



Assignment 2

Due date: Sunday 25 September, 2016 at 11:55 pm

Problem 1: Programming Languages

The purpose of this assignment is to create a Java model that allows you to generate working codes in two un-typed languages: **PHP** and **JavaScript**. We are providing you with a demo class and output samples. Your job is to create & implement the required classes.

Code Explanation:

- `PrintAction` is a class that takes a `String` variable `var` and generates the corresponding statement in PHP and js:
 - **PHP:**
`echo $var . "
";`
 - **JS:**
`document.write(var + "
");`
- `AssignmentAction` is a class that has two variables, a `target` of type `String` and a value of type `int`. Its job is to generate a statement with the following format:
 - **PHP:**
`$target = value;`
 - **JS:**
`target = value;`
- `ForLoop` is a class that has an integer variable representing the number of iterations and a body representing the code inside the for loop. Its job is to generate the for loop signature and body in the following formats:
 - **PHP:**

```
for($i=0 ; $i < n; $i++) {  
    PHP-statement;  
    PHP-statement;  
    ...  
}
```
 - **JS:**

```
for(var i=0; I < n; i++) {  
    JS-statement;  
    JS-statement;  
    ...  
}
```

Note that the body of the loop can be an `AssignmentAction`, another `ForLoop`, an `IfElseAction`, or a `Program`. ***In case of nested for loops, the indices of the loops should be different.***

- IfElseAction is a class that has a String condition of the form “var op int” (for example $x > 2$), a body for the if part, and an optional body for the else part. Note that the body of the loop can be an AssignmentAction, ForLoop, IfElseAction, or a Program. The job of this class is to generate the signature of the if-else statement and the corresponding bodies in the following formats:

- **PHP:**

```
if( $var op int) {
    Body of if
}
else {
    Body of else
}
```

- **JS:**

```
if(var op int) {
    Body of if
}
else {
    Body of else
}
```

- Program is a class that consists of a collection of the statements (classes) mentioned above i.e. PrintAction, AssignmentAction, ForLoop, IfElseAction, and Program.

All the classes above must contain a method generate that takes a String denoting the language (PHP or JS). The method should print the corresponding code given the language as argument. Note that, the generated code **should be properly indented**.

You can run your generated code online on the following links:

- PHP: <http://writecodeonline.com/php/>
- JS: <http://writecodeonline.com/javascript/>

On the next page are some sample runs.

Main program	Output
<pre> public static void main(String[] args) { Program p = new Program(); AssignmentAction assign1 = new AssignmentAction("x", 2); AssignmentAction assign2 = new AssignmentAction("y", 4); PrintAction p1 = new PrintAction("x"); PrintAction p2 = new PrintAction("y"); ForLoop forLoop1 = new ForLoop(3); ForLoop forLoop2 = new ForLoop(4); forLoop1.addAction(p1); forLoop2.addAction(p2); IfElseAction ifElseAction1 = new IfElseAction("x < 5", forLoop1, forLoop2); p.addAction(assign1); p.addAction(assign2); p.addAction(ifElseAction1); p.generate("php"); } </pre>	<pre> \$x = 2; \$y = 4; if(\$x < 5) { for(\$i0 = 0; \$i0 < 3; \$i0++) { echo \$x . "
"; } } else { for(\$i1 = 0; \$i1 < 4; \$i1++) { echo \$y . "
"; } } </pre>
<pre> public static void main(String[] args) { Program p = new Program(); AssignmentAction assign1 = new AssignmentAction("x", 2); AssignmentAction assign2 = new AssignmentAction("y", 4); PrintAction p1 = new PrintAction("x"); PrintAction p2 = new PrintAction("y"); ForLoop forLoop1 = new ForLoop(3); ForLoop forLoop2 = new ForLoop(4); forLoop1.addAction(p1); forLoop2.addAction(p2); IfElseAction ifElseAction1 = new IfElseAction("x < 5", forLoop1, forLoop2); p.addAction(assign1); p.addAction(assign2); p.addAction(ifElseAction1); p.generate("js"); } </pre>	<pre> x = 2; y = 4; if(x < 5) { for(var i0 = 0; i0 < 3; i0++) { document.write(x + "
"); } } else { for(var i1 = 0; i1 < 4; i1++) { document.write(y + "
"); } } </pre>
<pre> public static void main(String[] args) { Program p = new Program(); AssignmentAction assign1 = new AssignmentAction("x", 2); PrintAction p1 = new PrintAction("x"); ForLoop forLoop1 = new ForLoop(10); ForLoop forLoop2 = new ForLoop(10); forLoop1.addAction(forLoop2); forLoop2.addAction(p1); IfElseAction ifElseAction1 = new IfElseAction("x < 5", forLoop1); p.addAction(assign1); p.addAction(ifElseAction1); p.generate("php"); } </pre>	<pre> \$x = 2; if(\$x < 5) { for(\$i0 = 0; \$i0 < 10; \$i0++) { for(\$i1 = 0; \$i1 < 10; \$i1++) { echo \$x . "
"; } } } </pre>
<pre> public static void main(String[] args) { Program p = new Program(); AssignmentAction assign1 = new AssignmentAction("x", 2); PrintAction p1 = new PrintAction("x"); ForLoop forLoop1 = new ForLoop(10); ForLoop forLoop2 = new ForLoop(10); forLoop1.addAction(forLoop2); forLoop2.addAction(p1); IfElseAction ifElseAction1 = new IfElseAction("x < 5", forLoop1); p.addAction(assign1); p.addAction(ifElseAction1); p.generate("js"); } </pre>	<pre> x = 2; if(x < 5) { for(var i0 = 0; i0 < 10; i0++) { for(var i1 = 0; i1 < 10; i1++) { document.write(x + "
"); } } } </pre>

