© Manfred Kerber
Ian Batten
Joseph Gardiner

School of Computer Science
University of Birmingham

Worksheet 4

MSc/ICY Software Workshop

Assessed Exercise: 20% of this term's continuous assessment mark.

Submission: Sunday 30 November 2014 5pm

5% late submission penalty within the first 24 hours. No submission after 24 hours. JUnit tests and JavaDoc comments are mandatory. No tests will be provided for this worksheet. You will arrange a viva with your tutor. Follow the submission guidelines on `http://www.cs.bham.ac.uk/internal/courses/java/msc/submission.php`.

Exercise 1: (Basic, ICY: 40%, MSc: 30%)

(a) Write a class `Patient` with the following field variables `name`, `dateOfBirth`, and `gender` (which should take only the two values `"M"` or `"F"`). Write a constructor, the usual getters and setters, a `toString` methods and an `equals` method. If the constructor or the `setGender` method is not called with either `"M"` or `"F"` an exception should be thrown.

(b) Write a sub-class `PatientExtended` of the previous class with the additional field variable `lastSeen` of type `Calendar`. Assume that a medical practice keeps a list of all its patients in an `ArrayList<PatientExtended>`. Write a method that lists all patients that have not been seen within a year.

Exercise 2: (Medium, ICY: 30%, MSc: 20%) In the lecture (see `http://www.cs.bham.ac.uk/internal/courses/java/msc/handouts/1-08/README.html`) we have seen an abstract class `Employee` with two subclasses `HourlyEmployee` and `SalariedEmployee`. Adapt these classes so that the salary (and the `Payable`) are of type `double`. Add to the `Company` class a field variable `ArrayList<Employee> employees` with the usual getter and setter. Add to the `Company` class a method `public void increaseSalaries(double rate)` which increases the salaries of all employees (whether paid on an hourly rate or with a fixed monthly salary) by the fixed rate. (Note, it makes sense to have corresponding `increaseSalary` methods in the three classes dealing with employees.)
E.g., with a pay increase of 0.02, somebody earning £ 10 per hour would then earn £ 10.20 per hour and somebody earning £ 1800 a month would then earn £ 1836 per month. That is, your method should go over the whole `ArrayList` of all `employees` and increase all their salaries.

Exercise 3: (Advanced, ICY: 30%, MSc: 20%) Assume a real estate agent keeps lists of all the properties they sell or let. Distinguish between houses and apartments, and whether they are for sale or for rent. Each property goes with an asking price (or rent amount), the number of bedrooms, a picture, and some free text describing it. Write a program that generates two `html` files (see `http://www.cs.bham.ac.uk/internal/courses/java/msc/handouts/1-05/HtmlPages.java`), one for the properties to rent, one for the properties to buy, each sorted with the cheapest properties first.

Exercise 4: (Advanced, ICY: 0%, MSc: 30%)
THIS EXERCISE IS FOR MSc STUDENTS ONLY:

Explain in an executive summary the rationale for the use of object oriented programming (that is, which problem(s) does it address and how) and its limitations (to which degree are the problem(s) not fully solved this way). The explanation should be factual, well argued and supported by references. For answering this question, you should do background reading and make appropriate use of referencing the material that you have read.
Your executive summary – to be included in your zip file – should consist of two A4 pages with point size 11 and be submitted in accessible PDF format.