

# Programowanie Obiektowe

## Podstawy Java

### Zadanie oceniane nr 1a

29-03-2021

Po zakończeniu pracy konieczne jest wgranie zmian do repozytorium (add + commit + push) w katalogu o nazwie w stylu: zadanie\_oceniane\_1. Fakt wgrania plików do swojego repozytorium można sprawdzić samodzielnie logując się (via www) na swoje konto i sprawdzając czy pojawiły się tam wszystkie zmiany.

### "Kret"



W dniu dzisiejszym należy stworzyć mini symulator podziemnego dwuwymiarowego środowiska w którym pojawiły się krety. Model terenu działania gryzonia powinien być dwuwymiarową tablicą wypełnioną instancjami następujących klas obiektów (nazwijmy je Obiektami Ziemiennymi): glebę, dżdżownicami oraz larwami (które stanowią pożywienie dla kretów). Kret jest obiektem posiadającym referencję do tablicy Obiektów Ziemiennych, która na początku jest nimi całkowicie wypełniona. Ssak konsumuje losowo napotkane obiekty (dżdżownice, larwy i glebę) dopisując ich wartość odżywczą do swojej oraz wstawiając we wszystkie te odwiedzone miejsca wartości null. Oczywiście po zakończonej konsumpcji. Poniżej przedstawiony jest rysunek mówiący w jaki sposób można sobie tę sytuację wyobrazić.

	Dżdżow.								
								Larwa	
		null		null					
					null				
					null				
				null					
					Dżdżow.		Larwa		
	Dżdżow.								

## Prace do wykonania:

1. **Stworzyć poprawną\* hierarchię klas reprezentujących wspomniane Obiekty Ziemne wraz z danymi które one przechowują (podane w nawiasach):**

- gleba (gęstość – losowana z zakresu od 1.4 do 2.4; wartość odżywcza)
- dżdżownica (masa – losowana z zakresu 0.5-1.0; wartość odżywcza))
- larwa (masa – losowana z zakresu 2.0-3.0; wartość odżywcza)

Masa i gęstość jest zmiennoprzecinkowa, wartość odżywcza jest całkowita. Typy dobierać wedle uznania. Pola bez wytycznych co do ich tworzenia są przekazywane z zewnątrz, pozostałe kreowane są "na terenie" danej klasy w momencie jej instancjonowania.

2. **Stworzyć klasę Environment posiadającą wspomnianą dwuwymiarową tablicę. Klasa ta powinna:**

- podczas jej instancjonowania zainicjować tablicę i całkowicie (pseudolosowo) wypełnić ją obiektami (glebą, larwami, dżdżownicami) w oparciu o przekazane dane znane już na etapie tworzenia obiektu (rozmiar tablicy m x n, ilość larw, ilość dżdżownic). Wartość odżywcza larwy losowana jest z przedziału 3-5. Wartość odżywcza dżdżownicy pobierana jest ze specjalnie utworzonego pola związanego z klasą Environment (a nie z jej obiektami) i przekazywana do kreowanego obiektu reprezentującego dżdżownicę (w momencie jego tworzenia). Wartość ta jest równa 3 i nigdy nie będzie zmieniana. Lokalizacja na tablicy (indeksy x i y) każdego nowo utworzonego Obiektu Ziemnego jest losowana. Jeżeli miejsce na tablicy jest już zajęte (jest tam wstawiony jakiś obiekt), to trzeba wylosować je jeszcze raz. Do skutku. Ryzyko nieskończonego losowania pominąć. Cała reszta ma być wypełniona glebą (wartość odżywcza = 0).
- posiadać metodę wypisującą na konsoli schemat tablicy dwuwymiarowej z zawartością. Jeśli pod daną parą indeksów jest jakiś obiekt, to należy wypisać jego

**reprezentację tekstową** (gleba - "#", dżdżownica - "{}", larwa - "@"). Dla null – spacja.

### 3. Stworzyć klasę reprezentującą Kreta i wkomponować ją w klasę Environment operając się na poniższych wytycznych.

Kret (gatunek\*; wartość odżywcza – liczba całkowita losowana z zakresu 200-240; odnośnik do dwuwymiarowej tablicy z poprzedniego zadania, przekazany w momencie tworzenia)

\* może być jedna z trzech możliwych opcji: *cinerea*, *europaea* lub *velessiensis*

- Stworzyć interfejs MoleMotion zawierający metodę void goMole() i String getNutritionalValue(), który jest implementowany przez Kreta. Zaimplementować metody goMole() i getNutritionalValue() w ciele Kreta i narazie pozostawić je puste (getNutritionalValue niech narazie zwraca cokolwiek)
- Dodać do klasy Environment pole będące odnośnikiem do typu MoleMotion
- Stworzyć dodatkowy konstruktor klasy Environment dla przypadku gdy chcemy również utworzyć kreta. Jego gatunek jest przekazywany jako dodatkowy parametr (*europaea*). Przyporządkowujemy go do pola z poprzedniego podpunktu.
- Stworzyć metodę runMole(), która 10 razy na wspomnianym powyżej polu, wywołuje: metodę goMole() oraz metodę wypisującą zawartość tablicy dwuwymiarowej (stworzoną w poprzednim zadaniu)
- Metoda getNutritionalValue() zwraca bieżącą wartość odżywczą kreta
- Metoda goMole(), pochodząca z implementowanego interfejsu MoleMotion powinna:
  - ✓ losować miejsce w tablicy dwuwymiarowej
  - ✓ "zjadać" niepustą komórkę poprzez ustalenie jej wartości odżywczej i dodanie do wartości odżywczej kreta. Po takiej operacji kret na konsoli wypisuje: "Mniam!", skonsumowaną wartość odżywczą oraz lokalizację znaleziska (indeksy) w tablicy
  - ✓ wypisywać tekst "Pusto :(", jeśli komórka była pusta
  - ✓ ustawić null w każdej napotkanej niepustej komórce (oczywiście po jej "zjedzeniu")

Mile widziana jakaś demonstracja działania, tworząca obiekt klasy Environment i wywołująca metodę runMole().