Pracovný list č. 14 – Reťazcové metódy

|  |  |
| --- | --- |
| **Úloha 1** | Otvorte program **metody\_01.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()? Výsledkom je kópia vstupného reťazca, ktorá je upravená tak, že malé písmená sa zmenia na veľké a veľké na malé. |

|  |  |
| --- | --- |
| **Úloha 2** | Otvorte program **metody\_02.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 reťazce, ktoré spĺňajú dve podmienky: všetky znaky sú písmená a prvé písmeno je veľké.  Pre ktoré reťazce program vypíše vetu 'Zadané meno {meno} nie je v poriadku.'? Pre reťazce, ktoré nespĺňajú aspoň jednu podmienok z prvej otázky.  Čo je výsledkom príkazu retazec.isalpha()? Výsledkom je hodnota True, ak skúmaný reťazec obsahuje len písmená. Inak je výsledkom false.  Čo je výsledkom príkazu retazec.upper()? Výsledkom je kópia pôvodného reťazca, v ktorej sú všetky písmená prevedené na veľké. |

**Poznámky**

* retazec.count(podreťazec) – počet výskytov podreťazca v reťazci
* retazec.find(podreťazec) – zistí index prvého výskytu podreťazca v reťazci, ak sa v reťazci podreťazec nevyskytuje, vráti -1
* retazec.index(podreťazec) – ako find(), ak sa v reťazci podreťazec nevyskytuje, vráti chybu ValueErrr
* retazec.isalnum()- vráti True, ak reťazec obsahuje len alfanumerické znaky, inak vráti False, napr. print(‘Hviezdoslavova 6‘.isalnum()) ⇒ False, print(‘Hviezdoslavova6‘.isalnum()) ⇒ True
* retazec.isalpha()- vráti True, ak reťazec obsahuje len alfa znaky, inak vráti False, napr. print(‘Hviezdoslavova 6‘.isalpha()) ⇒ False, print(‘Hviezdoslavova‘.isalpha()) ⇒ True
* retazec.isnumeric ()- vráti True, ak reťazec obsahuje len numerické znaky, inak vráti False, napr. print(‘H56‘.isnumeric()) ⇒ False, print(‘456‘.isnumeric()) ⇒ True
* retazec.lower() – prevedie všetky písmená na malé
* retazec.upper() - prevedie všetky písmená na veľké
* retazec.replace(podreťazec1, podreťazec2) – nahradí podreťazec1 podreťazcom2
* retazec.capitalize() – vráti kópiu reťazca, v ktorej je prvý znak prevedený na veľké písmeno, ostatné sú malé
* retazec.swapcase() – vráti nkópiu reťazca, v ktorej sú všetky výskyty malých písmen nahradené veľkými písmenami a naopak

