

Worksheet 2

MSc/ICY SOFTWARE WORKSHOP

Assessed Worksheet: 2% of the module mark.

Submission Deadline is Thursday, 25 October, at 2pm via Canvas.

5% late submission penalty within the first 24 hours. No submission after 24 hours. Follow the submissions guidelines on Canvas. JavaDoc comments are mandatory.

All submissions must pass the tests provided on 18 October.

Exercise 1: (Basic, 30%) Define a class `Book` and a constructor to create it. A book should be constructed from the three field variables `title`, `year`, and `isbn` of types `String`, `int`, and `String`, respectively. Implement getter methods

- `public String getTitle()`,
- `public int getYear()`,
- `public String getIsbn()`, and a setter method
- `public void setTitle(String title)` that sets the title to `title`.

Furthermore write a

- `public String toString()` method that is used for printing objects of class `Book` in a user friendly way. (Note, the `toString()` method in this exercise WILL NOT BE tested, that is, you have flexibility how to write it.)

Note that you have always to comment and test your programs appropriately, not just for this exercise and not just for this worksheet. We will not write this to the exercises in future.

Exercise 2: (Medium, 30%) Assume that a `BankAccount` has three fields `private int accountNumber`, `private int balance`, and `private int setupFee`. You are supposed to write a constructor

- `public BankAccount(int accountNumber, int initialDeposit, int setupFee)`,
- getters for all three fields, and
- a setter for the balance. Furthermore write a
- `toString()` method that is used for printing objects of class `BankAccount` in a user friendly way. (Again, the `toString()` method in this exercise WILL NOT BE tested.)

The initial deposit is the amount of money that the customer pays in when they open their account. The setup fee is charged to the account upon construction.

For instance, assume you create an account with account number 54 by

```
BankAccount marysBankAccount = new BankAccount(54, 100, 20);
```

This means that after paying in £ 100 and being charged £ 20 for the setup of the account, Mary's account will have a balance of £ 80.

Exercise 3: (Advanced, 30%)

Write a class `Employee`. Each employee is represented by their `name`, their `monthlyGrossSalary`, and their `taxRate` (as a percentage) of types `String`, `double` and `double`, respectively. Write a class with

- a constructor,
- getters,
- setters, and a
- `toString()` method (which should return a `String` as specified in the example below).

Note that the naming of constructors, getters, and setter must follow the strict naming convention. Furthermore write two methods:

- `public double monthlyNetSalary()`, which computes for an `Employee` object their monthly net salary.
- `public void increaseSalary(double rate)`, which increases the monthly salary by the rate (as a percentage) given in the argument. That is, with a rate of 3 the monthly salary will be increased by 3 per cent.

For example, let us assume an employee

```
Employee john = new Employee("John", 2400, 20);
```

`System.out.println(john);` should give us:

```
"John has a monthly gross salary of £2400, a taxrate of 20%, and a monthly net salary of £1920."
```

After calling `john.increaseSalary(1)`, John's monthly salary is increased by one per cent and will be £2424 rather than £2400.

Note, in order to display the pound symbol you should NOT use the pound symbol on the keyboard, but the corresponding unicode symbol, that is, `\u00A3`.

Exercise 4: (Debugging, 10%) You have the task to evaluate the following piece of code which has problems. Submit the improved code with an assessment of the original code as a comment at the start.

```
/**
 * The following code is buggy. Why?
 *
 * We define a rectangle by the three field variables width, height,
 * and area, each of type double. Furthermore, we write getters and
 * setters for the three fields as well as a toString method. We test
 * it in a main method.
 *
 * @version 2018-08-31
 * @author Manfred Kerber
 */

public class Rectangle {

    private double width;
    private double height;
    private double area;

    /**
     * <pre>
     *
     *          width
     * +-----+
     * |               |
     * |   area = width * height   | height
     * |               |
     * +-----+
     * </pre>
     * @param width The width of the rectangle.
     * @param height The height of the rectangle.
     * @param area The area of the rectangle.
     */
    public Rectangle(double width, double height, double area) {
        this.width = width;
        this.height = height;
        this.area = area;
    }

    /**
     * Getter for the width.
     * @return The width of the rectangle is returned.
     */
    public double getWidth() {
        return width;
    }

    /**
     * Getter for the height.
     * @return The height of the rectangle is returned.
     */
    public double getHeight() {
        return height;
    }
}
```

```

/**
 * Getter for the area.
 * @return The area of the rectangle is returned.
 */
public double getArea() {
    return area;
}

/**
 * Setter for the width. The width of the rectangle is updated.
 * @param width The new width of the updated rectangle.
 */
public void setWidth(double width) {
    this.width = width;
}

/**
 * Setter for the height. The height of the rectangle is updated.
 * @param height The new height of the updated rectangle.
 */
public void setHeight(double height) {
    this.height = height;
}

/**
 * Setter for the area. The area of the rectangle is updated.
 * @param area The new area of the updated rectangle.
 */
public void setArea(double area) {
    this.area = area;
}

/**
 * @return A human readable description of the rectangle in form
 *         of the three field variables specifying it.
 */
public String toString() {
    return "The rectangle has a width of " + width +
        ", a height of " + height +
        ", and an area of " + area + ".";
}

/*
 * main method with a test of the setHeight setter and the
 * toString method.
 */
public static void main(String[] args) {
    Rectangle r = new Rectangle(2, 4, 8);
    System.out.println(r);
    r.setHeight(5);
    System.out.println(r);
}
}

```