Prof. Dr. Christoph Bockisch
MSc Steffen Dick
Fachbereich Mathematik und Informatik
AG Programmiersprachen und -werkzeuge

Philipps Universität Marburg

Companion exercises
Objektorientierte Programmierung: Wintersemester 2021/2022

No. 6, due until 06.12.2021

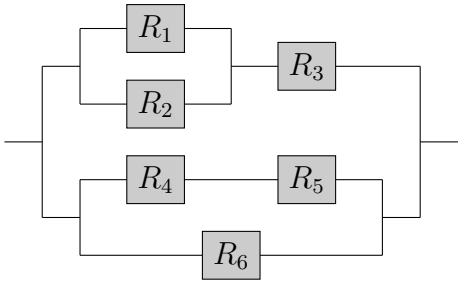
Task 6.1: Tron: Uprising — 3 Points

The resistance R of a circuit consists of two resistances, R_1 and R_2. A circuit can be connected in two different ways and be calculated as follows:

$$R = R_1 + R_2 \text{ Series Circuit}$$
$$R = \frac{R_1 * R_2}{R_1 + R_2} \text{ Parallel Circuit}$$

a) Implement a method `double seriesCircuit(double rOne, double rTwo)` that computes the total resistance of a given series circuit.
b) Implement a method `double parallelCircuit(double rOne, double rTwo)` that computes the resistance of a given parallel circuit.
c) Test your code by computing the total resistance of the circuit below. Use the following resistances in your computation:

$$R_1 = 60\Omega,\ R_2 = 40\Omega,\ R_3 = 50\Omega,\ R_4 = 50\Omega,\ R_5 = 70\Omega,\ R_6 = 80\Omega$$



Task 6.2: Optimus Prime | 5 Points

Professor Doctor Abdul Nightingale has a novel idea to compute prime numbers. He thinks that he can compute prime numbers by using other prime numbers. To do that he firstly writes all the numbers up to a chosen number, *n*, on a sheet of paper. He already knonws that 0 and 1 are not prime numbers, so he crosses them out. Next, he crosses out every number that is a multiple of 2, the smallest prime number. He then takes the next number that is not crossed out and crosses out every multiple of that number. This, he does for every number that is smaller than $\sqrt{n} + 2$. Every not crossed out number has to prime, according to his theory at least.

Always remember: Knowledge is night!

a) Implement Professor Nightingale's idea. Write a method `int[] primesUpTo(int n)` that returns an Array of all prime numbers up to and including *n*.

   Make sure that no negative numbers are accepted for *n*.

b) Test your implementation by calculating prime numbers up to 100. You may use `int[] primes = {2, 3, 5, 7, 11, 13, 17, 19, 23, 29, 31, 37, 41, 43, 47, 53, 59, 61, 67, 71, 73, 79, 83, 89, 97}` for testing purposes.

Task 6.3: El Psy Congroo | 4 Points

The mad scientist Hououin Kyouma needs a method to calculate the k-root of a number for his research. Generally, the k-root of a number can be approximated by using Heron's algorithm:

$$\sqrt[k]{a} = \begin{cases} x_0 = a \\ x_{n+1} = \frac{1}{k} * ((k-1) * x_n + \frac{a}{x_n^{k-1}}) \end{cases}$$

The algorithm completes when $|x_n - x_{n+1}| < d$ whereas `d` stands for delta. In this case it means that `d` is an upper bound and the margin between two computed values needs to be below that bound for the algorithm to complete.

a) Implement a method `double krt(double a, double k, double d)` that calculates the k-root by using Heron's algorithm. The method *krt* should only call a recursive help-method `double krtH(double a, double k, double d, double x_n)` with correct starting values.

   Attention: Use `x_n` to save the latest calculated value. Negative values for any argument should return 0 in *krt*.

b) Implement `double krtH(double a, double k, double d, double x_n)` in such a way that it computes Heron's algorithm the way it is described above.

c) Test *krt* with at least three different, positive values for *a*, *k* and *d*.