Programowanie Obiektowe
Podstawy języka Java
Zadanie oceniane nr 1a
31-03-2022

Po zakończeniu pracy konieczne jest wgranie zmian do repozytorium (add + commit + push) w katalogu o nazwie w stylu: zadanie_oceniane_1a. Fakt wgrania plików do swojego repozytorium można sprawdzić samodzielnie logując się (via www) na swoje konto i sprawdzając czy pojawiły się tam wszystkie zmiany.

"Grzybobranie"

W dniu dzisiejszym należy stworzyć mini symulator procesu zbierania grzybów. Terenem na którym proces ten będzia miał miejsce jest struktura, którą można wyobrazić sobie taką, jaka jest zaprezentowana na załączonym obrazku (oczywiście tym poniżej). Z uwagi na specyfikę nazewnictwa dobrym pomysłem jest użycie polskich nazw (ale bez znaków diakrytycznych).

	kolcz.				kolcz.			cesarski	
							sromotnik		
		sromotnik							
									szarawy
					szarawy				
		kolcz.							
							sromotnik		
					cesarski				kolcz.
	cesarski								
			kolcz.						

Prace do wykonania:

1. **Stworzyć hierarchię klas odzwierciedlającą elementy występujące w tym zadaniu grzyby wraz z danymi które przechowują (podane w nawiasach)**
   - muchomor sromotnikowy
     (twardość* - losowana; masa owocnika*- 25-75; toksyny* - *zabójcze* dla masy owocnika <50, w p.p. *wyjątkowo zabójcze*)
   - muchomor cesarski
     (twardość* - twardy; masa owocnika* - 50-75)
   - muchomor kolczastogłowy
     (twardość* - losowana; masa owocnika* - 25-50)
   - muchomor szarawy
     (twardość* - losowana; masa owocnika* - 25-75; toksyny* = trujące, rok odkrycia gatunku - 1783)

   * *twardość może przyjmować wartości takie jak: twardy, miękki, średniotwardy*
   * *masa owocnika losowana z podanego przedziału*
   * *toksyny mogą być wyjątkowo zabójcze, zabójcze, trujące lub nieobecne*

   Każdy grzyb sam ustala zawartość toksyn i twardość zgodnie z wytycznymi zadania.
   Żaden grzyb w tej hierarchii nie jest szczególnym przypadkiem innego z wyżej wymienionych.
   Każdy grzyb posiada reprezentację tekstową zawierającą nazwę i stan pól w dowolnym formacie.

   Należy wkomponować w hierarchię dwie metody:
   - ugotuj() - która twardość każdego grzyba redukuje do minimalnej, a w przypadku muchomora szarawego dodatkowo zmienia zawartość toksyn do stanu "*nieobecne*".
   - isTrujacy() zwracającą false dla muchomora cesarskiego, kolczastogłowego oraz

szarawego (jeśli w tym ostatnim poziom toksyn jest *nieobecny)*. W p.p. true.

2. **Stworzyć klasę modelującą las w którym rosną sobie w/w owoce runa leśniego.**
   Posiada ona:
   - dwuwymiarowką tablicę referencji na grzyby
   - konstruktor przyjmujący parametry będące wymiarami: x i y
   - dla podanych ujemnych bądź zerowych wartości używana jest domyślna wartość 100.
   - pseudolosowo rozrzucone referencje na grzyby w sposób następujący sposób:
     - ✔ jest jedna wylosowana na początku kolumna w której nie będzie grzybów
     - ✔ każde pole struktury posiada jakiegoś grzyba z P=0.2. W p.p. null.
     - ✔ jeżeli uznaliśmy że pole posiada któregoś z muchomorów to losujemy (z jednakowym prawdopodobieństwem) który to ma być, tworzymy go i wstawiamy.
   - Posiada metodę zwracającą grzyba z losowo wybranego pola, która dodatkowo wstawia tam wartość null oraz zwraca null (niestety), gdy pole będzie puste.

3. **Stworzyć klasę konsumenta grzybów**
   Posiada ona:
   - referencję do lasu z którego czerpać będzie smakołyki (przekazaną z zewnątrz podczas instancjonowania)
   - metodę konsumujGrzyby() która n razy (50 < n < 100):
     - ✔ wysołuje metodę zwracającą losowo wybranego grzyba
     - ✔ gotuje go
     - ✔ przekazuje do metody zjedz(grzyb), która w przypadku gdy grzyb jest trujący wypisuje na konsolę: "Uuuups!" i doprowadza (bez użycia wyjątków) do przerwania w/w pętli. Może być przez zwróconą wartość.

4. **Wprowadzić założenie**
   Konsumenta nie interesuje co dostarcza mu metodę zwracającą grzyba. Mogą to być nawet niespokrewnione ze sobą obiekty.

5. **Demonstracja działania**