

# Laboratorium 7

**Otwarto:** środa, 17 listopada 2021, 15:30

**Wymagane do:** środa, 17 listopada 2021, 17:05

Zaprojektuj dwa **typy wyliczeniowe**:

1. (2 pkt.) **Direction** ze stałymi **LEFT**, **RIGHT**, **UP**, **DOWN**, oznaczający **kierunek przejścia po planszy**, zawierający:
  - dwa prywatne pola przechowujące **współrzędne wektora przejścia**,
  - **konstruktor** ustawiający wartości pól,
  - metody typu **"get"** zwracające zawartości pól,
  - przesłoniętą metodę **toString** zwracającą dane w formie np. **[-1, 0]** dla **LEFT**.
2. (3 pkt.) **Option** ze stałymi **RESET**, **LEFT**, **RIGHT**, **UP**, **DOWN**, **EXIT**, oznaczający **opcję wybraną przez użytkownika**. Typ powinien zawierać:
  - trzy prywatne pola przechowujące: **znak** (odpowiadający żądanej opcji, np. **'q'** dla **EXIT**), **opis** opcji (np. **"Zakończenie gry"** dla **EXIT**), **kierunek** przejścia (czyli obiekt **Direction** lub **null** w przypadku opcji **EXIT** i **RESET**);  
*opisy opcji można wymyślić samodzielnie (podobnie znaki odpowiadające opcjom) albo skorzystać z tych, które wynikają z przykładowego wyjścia z programu*
  - **konstruktor** ustawiający wartości pól,
  - metody typu **"get"** zwracające zawartości pól,
  - przesłoniętą metodę **toString** zwracającą dane w formie np. **'d' ==> opcja DOWN, opis: Przesun w dol, wektor przesunięcia: [0, -1]** dla **DOWN**.

(5 pkt.) Napisz konsolową grę **labirynt** w uproszczonej wersji:

- Plansza **board** to **dwuwymiarowa tablica znakowa** o wymiarach **nx × ny**. Gracz może poruszać się tylko po elementach ze spacjami. Początkowe wypełnienie planszy:
  - znaki **'x'** we wszystkich **elementach brzegowych** poza jednym: spacja w środkowym elemencie górnego brzegu planszy (**wyjście** z labiryntu);
  - **wewnątrz planszy**: znaki **'x'** wypełniają resztę planszy z prawdopodobieństwem **p**, w pozostałych elementach są spacje;
  - znak **'o'** w dolnym lewym rogu planszy (oznaczenie **pozycji gracza**).
- **nx**, **ny** oraz **p** niech będą **parametrami wejściowymi** programu.
- Program powinien w pętli **pobierać od użytkownika znak**, interpretować go zgodnie z dostępnymi opcjami i wykonywać odpowiednie działanie na planszy (lub zakończyć program dla **'q'**). Użytkownik wygrywa, gdy przejdzie na pozycję jedyne wyjścia z labiryntu. Przykładowe wczytywanie znaku:

```
Scanner sc = new Scanner(System.in); // klasa z pakietu java.util
char z;
//...
z = sc.next().charAt(0);
//...
sc.close();
```

**Wymagania:**

3. Typ **Option** powinien być wykorzystany w celu **wypisania dostępnych opcji**.
4. Typ **Direction** powinien być wykorzystany w celu **przejścia** na planszy o odpowiedni wektor.

Przykład działania programu z parametrami 6 4 0.3:

```
Dostępne opcje:
'0' ==> opcja RESET, opis: Reset planszy
'1' ==> opcja LEFT, opis: Przesun w lewo, wektor przesunięcia: [-1, 0]
'r' ==> opcja RIGHT, opis: Przesun w prawo, wektor przesunięcia: [1, 0]
'u' ==> opcja UP, opis: Przesun w gore, wektor przesunięcia: [0, 1]
'd' ==> opcja DOWN, opis: Przesun w dol, wektor przesunięcia: [0, -1]
'q' ==> opcja EXIT, opis: Zakonczenie gry
```

```
XXX XX
XX  XX
Xo X X
XXXXXX
```

Wybierz opcje: r

```
XXX XX
XX  XX
X oX X
XXXXXX
```

Wybierz opcje: r  
NIE UDALO SIE WYKONAC TAKIEGO RUCHU.

```
XXX XX
XX  XX
X oX X
XXXXXX
```

Wybierz opcje: u

```
XXX XX
XXo XX
X  X X
XXXXXX
```

Wybierz opcje: r

```
XXX XX
XX oXX
X  X X
XXXXXX
```

Wybierz opcje: u

```
XXXoXX
XX  XX
X  X X
XXXXXX
```

GRATULACJE, WYGRALES!