

---

# PPPD - Lab. 13

Copyright ©2022 M. Śleszyńska-Nowak i in.

*Zadanie punktowane, lab 13, grupa B, 2022/2023, autor: Piotr Wolszakiewicz*

---

## Temat: Symulator sadu

### Treść zadania

Celem zadania jest implementacja prostego symulatora sadu poprzez uzupełnienie definicji klas `Orchard` i `AppleTree`. Symulacja będzie polegała na przechodzeniu kolejnych dni i wyświetlaniu stanu jabłek w sadzie. Drzewa usychają jeśli ich poziom wody spadnie poniżej zera. Po każdym dniu 1 jednostka wody zamienia się w jedno jabłko.

### Opis klas

W zadaniu będziemy wykorzystywali dwa typy obiektów reprezentujące sad i jabłoni.

Klasa `AppleTree` reprezentuje jabłoni. Jabłoni potrzebuje wody do przeżycia oraz rodzi jabłka. Jedno jabłko jednego dnia, ale wykorzystuje do tego wodę. Może uschnąć gdy zasoby wody się skończą.

Klasa posiada atrybuty:

- `fruits` - liczba owoców na drzewie
- `water_level` - zgromadzone zasoby wody
- `is_live` - wskaźnik mówiący, czy drzewo nie uschło

oraz metody:

- `next_day` - symuluje upływanie jednego dnia - służy do podlewania jabłoni

Klasa `Orchard` reprezentuje obiekt sad, w którym rosną jabłonie (każda jabłoni na konkretnej pozycji). W różnych obszarach sadu deszcz może padać z różną intensywnością, czyli drzewa nie muszą być tak samo podlewane. Klasa posiada atrybuty:

- `field` - pole reprezentowane przez macierz kwadratową. W komórkach macierzy będziemy sadzili drzewa - umieszczali obiekty `AppleTree`
- `size` - rozmiar pola

oraz metody:

- `is_over` - zwraca informację, czy wszystkie drzewa uschły
- `create_storm_matrix` - tworzy macierz o tym samym rozmiarze co sad. Wartości w jej komórkach oznaczają opady deszczu na danym polu.
- `next_day` - symuluje upływanie jednego dnia dla całego sadu. W każdym dniu generujemy macierz deszczu i podlewamy nim sad.

### Zaimplementuj brakujące elementy symulatora:

1. Konstruktor klasy `Orchard`, który jako argumenty przyjmie `size` - rozmiar pola, na którym będzie rósł sad. Pole ma kształt kwadratu o boku `size`.

---

W konstruktorze należy stworzyć i wypełnić atrybuty `size` (liczba całkowita) oraz `field` (macierz kwadratowa o wymiarze `side x side`). W 20% pól macierzy powinny znajdować się drzewka (obiekty `AppleTree`). Nieobsadzone pola w macierzy wypełniamy wartościami `None`. Drzewa sadzimy na polu w sposób losowy, ale w jednym polu możemy umieścić tylko jedno drzewo. Każde sadzone drzewo jest podlane na start losową wartością całkowitą z przedziału  $<1,3>$ .

2. Metodę `is_over` w klasie `Orchard`.

Metoda zwraca `True`, jeśli wszystkie drzewa na polu uschły, `False` w przeciwnym przypadku.

3. Metodę `next_day` w klasie `AppleTree` z jednym parametrem `rain_level`.

W ciągu dnia drzewo zużywa jedną jednostkę wody (pole `water_level`), natomiast zyskuje wodę z deszczu - `rain_level`. Drzewo wytwarza jeden owoc (pole `fruits`) jeśli stan wody po nawodnieniu drzewa jest większy od 0. Wytworzenie owocu kosztuje drzewo jedną jednostkę wody. Jeśli po całym dniu poziom wody jest mniejszy od 0 drzewo bezpowrotnie usycha, a razem z nim wszystkie jego owoce.

