

Cvičení 4

Úloha 1: (2 body)

Vytvořte třídu **ParkingSpace**, která představuje jedno parkovací místo na parkovišti. Třída obsahuje konstruktor, který má povinný argument **code** a nepovinný argument **car** s výchozí hodnotou **None**.

Atributy třídy:

- **code** – kódové označení parkovacího místa ve tvaru **'B15'**, kde písmeno označuje řadu (začínající od A) a číslo pozici parkovacího místa v řadě
- **car** – SPZ auta, které má dané místo rezervované (má hodnotu **None**, pokud dané místo není rezervované)

Metody třídy:

- **is_reserved** – metoda vrátí **True**, pokud je dané místo rezervované, jinak **False**
- **reserve** – metoda nastaví proměnnou **car** dle zadané hodnoty

Úloha 2: (3 bod)

Vytvořte třídu **ParkingLot**, která představuje parkoviště. Třída obsahuje konstruktor, který má dva povinné argumenty – **rows** a **cols**, které určují velikost parkoviště.

Atributy třídy:

- **rows** – počet řad na parkovišti
- **cols** – počet míst v jedné řadě
- **parking** – slovník, reprezentuje všechna parkovací místa
 - klíč: kód parkovacího místa ve tvaru **'B15'**
 - hodnota: instance třídy **ParkingSpace**

Metody třídy:

- **generate** – nemá žádné argumenty a vyplní proměnnou **parking** podle hodnot **rows** a **cols**
- **reserve** – má dva povinné argumenty (**code** a **car**) a rezervuje dané auto na příslušném parkovacím místě pomocí metody **reserve** ze třídy **ParkingSpace**
- **print_map** – vypíše mapu parkoviště (kódy parkovacích míst) ve tvaru:

```
A0 A1 A2 A3 A4 A5 A6 A7 A8 A9
B0 B1 B2 B3 B4 B5 B6 B7 B8 B9
C0 C1 C2 C3 C4 C5 C6 C7 C8 C9
D0 D1 D2 D3 D4 D5 D6 D7 D8 D9
```

- **print_reserved** – vypíše mapu parkoviště podle obsazenosti (volné místo jako **'..'** a rezervované jako **'XX'**)

```
.. .. .. XX .. .. .. .. ..
.. .. .. .. .. .. XX .. ..
.. .. XX .. XX .. .. .. ..
.. .. .. .. .. .. .. XX XX ..
```

- **get_code** – statická metoda, která dle zadaných hodnot **row** a **col** vrátí kód ve tvaru **'B15'**

Hint: Pro vytvoření kódu parkovacího místa použijte funkci **chr**, která vrátí znak podle zadaného ASCII kódu a funkci **ord**, která vrátí ASCII kód zadaného znaku.

```
chr(65) # 'A'  
ord('A') # 65
```

Hint: Pro generování sekvence čísel použijte funkci **range**.

```
range(1, 5) # [1, 2, 3, 4]
```

Hint: Statickou metodu vytvoříte tak, že nad její definicí napíšete **@staticmethod**. Nezapomeňte také, že se statická metoda volá na třídě a nikoli na instanci třídy.

```
@staticmethod  
def my_method():
```

Hint: Pro vytvoření řetězce z listu použijte metodu **join** na řetězci, který je oddělovačem.

```
",".join(['A', 'h', 'o', 'j']) # 'A,h,o,j'
```

Hint: Klíčové slovo **print** ve výchozím stavu automaticky přidá odřádkování. Pokud chcete toto chování změnit a místo nového řádku chcete vypisovat mezeru, přidejte na konec čárku.

Případně můžete použít funkci **print** s argumentem **end= ' '** (je dostupná od verze 2.6).

Kód:

```
print 'abc'  
print 'def'
```

Výstup:

```
abc  
def
```

Kód:

```
print 'abc',  
print 'def'
```

Výstup:

```
abc def
```

Kód:

```
from __future__ import print_function # není nutné pro Python 3.x  
  
print('abc', end = ' ')  
print('def')
```

Výstup:

```
abc def
```