

2023SP Data Structures
PA: Abstract Data Types with OOP
Points 100

Objective: Practice the use and creation of Abstract Data Types using OOP techniques.

Read the instructions carefully.

P2.36 - Write a **Python** program to **simulate** an ecosystem containing two types of animals, bears and fish.

The ecosystem consists of a river, which is modeled as a relatively large list.

Each element of the list should be a Bear object, a Fish object, or None.

In each time step of the simulation, in order, process each animal decision.

Based on a random process, each animal either attempts to move (e.g., option 0) into an adjacent list location or stay where it is (e.g., option 1).

If two animals of the same type are about to collide in the same cell, then they stay where they are, but they **create a new instance** of that type of animal, which is placed in a random empty (i.e., previously None) location in the list (i.e., river).

If a bear and a fish collide, however, then the fish dies (i.e., it gets removed from the list) and the bear does not move. If the bear, while moving, collides with a fish, the bear eats the fish at the fish location.

If the animal is at the end of the river and it decides to move, then the animal gets out of the river.

You need to create a **parent** class Animal, and **subclasses** for the Bear and Fish.

Initial class skeletons are provided to you. You need to use those skeletons to complete the missing code.

Similarly, you need to work on a **class** for the **Ecosystem** that will **simulate** this scenario. The skeleton for the class is provided to you. You need to complete the missing code.

Every animal has a location and a name that are initialized in the **constructor**.

Similarly, every animal has at least the following methods:

move(): to set the new location for the animal

get_location(): to return its location

__str__() : method to print the type of animal and its current location

get_class_name(): to return the class name

speak(): Allows the animal to speak

You can have other helper methods for this class (although not necessary).

The data and methods are inherited by all animals.

The **animal subclasses**, Bear and Fish, only assign the animal type (i.e., name) to each **new** animal.

Animals can speak. However, only the bears can say a different thing. See the examples below.

You need a **class** representing the **Ecosystem**.

This class will need to have a **constructor** to initialize the Ecosystem.

The constructor needs to initialize an empty river and randomly place animals into the river. The size of the initial river is a parameter for the creation of the object.

To initialize this random ecosystem, the constructor calls the method **initialize_random_ecosystem()**.

In this method you need to:

- 1) **initialize** a **random number** of **Bears** and **place** a **Bear** object in a **random empty position** in the river. You can generate between 1 to 4 bears.

- 2) **initialize** a **random number** of **Fish** and **place** a **Fish** object in a **random empty position** in the river. You can generate between 1 to 4 Fish.

Use a **river size** that is **larger** than the **probable** number of animals. For example, you could have 4 Bears and 4 Fish, therefore your river size should be larger than 8 positions.

For the simulation, you need to complete the **simulate()** method.

This method **receives** a **parameter** for the **number of iterations** for traversing the river. This is a parameter passed by the user.

This method is the **responsible** of **simulating** the scenario described above for the animal's behaviour.

Similarly, this method calls a method **animal_creation()**, if that is **required by the animal scenario**. This method receives an **animal as parameter**, and it **creates** a **new animal** of the **same type** as the received animal. Also, this method needs to **assign** the **new animal randomly** in a free space in the **river**.

Finally, the Ecosystem must have a method **show()** to show all the **information** (I.e., name and position) **for each animal** in the river.

You can have other helper methods for the Ecosystem.

You need to **use** my "**ecosystem_tester.py**" file to test your solution.

Making your simulation repeatable.

To make your simulation repeatable (i.e., the same result every time you run it), you need to uncomment the following line from the Simulation.py file.

If you leave this line as a comment, you will have different outputs every time.

```
# random.seed(42)    # Change this number for your different tests.
```

Show the Execution

Your program should output messages (i.e., text) to show the main stages of the simulation. Please, refer to the examples below to see how these texts are presented.

Class skeletons.

Download the file **simulator_templates.zip** to get the initial class skeletons.

