

lecture 2 - Exercises



Programming in geoinformatics
Autumn 2015

1.CONDITIONS

- zeptejte se uživatele na zeměpisnou šířku a délku pomocí dvou input()
- pokud je zadaná hodnota mimo reálný rozsah, změníme ji na nejbližší povolenou
- příklad:
 - šířka (rozsah -90 až +90): 220 -> 90
 - délka (rozsah -180 až 180): -253 -> -180

2. CONDITIONS

- definujte proměnné `no_people` a `no_cars` a určete jim číselné hodnoty
- v případě, že se lidé do aut nevejdou, vypište zprávu (`print()`) o stavu
- předpokládáme, že do auta se vleze 5 lidí

3. CONDITIONS

- pomocí příkazu if napište kód, který podle hodnoty proměnné elevation vypíše stupeň výškové členitosti
- stupně výškové členitosti
 - 0 - 300 -> nížiny
 - 300 - 800 -> nízké vysočiny
 - 800 - 1500 -> střední vysočiny
 - nad 1500 -> vysoké vysočiny

4. CONDITIONS

- máme proměnné “eggs” (int), “milk” (int), “flour” (bool), a “oven” (bool)
- k jídlu si můžeme dát následující:
 - koláč (3 eggs, 2 milk, flour, oven)
 - palačinky (2 eggs, 1 milk, flour)
 - omeleta (2 eggs, 1 milk)
 - sklenice mléka (1 milk)
- I) pro náhodně zvolené hodnoty, vypište (print()), jaké “nejkomplexnější” jídlo si můžeme dát
- II) vypište taky, kolik mlíka a kolik vajec nám zůstane

5. CONDITIONS

- vytvoříme si hru “kámen, nůžky, papír”.
 - I) Pro získání volby obou hráčů využijeme metody `input()`, v kterých informujeme hráče o následujících kódech pro dané volby:
 - 1 = kámen, 2 = nůžky, 3 = papír
- následně zkontrolujeme, jestli jsou vstupy v definovaném rozsahu (1, 2 nebo 3)
- II) do konzoly vypíšeme vítěze hry, případně informujeme o remíze

6. LISTS

- máme proměnnou `numbers` typu `list`, která obsahuje náhodné číselné hodnoty typu `int`
- `list` má vždy 4 nebo 5 elementů
- vytvořte novou proměnnou `sum`, která bude mít hodnotu součtu všech hodnot v `listu`
- příklad:
 - `numbers = [2, 1, 4, 5] -> sum = 12`
- použijte funkci `len()` a jednotlivé elementy získajte pomocí notace `[]`

7. LISTS

- zeptejte se uživatele (`input()`) na 4 číselné hodnoty `value1` až `value4`
- vytvořte list `filtered_values`, kde přidejte jenom hodnoty `value1` až `value4`, které spňují následující podmínky:
 - jsou dělitelné číslem 2
 - jsou větší než 10

8. LISTS

- máme danou definici topologií pomocí datového typu list:
 - bod - jednorozměrný list se dvěma hodnotami lat a lon, například [49,18]
 - linie - dvourozměrný list, kde každý element prvního listu bude znovu list definující hodnoty lat a lon (= list bodů). Například linie o dvou bodech bude [[49,18], [50,19]].
 - multi-linie - trojrozměrný list, kde první list bude obsahovat listy linií (= list linií). Například [[[49,18], [48,17]], [[45,17], [48,19]]].
- máme definované body $a = [47,18]$; $b = [49,19]$; $c = [48,18]$; $d = [44, 19]$; $e = [42, 17]$
- I) vytvoříme tři linie: $l1 = [a, d]$; $l2 = [c, d, b]$; $l3 = [e, a, c]$
- II) vytvoříme multilinii, která bude obsahovat linie $l1$, $l2$ a $l3$

9. CYCLES

- máme definovaný list hats = ["red", "black", "blue", "yellow"] a ids = [2, 7, 15, 22, 25, 34]
- I) definujte list ids_new, kde budou čísla sudá o 1 větší než původní, lichá o 1 menší než původní (například: 3 -> 2; 8 -> 9)
- II) vypište všechny kombinace listu ids_new a hats jako "person with id 5 has a red hat"

10. CYCLES

- máme karetní hru, ve které mají jednotlivé karty dané hodnoty:
 - 7 - 7 bodů; 8 - 8 bodů; 9 - 9 bodů; 10 - 10 bodů; J, Q, K mají 5 bodů a A 15 bodů
 - barva karty násobí počet bodů (srdce - 4x, káry - 3x, piky - 2x a kříže - 1x)
- I) napište kód, který v rámci vnořeného cyklu, vypíše (`print()`) jednotlivé karty v balíčku
- II) u každé karty vypočtete její hodnotu a pokud je hodnota větší než 20, vypíšte kartu do konzoly (`print()`)

HOMEWORK 1 - boundaries

- hranice krajiny A a krajiny B jsou definovány neuspořádanými listy s hraničními body
 - `countryA_bounds = ["A", "W", "X", "C", "M", "K"]`
 - `countryB_bounds = ["W", "O", "L", "A", "K", "U", "S"]`
- vytvořte nový list `countryAB_bounds`, do kterého uložíte hraniční body pro hypotetickou krajinu, která by vznikla spojením krajiny A a B
- hint: hraniční body, které se opakují v obou listech nebudou aktuální pro novou krajinu

HOMEWORK 2 - country statistics

- máme list s názvy krajin countries = ["Greece", "Czech Republic", "Portugal", "Hungary", "Sweden"], list s počtem obyvatel v 1000 population = [10769, 10535, 10311, 9835, 9794] a průměrný roční růst (v %) growth = [-0.11, -0.06, -0.56, -0.28, 1.03]
- definujte list population_estimation, kde pro každou krajinu vypočtete odhadovaný počet obyvatel o 100 let
- můžete počítat se stejným přírůstkem/úbytkem pro celé období 100 let (není potřeba 100letý součet počítat opakovaně vždy po jednom roce, ale hodnotu z prvního roku stačí vynásobit 100)
- předpokládáme, že hodnoty pro jednotlivé krajiny jsou v rámci listů vždy na stejném indexu (Greece - index 0, Czech Republic - index 1)

HOMEWORK 3 - minimum bounding rectangle

- definujte minimální obdélník, jehož strany jsou rovnoběžné s osami souřadnicového systému => je definovaný 4 body, které leží na souřadnicích maximálních nebo minimálních hodnot zeměpisné šířky a délky daného útvaru
- hranice Kambodže jsou definované pomocí těchto souřadnic
 - [[103.49728,10.632555], [103.09069,11.153661], [102.584932,12.186595], [102.348099,13.394247], [102.988422,14.225721], [104.281418,14.416743], [105.218777,14.273212], [106.043946,13.881091], [106.496373,14.570584], [107.382727,14.202441], [107.614548,13.535531], [107.491403,12.337206], [105.810524,11.567615], [106.24967,10.961812], [105.199915,10.88931], [104.334335,10.486544], [103.49728,10.632555]]
- vypočítejte souřadnice pro minimum bounding rectangle polygonu Kambodže
- hint: pro prohledání souřadnic použijte for cyklus a hledejte v listu maximální a