P2 - Exercise hour

Pooja Rani

2019-05-31
Exercise 10 Hints

- **Split lines:**
  ```plaintext```
aTurtleProgram lines
```
- **Split by whitespace:**
  ```plaintext```
aLine splitOn: Character space
```
- **Conditionals:**
  ```plaintext```
  (command = 'right') ifTrue: [ turtle right: amount ]
```
- **Regular expressions:**
  ```plaintext```
  'up 15' matchesRegex: '(left|right|up|down) \d+'
```
- ...
Sample exam questions

- What is the pattern of questions?
- How to approach the questions?
Why do god classes and data classes often occur together?
When should you call `super()` in a constructor and why?
What is iterative development, and how does it differ from the waterfall model?
What are the advantages of using the Model-View-Controller pattern?
Terminology

- You should be aware with all Object-oriented concepts.
- You should know what is the role of each concept.
Fix these JavaDoc comments.

/*
 * The <i>Algorithm</i> defines how a value
 * for a file is computed.
 * It must be sure that multiple calls for the
 * same file results in the same value.
 * The implementing class should implement
 * a useful toString() method.
 */

public interface Algorithm {
    // ...
}
Design By Contracts

Write JavaDoc comments for the given method.

```java
/*
 *
 * /
 public int updateAlgorithm(String name, int left) {
   // ...
 }
```
Write Dbc for the following methods.

```java
/*
 *
 * /
 public int updateAlgorithm(String name, int left) {
   // ...
 }
```
Design Patterns

Explain the observer pattern on an example use case of your choice. Include the following in your answer:

▶ Provide example code.
▶ Provide an UML diagram of the classes involved.
▶ State one advantages and one disadvantage of using the Observer pattern to implement a GUI. Use less than 100 words.
Explain the observer pattern on the given example use case. Include the following in your answer:

- Example code is provided, add something in existing code.
- Provide an UML diagram of the classes involved.
- State one advantages and one disadvantage of using the Observer pattern to implement a GUI. Use less than 100 words.
Design Patterns

You should be able to do this for all the patterns from the lecture and from the lab, for example, adapter, proxy, Composite, observer, null object, builder, chain of responsibility.. (and more!)
Write a JUnit test that verifies that line 10 works as expected.

```java
public class Spreadsheet {
    private int[][] contents;
    private int rows;
    private int cols;

    /** JavaDoc omitted */

    public void setCellValue(int row, int col, int value){
        if (row < 0 || row > this.rows-1) {
            throw new IllegalArgumentException();
        }
        if (col < 0 || col > this.cols-1) {
            throw new IllegalArgumentException();
        }
        this.contents[row][col] = value;
    }
}
```
What is the Law of Demeter? Does the following code satisfy the Law of Demeter? If not, where does it violate it?

```java
/**
 * Play the game with the given scripted player.
 * // more JavaDoc omitted.
 */

public void runWithScriptedPlayer(ScriptedPlayer player){
    assert isValidGame();
    Queue<Command> commands = player.getInputQueue();
    while (!isOver() && !commands.isEmpty()) {
        execute(commands.top());
        commands.pop();
    }
    if (isOver()) {
        print("The player successfully solved");
    } else {
        print("The player failed to solve");
    }
}
```
Explain what the following Smalltalk code result into and why?

```smalltalk
rows: rows columns: columns tabulate: aBlock
| a i |
a := Array new: rows*columns
i := 0.
1 to: rows do: [ :row |
    1 to: columns do: [ :column |
        a at: (i := i+1) put:
            (aBlock value: row value: column) ] ].
^ a
```
This is just a selection of topics.

Everything that was covered in the lectures and exercises can appear in the exam.
Final Remarks

▶ Please fill out the (official) evaluation form after the exam!
▶ Check whether you got the Testat.
▶ The exam takes place on Wednesday, 5 June, 10:00–12:00 (You get 90 minutes!) in ExWi A6