

Úloha 1 – Klasifikátor studentů

24 bodů

Úkolem je vytvořit **konzolovou** aplikaci v jazyce Python, která zobrazí přehlednou tabulku známek a aritmetických průměrů všech studentů.

a) Hlavní menu

Po spuštění programu se vypíše do konzole nabídka hlavního menu, které obsahuje tři možnosti přesně podle ukázkového výstupu. Následně program umožní uživateli zadat jedno z písmen a tím ovlivnit, kterou částí bude program pokračovat. Po skončení dané části se program znovu vrátí do hlavního menu a znovu se vypíše nabídka. Pokud uživatel zadá cokoliv jiného než jedno z písmen v nabídce, program vypíše výzvu znovu. Program končí po zadání písmene K.

Využívané znalosti: výstup do konzole, vstup z klávesnice

6 bodů

b) Přidání studenta

Po zadání písmene S program přidá do tabulky jeden řádek, tedy nového studenta. Za tímto účelem se uživatele zeptá na jeho jméno. Pokud uživatel nic nezadá, výzva se opakuje znovu. Následně se program zeptá na známky daného studenta ze všech předmětů v tabulce. Pokud zatím v tabulce nejsou žádné předměty, žádné známky se zadávat nebudou. Zadávání známek probíhá postupně podle pořadí předmětů v tabulce. Program vypíše výzvu se zkratkou konkrétního předmětu a očekává zadání celého čísla od 1 do 5. Pokud vstup uživatele neodpovídá těmto pravidlům, výzva se opakuje znovu.

Využívané znalosti: dvourozměrné pole, cyklus FOR

6 bodů

c) Přidání předmětu

Po zadání písmene P program přidá do tabulky jeden sloupec, tedy nový předmět. Za tímto účelem se uživatele zeptá na jeho zkratku. Pokud uživatel nic nezadá, výzva se opakuje znovu. Následně se program zeptá na známky z daného předmětu pro všechny studenty v tabulce. Pokud zatím nejsou v tabulce žádní studenti, žádné známky se zadávat nebudou. Zadávání známek probíhá podle pořadí studentů v tabulce za dodržení stejných pravidel jako v předchozí části.

Využívané znalosti: dvourozměrné pole, cyklus FOR

6 bodů

d) Zobrazení tabulky

Po zadání písmene T program vytiskne do konzole tabulku se všemi dosud zadanými známkami. Tabulka obsahuje záhlaví, v prvním sloupci je uvedeno jméno studenta, v posledním sloupci jeho aritmetický průměr. Zbývající sloupce odpovídají jednotlivým předmětům, přičemž v záhlaví jsou uvedeny jejich zkratky a ve zbývajících buňkách tabulky jsou známky jednotlivých studentů. Řádky tabulky jsou graficky odděleny pomocí pomlček, sloupce tabulky jsou graficky odděleny pomocí svislic.

Využívané znalosti: výstup do konzole ve formě tabulky

6 bodů

Celostátní kolo soutěže Mladý programátor 2021, kategorie D - Python

Níže najdete ukázkový výstup vašeho programu:

```
Co chcete udělat? (Student/Předmět/Tabulka) S
Zadejte jméno studenta: Tomáš
```

```
Co chcete udělat? (Student/Předmět/Tabulka) P
Zadejte zkratku předmětu: PRG
Zadejte známku studenta: Tomáš: 1
```

```
Co chcete udělat? (Student/Předmět/Tabulka) P
Zadejte zkratku předmětu: MAT
Zadejte známku studenta: Tomáš: 2
```

```
Co chcete udělat? (Student/Předmět/Tabulka) S
Zadejte jméno studenta: Marek
Zadejte známku z předmětu: PRG: 1
Zadejte známku z předmětu: MAT: 1
```

```
Co chcete udělat? (Student/Předmět/Tabulka) P
Zadejte zkratku předmětu: FYZ
Zadejte známku studenta: Tomáš: 1
Zadejte známku studenta: Marek: 3
```

```
Co chcete udělat? (Student/Předmět/Tabulka) S
Zadejte jméno studenta: Zuzka
Zadejte známku z předmětu: PRG: 1
Zadejte známku z předmětu: MAT: 1
Zadejte známku z předmětu: FYZ: 1
```

```
Co chcete udělat? (Student/Předmět/Tabulka) T
/-----\
| student | PRG | MAT | FYZ | průměr |
|-----|
| Tomáš   | 1   | 2   | 1   | 1,33   |
| Marek   | 1   | 1   | 3   | 1,67   |
| Zuzka   | 1   | 2   | 2   | 1,67   |
\-----/
```

Autorem úlohy je Tomáš Veselý.