|  |  |
| --- | --- |
| **Úloha 3** | Poznáte svoje šťastné číslo? Existujú rôzne návody, ako ho vypočítať. Tu je jeden z nich:  Svoje šťastné číslo vypočítame tak, že spočítame ciferný súčet nášho dátumu narodenia a takto získane číslo upravíme podľa toho, o koľkej sme sa narodili:   1. ak sme sa narodili od 00:00 hod. vrátane do 06:00 hod., k číslu pripočítame číslo 6, 2. ak sme sa narodili od 06:00 hod. vrátane do 13:00 hod., k číslu pripočítame číslo 3 3. ak sme sa narodili od 13:00 hod. vrátane do 20:00 hod., k číslu pripočítame číslo 4, 4. ak sme sa narodili od 20:00 hod. vrátane do 00:00 hod., k číslu pripočítame číslo 7.   Napríklad ak sme sa narodili 5. 12. 2003 o 14:35, program dostane na vstupe dva reťazce: ‘5.12.2003’ a ’ 3’ , naše šťastné číslo je výsledkom výpočtu 5+1+2+2+0+0+3 + 4, teda 17.  Otvorte program **stastie.py**. Definujte funkciu stastne\_cislo(), ktorá pre zadaný reťazec reprezentujúci dátum a pre číslo reprezentujúce časový úsek, v ktorom sme sa narodili, vypíše do konzoly šťastné číslo. Predpokladajte, že dátum je reprezentovaný v tvare den.mesiac.rok.  Riešenie:  def stastne\_cislo(datum, cas):  upraveny\_datum = datum.replace('.', '') #odstranime bodky z datumu  if upraveny\_datum.isnumeric() and cas in '1234': #ak takto upraveny datum obsahuje len cislice a casovy usek je urceny spravnym kodom ...  sucet = 0  for i in upraveny\_datum: #prejdeme retazec po znaku a urobime ciferny sucet ...  sucet = sucet + int(i) #tak, znaky pretypujeme na cele cislo a pripocitame ho k vyslednej hodnote  if cas == '1': #testujeme kod casoveho useku dna a podla neho upravime ciferny sucet  sucet = sucet + 6  elif cas == '2':  sucet = sucet + 3  elif cas == '3':  sucet = sucet + 4  else:  sucet = sucet + 7  else:  raise ValueError('Dátum alebo časový úsek dňa obsahuje neočakávané znaky')  return sucet  datum = input('Zadajte dátum v tvare den.mesiac.rok: ') #nacitame datum narodenia pouzivatela  print('Ak ste sa narodili od 00:00 hod. vrátane do 06:00 hod., stlačte 1.') #vypiseme casove useky dna, ktore sledujeme  print('Ak ste sa narodili od 06:00 hod. vrátane do 13:00 hod., stlačte 2.')  print('Ak ste sa narodili od 13:00 hod. vrátane do 20:00 hod., stlačte 3.')  print('Ak ste sa narodili od 20:00 hod. vrátane do 24:00 hod., stlačte 4.')  cas = input('Vaša voľba: ') #nacitame kod zvoleneho casoveho useku dna  try:  print(f'Vaše šťastné číslo: {stastne\_cislo(datum, cas)}') #vypiseme navratovu hodnotu funkcie stastne\_cislo()  except ValueError as chyba:  print(chyba) #niektory z udajov bol zadany nespravne |

|  |  |
| --- | --- |
| **Úloha 4** | Pani učiteľka slovenského jazyka posudzuje náročnosť diktátu podľa počtu písmen i, í, y, ý v diktovanom texte. Definujte funkciu pocet\_iy(), ktorá pre zadaný reťazec vypíše, koľko písmen i, í, y, ý reťazec obsahuje. Program uložte pod názvom **diktat.py.**  Riešenie:  def pocet\_iy(text):  pismena = 'iyíýIÍYÝ'  pocet = 0  for znak in pismena:  pocet = pocet + text.count(znak)  return pocet  vstup ='V našej peci myši pištia, v našej peci psík spí.'  print(pocet\_iy(vstup)) |

|  |  |
| --- | --- |
| **Úloha 5** | Vo vstupnom formulári pre evidenciu nových zamestnancov programátori vytvorili textové pole pre zadanie mena a priezviska zamestnanca. Neskoro si uvedomili, že pre potreby databázy zamestnancov potrebujú samostatne meno a samostatne priezvisko.  Definujte funkcie meno() a priezvisko(), ktorých vstupným parametrom je reťazec v tvare „meno priezvisko“ (oddelené práve jednou medzerou). Návratovou hodnotou funkcie meno() je meno zamestnanca (prvé písmeno veľké, ostatné malé), návratovou hodnotou funkcie priezvisko() je priezvisko zamestnanca (všetky písmená veľké). Program uložte pod názvom **formular.py.**  Riešenie:  def meno\_zamestnanca(retazec): #navratovou hodnotou funkcie je meno zamestnanca  i = retazec.find(' ') #najdeme poziciu medzery / pred nou konci meno  meno = retazec[:i] #vyrez prepojime s premennou meno (od prveho pismena po pismeno pred medzerou)  meno = meno[0].upper() + meno[1:].lower() #prve pismeno prevedieme na velke, ostatne na male  return meno  def priezvisko\_zamestnanca(retazec): #navratovou hodnotou funkcie je priezvisko zamestnanca  i = retazec.find(' ') #najdeme poziciu medzery / za nou zacina priezvisko  priezvisko = retazec[i+1:] #vyrez prepojime s premennou priezvisko (od prveho pismena za medzerou po posledne pismeno)  priezvisko = priezvisko.upper() #vsetky pismena prevedieme na velke  return priezvisko  zamestnanec = 'JOzEf vESELý'  print(meno\_zamestnanca(zamestnanec))  print(priezvisko\_zamestnanca(zamestnanec)) |

