

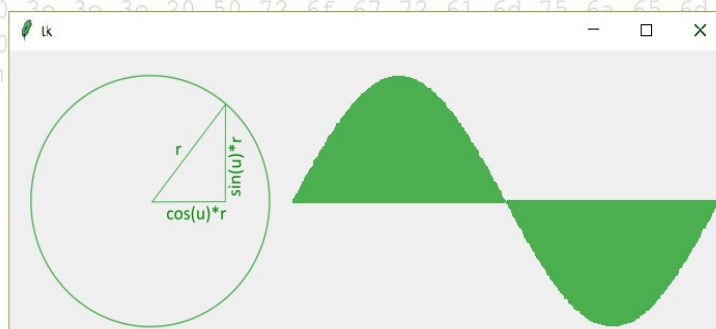
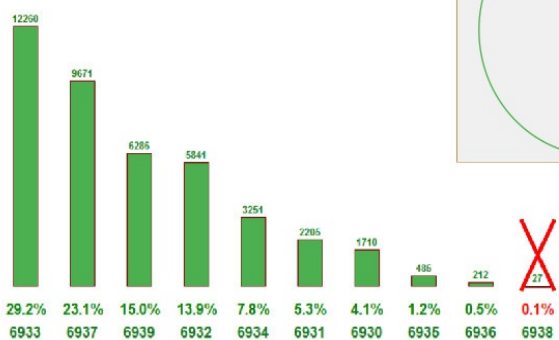
Programujeme v Pythone

učebnica informatiky pre stredné školy

Peter Kučera, Jaroslav Výboštok

2

Hlasovalo: 41948 ľudí.



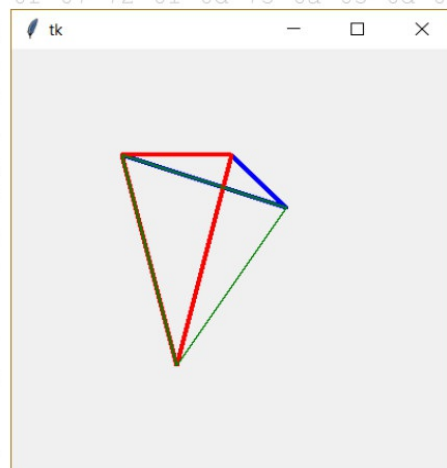
hod = 3



(0, 0, 0, 0, 0, 0)

[:2] [2] [3:]

[:hod-1] [hod] [hod:]



```
score.txt - Poznámkový blok
Súbor Úpravy Formát Zobrazit Pomocník
High score
=====
Alena 120
Bob 115
Charles 80
Dana 75
Edo 70
```



Programujeme v Pythone 2

učebnica informatiky pre stredné školy

Autori © Mgr. Peter Kučera, Mgr. Jaroslav Výboštok

Design © Mgr. Peter Kučera

Jazyková korektúra: Mgr. Katarína Kučerová

Prvé vydanie, 2017

Verzia číslo: 20171115

Vydavateľ: Mgr. Peter Kučera

Ukážka z e-knihy

Upozorňujeme, že elektronická kniha je dielom chráneným podľa autorského zákona a je určená len pre osobnú potrebu kupujúceho. Kniha ako celok ani žiadna jej časť nesmie byť voľne šírená na internete, ani nijako ďalej zverejňovaná. V prípade ďalšieho šírenia neoprávnene zasiahnete do autorského práva s dôsledkami podľa platného autorského zákona a trestného zákonníka.

Veľmi si vážime, že e-knihu ďalej nešírite. Len vďaka vašim nákupom dostanú autori, vydavatelia a kníhkupci odmenu za svoju prácu. Ďakujeme, že tak prispievate k vzniku ďalších skvelých kníh.

učebnicu a ďalšie materiály si môžete zakúpiť aj priamo na stránkach autora:

<http://www.programujemevpythone.sk/> a <https://www.facebook.com/programujemevpythone>

ISBN 978-80-972779-1-8 (pdf)

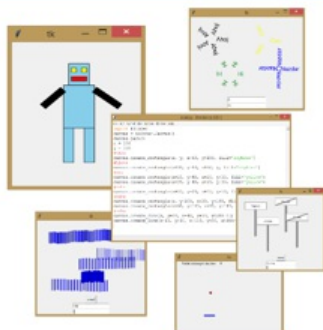
ISBN 978-80-972779-2-5 (epub)

ISBN 978-80-972779-3-2 (mobi)