Úloha 2 – Generátor autentizačních údajů

24 bodů

Úkolem je vytvořit **konzolovou** aplikaci v jazyce Python, která náhodně vygeneruje uživatelské jméno a heslo na základě zadaných osobních údajů.

a) Zadání osobních údajů

Po spuštění programu se vypíše úvodní hláška a program se bude postupně dotazovat uživatele na osobní údaje, které v následujících částech využije pro vygenerování přihlašovacích údajů. Pokud uživatel odmítne zadat některé údaje, program pokračuje dál a výzvu znovu neopakuje, musí to však v dalších částech zohlednit. V případě, že uživatel zadá méně než dva osobní údaje, program vypíše chybovou hlášku a skončí.

Program postupně pokládá následující otázky:

- Jaké je vaše jméno?
- Jaké je vaše příjmení?
- V jakém roce jste se narodili?
- Jak se jmenuje vaše matka?
- Jak se jmenuje váš otec?
- Jak se jmenuje váš sourozenec?

Využívané znalosti: výstup do konzole, vstup z klávesnice

6 bodů

b) Generování uživatelského jména

Program nejprve náhodně vybere jeden z formátů, který využije pro vytvoření uživatelského jména. Pro všechny formáty platí, že vygenerované uživatelské jméno může obsahovat pouze povolené znaky. Pokud některý ze zadaných osobních údajů obsahuje také jiné znaky, tyto znaky musí být vynechány. Mezi povolené znaky patří malá písmena bez diakritiky, číslice, tečka, pomlčka a podtržítka.

Znaky s diakritikou mohou být vynechány, ale lepším řešením je jejich nahrazení za odpovídající znaky bez diakritiky. *Takové řešení je složitější, ale bude hodnoceno lépe.*

Program podporuje následující formáty uživatelského jména:

- Celé jméno, náhodný symbol, celé příjmení (*jan_novak, marie.novakova*)
- Prvních pět písmen příjmení, první tři písmena jména (*novakjan*)
- První písmeno jména, symbol, celé příjmení, symbol, rok narození (*j.novak.1988*)
- První tři písmena jména, první tři písmena příjmení, zkrácený rok narození (*jannov88*)

Využívané znalosti: náhodné číslo, práce s řetězcem

8 bodů

c) Generování hesla

Program vybere náhodně jeden z formátů pro generování hesla podobně jako v předchozí části. Na rozdíl od uživatelského jména nemá vygenerované heslo žádné omezení z hlediska použitých znaků. Kromě písmen a číslic tedy může obsahovat libovolné symboly včetně mezer.

Program podporuje následující formáty hesla:

- Jméno otce, náhodný symbol, jméno matky (*Jirka&Jana, Karel-Klára*)
- Jméno sourozence, náhodný symbol, rok narození (*Marie/1988*)
- Rok narození, symbol, křestní jméno, symbol, jméno sourozence (*1988+Jan+Marie*)
- Iniciály, symbol, rok narození, symbol, jméno jednoho z rodičů (*JN*1988*Jiří*)

Využívané znalosti: náhodné číslo, práce s řetězcem

6 bodů

Celostátní kolo soutěže Mladý programátor 2021, kategorie D - Python

d) Uložení přihlašovacích údajů

V poslední části program uloží vygenerované přihlašovací údaje do textového souboru s názvem **autentizace.txt**. Pokud takový soubor ve složce s programem neexistuje, program ho nejprve vytvoří. Následně do něj na jeden řádek zapíše uživatelské jméno a heslo oddělené dvojtečkou. Pokud soubor již existoval, nové přihlašovací údaje se uloží na další řádek na konci souboru.

