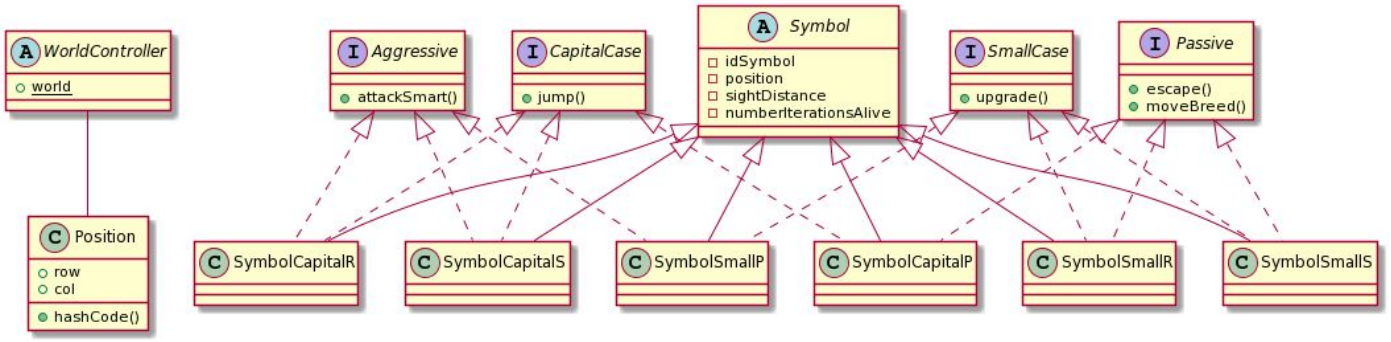
Introduction to Programming I 2020 - Assignment #2

Deadline: 13/11/2020

One fundamental concept in computer simulations is Rock-Paper-Scissors relationships. In this assignment, you are asked to implement an ecosystem involving symbols (R, r, P, p, S, s) in a 10x10 world. Different types of symbols implement different behaviours. Therefore, we can expect complex interactions happening in our simulated world.



Your project should utilize the classes and interfaces made available on Moodle. Some part of the code should not be changed, even if you need it. Also, don't use external libraries that will possibly cause issues during the grading of your work.

PS.: Hash Maps appear in the code provided and will be explained in a future lecture and/or lab.

1. Software Requirements

a. Extend the given abstract classes and implement the given interfaces. ALL your additional classes, code and Simulator class, EVERYTHING, should be in the package **simulator.yourname_yoursurname**

b. Implement the classes SymbolCapitalR, SymbolSmallR, SymbolCapitalP, SymbolSmallP, SymbolCapitalS, SymbolSmallS

c. The following rules apply for a meeting of symbols in the same position:

   i. If there's at least one P or p symbol at the same position as one or more R or r, all the R/r symbols are killed. This rule extends for other symbols: Paper (capital or small) > Rock (capital or small) > Scissor (capital or small) > Paper (capital or small)

   ii. For each pair of similar symbols (r and r, R and R, but not r and R) at the same position, a new symbol (always in the small case and never in the capital case, even if two capital symbols meet) is born in the world in an adjacent position

   iii. After a certain number of iterations, small symbols are upgraded to capital symbols

   iv. No symbol lives forever. After a certain number of iterations, capital symbols die naturally

   v. Symbols don't see the entire world, but only what is in their sight distance

   vi. At each iteration:

      1. symbols move randomly
      2. symbols die
      3. small case symbols upgrade
      4. capital case symbols jump
      5. passive symbols escape
      6. passive breed

7. aggressive symbols attack smartly

d. You are not allowed to change the files in **simulator.do_not_change**, but you can extend/implement and then use the subclasses
e. A certain number of pairs of each symbol start on the board. Some might disappear after several iterations (life is cruel)
f. The simulation ends after 1,000 iterations
g. Your project should be compatible with JDK 1.8

2. Assignment Requirements

a. Your program should somehow plot (screen, TXT file in the proper folder - see within the project, XML, JSON, etc) the world in each interaction
b. There should be a comment (/** ... */) before each method explaining what it does
    i. See https://www.tutorialspoint.com/java/java_documentation.htm
    ii. Good IDEs can generate a template automatically
c. Follow the given Class Diagram
d. The class with the main() method should be named as **Simulator**
e. Submit to Moodle ONLY the zipped (.ZIP) directory **simulator.yourname_yoursurname**

3. Grading schema (total of 10 points)

- 3 points for code quality (structure, naming, indentation, formatting, no extremely long methods, no magical numbers)
- 2 points for clear comments for java documentation according to the standard mentioned in (2b)
- 3 points for the general impression about the execution of the program, presentation, plotting and creativity
- 2 points for robustness. For example, replacing your class SymbolCapitalR by the same class implemented by one of your colleagues should not break your code.
- -5 for failing in implementing the Class Diagram
- -1 for missing (2e)
- Total grade equals to 0 if your program does not compile
- Total grade equals to 0 if you changed one of the classes/interfaces you were told not to change
- Total grade equals to 0 if caught in the automatic plagiarism check