Z NAŠEJ PONUKY

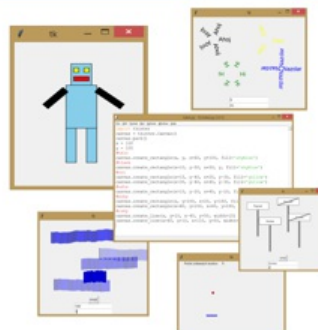
Programujeme v Pythone

učebnica informatiky pre stredné školy
Peter Kučera



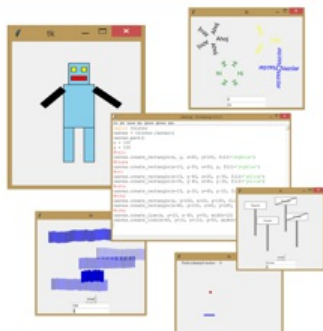
Príručka pre učiteľa

k učebnici Programujeme v Pythone
Peter Kučera



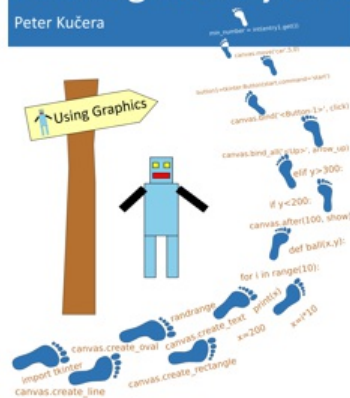
Testy k učebnici

Programujeme v Pythone
Peter Kučera



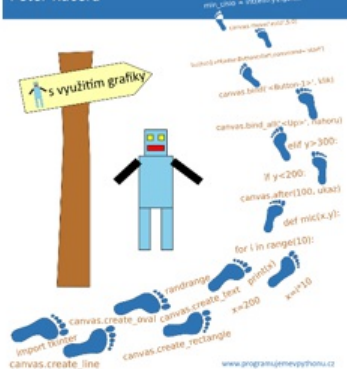
Creating with Python

Peter Kučera



Programujeme v Pythonu

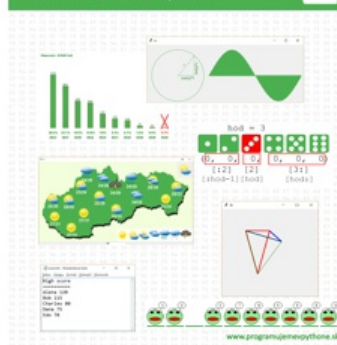
Peter Kučera



Programujeme v Pythone

učebnica informatiky pre stredné školy
Peter Kučera, Jaroslav Výboštok

2



www.programujemevpythone.sk
www.facebook.com/programujemevpythone

Napísali o našich e-knihách...

Učebnica Programujeme v Pythone (učebnica pre stredné školy) si rýchlo vydobyla miesto vo vyučovaní programovania naprieč celým Slovenskom vďaka jednoduchému prístupu k celej škále programovacích prostriedkov zavedených prostredníctvom grafického prostredia tkinter. Tento prístup podporuje vizualizáciou výsledkov v interaktívnom alebo programovom režime pochopenie programovacích techník, priam vyzýva študentov na experimentovanie pri modifikácii množstva gradovaných nematematických úloh a umožňuje individuálne tempo v štúdiu rovnako dievčatám aj chlapcom, poskytuje študentom radosť z nadobúdania vedomostí. Vrelo odporúčam všetkým záujemcom o vniknutie do tajov programovania vo veku od 9 do 99 rokov, nielen stredoškólakom. Prvýkrát vo svojej dlhoročnej praxi učiteľa informatiky som sa stretla s metodickou príručkou k učebnici – pomôže nielen učiteľom, ale aj samoukom.

*RNDr. Eva Hanulová,
Gymnázium Jura Hronca, Bratislava*

Teším sa z každej novej učebnice, a tých vhodných na vyučovanie programovania na strednej škole je ako šafránu. Preto som sa jej veľmi potešila, no nielen ja, ale aj moji študenti. Veľmi oceňujem aj príručku pre učiteľa a testy, ktoré sú úplne perfektné - sú to ďalšie úlohy a námety na vyučovanie. Na druhý diel sa teším odvtedy, odkedy som sa dozvedela, že ho autori píšú.