|  |  |
| --- | --- |
| **Úloha 6** | Napíšte program, ktorý nahradí všetky podreťazce:   1. novým reťazcom, 2. prázdnym reťazcom.   Program uložte pod názvom **novy.py** a **prazdny.py.**  Riešenie:   1. nový reťazec   s = input('Zadaj text: ') 'Okolo kolovrátka sedeli v kole všetci z okolia.'  s = s.replace('kol', '\*')  print(s)   1. prázdny reťazec   s = input('Zadaj text: ') 'Okolo kolovrátka sedeli v kole všetci z okolia.'  s = s.replace('o', '')  print(s) |

|  |  |
| --- | --- |
| **Úloha 7** | Napíšte program **cvicenie.py**, ktorý v zadanej vete nahradí všetky výskyty y, Y, i, a I pomlčkami, a vašou úlohou je napísať vetu aj so správne doplnenými i a y. Program našu odpoveď skontroluje a nejako označí nesprávne doplnenia.  Riešenie:  vstup = input('Zadaj vetu:')  zadanie = vstup  zamenit = 'iIyY'  for znak in zamenit:  zadanie = zadanie.replace(znak, '-')  print(zadanie)  riesenie = input('Napíš vetu aj so správnym i / I, y / Y:')  odpoved = ''  for i in range(len(riesenie)):  if vstup[i] != riesenie[i]:  odpoved += '!'  else:  odpoved += vstup[i]  print(odpoved) |

Sebahodnotiaci test

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 1. | Pán učiteľ napísal na tabuľu nasledujúce programové kódy. Žiaci majú rozhodnúť, ktorý z nich upraví reťazec, na ktorý odkazuje premenná s tak, že jeho prvé písmeno je malé a všetky ostatné veľké:   |  |  | | --- | --- | | **Kód 1** | **Kód 2** | | s = 'PeRINBabA'  s = s[0].lower()+s[1:].upper() | s = 'PeRINBabA'  s = s.capitalize().swapcase() |   Vyberte pravdivé tvrdenie:  **(A)**   Zadanú úlohu rieši len kód 1. **(B)**   Zadanú úlohu rieši len kód 2.  **(C)**   Obidva kódy riešia zadanú úlohu. **(D)**   Ani jeden z kódov nie je riešením zadanej úlohy. |
| 2. | 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.“   |  |  | | --- | --- | | **Kód 1** | **Kód 2** | | text = **'V koryte spí milá myš.'** nahrad = **'iíyýIÍYÝ'**  **for** znak **in** nahrad:  text = text.replace(znak, **'\_'**)  print(text) | text = **'V koryte spí milá myš.'** nahrad = **'iíyýIÍYÝ'**  **for** znak **in** nahrad:  text.replace(znak, **'\_'**)  print(text) | | **Kód 3** | **Kód 4** | | text = **'V koryte spí milá myš.'** nahrad = **'iíyýIÍYÝ'**  **for** znak **in** text:  znak.replace(nahrad, '\_')  print(text) | text = **'V koryte spí milá myš.'** nahrad = **'iíyýIÍYÝ'**  **for** znak **in** text:  znak = znak.replace(nahrad, '\_')  print(text) |   Správny je kód číslo 1 . |