Lab Sheet: C# Fundamentals

This lab sheet covers the sections on C# Fundamentals and C# Objects and Classes from the course on *Industrial Programming* (F21SC): http://www.macs.hw.ac.uk/~hwloidl/Courses/F21SC#slides. The tasks should be performed on the Windows lab (EM 2.45) machines in the department.

C# Fundamentals (Week 2)

Relevant Slides

- Define `weekday` as an enumeration type and implement a `nextday` method
- Implement a `whatday` method returning either `workday` or `weekend` (use another enum)
- Write a method calculating the sum from 1 to n, for a fixed integer value n
- Write a method calculating the sum over an array (one version with foreach, one version with explicit indexing)
- Use the `setstep` method to implement a method `set0`, which sets all array elements to the value 0.
- Implement a method, reading via readline, and counting how many unsigned short, unsigned int and unsigned long values have been read.
- Define complex numbers using structs, and implement basic arithmetic on them.
- **Implement Euclid's greatest common divisor algorithm as a static method over two `int` parameters.**
- **Implement matrix multiplication as a static method taking two 2-dimensional arrays (of any size) as arguments.**

C# Objects and Classes (Week 3)

Relevant Slides

- Implement the bank account example as discussed in the C# slides on classes
- Complete the points example and implement access to the x- and y-fields, using direct access, public methods, and (automatic) properties, respectively.
- Use inheritance and overloading to define a method area, that works on different shapes, namely circles, rectangles and squares.
- Let the user decide how many of these objects to construct, with which parameters, calculate the overall and per-shape area and print it
- Write a (polymorphic) function that takes an array of shapes and calculates the total area covered by all elements.
- Modify the ReadLine exercise from the previous lecture to generate instances of classes containing an unsigned short, unsigned int and unsigned long field, respectively.
- Implement basic arithmetic on complex numbers using operator overloading.
- **Implement the data structure of binary search trees with operations for inserting and finding an element.**

In-lab demo of the bold-face exercises, **gcd, mat-mult, trees**, in Week 3.