*RNDr. Eva Stanková,
Gymnázium Ivana Horvátha, Bratislava*

Veľmi oceňujem učebnicu Programujeme v Pythone. Ponúka študentom prehľadným spôsobom dostať sa do programovania. Na hodinách môžu študenti s učebnicou pracovať aj samostatne vlastným tempom. Veľkým prínosom je aj príručka pre učiteľa, kde je prehľadne spracovaná celá metodika cez vzdelávacie plány až po prehľadne spracované učivo s riešenými úlohami a praktickými metodickými poznámkami ku každej úlohe. Hodnotný je aj súbor testov.

RNDr. Oľga Poliaková, SPŠ, Bardejov

Informatika je v porovnaní s inými predmetmi predmetom, ktorého veľká časť učiva sa z roka na rok mení dosť výrazným tempom, čo vyvíja na učiteľa tlak venovať veľkú časť prípravy samoštúdiu. Učiteľ informatiky na základe "podpory" štátu (chýbajúce, resp. neaktuálne učebnice, starý hardvér, ...) v takejto situácii zisťuje, že informatika v tejto spoločnosti nepatrí medzi kľúčové predmety. Učebnica mi ušetrila množstvo času a energie, ktorú som mohol venovať iným aktivitám v škole. Oceňujem príručku pre učiteľa s vyriešenými zadaniami z učebnice. Veľakrát som našiel v príručke iné riešenia ako tie moje, mohol som ich predstaviť žiakom, a tým ich myslenie, uvažovanie posunúť do iného, ďalšieho levelu. Myslím, že vďaka učebnici sa skvalitnila výučba informatiky na našej škole.

*Mgr. Peter Nemeč,
Spojená škola sv. Vincenta de Paul, Bratislava*

Učebnica Programujeme v Pythone je výbornou pomôckou pre učiteľov informatiky pri výučbe programovania. Metodika použitia grafických úloh v jazyku Python je vhodná na názorné osvojenie základov programovania. Prínosom učebnice je metodická príručka pre učiteľa a súbor testov. Učebnicu použijeme pri príprave budúcich učiteľov informatiky, ktorí ju môžu overiť v rámci svojej pedagogickej praxe na cvičných školách.

*Ing. Janka Majherová, PhD., Katedra informatiky,
Pedagogická fakulta, Katolícka univerzita v Ružomberku*

Obsah

Obsah

Úvod

1 Textové reťazce

1.1 Čísla vs. textové reťazce, načítanie vstupu

1.2 Prechádzanie znakmi reťazca, konštruovanie nového reťazca

1.3 Podreťazce, rezy

1.4 Znaký a ich kódy

1.5 Pracujeme s textovými reťazcami

Cézarova šifra

Náhodné zamiešanie znakov v textovom reťazci

1.6 Logické operácie a textové reťazce

Operácia in a pravdivostné hodnoty (boolean)

1.7 Ďalšie užitočné funkcie na prácu s textovými reťazcami

Niektoré metódy textových reťazcov

1.8 Formátovanie reťazcov

2 N-tice (tuple)

2.1 N-tice textových reťazcov - farby a slová

2.2 Body v rovine

2.3 N-tice ako parameter

2.4 Viacnásobné priradenie

3 Textové súbory

3.1 Zápis do textového súboru

3.2 Pridávanie riadkov do textového súboru

3.3 Čítanie z textového súboru

3.4 Iné spôsoby čítania textového súboru

Cyklus s podmienkou (while cyklus)

