16.04.19

Lab Report

Info 1

Exercise 0: Figures

Name:	Pavel Tsvyatkov
E-Mail:	Pavel.Tsvyatkov@Student.HTW-Berlin.de
Lab-Partner:	Martin Adam
E-Mail:	Martin.Adam@Student.HTW-Berlin.de

Lab report index

1. Overview

Exercise 0 was about objects and classes in Java. For preparation we had to find out more about primitive data types in Java and there was an exercise where we had to write the types of some given values. Also we had to think about if a book is a class or an object and give our reasoning. Then there was a constructor given and we had to count its formal parameters and name their types.

For the lab we had to get BlueJ started and download the project files called „figures“ from Moodle. The first thing was getting BlueJ started, we had the program on the taskbar so we started it from there. After that we downloaded the figure.zip from Moodle and tried to open it in BlueJ, but it didn’t work. We had to extract the .zip file first so BlueJ could open the project. In BlueJ then there were five classes Circle, Square, Triangle and Person as figures which were all connected to the Canvas class. After that we opened the exercise webpage and started with the tasks.

The pdf was produced by first writing everything in a Word file and then converting the Word document to PDF with the help of Adobe Acrobat DC. This program has a built-in word to pdf converter.

2. Lab exercises

2.1 Differences and similarities between Persons, Circles, Squares and Triangle

We spent around ten minutes discussing about the first exercise. We had some issues from the beginning, because we didn’t know if we are allowed to create new objects of the classes. That was mentioned in one of the following tasks,so we just thought about what these figures have in common in general and that in BlueJ they were all classes which are connected to the Canvas class. Then we created objects of the figures and tried to find out what the figures had in common. To create an object we had to right-click for example on the Circle class and select new Circle(). We are then prompted to select a name, but the default name was fine. The object appears in the bottom left of the BlueJ window, which is called the object bench. Double-clicking on an object in the object bench opens the inspection window which shows its fields and values, we also call that the object's **state**. We immediately started being curious and saw that all the classes had x and y position fields of type int, however with different default values. They all had a field for color of type String, but with a different default value. For example the Circle class had blue as default value for color, but the Square class had red. They all had a field which was of type boolean and stated whether the object is visible or not. We noticed that the Circle class was the only one with a diameter field, the Square class was the only one with a size field and the Triangle class had fields for width and height.

2.2 How is a Person similar to Circles, Triangles, Squares

The second exercise was easier than the first and we spent just two or three minutes. After we created a person1 object, we inspected it and noticed that it also has fields for height and width, just like the triangle. The person1 object also had a field of type boolean about whether it's visible or not and a field of type String for the color like the other figures.

