

## Zadanie skúšky z programovania 18.1.2017

### Sokoban

Sokoban je hra pre jedného hráča, ktorého cieľom je presunúť v štvorcovej sieti všetky škatule na vyznačené cieľové políčka. Hráč pri prechádzaní hracej plochy (pomocou šípok 'l' vľavo, 'p' vpravo, 'h' hore, 'd' dole) môže škatuľu, ktorá leží pred ním, potlačiť v danom smere na políčko za škatuľou, ale len vtedy, keď je toto políčko prázdne (nie je tam ani stena ani iná škatuľa).

Cieľom v programe je odkontrolovať, kam sa hráč po zadanej postupnosti šípok dostane, ktoré škatule kam presunie a na koľkých cieľových políčkach sa pritom nachádzajú škatule a na koľkých nie.

Textový súbor popisuje hraciu plochu ako štvorcovú sieť (riadky môžu byť rôzne dlhé). Riadky v prvej časti súboru obsahujú priamo riadky hracej plochy. Ďalej za jedným prázdny riadkom nasledujú pozície cieľových políčok ako dvojice celých čísel. Znaky popisujúce hraciu plochu majú tento význam:

- '\*' je pozícia hráča
- 'o' je škatuľa
- 'M' je stena
- ' ' je voľné políčko

Zrejme na niektorých zadaných cieľových políčkach sa na začiatku hry môže nachádzať nielen hráč ale aj škatuľa.

Vašou úlohou je prečítať textový súbor s popisom plochy a odsimulovať zadanú postupnosť stlačení šípok. Samotný pohyb hráča sa bude riadiť podľa týchto pravidiel:

- šípka na voľné políčko (môže byť cieľové) presunie hráča
- šípka na políčko so stenou sa ignoruje (hráč zostane stáť na mieste); môžete počítať s tým, že celá hracia plocha je obkolesená stenami a hráč sa nemôže dostať za okraj plochy
- šípka na políčko so škatuľou bude mať účinok (posunie túto škatuľu) jedine vtedy, keď je za škatuľou v danom smere voľné políčko, inak sa šípka ignoruje; zrejme posunutie škatule znamená, že škatuľa sa posunie na voľné miesto a hráč sa presunie na pôvodnú pozíciu škatule

Riešenie zapíšete do triedy **Sokoban** s týmito metódami:

```
class Sokoban:

    def __init__(self, meno_suboru):
        ...

    def __str__(self):
        return ''

    def rob(self, postupnost):
        return 0

    def kontrola(self): # na koľkých cieľových políčkach nie je škatuľa
        return None
```

kde

- metóda `__str__()`: vráti znakový reťazec, ktorý reprezentuje momentálny stav hracej plochy aj s pozíciou hráča (znak `'*'`), všetkých škatúl (znak `'O'`) a všetkých cieľových políčok (znak `'+'`); medzi riadkami je znak `'\n'`;
- metóda `rob(postupnost)`: kde parameter `postupnost` je znakový reťazec s postupnosťou príkazov, teda stláčaných šípok (znaky `'l'`, `'p'`, `'h'`, `'d'`) – hráč sa v ploche pohybuje podľa týchto zadaných príkazov; ak sa daným smerom nemôže pohnúť (stena alebo prekážka za tlačenu škatuľou), tento konkrétny príkaz sa ignoruje; metóda vráti počet naozaj vykonaných krokov
- metóda `kontrola()`: vráti dvojicu (typ `tuple`) celých čísel: (na koľkých cieľových políčkach je škatuľa, na koľkých cieľových políčkach nie je škatuľa) – zrejme v súčte tieto dve čísla dávajú celkový počet cieľových políčok

Napr. pre súbor

```
MMMMM
M  M
M  MMMMM
M O  M
M *O  M
MMMMMMMMM

2 2
4 4
```

vypíše:

```
MMMMM
M  M
M + MMMMM
M O  M
M *O+  M
MMMMMMMMM
kontrola (0, 2)
rob 5
MMMMM
M  M
M O MMMMM
M  M
M *O  M
MMMMMMMMM
kontrola (2, 0)
```

takýto test:

```
if __name__ == '__main__':
    s = Sokoban('subor1.txt')
    print(s)
    print('kontrola', s.kontrola())
    print('rob', s.rob('hddllpp'))
    print(s)
    print('kontrola', s.kontrola())
```

Z úlohového servera L.I.S.T. si stiahnite kostru programu `skuska.py`. Pozrite si testovacie dáta v súboroch `'subor1.txt'`, `'subor2.txt'`, `'subor3.txt'`, ..., ktoré bude používať testovač.

Aby ste mohli spúšťať skúškové testy, program uložte do súboru `skuska.py`.

Riešenie (len súbor `skuska.py` bez dátových súborov) odovzdajte 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**.