Advanced Software Development DE Midterm Exam

May 10, 2014

PRIVATE AND CONFIDENTIAL

1. **Allotted exam duration is 2 hours.**
2. **Closed book/notes.**
3. **No personal items including electronic devices (cell phones, computers, calculators, PDAs).**
4. **Cell phones must be turned in to your proctor before beginning exam.**
5. **No additional papers are allowed. Sufficient blank paper is included in the exam packet.**
6. **Exams are copyrighted and may not be copied or transferred.**
7. **Restroom and other personal breaks are not permitted.**
8. **Total exam including questions and scratch paper must be returned to the proctor.**

7 blank pages are provided for writing the solutions and/or scratch paper. All 7 pages must be handed in with the exam

BE VERY CAREFUL WITH THE GIVEN 2 HOURS AND USE YOUR TIME WISELY. THE ALLOTED TIME IS GIVEN FOR EVERY QUESTION.

Write your name and student id at the top of this page.

Question 1 [30 points] {30 minutes}

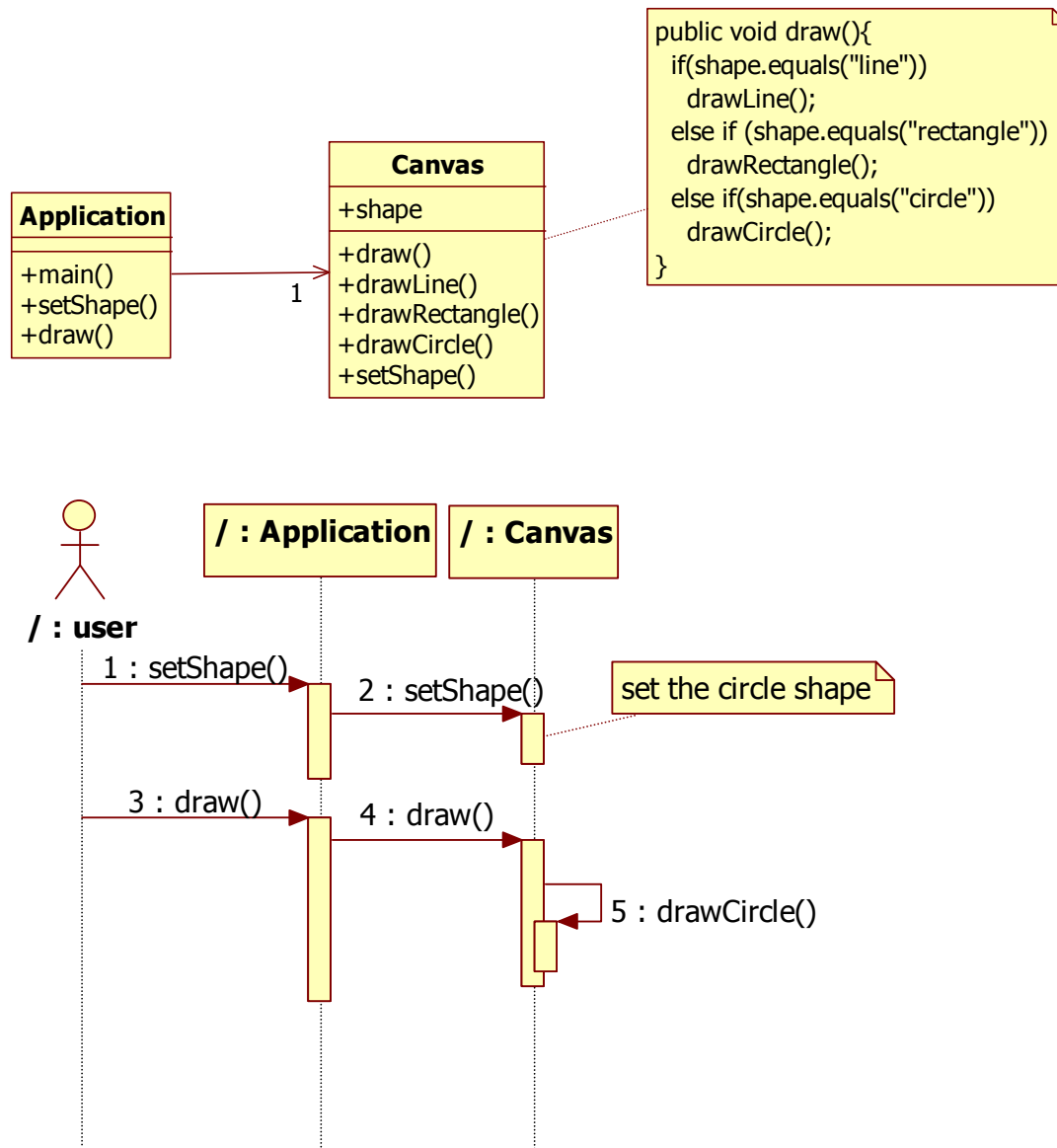Draw the class diagram of a product catalog application with the following requirements:

- You can add and delete different product categories (like electronics, books, cars)
- For every category we store its name.
- A product category can have subcategories (and subcategories can also have subcategories, etc.)
- You can add and delete products in/from categories
- For every product we store its productnumber, name and price.
- You can add customer reviews to products
- You can view all reviews for a certain product
- A review consists of a description and a certain integer number between 1 and 5 that shows how much stars a product can get from a customer. 1 star means that the customer was not very happy with the product and 5 stars means the customer was very happy with the product.
- All the products in this product catalog have a certain owner
- Because these products change from owner very quickly, we can see the history of all previous owners of a certain product
- All data is stored in a database.

Your design should implement the appropriate design principles we studied in the course.

Make sure you add all necessary UML elements (attributes, multiplicity, etc.) to communicate the important parts of your design. IMPORTANT: Do not write methods in this class diagram (for this question only).

Question 2 [40 points] {50 minutes}

Suppose we have written the following drawing application:

```
public void draw(){
  if(shape.equals("line"))
    drawLine();
  else if (shape.equals("rectangle"))
    drawRectangle();
  else if(shape.equals("circle"))
    drawCircle();
}
```

Application

+main()
+setShape()
+draw()

1

Canvas

+shape

+draw()
+drawLine()
+drawRectangle()
+drawCircle()
+setShape()

/ : Application   / : Canvas

/ : user

1 : setShape()

2 : setShape()

set the circle shape

3 : draw()

4 : draw()

5 : drawCircle()

The problem with this solution is that the canvas class needs to change every time we add a new shape. We need to redesign our application such that our new design meets the following requirements:

1. It should be easy to add new shapes without changing the canvas class
2. The algorithms to draw a certain shape should be reusable by other classes
3. We should be able to undo and redo our previous drawing actions.

a) Redraw the given class diagram so that our new design meets the given requirements. **Make sure you add all necessary UML elements (attributes, methods, multiplicity, etc.) to communicate the important parts of your design.**

b) Redraw the sequence diagram with the following scenario:
1. the actor first draws a circle on the canvas
2. The actor then calls undo

Make sure you add all necessary UML elements to communicate the important parts of your design.