Problem 2: Robot War

In this problem, you will have to write a program to simulate a fight between *soldiers* and *robots*. The battleground is assumed to be a grid that can be represented as a 2D array, where a cell in the grid can be `null`, or it can contain a *soldier*, a *robot*, or a *mine*.

Soldiers are of two types: *offensive* soldiers and *defensive* soldiers. An offensive soldier performs an action by placing a mine in his current position and then moving randomly either up, down, left, or right. **Hint:** First generate the random move, and then check if the move is allowed before performing the move. Here are the move rules of an offensive soldiers:

- A move is allowed if the new position in the grid is either null or has a robot. If the new position is null, update the soldier's position i.e. the old position becomes a mine and the new position becomes the soldier. If the new position has a robot, both the soldier and the robot collide and die immediately i.e. programmatically you should replace the new position in the grid with `null`; the position becomes a mine.
- A move is not allowed if the new position contains a soldier, a mine, or if the soldier crosses the boundaries of the grid. In this case, the move is ignored.
- Note that, if the soldier couldn't move for any reason, he does not place the mine.

A defensive soldier performs an action by killing all the robots that are to his direct left, right, up, and down. He does not move

Robots are also of two types: *smart* robots and *dumb* robots. A smart robot performs an action by moving randomly up, down, left, or right, but it avoids mines when doing so, i.e. if the new position has a mine, another robot, or it crosses the grid boundaries, the move is ignored. If the new position is null, the robot moves to the new position. If the new position has a soldier, both the robot and soldier die immediately (replaced with `null` as discussed above).

Similar to the smart robot, a dumb robot performs an action by moving randomly up, down, left, or right. The only difference is that the dumb robot does not avoid mines. If it steps on a mine, both the mine and the robot are destroyed immediately (i.e. set to null).

Create a game, represented as a `Game` class that simulates a battle between soldiers and robots. The `Game` class contains a 2D array of size $n \times n$, where n is a parameter provided by the user that is greater than or equal to 2. When a game is created, you should create n soldiers (randomly chosen as defensive and offensive) and n robots (randomly chosen as smart or dumb). Place the n soldiers in the first row of the grid and place the n robots in the last row of the grid.

Now it is time to play: go through the 2D grid, ***one column at a time visiting every cell***. If the cell is occupied by a soldier or a robot, perform the corresponding action. Note that the game might not end or either the soldiers will win (all robots are dead) or a tie will happen (all soldiers and robots are dead).

Use the driver below when testing your code. Create appropriate print statements to printout the status of the game.

```
public class RobotWar {  
    public static void main(String[] args) {  
        // use parameters 2, 3, 4, and 5 for testing  
        Game firstGame = new Game(4);  
        firstGame.run();  
    }  
}
```