

Homework Assignment 2





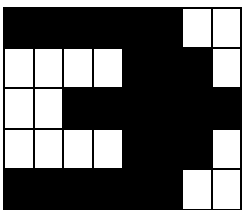

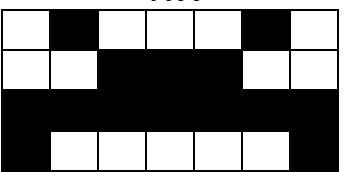
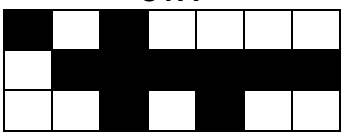
General information:

- Responsible teaching assistant: Moshe Davidian.
- To get quick answers regarding this assignment please ask in Piazza under label HW2.
- If necessary, zoom hours are on Tuesday, 14:00-15:00 please send an email to coordinate prior.

The OOP aquarium



Our aquarium will have 4 types of animals, two types of fish and two types of crabs. by the following specification:

fish		crab	
Scalar	Moly	Ocypode	Shrimp
			
5 X 8	3 X 8	4 X 7	3 X 7
			

-

	x=0							x=7
Y=0								
Y=2		~	~	~	~	~	~	
Y=7	\	/	/	/	/	/	/	/

Notes: The water line is always in the third row ($y = 2$) from the top (regardless of the size of the aquarium). Pay attention to the lower corners of the aquarium (not a straight line).

Screen printing (of both the aquarium and the animals) is done with a space " " between any two horizontal characters.

Program structure:

General menu: At the beginning of the run, the program will ask for dimensions for the aquarium as follows:

```
Welcome to "The OOP Aquarium"
The width of the aquarium (Minimum 40):
The height of the aquarium (Minimum 25):
```

After absorbing valid values, the following main menu will be displayed:

```
Main menu
-----
1. Add an animal
2. Drop food into the aquarium
3. Take a step forward
4. Take several steps
5. Demo
6. Print all
7. Exit
What do you want to do?|
```

When the option selected by the user is completed, the aquarium will be printed on the screen and the main menu will appear again.

Main menu:

- 1) Add an animal - Selecting this option will display a submenu:

```
Please select:
1. Scalare
2. Moly
3. Ocypode
4. Shrimp
What animal do you want to put in the aquarium?|
```

- Each choice will add the right animal to the aquarium.
- For each animal the user will have to enter a name, age, place where it will appear and initial swimming direction.
- For fish, the user will enter X and Y coordinates.
- For crabs, the user will only enter the X coordinate. since the crabs walk on the floor and do not swim in the aquarium. (The program will determine the Y value according to the type of crab selected)
- If the place is not available at that moment, the following message will be displayed: **"The place is not available! Please try again later. "** And the user will have to enter new coordinates.
- Example for crab:

```
Please enter a name:MyShrimp
Please enter age:5
Please enter an X axis location (1 - 59):12
Please enter horizontal direction (0 for Left, 1 for Right):1
```

- Example for fish:

```
Please enter a name:MyScalar
Please enter age:12
Please enter an X axis location (1 - 59):40
Please enter an Y axis location (3 - 29):20
Please enter horizontal direction (0 for Left, 1 for Right):1
Please enter vertical direction (0 for Down, 1 for Up):0
```

Notes:

- The possible age is between 1 and 100 (inclusive)
- The user should be provided with the legal values he can enter depending on the size of the aquarium.
- The coordinates (for animal position) are for the upper left corner of the animal.
- Of course, even if the user enters the last column number (59 for the example above), the whole fish will appear inside the aquarium (the program will check which is the rightmost coordinate possible)
- Each animal starts with a fixed amount of food (STARTING_FOOD = 5)
- The first step (number 0) should reduce the amount of food and promote the age of the animal.

