

Zadanie skúšky 1.2.2016

Wurmi

Hadík **Wurmi** sa pohybuje po ploche, pričom pojedá chutné ovocie (tzv. potrava) a raz začas prehltnie aj nejaký nechutný hríbik (jed), ktorý mu ale nerobí dobre. Hadík sa skladá z niekoľkých dielikov, pričom, keď zje nejakú potravu, predĺži sa o jeden dielik, keď zje nejaký jed, skráti sa o jeden dielik. Ak by sa hadík skladal len z jedného dielika a prehltol by jed, umiera (jeho dĺžka bude 0) a zrejme sa ďalej už nehýbe. V hracej ploche sa môžu nachádzať aj nejaké prekážky, cez ktoré ale neprejde. Podobne sú pre hadíka prekážkou aj jeho vlastné dieliky, cez ktoré tiež nevie prejsť. Hadíka 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 **Wurmi**:

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

    def __repr__(self):
        ...

    def zisti(self):
        ...

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

kde

- metóda **__init__(self, meno_suboru)**: prečíta súbor a vytvorí z neho hraciu plochu ako dvojrozmerné pole znakov (atribút **pole**) s počiatočným rozložením potravy, jedu a prekážok; tiež prečíta pozíciu hadíka;
- metóda **__repr__(self)**: vráti znakový reťazec, ktorý reprezentuje momentálny stav hracej plochy, pričom sa zobrazí aj poloha hadíka: hlava ako 'o', zvyšné dieliky ako '*', medzi riadkami je znak '\n';
- metóda **zisti(self)**: vráti 3 hodnoty (trojicu): množinu políčok s potravou (ako množinu dvojíc (riadok, stĺpec)), množinu políčok s jedom (tiež ako množinu dvojíc) a pole pozícií dielikov hadíka (ako pole dvojíc, pričom prvý prvok je pozícia hlavy hadíka);
- metóda **krok(self, sipka)**: pohne hadíka v jednom zo zadaných smerov ('l' pre vľavo, 'p' pre vpravo, 'h' pre hore, 'd' pre dolu), pričom, ak je na danom políčku potrava alebo jed, hadíka predĺži alebo skráti; ak sa tento príkaz vykonať nedá, metóda ho odignoruje; parameter **sipka** môže obsahovať aj celú postupnosť šípkov, vtedy tieto šípky hadík postupne vykoná, pričom niektoré z nich možno odignoruje, ak sú v smere nejakej prekážky; v tejto postupnosti šípkov sa môžu nachádzať aj jednociferné čísla, ktoré potom označujú, koľkokrát sa má vykonať za nimi nasledovná šípka; metóda vráti počet naozaj vykonaných krokov hadíka.

Vstupný formát súboru:

- na začiatku obsahuje popis hracej plochy ako dvojrozmerné pole znakov (tieto riadky majú rovnakú dĺžku)
- znaky v tejto hracej ploche majú tento význam:
 - ' ' voľné políčko;
 - '.' políčko s potravou;
 - '+' políčko s jedom;
 - všetky zvyšné znaky sú pre hadíka prekážkou;
- za popisom hracej plochy je v súbore jeden prázdny riadok
- až do konca súboru nasledujú riadky s popisom hadíka: každý riadok obsahuje dvojicu celých čísel pre riadok a stĺpec políčka s dielikom hadíka, prvý dielik označuje hlavu hadíka; všetkým týmto políčkam v hracej ploche zodpovedajú voľné políčka
- predpokladajte, že súbor je zadaný korektne

Uvedomte si, že ak sú napr. v poli **had** postupne uložené súradnice dielikov hada (na začiatku je hlava), potom, keď hlava príde na políčko, ktoré nie je prekážkou, toto políčko (jeho súradnice) sa stáva hlavou, a teda všetky doterajšie políčka sa v poli posunú dozadu (had sa predlžil o 1) a potom pre políčko hracej plochy:

- ' ' - sa posledné políčko hada zruší
- '. ' - posledné políčko hada sa nemení
- '+ ' - posledné dve políčka hada sa zrušia

Tiež si uvedomte, že ak mal had dĺžku 1 (mal iba hlavu) a prešiel cez jed, had umiera, t.j. pole jeho súradníc bude prázdne.

Napr. pre takéto zadanie hracej plochy a súradníc hadíka:

```
XXXXXXXXX
X      X
X X   X
X .   X
X X   X
X XX. X
X +   X
XXXXXXXXX
```

```
1 3
1 4
2 4
2 5
```

tento test:

```
if __name__ == '__main__':
    w = Wurmi('subor.txt')
    print(w)
    m1, m2, had = w.zisti()
    print('potrava =', m1)
    print('jed =', m2)
    print('had =', had)
    print(w.pole)
    print(w.krok('h'))
    print(w.krok('l16d5ph5l'))
    print(w)
    m1, m2, had = w.zisti()
    print('potrava =', m1)
    print('jed =', m2)
    print('had =', had)
```

vypíše:

```
XXXXXXXXX
X o* X
X X** X
X .   X
X X   X
X XX. X
X +   X
XXXXXXXXX
potrava = {(5, 4), (3, 4)}
jed = {(6, 3)}
had = [(1, 3), (1, 4), (2, 4), (2, 5)]
[['X', 'X', 'X', 'X', 'X', 'X', 'X', 'X'], ['X', ' ', ' ', ' ', ' ', ' ', ' ', 'X'], ['X', ' ', ' ', ' ', 'X', ' ', ' ', ' '], [' ', ' ', 'X'], ['X', ' ', ' ', ' ', ' ', ' ', 'X'], ['X', ' ', ' ', ' ', ' ', ' ', 'X'], ['X', 'X', 'X', 'X', 'X', 'X', 'X', 'X'],
['X', 'X', ' ', ' ', ' ', 'X'], ['X', ' ', ' ', ' ', '+', ' ', ' ', 'X'], ['X', 'X', 'X', 'X', 'X', 'X', 'X', 'X'],
['X']]
0
15
XXXXXXXXX
X      X
X X   X
X .   X
X X   X
X Xoo**X
X      *X
XXXXXXXXX
potrava = {(3, 4)}
jed = set()
had = [(5, 4), (5, 5), (5, 6), (6, 6)]
```

Aby ste mohli spúšťať skúškové testy, program uložte do súboru **skuska.py**. Riešenie (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**.