Tema 1 - Maze

- Responsabili: Mihaela Pînzaru Cristi Alexandru Vasile Cosmin Petrişor Dragos Dimitriu
- **Deadline:** 13.11.2016
- Data publicării: 24.10.2016
- Data ultimei modificări: 24.10.2016
- Data tester-ului: 27.10.2016

Obiective

- Familiarizarea cu limbajul **Java**
- Dezvoltarea abilităţilor de bază de organizare a surselor şi controlul accesului
- Utilizarea unui **design orientat-obiect** şi exploatarea conceptului de **încapsulare**
- Respectarea unui **coding-style** adecvat

Descrierea problemei

Romeo și Julieta vor să se întâlnească în labirinticul oraș Verona. Cei doi trebuie să se vadă până apune soarele, de aceea au nevoie de ajutor pentru a stabili locul cel mai bun pentru întâlnire. Atât Romeo, cât și Julieta, se pot deplasa prin oraș din poziția curentă în oricare din cele 8 poziții învecinate (dacă harta le permite).

Descrierea datelor de intrare

Orașul Verona este reprezentat printr-o hartă de *n* x *m* celule. Poziția de unde pleacă **Romeo** este reprezentat pe hartă prin literă R, iar cea a **Julietei** prin literă J. Zidurile orașului sunt reprezentate prin X (nu se poate trece prin acel loc din matrice). Zonele prin care pot trece sunt marcate cu spațiu.

Cerințe

Implementați folosind paradigma OOP un algoritm care să aleagă un punct de întâlnire în care atât Romeo, cât și Julieta, să poată ajunge în același timp plecând de acasă (același număr de pași), iar acest **timp** să fie **minim**.

Structura fișierului de intrare

Fișierul de intrare `maze.in` conține:

- pe prima linie numerele naturale `N M`, care reprezintă numărul de linii și respectiv de coloane ale matricei, separate prin spațiu.
- pe fiecare dintre următoarele N linii se află M caractere (care pot fi doar 'R', 'J', 'X' sau spațiu), reprezentând matricea.

```
5 5
R XX
X X X
X XXX
X X X
X J X
```

Datale de iesire

Fișierul de ieșire `maze.out` va conține câte un răspuns pe linie, pentru fiecare soluţie găsită. Un răspuns este format din trei numere naturale separate prin câte un spațiu: *t x y*, având semnificația:

- *t* este timpul minim în care Romeo, respectiv Julieta, ajunge în punctul de întâlnire (*t* porneşte din 1).
- *x y* reprezintă coordonatele punctului de întâlnire (acestea, ca şi *t*, pornesc din [1, 1]).

În cazul în care nu există o cale între cei doi se va afișa doar INF.

```
 3 3 2
```

Reprezentarea datelor

Acesta reprezentare este doar o sugestie.

Pentru a-i reprezenta pe Romeo și Julieta vom avea o clasă **Character** care va avea următoarele informații:

- nume
- poziție pe hartă

Orașul va fi reprezentat de clasa **CityMap** (numele **Map** este luat deja) care va conține:

- o matrice *n* x *m* care conține detalii despre accesibilitatea punctului.

Determinarea poziției de întâlnire cu cel mai mic cost, cât și efortul depus de fiecare personaj se vor

implementa în clasa **Game**. Această pornește de la următoarea structură:

- character1
- character2
- map
- puteți avea o metodă `play()` care va implementa algoritmul de găsire a punctului de întâlnire.
- **această clasă va implementa și metoda *main***

Punctaj

- **90p** trecerea testelor pe Vmchecker
- **10p** coding-style, readme, comentarii, organizarea surselor și a codului, aspect general

Comentariile tuturor claselor și metodelor relevante vor trebui să respecte formatul Javadoc.

Nu uitați de paginile wiki: indicatii pentru teme și coding style.

Temele vor fi testate împotriva plagiatului. Orice tentativă de copiere va duce la **anularea punctajului** de pe parcursul semestrului și **repetarea materiei** atât pentru sursă(e) cât și pentru destinație(ii), fără excepție.

Tema nu va fi luată în considerare dacă nu respectă paradigma OOP (adică primeşte **0** puncte, adica aţi muncit degeaba). Fiecare metodă/clasă trebuie să respecte condiția “Do one thing! Do one thing well! ”

Format arhivă

Arhiva pe care o veţi urca pe **Vmchecker** va trebui să conţină în **directorul rădăcină**:

- directorul cu pachete şi fişiere sursă `src`, inclusiv
- directorul `doc`, generat de javadoc
- fişierul `README` cu descrierea implementării

Resurse

- PDF enunţ temă
- Tester v1.1
- Soluţie

From:
http://elf.cs.pub.ro/poo/ - **Programare Orientată pe Obiecte**

Permanent link:
http://elf.cs.pub.ro/poo/teme/tema1

Last update: **2016/12/04 16:59**