

## Zadanie skúšky 8.2.2016

### Pet detective

V malom mestečku v niektorých domoch chovajú domáce zvieratá. Jedného dňa dostal zvierací detektív informácie o tom, kde v meste videli stratené zvieratá a kam ich treba odviezť. Nasadol do svojho auta, obchádzal uličky mesta, zbieral do auta nájdené zvieratá a pri správnych domoch ich opäť vykladal. Lenže jeho auto nemalo neobmedzenú kapacitu, takže, keď už bolo plné, aj keď prešiel okolo strateného zvieratá na ceste, nenaložil ho, ale išiel ďalej. Auto budeme riadiť šípkami ('l' pre vľavo, 'p' pre vpravo, 'h' pre hore, 'd' pre dolu), a keď sa v danom smere nebude môcť pohnúť, tak tento príkaz odignoruje.

Zadefinujte triedu PetDetective:

```
class PetDetective:
    def __init__(self, meno_suboru):
        self.pole = []
        ...

    def __repr__(self):
        ...

    def kapacita(self, kapa):
        ...

    def zisti(self):
        ...

    def krok(self, smer):
        ...
```

kde

- metóda **\_\_init\_\_(self, meno\_suboru)**: prečíta súbor a vytvorí z neho hraciu plochu ako dvojrozmerné pole znakov (atribút **pole**) s políčkami zvierat a domov, toto pole nemusí mať rovnako dlhé riadky; tiež prečíta pozíciu auta, ktoré ale nezapisujete do tohto poľa (testovač bude kontrolovať správny tvar a obsah poľa);
- metóda **\_\_repr\_\_(self)**: vráti znakový reťazec, ktorý reprezentuje momentálny stav hracej plochy, pričom sa zobrazí aj poloha auta ako '\*', medzi riadkami je znak '\n';
- metóda **kapacita(self, kapa)**: nastaví kapacitu auta, bez tohto nastavenia by bola kapacita auta 1;
- metóda **zisti(self)**: vráti 2 hodnoty (dvojicu): znakový reťazec s písmenami práve naložených zvierat na aute (nezáleží na poradí) a množinu políčok so zvieratami, ktoré sa ešte nenakladali na auto (ako množinu trojíc (písmeno, riadok, stĺpec));
- metóda **krok(self, smer)**: pohne auto v jednom zo zadaných smerov ('l' pre vľavo, 'p' pre vpravo, 'h' pre hore, 'd' pre dolu) o jedno políčko, pričom, ak je na danom políčku zviera (písmeno od 'a' po 'z') a kapacita auta to ešte povoľuje, tak ho naloží a z políčka ho odstráni (v poli ho nahradí '.'), ak je na danom políčku dom (písmeno od 'A' po 'Z') a na aute má príslušné zvieratá, tak zo zoznamu zvierat na aute ich odstráni (písmeno domu v ploche ponechá); ak sa tento príkaz vykonať nedá (políčko je mimo poľa alebo obsahuje medzeru), metóda ho odignoruje; parameter **smer** môže obsahovať aj celú postupnosť šípkov, vtedy auto tieto šípkov postupne vykoná, pričom niektoré z nich možno odignoruje (ak sú v smere nepriechodného políčka); metóda vráti počet naozaj vykonaných krokov auta.

Vstupný formát súboru:

- na začiatku obsahuje popis hracej plochy ako dvojrozmerné pole znakov (tieto riadky nemusia mať rovnakú dĺžku)
- znaky v tejto hracej ploche majú tento význam:
  - '.' priechodné políčko s cestou;
  - ' ' nepriechodné políčko;
  - 'a' až 'z' priechodné políčko so zvieratom – rovnakých zvierat môže byť aj viac;
  - 'A' až 'Z' priechodné políčko s domom pre zodpovedajúce zviera (pre príslušné malé písmeno) – domov s rovnakým písmenom môže byť aj viac;
- za popisom hracej plochy je v súbore jeden prázdny riadok;
- nasledujúci riadok obsahuje dvojicu celých čísel pre riadok a stĺpec políčka s autom;
- predpokladajte, že súbor je zadaný korektné.

Počítajte s tým, že v ploche sa môže nachádzať aj viac rovnakých zvierat a auto ich zrejme všetky bude prevážať do domu s rovnakým veľkým písmenom (hoci aj domov s týmto písmenom môže byť aj viac). Zrejme závisí od kapacity auta, koľko zvierat sa odvezie naraz. Ak auto vezie viac rovnakých zvierat a príde k príslušnému domu, naraz ich všetky vyloží.

Napr. pre takéto zadanie hracej plochy a polohy auta:



tento test:

```

if __name__ == '__main__':
    p = PetDetective('subor2.txt')
    #p.kapacita(2)
    print(p)
    print(p.zisti())
    print(p.krok('hphpppppddd'))
    print(p)
    print(p.zisti())
    print(p.krok('dhhllllldddllllll'))
    print(p)
    print(p.zisti())

```

vypíše:

```

.....a..
. . . . .
....* ..b
. . . . .
B..... A
('', {'b', 2, 8}, {'a', 0, 6})
9
.....
. . . . .
..... ..b
. . . . *
B..... A
('a', {'b', 2, 8})
13
.....
. . . . .
..... ..b
. . . . *
*..... A
('', set())

```

Uvedomte si, že ak by bola nastavená kapacita auta na 2, tak druhý výpis `print(p.zisti())` by vypísal `('ab', set())`.

Aby ste mohli spúšťať skúškové testy, program uložte do súboru **skuska.py**. Riešenie (bez dátových súborov) odozdajte na úlohový server <http://capek.ii.fmph.uniba.sk/list>.

Praktická časť končí o 11:00 a skúška ďalej pokračuje od 12:00 vyhodnotením v kancelárii **m162**.