Konštrukcia with

3.5 Práca s viacerými textovými súbormi

4 Funkcie s návratovou hodnotou

5 Práca s viacerými údajmi (zoznam)

5.1 Hádzanie hracími kockami

Spoločné vlastnosti textových reťazcov, n-tíc a zoznamu

5.2 Užitočné funkcie a metódy na prácu so zoznamom

Vizualizácia údajov programu

5.3 Rozdelenie textového reťazca do zoznamu

5.4 Využívame zoznam v programoch

Need for Speed

Vyhľadávanie prvku s požadovanými vlastnosťami

SMS hlasovanie, metóda sort

Žaby, vymieňanie prvkov v zozname

6 Obrázky

6.1 Načítavanie a kreslenie obrázkov gif a png

6.2 Zoznam obrázkov

6.3 Vizualizácia predpovede počasia

Lambda funkcie

7 Matematické výpočty a geometria

8 Asociatívne polia (slovník - dictionary)

8.1 Vytvorenie slovníka, metódy na prácu so slovníkom

Šifrovanie náhodnou substitúciou

8.2 Frekvencia výskytov

Frekvencia znakov

Frekvencia slov v slovných hodnoteniach

8.3 Zoznam asociatívnych polí

9 Vlastnosti útvarov nakreslených v canvase

9.1 Zisťovanie a zmena nastavení útvarov

9.2 Značky útvarov a ich využitie na pamätanie si informácie

10 Upravujeme vzhľad aplikácií

Metódy zobrazenia widgetov a ich umiestnenie

Listbox

Posúvač

Textová plocha

Radiobutton

Checkbutton

10.1 Menu

10.2 Dialógové okná

11 Úlohy na opakovanie

Príkazovník

Použitá literatúra

Úvod

Sme radi, že elektronická učebnica informatiky pre stredné školy - Programujeme v Pythone si našla svojich priaznivcov. Prvý diel bol základným kurzom programovania, určeným všetkým študentom strednej školy, a napíňal Štátny vzdelávací program. Druhý diel je plynulým pokračovaním nielen pre študentov, ktorí sa chcú venovať programovaniu hlbšie alebo sa pripravujú na maturitu z informatiky, ale aj pre ďalších záujemcov či samoukov.

Učebnica spĺňa požiadavky na maturitu z informatiky podľa Cieľových požiadaviek na vedomosti a zručnosti maturantov z informatiky pre 1. oblasť s názvom Algoritmické riešenie problémov, ktoré sú platné od školského roka 2018/2019, ale aj v súčasnosti. Okrem tém (kapitol), ktoré sú priamo potrebné na maturitu, nájdete v učebnici aj témy, ktoré tento obsah dopĺňajú a pomôžu vám porozumieť dôležitým súvislostiam v Pythone, primerane na maturitnej úrovni (resp. na úrovni pre mierne pokročilých). Tieto témy vám pomôžu tvoriť komplexnejšie a zaujímavejšie programy.

V učebnici nájdete množstvo praktických a riešených príkladov, úlohy na precvičenie, ale aj otázky, ktoré vás majú nabádať na premýšľanie, objavovanie súvislostí, diskusiu v skupine, experimentovanie, ale aj hľadanie chýb a intuitívne hľadanie optimálneho riešenia.

Desiata kapitola je zameraná na upravovanie vzhľadu aplikácií - na ovládacie prvky z knižnice tkinter. Obsahuje praktické ukážky na použitie widgetov (ovládacích prvkov), nie je však nutné ovládať všetky tieto poznatky, skôr je to rýchla pomôcka na ich používanie. Aj preto táto kapitola neobsahuje úlohy a otázky.

Po preštudovaní si tejto učebnice a hodinách strávených praktickým programovaním by ste mali zvládnuť riešiť aj stredne náročné úlohy s použitím zložených dátových údajov a tiež by ste mali zvládnuť naprogramovať komplexnejší program väčšieho rozsahu, napríklad seminárnu prácu.

Aj keď v skutočnosti je táto e-kniha len postupnosťou núl a jednotiek, veríme, že v tejto postupnosti prevážia tie zaujímavé veci a učebnica sa stane vašou jednotkou pri štúdiu programovania.

Prajeme vám príjemné chvíle pri programovaní :)
autori

1 Textové reťazce

1.1 Čísla vs. textové reťazce, načítanie vstupu

V minulosti sme už pracovali so znakovými reťazcami a tvorili sme náhodné vety. V tejto kapitole sa pozrieme na textové reťazce podrobnejšie. Zatiaľ vieme:

- spájať reťazce operáciou `+`

```
slovo = 'Py'+ 'thon'
print(slovo)
```

- používať funkciu `int()`, ktorá z reťazca vytvorí číslo (ak obsahuje len znaky, ktoré môže obsahovať číslo). Používali sme ju pri zadávaní vstupu z `entry`, napríklad `od = int(entry1.get())`.

```
a = '1250'
b = int(a)
b = b+10
print(b)
```

Ak sme chceli vypísať `canvas.create_text(100, 200, text='Počet bodov: '+body)`, museli sme `body` (číslo) premeniť na text. Zatiaľ sme to vedeli vyriešiť tým, že sme zvlášť jedným príkazom vypísali text a ďalším príkazom číselnú hodnotu.

V takýchto prípadoch môžeme použiť funkciu `str()`, ktorá vytvorí z čísla textový reťazec.

