Christian-Albrechts-Universität zu Kiel
Institut für Informatik
Lehrstuhl für Echtzeitsysteme und Eingebettete Systeme

Prof. Dr. Reinhard von Hanxleden
Christoph Daniel Schulze, Steven Smyth

Institut für Informatik

1. Übung zur Vorlesung „Programmierung“

First Steps
Abgabe am Sonntag, 06. November 2016 - 23:55

This is the first set of homework assignments for you to work on, so let's make sure you know about the following things:

- You have just over a week to submit your solutions to the iLearn system; after the submission date, the system won't accept submissions anymore.
- Besides working at home, you should use the practical classes to work on your homework assignments. There, our staff is available for you to ask questions.
- To solve these assignments, remember that you will need to download the ACM library. See this guide for details on how to set everything up to get working.

Further comments, from the class homepage:

- Each assignment consists of a number of problems to solve, and each problem is weighted with a number of points. The points of the problems of each assignment (including bonus problem) add up to 100 points.
- We recommend that you give all problems at least a try. At the minimum, however, you must hand in "solutions" (see below) for problems that add up to at least 52 points *in each series*, except one series. (You can think about why 52 . . .) You must also hand in a solution for the project. An assignment may include one or two "mandatory problems" that by themselves have around 50 points. A mandatory problem does not have to be particularly difficult, however, it is particularly important for us that you solve that specific problem.
- An assignment may also contain a "Bonus Problem" that is more difficult than the others. You are not expected to solve this, but we of course invite you to try.
- A *solution* is a meaningful program that shows that you yourself seriously worked on the problem (see also "plagiarism" below). Usually there is an automatic test that your program has to pass, often there is an additional requirement such as writing a comment that explains how your program relates to the concepts taught in class. (Hint: these comments are of particular importance to us . . .)
- Again, if you have problems fulfilling this requirement (e.g. due to extended illness), contact us asap and we will find a solution. If you follow class, spend reasonable effort on the homework and use the practical classes to clear up any remaining issues, you should have no trouble to fulfill this requirement.

Aufgabe 1 - Getting Everything Working 1 Punkt

The first example of a Java program can be found in Figure 2-1 in our book. It is called `HelloProgram` and prints the words `hello, world` to the screen. This is admittedly a bit boring, because everyone uses those words for the first program in any given programming language. That's why you're going to change what the program prints to the screen. The real goal of this assignment, however, is to get a first program running with Eclipse.

Do the following:

1. Create a new project in the *Package Explorer*. We don't really care about the project's name, but you will probably want to choose a name that makes clear that it's the project that contains the solution to assignment 1 of the first set of assignments.

2. Create a new class called `HelloProgram` and fill it with the example in Figure 2-1.
3. Finally, make the class slightly more interesting by replacing the text `hello, world` with your favourite movie quote. Everyone has a favourite movie quote!

To hand in your solution, either copy-and-paste the entire source code you've written, or upload the `.java` file.

Aufgabe 2 - Three From Two — 49 Punkte

Figure 2-2 in our programming book shows a program called `Add2Integers` that asks the user to input two numbers and displays their sum. Write a program called `Add3Integers` that asks the user to input three numbers and displays their sum.

Make sure that your program passes the automatic test. *Hint:* The test requires the text "The total is" at the beginning of the output line, followed by the result.

Write a Java comment at the beginning of the program in which you answer the following questions.

1. Which variables do you use? For each variable individually, write what their three properties are according to what you learned in the lecture.
2. What classes does your program text mention? What is a superclass of what, what is a subclass of what?

Aufgabe 3 - Architecture — 49 Punkte

Our programming course does not just teach you the basics of programming. We also teach other important skills. In this assignment, for example, you will learn everything there is to be learned about architecture.