c) Explain clearly in your own words how we can implement the **undo** functionality of this application. For example, we first draw a line, then a circle and then the rectangle:
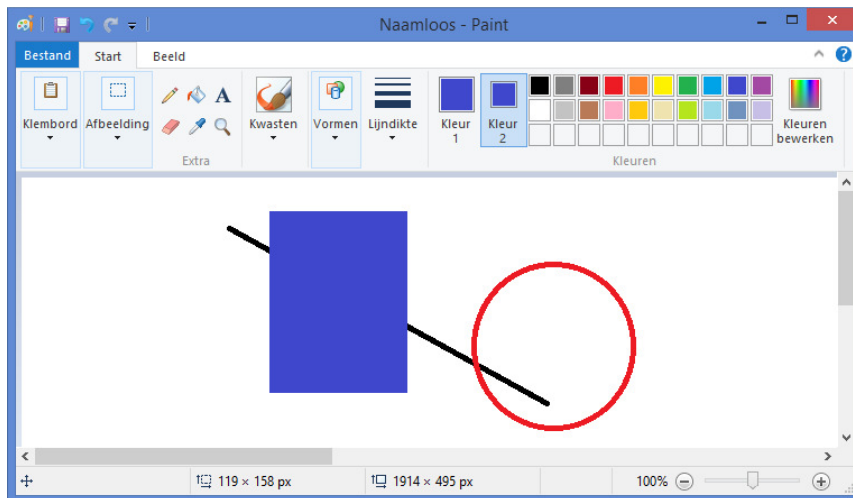
Then if we execute the undo action, the rectangle is removed from the drawing:
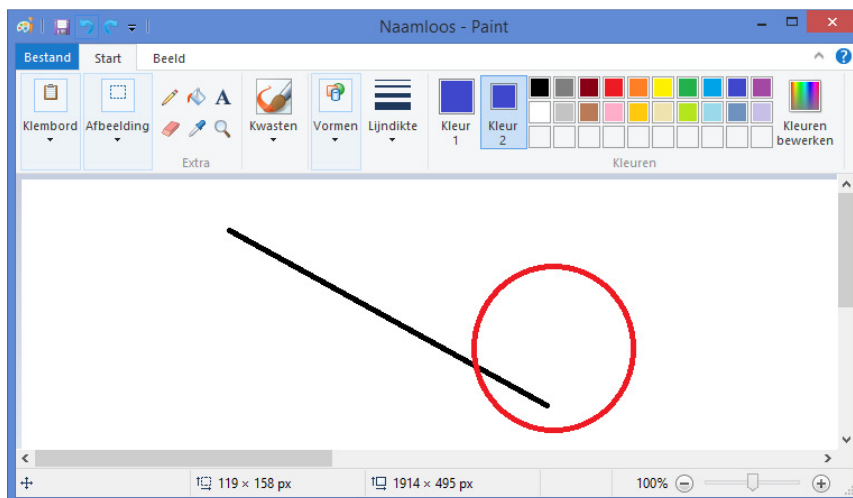
Question 3 [25 points] {30 minutes}

Suppose we need to design a racecar game for a game console. The game console contains a display and buttons:

For our left hand we have only one button.

For our right hand we have 2 buttons: up and down.

The buttons work as follows:
The game starts in the speed mode.
If you are in the speed mode and click the left button, you move to the steering mode.
If you are in the steering mode and click the left button, you move to the turbo speed mode.
If you are in the turbo speed mode and click the left button, you move to the speed mode.

If you are in speed mode, and you click the up button, then you increase your speed with one mile/hour.
If you are in speed mode, and you click the up button, then you decrease your speed with one mile/hour.
If you are in steering mode, and you click the up button, then turn left.
If you are in steering mode, and you click the up button, then turn right.
If you are in turbo mode, and you click the up button, then you increase your speed with three miles/hour.
If you are in speed mode, and you click the up button, then you decrease your speed with three miles/hour.

Your colleague starts to design this game, and he shows you the following simple design:
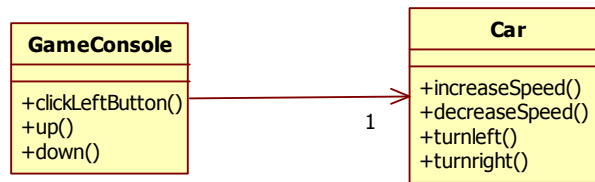
When you look at the design you think back at the design principles you learned in your studies, and you notice that this simple design is not very flexible. It is very likely the game will evolve in time and future releases might need support for more game modes (like beginner mode, expert mode, etc.).

Draw the class diagram of a new design of this race game that allows us to add new game modes easily.
Make sure you add all necessary UML elements (attributes, methods, multiplicity, etc) to communicate the important parts of your design.

Question 4 [5 points] {10 minutes}

Describe how we can relate the **Service class** to one or more principles of SCI. Your answer should be about half a page, but should not exceed one page (handwritten). The number of points you get for this questions depends how well you explain the relationship between the **Service class** and one or more principles of SCI.