utbm
université de technologie
Belfort-Montbéliard

LP24 Object Oriented Programing
Final Exam - Spring 2016

Documents, Computer, Calculator unauthorized

⇨ Theoretical questions (3 pts)

A. What is the purpose of interfaces? How to write a class which implements an interface in java? How many interfaces can implement a class?

B. What is polymorphism? How is it made in Java?

C. Explain the difference between a class and an object.

⇨ Exercise #1: I-Robot (6 pts)

Let consider a class named Robot which is aimed at modeling the state and the behavior of a virtual robot. Each robot has got:
- a name,
- a position in a 2D world,
- an orientation which could be North, South, East or West

It can go one step forward or backward, can turn of 90 degrees left, can print its state on screen (name, position, orientation)

A. Write the class Robot strictly respecting data encapsulation.
B. One want to improve these robots by creating a new generation (RobotNG) that can coexist with the old models. Basically, they can do the same but, they can go forward several steps in one time step, can turn right of 90 degrees and can make a u-turn. Write this new class:
   1) using calls to former methods.
   2) accessing directly to the robot state.

C. All robots are put into a standard array. How can we print on screen the state of all robots?

⇨ Exercise #2: ObservableWorldList (7 pts)

Goal: We want to create a new type of collection: the "ObservableWordList". This collection is close to the standard « java.util.List » but it differs in that the contained elements are necessarily of type « String » ; and it's possible to be informed with events when new words are added in the list.

This new collection must only exhibit the following methods:
- to add a new word at the end of the list
- to add a new word in the list at a specified index

To store data of this new collection, we will use an object of type « ArrayList ». It's forbiden to use another kind of object in place of this ArrayList.

✶ Question 1: Write in Java the source code of this new class.
Explain you design choices between the inheritance-based approach where you extend from the class « ArrayList » or the aggregate-based approach where you have a dedicated attribute to store the list.

One of the particularity of this new collection, is to trigger events when a new word is added to the list.

✶ Question 2: Complete the source code of question 1 to manage this new functionality.
You could create new classes or interfaces if needed...
All modifications must be clearly visible by changing writing color or by rewriting the entire program.

✶ **Bonus question:** With the minimal effort, provide a version of this program able to manage a list of any kind of object, not only "String".
Once again, these modifications must be clearly visible in your previous source code or write it again.

⇨ Exercise #3: Correct this source code (4 pts)

The following Java source code contains many mistakes. Underline the errors on the sheet containing the source code and for each suggest a correction.

Exercise #3

```
package fr.utbm.lp24;

public class StringPair throws ClassCastException {

	private final String data1;
	private final String data2;
	private final int data1Lenght;
	private final int data2Length;

	private enum DATA {
		DATA1,
		DATA2,
	}

	/**
	 * Constructor
	 */
	private StringPair(String data1, String data2) {
		this.data1 = data1;
		this.data2 = data2;
	}

	public void setData(DATA d, Object value) {
		if (value instanceof String) {
			if (d == DATA1) data1 = value;
			else data2 = value;
		} else {
			if (d == DATA1) data1 = value.toString();
			else data2 = value.toString();
		}
	}

	/**
	 * Return in a list the length of the two elements of this pair
	 */
	public List<int> getDataLength() {
		List result = new List<>();
		if (data1Length == null && data2Length == null) {
			data1Length = data1.length();
			data2Length = data2.length();
		}
		result.add(data1Length);
		result.add(data2Length);
		return result;
	}

	/**
	 * Inverts the two elements of this pair
	 */
	public final void invert() {
		data1 = data2;
		data2 = data1;
		return;
	}

	/**
	 * Return an array containing the two elements of this pair
	 */
	public String[] getArray() {
		String result[] = new String[2];
		result[1] = data1;
		result[2] = data2;
	}
}
```

Signature :