Čiže `canvas.create_text(100, 200, text='Počet bodov: '+str(body))`.

```
body = 12
oznam = 'Tvoj počet bodov: '+str(body)
print(oznam)
```

Otázky:

1. Podľa čoho spoznáme, či je v premennej číslo alebo znakový reťazec?
2. Čo sa stane, ak použijeme operáciu `+` na text a číslo zároveň?
3. Čo sa stane, ak použijeme operáciu `*` na text a číslo zároveň? (Napríklad: `s = 'abc'*4`)

Ak chceme presne zistiť, či si v premennej pamätáme číslo alebo znakový reťazec, použijeme funkciu `type(meno_premennej)`. Funkcia zistí typ informácie, ktorú si pamätáme v premennej, čiže dátový typ.

```
>>> a = 12
>>> type(a)
<class 'int'>
>>>
```

Vidíme, že číslo je dátového typu `int`, čo je skratka anglického `integer`, čiže celé číslo.

```
>>> s = 'Python'
>>> type(s)
<class 'str'>
>>>
```

Premenná `s` je dátového typu `str`, čo je skratka anglického `string`, čiže reťazec.

Otázky:

4. Do premennej `oznam` sme priradili: `oznam = ''`. Akého dátového typu bude premenná `oznam`?
5. Vykonali sme tieto príkazy: `pocet = '12'` `vysledok = pocet*5`. Čo bude v premennej `vysledok` a akého dátového typu bude premenná `vysledok`?
6. Vykonali sme tieto príkazy: `pocet = '12'` `vysledok = pocet*5`. Čo sa zmení, ak by sme `pocet*5` nahradili napr. `pocet*-1` alebo `pocet*-5`?

Doposiaľ sme väčšinu úloh riešili v grafickom prostredí knižnice `tkinter`. Vstup sme načítali pomocou súčiastky (widgetu) `entry`. Mnohé programy ale nepotrebujú okno s grafickým rozhraním, postačuje im textový režim (okno shell). S takým programom sme sa už stretli v minulosti pri tvorení náhodných viet. V textovom režime môžeme programu zadávať vstup (od používateľa programu) funkciou `input`. Funkcia `input` má jeden parameter - v ňom je napísaný text, ktorý sa zobrazí používateľovi, a program čaká na zadanie vstupu, zadávanie ukončíme stlačením klávesu `enter`.

```
#program rozhovor
from random import *
meno = input('Ako sa voláš?')
print('Ahoj '+meno+' rád Ťa spoznávam :)')
roknarodenia = input(meno+', v ktorom roku si sa narodil?')
meno2 = choice(('Alena', 'Barbora', 'Eva', 'Sofia'))
print('Á spomínam si, v roku '+roknarodenia+' sa narodila aj '+meno2)
```

Otázky:

7. Čo bude robiť predchádzajúci program?
8. Akého dátového typu sú premenné: `meno`, `roknarodenia` a `meno2`?

Úlohy:

- 1 Upravte program `rozhovor` tak, aby nám počítač vypočítal a napísal náš vek (približne, keďže nezadáваме presný dátum narodenia).
- 2 Dopĺňte do programu `rozhovor` ďalšie otázky a odpovede.
- 3 Vytvorte program, ktorý vyzve používateľa na zadanie svojho mena a veku. Program následne používateľa pozdraví, napr. `Ahoj Filip` a vypíše, koľko rokov mu chýba do dovŕšenia 100 rokov.
- 4 Vytvorte program, v ktorom používateľ zadá svoje meno. Program potom vypíše „štítok“ v tvare:

```
*****
*           *
*  Andrea  *
*           *
*****
```

1.2 Prechádzanie znakmi reťazca, konštruovanie nového reťazca

Vypíšme jednotlivé znaky reťazca pod seba, každý znak na samostatný riadok. Na postupné prechádzanie reťazcom po jednotlivých znakoch môžeme použiť for cyklus:

```
for znak in 'Python':  
    print(znak)
```

alebo:

```
retazec = 'Python'  
for znak in retazec:  
    print(znak)
```

Otázky:

9. Čo bude robiť nasledujúci program? Čo vypíše na obrazovku?

```
retazec = 'Python'  
poradie = 0  
for znak in retazec:  
    poradie += 1      #je to isté ako zápis poradie = poradie+1  
    print(znak)  
print(poradie)
```

Funkcia `len (reťazec)` zistí dĺžku reťazca.

```
veta = input('Napíš nejakú vetu:')  
dlzka = len(veta)  
print('Počet znakov v tvojej vete je:', dlzka)
```

K jednotlivým znakom reťazca `retazec` vieme pristupovať pomocou zápisu `retazec[číslo_znaku]`. Znaký sú číslované (indexované) postupne od `0` po `len(s)-1`. Číslu, ktorým pristupujeme k znaku reťazca, hovoríme aj `index`. Napríklad:

```
retazec = 'Python'  
print(retazec[0]) # vypíše 'P'  
print(retazec[1]) # vypíše 'y'
```