Output Examples

Example 1.

This example shows the simulation for **one iteration**. The river size is **10**.

```
eduardo@Edgars-MacBook-Air OOP % python3 ecosystem_tester.py
```

```
Initializing Ecosystem
Initial number of bears: 1
Initial number of fish: 3
Bear assigned to position 8
Fish assigned to position 5
Fish assigned to position 1
Fish assigned to position 7
```

Output from **initialize_random_ecosystem()**

```
Ecosystem BEFORE simulation
None
Fish at location 1
None
None
None
Fish at location 5
None
Fish at location 7
Bear at location 8
None
```

Output from **show()**

```
~~~ Initializing the Simulation ~~~  
~~~ Iteration 1 ~~~~
```

```
Processing Animal at location 1  
    Animal chose to 1  
    Stay  
Processing Animal at location 5  
    Animal chose to 1  
    Stay  
Processing Animal at location 7  
    Animal chose to 0  
Move Fish at location 7 to 8  
Processing Animal at location 8  
    Animal chose to 1  
    Stay
```

Output from **simulate()**

```
Ecosystem AFTER simulation  
None  
Fish at location 1  
None  
None  
None  
Fish at location 5  
None  
None  
Bear at location 8  
None
```

Output from **show()**

Example 2.

This example shows the simulation for **two iterations**. The river size is **10**.

```
eduardo@Edgars-MacBook-Air OOP % python3 ecosystem_tester.py
```

```
Initializing Ecosystem  
Initial number of bears: 1  
Initial number of fish: 4  
Bear assigned to position 5  
Fish assigned to position 6  
Fish assigned to position 0  
Fish assigned to position 4  
Fish assigned to position 1
```

```
Ecosystem BEFORE simulation  
Fish at location 0  
Fish at location 1  
None
```

None
Fish at location 4
Bear at location 5
Fish at location 6
None
None
None
~~~ Initializing the Simulation ~~~  
~~~ Iteration 1 ~~~~

Processing Animal at location 0
 Animal chose to 0
Stay Fish at location 0 BUT
 Creating a new Fish
New Fish assigned to position 2
Processing Animal at location 1
 Animal chose to 1
 Stay
Processing Animal at location 2
 Animal chose to 1
 Stay
Processing Animal at location 4
 Animal chose to 1
 Stay
Processing Animal at location 5
 Animal chose to 0
Move Bear at location 5 to 6
Bear eats Fish
 It says: I am eating!

Processing Animal at location 6
 Animal chose to 1
 Stay
~~~ Iteration 2 ~~~~

Processing Animal at location 0  
    Animal chose to 1  
        Stay  
Processing Animal at location 1  
    Animal chose to 0  
Stay Fish at location 1 BUT  
    Creating a new Fish  
New Fish assigned to position 8  
Processing Animal at location 2  
    Animal chose to 1  
        Stay  
Processing Animal at location 4  
    Animal chose to 1  
        Stay  
Processing Animal at location 6

```

    Animal chose to 0
        Location 7 is empty, so move it
Processing Animal at location 7
    Animal chose to 0
Move Bear at location 7 to 8
Bear eats Fish
    It says: I am eating!

Processing Animal at location 8
    Animal chose to 0
        Location 9 is empty, so move it
Processing Animal at location 9
    Animal chose to 1
        Stay

Ecosystem AFTER simulation
Fish at location 0
Fish at location 1
Fish at location 2
None
Fish at location 4
None
None
None
None
None
Bear at location 9

```

### **Example 3.**

This example shows the simulation when a **Bear eats a Fish**

```
eduardo@Edgars-MacBook-Air OOP % python3 ecosystem_tester.py
```

```

Initializing Ecosystem
Initial number of bears: 3
Initial number of fish: 1
Bear assigned to position 4
Bear assigned to position 0
Bear assigned to position 9
Fish assigned to position 7