4. Metodę `create_storm_matrix` w klasie `Orchard`.

Metoda przy każdorazowym wykonaniu tworzy nową macierz tego samego rozmiaru co `field`. W macierzy będą przechowywane wartości typu `int` - oznaczające intensywność deszczu. Przy czym ziemia w górnej prawej (górna macierz trójkątna,  $u_{i,j} = 0$  dla  $i > j$ , gdzie  $i$  - wiersz,  $j$  - kolumna) części pola jest bardziej żyzna. To oznacza, że deszcz pada tam obficie niż w dolnej lewej części pola (dolna macierz trójkątna bez diagonal,  $l_{i,j} = 0$  dla  $i \leq j$ ). W macierzy deszczu żyzność objawia się tym, że wartości w górnej prawej części macierzy mają losową wartość deszczu z zakresu  $<0, k>$ , a w lewej dolnej części macierzy z  $<0, k/2>$ , gdzie  $k$  jest losową wartością z liczb  $\{2,4,6\}$ . Dodatkowo w macierzy deszczu należy umieścić losowo  $k/2$  opadów typu `grad`, na różnych pozycjach. Oznacz `grad` jako wartość `-float('Inf')`

5. Metodę `next_day` w klasie `Orchard`.

Wykorzystaj już zaimplementowaną metodę `create_storm_matrix`, a następnie spuść deszcz na wszystkie drzewka w twoim sadzie. Wartość komórki  $i, j$  z macierzy deszczu, odpowiada ilości wody, która spadnie na pole o indeksie  $i, j$ . Dla każdego z drzew wywołaj metodę `next_day` z odpowiednią wartością poziomu deszczu.

6. Funkcję `main`, a w niej:

- wczytanie `size`
- stworzenie obiektu `Orchard`
- dla 20 pierwszych dni bądź krócej, jeśli umarło ostatnie drzewo:

zasymuluj upływający czas wywołując metodę `next_day`. Po każdym dniu wyświetl planszę sadu.

## Szczegóły

- Plik `2022_IAD_13B_stud.py` zawiera definicje klas do uzupełnienia.
- Liczbę drzewek obcinamy do liczb całkowitych np. dla planszy o rozmiarze 49 - liczba drzewek to 9.

## Punktacja

Za poszczególne elementy można uzyskać następującą liczbę punktów:

- konstruktor klasy `Orchard` - 2 punkty
- metoda `is_over` w klasie `Orchard` - 1 punkt
- metoda `next_day` w klasie `AppleTree` - 2 punkty
- metoda `create_storm_matrix` w klasie `Orchard` - 2 punkty
- metoda `next_day` w klasie `Orchard` - 2 punkty
- funkcja `main` - 1 punkt

## Uwagi

- Jeśli program się nie kompiluje (interpretuje), ocena jest zmniejszana o połowę
- Jeśli kod programu jest niskiej jakości (nieestetycznie formatowanie, mylące nazwy zmiennych itp.), ocena jest zmniejszana o 2p

### Przykład interakcji użytkownika z programem

Please provide orchard size: 10

After planting:

[illegible]

Day: 1

1		1		1		1
	1					
1	1			1		
				1		1
			1	1	1	1
		1			1	
			1			
1		1				

Day: 2

2		2		X		2
	2					
2	2			2		
				2		2
			2	1	2	1
		2			2	X
			2			
2		2				

Day: 3

2	3	X	3
2			

---

	3	3			3			
					3		3	
				3	X	3	2	
		3			3		X	
				3				
	3	3						

---

...(kolejne iteracje)

Day: 18

---

	7		X		X		X	
		X						
	X	X			X			
					X		16	
				X	X	18	X	
		X			18		X	
				X				
	18	X						

---

Day: 19

---

	8		X		X		X	
		X						
	X	X			X			
					X		17	
				X	X	19	X	
		X			19		X	
				X				
	19	X						

---

Day: 20

---

	9		X		X		X	
		X						
	X	X			X			
					X		18	
				X	X	20	X	
		X			20		X	
				X				
	20	X						

---