Celý reťazec môžeme vypísať aj takto:

```
retazec = 'Python'  
for i in range(len(retazec)):  
    print(retazec[i])  
    print(i)
```

V shelli uvidíme:

```
===== RESTART: vypis.py =====  
P  
0  
Y  
1  
t  
2  
h  
3  
o  
4  
n  
5  
>>>
```

Prvý znak reťazca má index `0` a posledný `len(s)-1`. Často potrebujeme pristupovať k poslednému znaku. Na pristupovanie k znakom z opačnej strany reťazca (od konca) môžeme používať na indexovanie aj záporné čísla. Zápornými indexmi označujeme znaky od konca reťazca (viď obrázok).

P	y	t	h	o	n
0	1	2	3	4	5
-6	-5	-4	-3	-2	-1

Otázky:

10. Prečo posledný znak z reťazca `r` nezistíme zápisom `r[len(r)]`, ale zápisom `r[len(r)-1]`?
11. Čo sa stane, ak použijeme index, ktorý je mimo dĺžky reťazca? Napríklad: `retazec = 'Python'`
`retazec[10]` alebo `retazec[-10]`.
12. Vieme zistiť, vypísať konkrétny znak reťazca. Čo sa stane, ak sa ho pokúsime zmeniť? Napríklad zápisom `retazec[2] = 'i'`.

Reťazec (**string**) je v Pythone nemenný typ (**immutable**), to znamená, že sa nám nepodarí zmeniť znak, napr. `retazec[1] = '-'` (Python vtedy ohlásí chybu).

```
>>> retazec = 'Python'
>>> retazec[1] = '-'
Traceback (most recent call last):
  File "<pyshell#1>", line 1, in <module>
    retazec[1] = '-'
TypeError: 'str' object does not support item assignment
>>>
```

Namiesto zmeny reťazca musíme vždy konštruovať nový reťazec (môže mať aj rovnaké meno).

```
>>> meno = 'Hana'
>>> meno = 'J'+meno[1]+meno[2]+meno[3]
>>> meno
'Jana'
>>>
```

V nasledujúcom príklade vytvoríme nový reťazec, v ktorom všetky znaky `'o'` zameníme za hviezdičku `'*'`.

```
veta1 = 'Programujeme v Pythone'
veta2 = ''
for i in range(len(veta1)):
    if veta1[i] == 'o':
        veta2 = veta2+'*'
    else:
        veta2 = veta2+veta1[i]
print(veta1)
print(veta2)
```

Alebo môžeme niektoré znaky aj vynechať (napríklad medzery).

```
veta1 = 'Programujeme v Pythone'
veta2 = ''
for i in range(len(veta1)):
    if veta1[i] != ' ':
        veta2 = veta2+veta1[i]
print(veta1)
print(veta2)
```

Úlohy:

- 5 Vytvorte program, ktorému na vstupe zadáme dva rovnako dlhé reťazce a program z nich vytvorí jeden spojený reťazec, v ktorom sa striedajú znaky z prvého a druhého reťazca. Napríklad zadáme

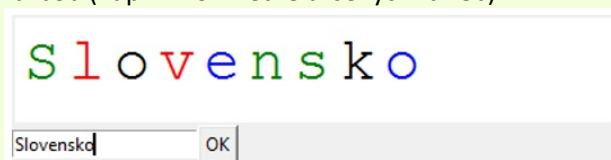
'Andrea' a 'Michal' a vznikne: 'AMnidcrheaal'.

6 Vytvorte program, ktorému na vstupe zadáme reťazec, ktorý vznikol prekladaním písmen z dvoch rôznych reťazcov (viď predchádzajúcu úlohu), a program nám vypíše dva pôvodné reťazce. Napríklad zadáme 'AMnidcrheaal' a vznikne 'Andrea' a 'Michal'.

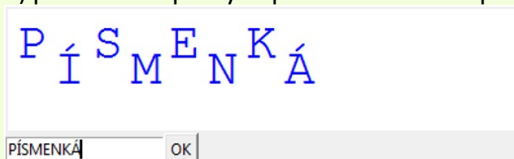
7 Vytvorte program, ktorému na vstupe zadáme dva rôzne dlhé reťazce a program z nich vytvorí jeden spojený reťazec, v ktorom sa striedajú znaky z prvého a druhého reťazca. Keďže jeden z nich je kratší, na konci sú už len prilepené zostávajúce znaky z dlhšieho reťazca. Napríklad zadáme 'Diana' a 'Ferdinand' a vznikne: 'DFiearndainand'.