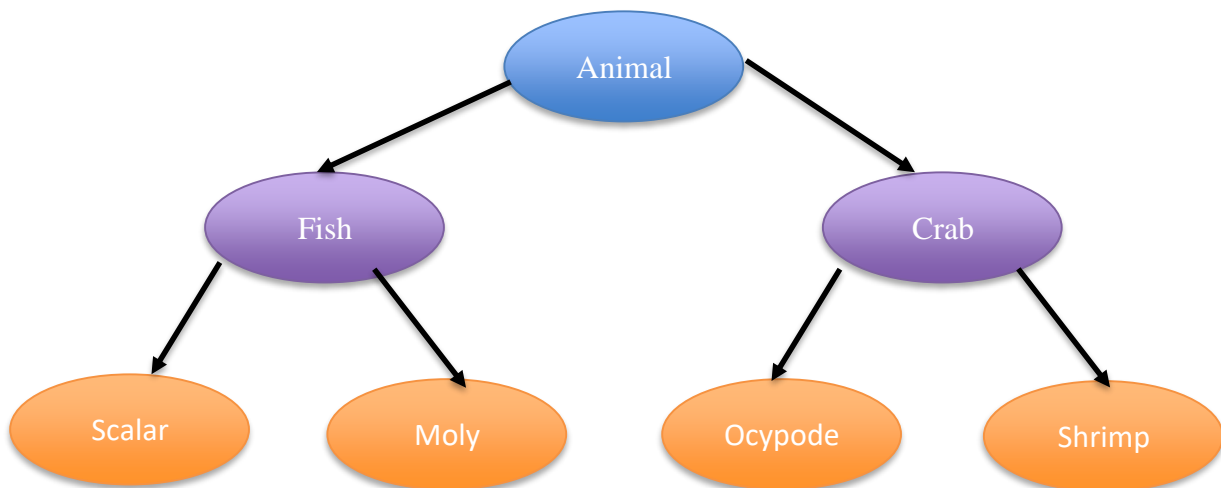


Table 2 - The upper left corner of the animal in yellow

- 2) Drop food into the aquarium - Selecting this option will add a fixed amount of food to all the animals in the aquarium (FEED_AMOUNT = 10).
- 3) Take a step forward - Selecting this option will advance the aquarium one unit of time forward (one step forward):
 - At each step, each of the animals swims one step in the direction of its swimming.
 - Every 10 steps, the amount of food of each animal decreases by 1.
 - Every 100 steps, the age of each animal increases by 1.
 - When an animal reaches the age of 120 and dies, the message will appear:
%AnimalName died in good health
 - When an animal dies due to lack of food, the following message will appear:
For fish:
**The fish %Name died at the age of %Age years
Because he ran out of food!**
For crab:
**The crab %Name died at the age of %Age years
Because he ran out of food!**
- 4) Take several steps - after selecting this option, the following message will appear:
How many steps do you want to take?
And the user will enter how many steps he wants to proceed. (The aquarium will advance a number of steps as required.)

- 5) Demo - Selecting this option will display a 120-step run when the aquarium has one animal of each type (4 animals in total). In the demo run, the animals must live at least 100 steps. Between each step there will be a 0.5 second pause and printing of the aquarium to the screen.
- 6) Print all - Selecting this option will display a list of all the animals in the aquarium. For each animal the following line will be printed:
For fish:
The fish %Name is %Age years old and with %Food food
For crab:
The crab %Name is %Age years old and with %Food food
- 7) Exit - Selecting this option will display the message: "Bye bye" and end the program.

The structure of the classes you need to use:



In addition, there is an Aqua class that manages the entire aquarium, among other things it contains an Animal-type array that contains all the animals in the aquarium

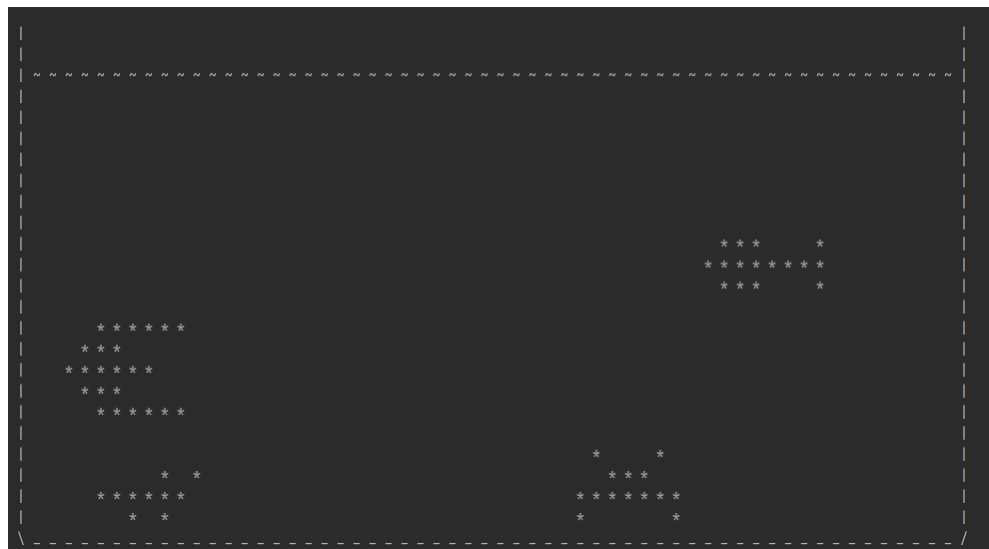
Directions for thought, where to write each method:

- All animals can move left and right
- Only fish can move up and down.
- The methods that should be for each class are the animal print on the board and the constructor.
- The amount of food of all the animals decreases similarly, as does their growth rate (over the years).
- All animals die for the same reasons and in a similar way.
- **To help you write the code, we have added files to all classes with the required method names to the work submission box.**

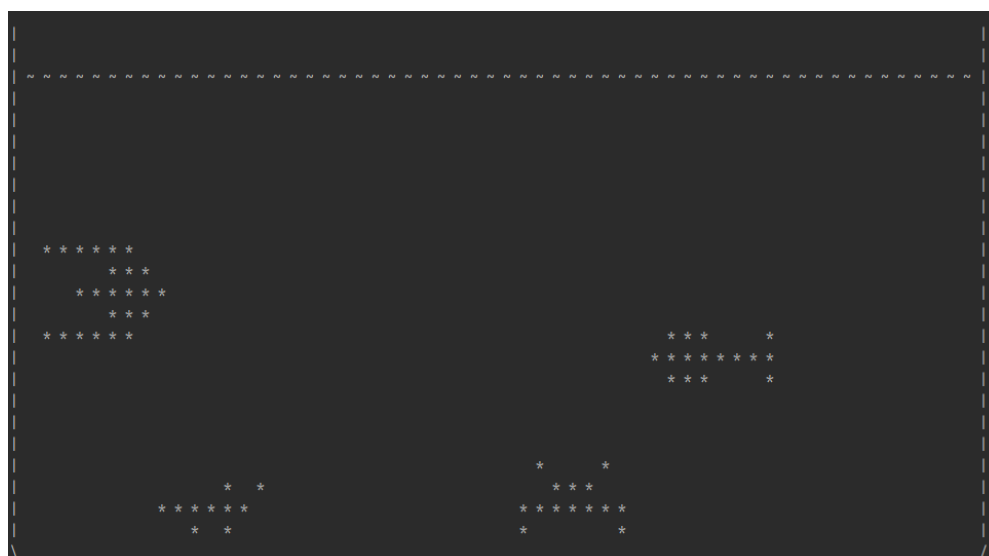
Pay Attention:

- You must use `input()` where the program needs to receive an input. This will allow the input from the auto test methods.
- If the input is invalid, the program must ask the user to re-enter the input (by repeating the original request)
- You are allowed to use built in methods that we did not study in class. (but not external libraries!)
- You need to use **"import time"** to pause between steps.
- You may **not** use other external libraries (no import statements allowed). This will be checked and grade will be reduced.
- Some tests will be visible for your convenience, others will not be and will be performed during the testing process.
- You must code this yourself. Similarity tests will be performed automatically and codes that are similar will be **automatically graded 0**.
- Good luck!!!

Example:

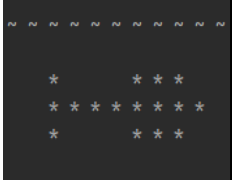
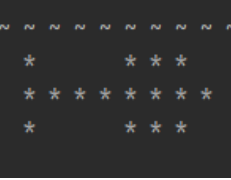
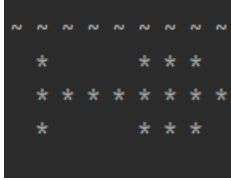
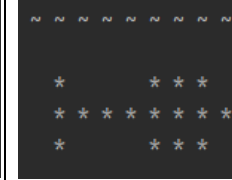
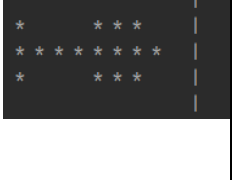
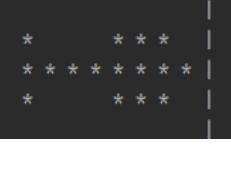
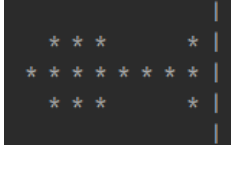

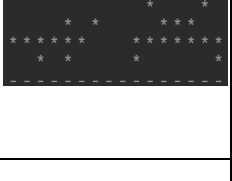
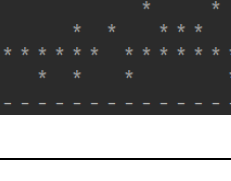
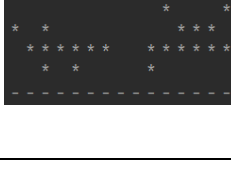


A few steps later:



Appendix

- When an animal collides with the aquarium, it first changes direction (without moving) and only in the next step does it move in the new direction.
- When crabs collide they change direction and move in the new direction in the same step.
- Fish swim on top of each other

Example \ step	1	2	3	4
1 From up to down				
2 From right to left				
3 Two crab collide				
4 Two fish	