2.3 Creating a Circle, a Triangle, two Squares and a Person

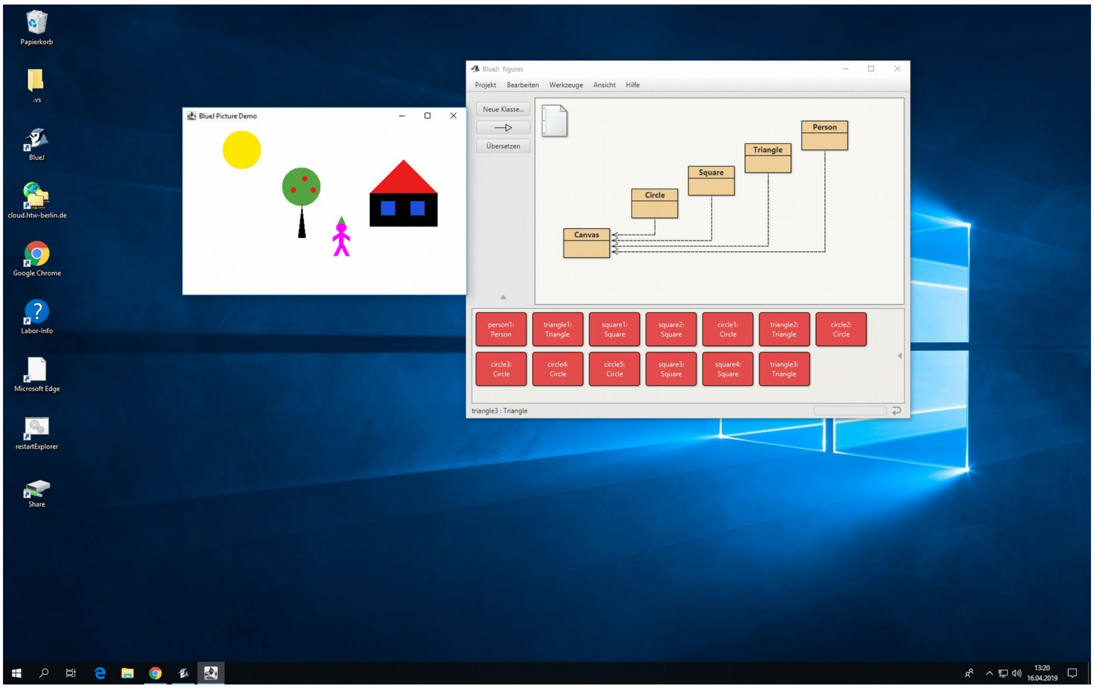
For the third exercise we needed around ten minutes. We had to create a circle, a triangle and two squares. As we did before we needed to right-click on the class we wanted an instance of and for example select new Circle(). We created a circle, a triangle, a person and a square this way. We then had to right-click and invoke the void makeVisible() method of each object in the object bench. We noticed that after making them visible, they all had different default colors. The square was red, the circle was blue, the triangle was green and the person was black. After that we created a second square object and passed it the name square2. We made it visible by calling the void makeVisible method, but we noticed that there wasn't a second square on the picture. That was because both of the squares had the exact same default position and also the same default color when we created them. To make the second one visible we had to change its position. We did that by invoking the void moveRight() method of square2 several times until there was a distance between the two squares. We can also move the objects up, down, left, horizontally and vertically by calling the respective methods. However, to move an object horizontally or vertically we needed to input an integer value to move a distance of our choice.

2.4 Changing all figures to the same color

The fourth exercise where we had to change all figures to the same color took us around five minutes. By invoking the void changeColor() method of each object, a window appears where we need to pass the object a value of type String for the color and it says we have several allowed colors. We need to be careful that we use the double quotes when we pass a color and that we use colors which are allowed. If we assign a different from the mentioned colors, BlueJ just passes the default color which is black.

2.5 Creating an interesting picture with at least seven objects

To set up our picture we needed around thirty minutes. We decided to use a total of 13 objects. We had to change the color of every object by invoking the changeColor() method and assigning it the value for the color we wanted. First we decided that our circle1 object will be the sun and changed its size by calling the changeSize() method and assigning it a diameter value big enough to be a sun. After that we called the changeColor() method and gave it the value "yellow", which changed the object's color to yellow. After that we had to call the moveUp() and moveLeft() methods several times until we aligned it well to serve as a sun for our picture. For our house we used two squares and a triangle and for the windows we needed two additional squares. First we had to invoke the changeColor() method of square1 and give it the value "black". Then we called the changeSize() method of square1 to change the object's size. We did the exact same thing with our square2 object, but we also had to call the moveUp() and moveRight() method until it was well aligned right next to square1, so they form a big rectangle together. To perfectly align it to square1, we had to call the moveHorizontal() method and give a certain value as a distance. When we used the moveHorizontal() method, we had to give a positive value to move the object to the right and a negative value to move it to the left. Next thing we needed for our house was the roof. We called the changeColor() method of triangle1 and gave it the value "green". We then invoked the changeSize() method and we had to assign two different values for width and height. Again by calling the moveUp() and moveHorizontal() methods, we aligned it well with our house. We then created two more squares for the windows. By calling the same methods like for the other squares, we changed their color to blue, gave them smaller value for the size and moved them so that they fit well on the house. After that we created our tree. We used a triangle and a few circles. We had to call the changeColor() and changeSize() methods of the objects we needed for the tree to make it as we want and then align them together by calling the moveUp() and moveLeft() methods. Last thing in our picture was the person. We assigned our person object a magenta color and used a small green triangle for the hat. After that our picture was ready.



