© Manfred Kerber
Alexandros Evangelidis

School of Computer Science
University of Birmingham
5 November 2019

Worksheet 4

MSc/ICY Software Workshop

Assessed Worksheet: 3% of the module mark (5% for the 20cr version).
Submission Deadline is Tuesday, 19 Nov 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 12 Nov. For Exercises 1 – 3 submit own tests.

<u>Exercise 1:</u> **(Basic, MSc: 30%, ICY: 40%)** Write a class `Aircraft.java` with the two field variables `passengerNumber` and `maxSpeed` both of type `int` with a standard constructor, getter(s), setter(s), and a `toString` method. Furthermore write a subclass `Aeroplane.java` with the additional field variable `fuelConsumption` of type `double`. The class should contain a constructor, getter(s) and setter(s), and a `toString` method. Make use of inheritance as far as possible.

<u>Exercise 2:</u> **(Medium, 30%)**
Let two interfaces for the two classes `Subscription.java` and `SubscriptionPrime.java` be given as `SubscriptionInterface.java` and `SubscriptionPrimeInterface.java`. You find the interfaces in the archive `WS1-04-interfaces.zip` on the Canvas page for the worksheet.
`Subscription.java` describes the subscription to a journal by the 3 field variables `title`, `email`, and `cost` of types `String`, `String`, and `int`, respectively. The subclass `SubscriptionPrime.java` has the additional field `deliveryAddress` of type `String`, indicating where hard copies of the journal should be sent to. Make use of inheritance as far as possible.

<u>Exercise 3:</u> **(Advanced, 30%)** Assume a website for downloading music which has four classes, one for representing a music title, `MusicTitle.java`, and three for user accounts. The class `Account.java` is `abstract` with two subclasses `AccountStandard.java` for customers, and `AccountAdministrator.java` for administrators. Each of the four classes is to implement a corresponding `interface` with the names `MusicTitleInterface.java`, `AccountInterface.java`, `AccountStandardInterface.java`, and `AccountAdministratorInterface.java`, respectively.
You can find the interfaces in a `WS1-04-interfaces.zip` file on the Canvas page for the worksheet.
You are supposed to write the four classes to add a (non-interactive) implementation of a very rudimentary system.
The class `Account` has to have five field variables: `name`, `salutation`, `email`, `password`, and `loggedIn` of types `String`, `String`, `String`, `String`, and `boolean`, respectively.
The subclass `AccountStandard` has the field variables `balance`, `titlesBought`, and `failedLoginAttempts` of types `int`, `ArrayList<MusicTitle>`, and `int`, respectively. Furthermore, assume a fixed number of 3 maximal login attempts, stored in a `public static final int MAXIMAL_LOGIN_ATTEMPTS` variable.
The subclass `AccountAdministrator` has the field variable `accounts` of type `ArrayList<Account>`. An administrator shall have the possibility to reset an account of a user by a method
`public void resetAccount(Account account, String password)`. The method shall first reset the password. For standard users (and only for standard users) it shall also reset the counter
`failedLoginAttempts` of failed login attempts to `0`.
For the latter, you may make use of `account instanceof AccountStandard` to check whether the object `account` is an instance of the subclass `AccountStandard`. If it is, it is still of type `Account`, but it can be cast to the type `AccountStandard` using the standard type casting approach: `(AccountStandard) account`.
Make use of inheritance as far as possible.

<u>Exercise 4:</u> **(Executive Summary, MSc Students Only 10%)** Write an executive summary on the topic: “What is the rationale for the use of object oriented programming and which limitations are overcome by the use of functions in Java?” The submission must be an accessible PDF document of two pages (including title, author, date, and at least three references) using an 11pt font.