To that end, write a class `Architecture` that extends `GraphicsProgram` and draws a picture of a simple house with a peaked roof, two windows, and a door with a circular doorknob, pretty much like this one:

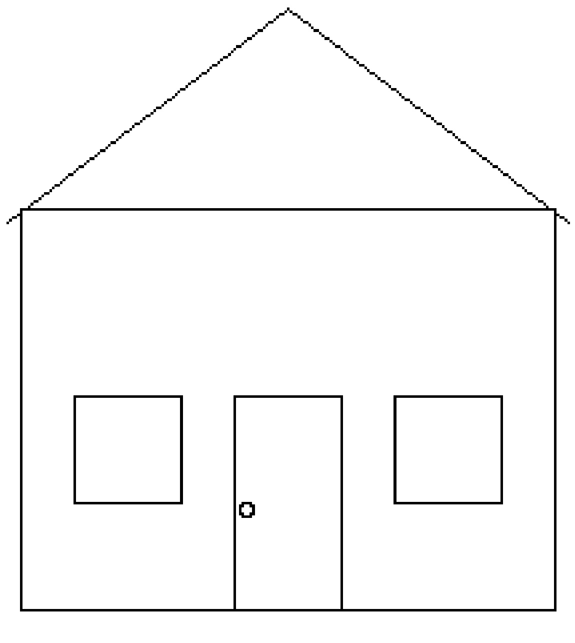

Abbildung 1:

Chapter 2 contains a lot of example programs you can look at for inspiration. For each of the components of the house, you should ask yourself which of the graphical objects you have at your disposal (`GLabel`, `GRect`, `GOval`, and `GLine`) is the best fit.

Zusatzaufgabe 4 - Advanced Drawing 1 Punkt

(Remember that this is a more difficult task aimed at students that already have Java experience. You are not expected to solve it, but of course are invited to try.)

Write a program `Polygon` that extends the class `GraphicsProgram` that, upon execution:

- Asks the user for a number n
- Use `GLine`s to draw a polygon with n corners regularly distributed over a circle.

For $n = 15$, it could look like this:

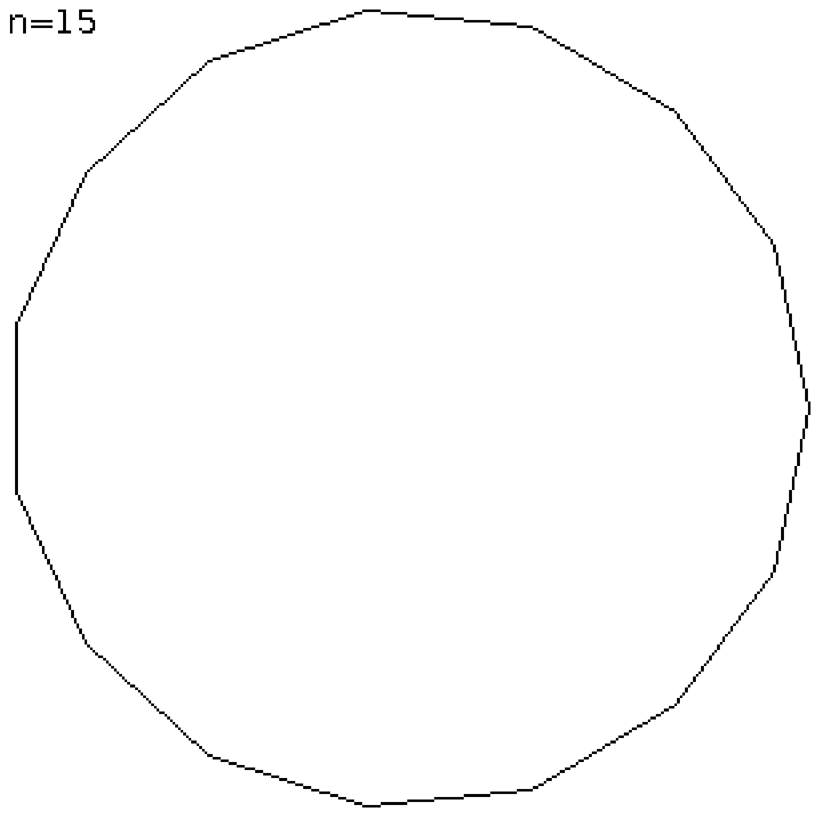

Abbildung 2: