iíyýIÍYÝ'****for** znak **in** pismena: text = text.replace(znak, **'\_'**)print(text) | text = **'V koryte spí milá myš.'**pismena = **'iíyýIÍYÝ'****for** znak **in** pismena: text.replace(znak, **'\_'**)print(text) |

Správny je kód číslo . |
| 2. | Preštudujte si nasledujúci program:vstup = input(**'Zadajte vstupný reťazec: '**)**if** vstup[0] == **'L' and** vstup[-3:] == **'035'**: **if** len(vstup) < 8: print(**'Výpis1'**) **else**: print(**'Výpis2'**)**else**: print(**'Výpis3'**)1. Aký bude výstup nasledujúceho programu, ak na vstupe zadáme reťazec ‘Letisko035’?
2. Vytvorte taký vstupný reťazec, aby výstupom programu bol *Výpis1*.
 |

|  |  |
| --- | --- |
| Vedomosti v kocke |  |
| Použitie reťazcových metód nám môže uľahčiť riešenie daného problému. Namiesto vytvárania vlastného algoritmu riešiaceho daný podproblém zvolíme vhodnú metódu.Dôležité je zapamätať si, že **reťazcová metóda nemení reťazec**, na ktorý ju použijeme – **vytvára jeho upravenú kópiu** (podľa zvolenej metódy) a my túto **kópiu musíme spracovať**. Môžeme ju prepojiť s novou alebo pôvodnou premennou, prípadne ju použiť ako parameter niektorej funkcie:

|  |  |
| --- | --- |
| s = **'kaTKa'**nove = s[0].upper() + s[1].lower() | s = **'kaTKa'**print(s[0].upper() + s[1].lower()) |

Pri úpravách reťazcov často využívame podmienky – zložené alebo vnorené.**Zložené podmienky** vytvárame pomocou logických operátorov and, or a not:

|  |  |
| --- | --- |
| Logický operátor | Popis |
| *and* | Vráti *True*, ak sú všetky logické výrazy pravdivé. |
| *or* | Vráti *True*, ak je pravdivý aspoň jeden z logických výrazov. |
| *not* | Neguje výsledok, vracia *False*, ak bol výsledok *True* a naopak. |

Často krát je vhodnejšie použiť namiesto zložených podmienok **vnorené podmienky** (nahrádzame nimi najmä viacnásobné použitie logického operátora and). Napr. ak chceme od používateľa získať jeho vek a výšku, budeme pravdepodobne vyžadovať, aby obe čísla boli kladné. Je pre používateľa pohodlnejšie, ak mu oznámime hneď po načítaní jeho veku, že zadal nekorektné číslo, a len v prípade správne zadaného veku budeme žiadať zadanie výšky:

|  |  |
| --- | --- |
| Riešenie úlohy pomocou zložených podmienok | Riešenie úlohy pomocou vnorených podmienok |
| vek = input(**'Vek: '**)vek = int(vek)vyska = input(**'Výška (cm): '**)vyska = int(vyska)**if** vek > 0 **and** vyska > 0: print(**'Údaje sú OK.'**)**else**: print(**Chybný vstup.'**)Riešenie je stručnejšie, núti však používateľa zadať obidve vstupné hodnoty, aj keď hneď prvá je chybná. Navyše používateľ nevie, pri ktorom vstupe zadal chybný vstup. | vek = input(**'Vek: '**)vek= int(vek)**if** vek > 0: vyska = input(**'Výška (cm): '**) vyska = int(vyska) **if** vyska > 0: print(**'Údaje sú OK.'**) **else**: print(**'Chybná výška.'**)**else**: print(**'Chybný vek.'**)Použitie vnorených podmienok predpokladá, že programátor nad úlohou premýšľal aj z pohľadu používateľa. |

 |