© Manfred Kerber
Alexandros Evangelidis

School of Computer Science
University of Birmingham
8 October 2019

Worksheet 2

MSc/ICY Software Workshop

Assessed Worksheet: 3% of the module mark (5% for the 20cr version).
Submission Deadline is Tuesday, 22 Oct 2019, at 12:00 noon via Canvas.
Follow the submissions guidelines on Canvas. JavaDoc comments are mandatory.
All submissions must pass the tests provided on 15 Oct. For Exercises 1 – 3 submit own tests.

Note that in the following you should follow the strict naming convention for the constructors, getters, and setters.

Exercise 1: (Basic, 30%)

- Write a class `Car` with the four field variables `make`, `price`, `maxSpeed`, and `colour` of types `String`, `int`, `int`, and `String`, respectively.
- Write a constructor to create `Car` objects.
- Implement all getter and setter methods.

Furthermore write a

- `public String toString()` method that is used for printing objects of class `Car` 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%)

- Write a class `Film` with the three fields `private String title`, `private Date releaseDate`, and `private int length`. `Date` should be the class defined in the lecture.
- Write a constructor `public Film(String title, Date releaseDate, int length)`.
- Write getters for all three fields.
- Write a setter for the `releaseDate`.
- Furthermore write a `toString()` method that is used for printing objects of type `Film` in a user friendly way. (Again, the `toString()` method in this exercise WILL NOT BE tested.)

In the `main` method create an example object `Film adAstra` with title `"Ad Astra"`, release date `new Date(18, "September", 2019)`, and length `123` minutes.

Exercise 3: (Advanced, 30%)

- Write a class `Good`. Each `Good` is represented by its `name` and its `netPrice` of types `String` and `double`, respectively. The class should contain
- a constructor,
- all getters,
- all setters, and a
- `toString()` method (which should return a String as specified in the example below).
- Furthermore assume a fixed VAT rate of `20` percent stored in a `final static double VAT_RATE` variable. Use this to write a method `public double grossPrice()`.

For example, let us assume a good `Good milk = new Good("Milk", 0.50);`.
`System.out.println(milk);` should give us:
`"The Milk has a gross price of £ 0.60."`

- Write a method `public void discount(double rate)` that reduces the price of the object by the given rate (between 0 and 100 percent). For instance, after calling the method `milk.discount(20);` `System.out.println(milk);` should give us:
`"The Milk has a gross price of £ 0.48."`

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 perimeter, 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 2019-09-26
 *  @author Manfred Kerber
 */
public class Rectangle {
    private double width;
    private double height;
    private double perimeter;
    /**
     *  <pre>
     *                width
     *  +--------------------------------------+
     *  |                                      |
     *  |                                      |
     *  |                                      |  height
     *  |                                      |
     *  |                                      |
     *  +--------------------------------------+
     *  </pre>
     *  @param width The width of the rectangle.
     *  @param height The height of the rectangle.
     *  @param perimeter The perimeter of the rectangle as 2
     *  times the width plus the height.
     */
    public Rectangle(double width, double height, double perimeter) {
        this.width     = width;
        this.height    = height;
        this.perimeter = perimeter;
    }
    /**
     *  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 perimeter.
     *  @return The perimeter of the rectangle is returned.
     */
    public double getPerimeter() {
        return perimeter;
    }
    /**
     *  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 perimeter. The perimeter of the
     *  rectangle is updated.
     *  @param perimeter The new perimeter of the updated rectangle.
     */
    public void setPerimeter(double perimeter) {
        this.perimeter = perimeter;
    }
    /**
     * @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 a perimeter of " + perimeter + ".";
    }
    /*
     *  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, 12);
        System.out.println(r);
        r.setHeight(5);
        System.out.println(r);
    }
}
```