Ecosystem BEFORE simulation
Bear at location 0
None
None
None
Bear at location 4
None
None
Fish at location 7
None
Bear at location 9
~~~ Initializing the Simulation ~~~

```

~~~ Iteration 1 ~~~~

```
Processing Animal at location 0
 Animal chose to 1
 Stay
Processing Animal at location 4
 Animal chose to 0
 Location 5 is empty, so move it
Processing Animal at location 5
 Animal chose to 0
 Location 6 is empty, so move it
Processing Animal at location 6
 Animal chose to 0
 Move Bear at location 6 to 7
Bear eats Fish
 It says: I am eating!
Processing Animal at location 7
 Animal chose to 1
 Stay
Processing Animal at location 9
 Animal chose to 1
 Stay
```

```
Ecosystem AFTER simulation
Bear at location 0
None
None
None
None
None
None
None
Bear at location 7
None
Bear at location 9
```

#### **Example 4.**

This example shows the simulation when a **Fish is eaten by a Bear**

```
eduardo@Edgars-MacBook-Air OOP % python3 ecosystem_tester.py
```

```
Initializing Ecosystem
Initial number of bears: 3
Initial number of fish: 3
Bear assigned to position 9
Bear assigned to position 4
Bear assigned to position 7
Fish assigned to position 3
Fish assigned to position 1
Fish assigned to position 5
```

Ecosystem BEFORE simulation  
None  
Fish at location 1  
None  
Fish at location 3  
Bear at location 4  
Fish at location 5  
None  
Bear at location 7  
None  
Bear at location 9  
~~~ Initializing the Simulation ~~~  
~~~ Iteration 1 ~~~

Processing Animal at location 1  
    Animal chose to 0  
        Location 2 is empty, so move it  
Processing Animal at location 2  
    Animal chose to 0  
Stay Fish at location 2 BUT  
    Creating a new Fish  
New Fish assigned to position 1  
Processing Animal at location 3  
    Animal chose to 1  
        Stay  
Processing Animal at location 4  
    Animal chose to 1  
        Stay  
Processing Animal at location 5  
    Animal chose to 0  
        Location 6 is empty, so move it  
**Processing Animal at location 6**  
    **Animal chose to 0**  
    **Move Fish at location 6 to 7**  
    **Fish is eaten by Bear**  
    **It says: \*Silence...\***  
Processing Animal at location 7  
    Animal chose to 1  
        Stay  
Processing Animal at location 9  
    Animal chose to 1  
        Stay

Ecosystem AFTER simulation  
None  
Fish at location 1  
Fish at location 2  
Fish at location 3  
Bear at location 4  
None  
None  
Bear at location 7



None  
Bear at location 9

## Submission

You need to use at least **Python 3.9x**

Upload a **single ZIP** file.

Name your file as follows: **PA1\_Lastname\_Firstname.zip**

Your **.zip** file must contain the following:

1. Your **.py** source files for each class with your solutions (Do not sent my tester program)
2. A single PDF with screenshots from your program running (10% points deduction if you don't submit a **SINGLE** PDF file). **Do not send .jpg files.**

In each **.py** file, **write as a multiline comment** at the beginning of the file the following:

1. Your name

The **zip** file must be uploaded to Canvas. I do not accept answers via email. I do not accept image files; it must be a PDF file.

Make sure to check the **due date** for this activity on Canvas. Try to submit it before the due date so you can have time to check for improvements.

Make sure you are **submitting the correct files**. I will grade the files uploaded to Canvas.

Make sure you **test your class** with a similar **tester program** as the one I am giving you in this programming assignment.

**Make sure to review the grading rubric.**

**Read all the instructions carefully.**

Late submissions are not allowed.

If you have questions, contact me before making assumptions about what you need to do for solving this assignment.

More questions? Contact me before going with a tutor. I need to know what issues you are having to adjust the content of my lectures.

**Do not assume requirements, contact me if you have doubts!**