Object Oriented Programming

—

Exercise series 4

Exercise 1 This short exercise illustrates how one can read and write on the `Standard Input`.

First, get a look at the file `Walter.java`, and try to figure out what the code does without running it. Use the `Java` documentation for the classes and methods you are not familiar with.

There is, however, a mistake in the program that prevents from exiting the *while loop*. Find it and correct it.

Exercise 2 A *simply linked list* is a data structure used to store a sequence of elements. It consists in a collection of nodes arranged sequentially, such that each node stores an element, as well as the location of the next node in the sequence (which does not exist in the case of the last one).

In Java, define a `Node` class suited for representing the nodes of a simply linked list.

Exercise 3 A *FIFO queue* is a data structure that represents a set of values accessed using a FIFO (*First In, First Out*) policy : the first value added to the queue will be the first to leave it.

Reusing the Node class from Exercise , define a Queue class implementing a FIFO queue with the help of a simply linked list. This class should provide the following interface:

- `put(v)`: appends a new value v to the queue.
- `get()`: extracts the first value from the queue.
- `first()`: reads the first value of the queue without removing it.
- `size()`: returns the number of values currently stored in the queue.

Notes :

- All operations involving a queue should be efficient! (In other words, they should run in $O(1)$ time.)
- Consider writing an auxiliary program for testing your code.

Exercise 4 Following the same approach as in Exercise 3, implement the Stack data structure studied in the theoretical part of the course with the help of a simply linked list.