8 Vytvorte program, v ktorom používateľ zadá vetu v slovenčine ukončenú interpunkčným znamienkom (bodkou, otáznikom alebo výkričníkom). Program na základe ukončovacieho znamienka vypíše, či ide o vetu opytovaciu, rozkazovaciu alebo oznamovaciu.

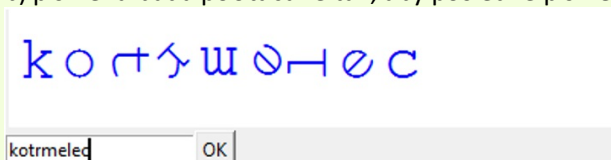
9 Vytvorte program, v ktorom používateľ zadá slovo. Toto slovo sa do grafického plátna vypíše v podobe „reklamného nápisu“, kde každý znak zadaného slova bude vypísaný náhodne vybranou farbou (napr. z konkrétne určených farieb).



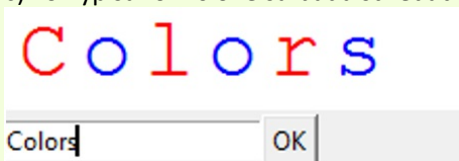
10 Vytvorte program, v ktorom používateľ zadá slovo. Toto slovo sa do grafického plátna vypíše tak, že:
a) písmená na párných pozíciách budú napísané nižšie ako na nepárných pozíciách,



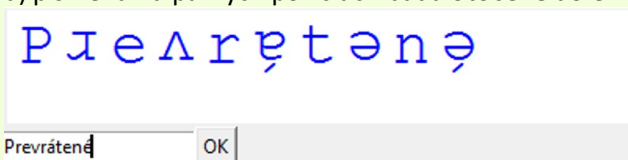
b) písmená budú pootáčané tak, aby posledné písmeno bolo opäť natočené „normálne“,



c) vo vypísanom slove sa budú striedať dve farby,



d) písmená na párných pozíciách budú otočené dole hlavou.



11 Vytvorte program, v ktorom používateľ zadá slovo.
a) Toto slovo sa vypíše postupne po jednotlivých znakoch tak, že po každej sekunde sa objaví ďalšie písmeno.

Informa

Informat

Informati

Informatika OK

Informatika OK

Informatika OK

b) Jednotlivé písmená sa postupne po sekundách pridávajú k výslednému slovu tak, že prichádzajú sprava.

c) Keď už bude vypísané celé slovo, po chvíli nápis „zhasne“ a začne sa vypisovať odznova.

12

Napište program, v ktorom používateľ zadá vetu. Program vypíše:

a) počet slov vo vete,

b) dĺžku a poradové číslo najdlhšieho slova vo vete – ak ich je viac (s rovnakou dĺžkou), stačí vypísať jedno,

c) najdlhšie slovo vo vete – ak ich je viac, vypíšte všetky.

13

Napište program, v ktorom používateľ zadá slovo. Program ho potom vypíše v takomto tvare:

A

H

O

J

Otázky:

13. V jednej z úloh sme navzájom spájali (prekladali znaky z jedného reťazca a druhého reťazca) dva nerovnako dlhé reťazce. Vedeli by sme výsledný reťazec rozdeliť na pôvodné reťazce? Svoju odpoveď zdôvodnite. Ako by sme mohli zlepšiť spôsob prekladania znakov tak, aby sme vedeli jednoducho rozdeliť aj reťazec, ktorý vznikol z dvoch rôzne dlhých reťazcov?

1.3 Podreťazce, rezy

Nové reťazce môžeme vytvárať aj pomocou rezov - indexovaním znakov s viacerými indexmi - **reťazec [odkiaľ : pokiaľ]**. Znak s indexom **pokiaľ** sa už nebude nachádzať vo výsledku. Vždy sem musíme písať index o jeden viac (podobne ako sme dávali hranice vo for cykle pre **range**).

```
>>> s = 'Programujeme v Pythone'
>>> s[3:7]
'gram'
>>>
```

P	r	o	g	r	a	m	u	j	e	m	e		v		P	y	t	h	o	n	e
0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21
-22	-21	-20	-19	-18	-17	-16	-15	-14	-13	-12	-11	-10	-9	-8	-7	-6	-5	-4	-3	-2	-1

```
>>> s = 'Programujeme v Pythone'
>>> s[15:21]
'Python'
>>> s[15:22]
'Pythone'
>>>
```