Využívané znalosti: zápis do textového souboru

4 body

Níže najdete ukázkový výstup programu při zadání všech údajů:

```
Potřebujeme se vás zeptat na pár věcí:  
Jaké je vaše jméno? Jan  
Jaké je vaše příjmení? Novák  
V jakém roce jste se narodili? 1988  
Jak se jmenuje vaše matka? Jana  
Jak se jmenuje váš otec? Jiří  
Jak se jmenuje váš sourozenec? Marie  
  
K přihlášení použijte následující údaje:  
Uživatelské jméno: jan_novak  
Heslo: 1988+Jan+Marie  
  
Stejné údaje jsme pro vás uložili do souboru.  
Děkujeme za použití našeho generátoru!
```

Níže najdete ukázkový výstup programu při zadání části údajů:

```
Potřebujeme se vás zeptat na pár věcí:  
Jaké je vaše jméno? Marie  
Jaké je vaše příjmení? Nováková  
V jakém roce jste se narodili?  
Jak se jmenuje vaše matka? Jana  
Jak se jmenuje váš otec? Jirka  
Jak se jmenuje váš sourozenec?  
  
K přihlášení použijte následující údaje:  
Uživatelské jméno: marie.novakova  
Heslo: Jirka&Jana  
  
Stejné údaje jsme pro vás uložili do souboru.  
Děkujeme za použití našeho generátoru!
```

Níže najdete ukázkový soubor, který vzniknul na základě dvou předchozích vstupů:

```
jan_novak:1988+Jan+Marie  
marie.novakova:Jirka&Jana
```

Autorem úlohy je Tomáš Veselý.

Úloha 3 – Animátor objektů

24 bodů

Úkolem je vytvořit **grafickou** aplikaci v jazyce Python s využitím knihovny Pygame, ve které se bude pohybovat několik barevných geometrických obrazců.

a) Příprava scény

Po spuštění programu se vytvoří okno o šířce 800 pixelů a výšce 600 pixelů. Na pozadí okna je barva RGB (220, 220, 240) a v okně jsou ve směru zleva doprava vykresleny tři objekty:

- Žlutý čtverec – barva RGB (240, 240, 20)
- Oranžový kruh – barva RGB (240, 160, 20)
- Červený čtverec – barva RGB (240, 40, 20)

Všechny tři objekty mají šířku a výšku shodně 100 pixelů a navíc černý obrys o tloušťce 5 pixelů. Tloušťka obrysu je součástí celkových rozměrů každého objektu.

Využívané znalosti: vytvoření okna, vykreslení čtverce a kruhu

6 bodů

b) Pohyb kruhu

Oranžový kruh se ihned po spuštění programu začne pohybovat po obrazovce směrem šikmo doprava dolů pod úhlem 45 stupňů. Za každých 100 milisekund se kruh pohne přesně o 50 pixelů ve svém aktuálním směru pohybu. To znamená, že pokud bychom nepočítali s odrazy, každou sekundu kruh urazí dráhu 500 pixelů. Úkolem je vytvořit plynulou animaci, polohu kruhu tedy bude potřeba aktualizovat častěji než desetkrát za sekundu.

Jakmile se kruh dotkne okraje obrazovky, změní se směr jeho pohybu a kruh se odrazí pod stejným úhlem. Pokud se původně pohyboval směrem šikmo doprava dolů, po odrazu se bude pohybovat směrem šikmo doprava nahoru a podobně.

Využívané znalosti: celočíselné proměnné, mazání obrazovky

8 bodů

c) Pohyb čtverců

Žlutý a červený čtverec budou reagovat na stisknutí příslušné klávesy na klávesnici. Začnou se pohybovat právě ve chvíli, kdy klávesu stiskneme a držíme, a přestanou se pohybovat ve chvíli, kdy danou klávesu zase pustíme. Za každých 100 milisekund se čtverec pohne přesně o 50 pixelů ve směru pohybu daném stisknutou klávesou. To znamená, že přesně za jednu sekundu se čtverec přesune od spodního okraje obrazovky až k hornímu okraji obrazovky a naopak.

Čtverce se mohou po obrazovce pohybovat pouze vertikálně pomocí čtyř kláves:

- Klávesa **W** posouvá žlutý čtverec směrem nahoru, klávesa **S** směrem dolů
- Klávesa **I** posouvá červený čtverec směrem nahoru, klávesa **K** směrem dolů

Jakmile se čtverec dotkne okraje obrazovky, zastaví se a v pohybu daným směrem není možné dál pokračovat ani při opětovném stisknutí příslušné klávesy.