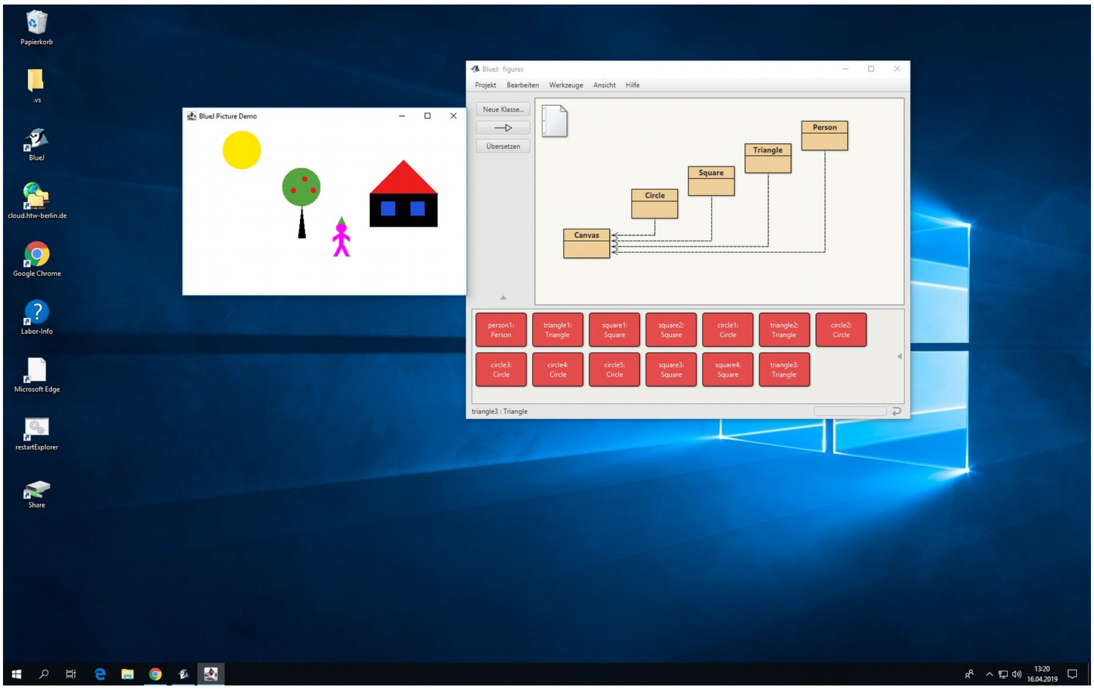
2.6 Errors, mistakes and how things worked or didn’t work while creating the picture

This part took us around five to ten minutes. We wanted the trunk of the tree to be brown, but when we passed it the brown color BlueJ gave it the default black color. After that we created an object and made it visible, but then we decided we don't need it and wanted to remove it. We deleted the object, but it was still visible on the picture. We realized that now there is no way we can delete the figure from the picture, because we deleted the object which had the method to make it invisible. In the end we decided to save our project. We navigated to the bar in BlueJ and clicked on Project, from there we chose to save it in the home directory and it worked without problems. We wanted to check if everything is fine with our project file and opened it. We were very surprised, because the project loaded fine, but all of our created objects were missing. It was still okay, because we had already made our screenshot before that, but we learned something new in that moment.

3. Reflection of what I learned

While working on the figures exercise I learned that after creating an object and making it visible, if I delete the object from the object bench, then there is no way anymore to make the object invisible or interact with it in any way. If I have an object that I don't need, I need to first make it invisible and then delete it from the object bench. I learned that when I save a project and try to open it after, all the objects that i created in this project are no more there. From the pre-lab exercises I learned about the eight primitive data types.

4. Appendix



Screenshot of our scene in BlueJ. We made the screenshot by holding the Windows key and pressing „Druck". We then included it in our lab report by pressing Strg + V (paste) and formatting as we wanted.

Pre-lab exercises

P1. In the lecture we have talked about data types called int and String. Java has more predefined primitive data types. What are they? Record where you found this information.

The other primitive data types, except int and String, are: byte, short, long, float, double, boolean, char

I found the information here:

https://docs.oracle.com/javase/tutorial/java/nutsandbolts/datatypes.html

P2. What are the types of the following values?

0 – int	"hello" – String	101 – int
-13 – int	true – boolean	"true" – String
"61" – String	'7' – char	3.1415 – float

P3. Pick up a book—you should have at least one at home. Is it an object or a class? If it is a class, name some objects. If it is an object, name the class. Give your reasoning.

I think the specific book I picked up is an object, because there are many other books with different titles, genres and color. The class should be Books and every other specific book is an instance of class Books, which has its own color, title or genre.

P4. What class does the following constructor belong to? How many formal parameters does it have? What are their types?

public Book (String title, double price)

The constructor belongs to class Book and it has two formal parameters one of type String and one of type double.