Rezy môžeme použiť aj pri zmene formátu dátumu. Napríklad dátum 31. 12. 2017 chcem prepísať do tvaru rrrr-mm-dd, teda 2017-12-31:

```
>>> datum = '31-12-2017'
>>> datum_en = datum[6:10]+'-'+datum[3:5]+'-'+datum[0:2]
```

```
>>> datum_en
'2017-12-31'
>>>
```

Otázky:

14. Ktorý formát dátumu je vhodnejší pre informatiku? Prečo?
15. V programoch sa často používajú aj takéto zápisy. Vyskúšajte ich a zistite, čo znamenajú.
 - a) `s[:5]` `s[:2]`
 - b) `s[2:]` `s[3:]`
 - c) `s[:]`
16. Ktoré z týchto zápisov budú fungovať a ktoré vrátia neprázdny reťazec?
`s[-5:-2]` `s[5:2]` `s[-2:-5]` `s[-2:]` `s[:-2]`

V reze môžeme určiť aj krok indexov (podobne ako v príkaze `range()`, ktorý sme používali vo for cykle). Krok určuje, o koľko sa budeme v indexoch posúvať od prvého uvedeného indexu. Ak krok nevedieme, štandardne je krok 1.

```
>>> s = 'Programujeme v Pythone'
>>> s[0:10:2]
'Pormj'
>>> s[0:22:2]
'Pormjm yhn'
>>> s[-1:-10:-1]
'enohtyP v'
>>> s[::-1]
'enohtyP v emejumargorP'
>>> s[::-1]
'enohtyP v emejumargorP'
>>>
```

Rez okrem konkrétnych čísel môže obsahovať aj premenné alebo nejaký výraz, ktorého výsledkom je číslo použiteľné v reze. Môžeme ich vytvárať aj takto:

```
s = 'Python'
for i in range(len(s)):
    print(s[:i+1])
```

```
P
Py
Pyt
Pyth
Pytho
Python
```

```
s = 'Python'
for i in range(len(s)):
    ns = '-'*i+s[:i+1]
    print(ns)
```

```
P
-Py
--Pyt
---Pyth
----Pytho
-----Python
```

Pomocou rezov vytvoríme nový reťazec, v ktorom môžeme nahradiť niektorý znak alebo viacero znakov. Napríklad:

```
>>>s = 'Python'
>>>s = s[:2]+'T'+s[3:]
>>>s
'PyThon'
>>>
```

Úlohy:

14 Na vstupe zadáme reťazec, ktorý vznikol prekladaním písmen z dvoch rôznych reťazcov s rovnakou dĺžkou. Napríklad zadáme 'AMnidcrheaal'. Pomocou rezov (bez použitia for cyklu) vytvorte zo vstupu prvé a druhé slovo, teda: 'Andrea' a 'Michal'.

15 Vytvorte program, ktorý zo vstupného reťazca nakreslí (ukážka je pre vstupný reťazec 'Python'):

```
.....P
.....Py
....Pyt
...Pyth
..Pytho
.Python
```

16 Vytvorte program, ktorý zo vstupného reťazca nakreslí (ukážka je pre vstupný reťazec 'Python'):

```
.....PP.....
.....yPPy.....
....tyPPyt....
...htyPPyth...
..ohtyPPytho..
.nohtyPPython.
```

17 Napíšte program, v ktorom používateľ zadá platnú e-mailovú adresu. Program následne vypíše:

- doménu najvyššej úrovne (TLD) aj s bodkou,
- adresu mailového servera,
- meno používateľa,
- zoznam všetkých domén v poradí od domény prvej úrovne po doménu najnižšej úrovne.

```
Zadajte email: michal.velky@mail.pythonsoftware.net
TLD: .net
Server: mail.pythonsoftware.net
User: michal.velky
Domény:
Doména 1. úrovne je: net
Doména 2. úrovne je: pythonsoftware
Doména 3. úrovne je: mail
```

18 Napíšte program, v ktorom používateľ zadá platné rodné číslo z dvadsiateho storočia. Program následne vypíše:

- dátum narodenia v tvare deň.mesiac.rok,
- pohlavie.

Ukážka:

```
Rodné číslo: 806202/5675
Dátum narodenia: 2.12.1980
Pohlavie: Žena
```

(Pozn. Ženy majú v rodnom čísle k číslu mesiaca prirátané číslo 50)