Využívané znalosti: čtení událostí, kontrola stisknutých kláves

6 bodů

d) Zastavení animace

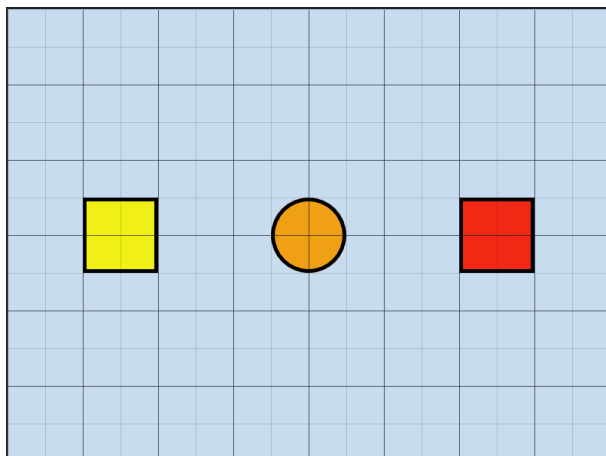
Upravte program tak, aby po stisknutí klávesy P došlo k zastavení všech animací, podobně jako při pozastavení počítačové hry. Tedy pokud je program v tomto stavu, přestane se pohybovat kruh a není možný ani pohyb čtverců pomocí klávesnice. Po opětovném stisknutí klávesy P se program vrátí do normálního stavu a animace pokračují tam, kde skončily.

Využívané znalosti: jednoduché podmínky

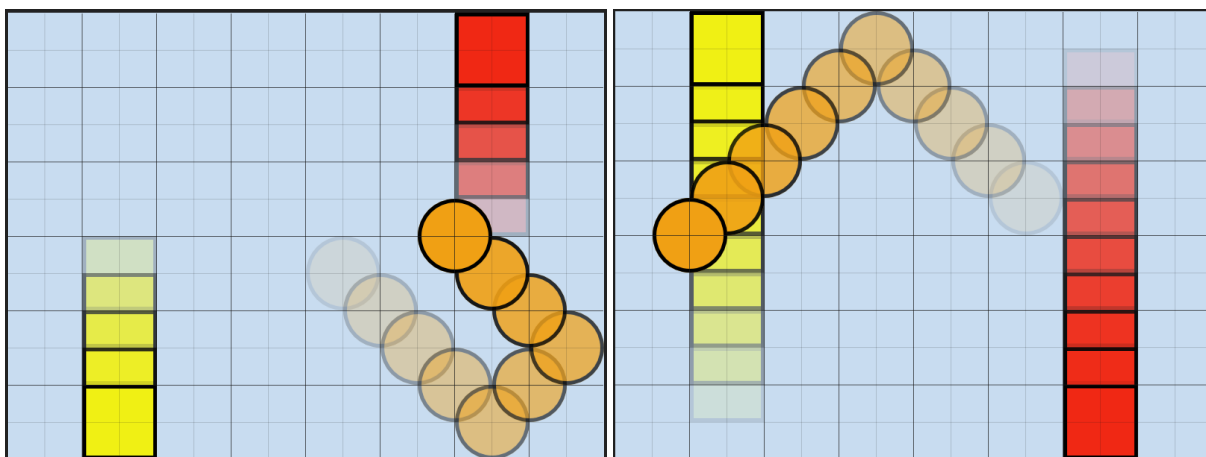
4 body

Celostátní kolo soutěže Mladý programátor 2021, kategorie D - Python

Po spuštění programu by měla obrazovka vypadat podle následujícího obrázku. Mřížka je vykreslena pouze pro vaši lepší orientaci a není součástí programu.



Průběh animace je znázorněn na dvou následujících obrázcích. Stopa objektů je vykreslena pouze pro znázornění animace a není součástí programu.



- První obrázek odpovídá první sekundě běhu programu, přičemž jsou po celou dobu drženy klávesy **S** a **I**. Čtverce se přitom pohybují jen do té doby, než narazí na okraj obrazovky.
- Druhý obrázek odpovídá druhé sekundě běhu programu, přičemž jsou po celou dobu drženy klávesy **W** a **K**. Oba čtverce za tu dobu dorazí z jednoho konce obrazovky na druhý.

Autorem úlohy je Tomáš Veselý.