h_da	Fachbereich Elektrotechnik und Informationstechnik	Advanced Programming Techniques (APT) Module Exam WiSe2024 (PO2019) Prof. Dr. Lipp

Name	Matr.-Nr.	Computer	Signature (*)

(*) I hereby declare that I am able and willing to take the examination with my signature.

Rules:

- Use the prepared project “Exam-2024WiSe”.
- **In `main.cpp` replace the place holders with your name and matrikel number.**
- Compile your project once (there may be warnings) and execute it. Check the output.
- Access to
- The snapshot of cppreference.com that you have on the desktop does not have a search facility. However, it has a link “std Symbol Index” which comes close. Make sure that you have found the link on the index page.
- Create all classes in the folder „`myCode`“.
- “`using namespace`” statements in header files are forbidden.
- Code that implements methods (and functions) must be written in the proper *.cpp-file (no implementation code in *.h-files).
- Not using defined methods when appropriate (copying code instead) results in a reduction of points.
- Leaving automatically generated, not required code such as default constructors in the code results in a reduction of points.
- Code will only be graded up to the first compiler error.
- Code in comments will not be graded.
- Answers to questions will only be accepted when provided in the specified comments. Answers found elsewhere in your code will gain no points, even if they are correct.

Hints:

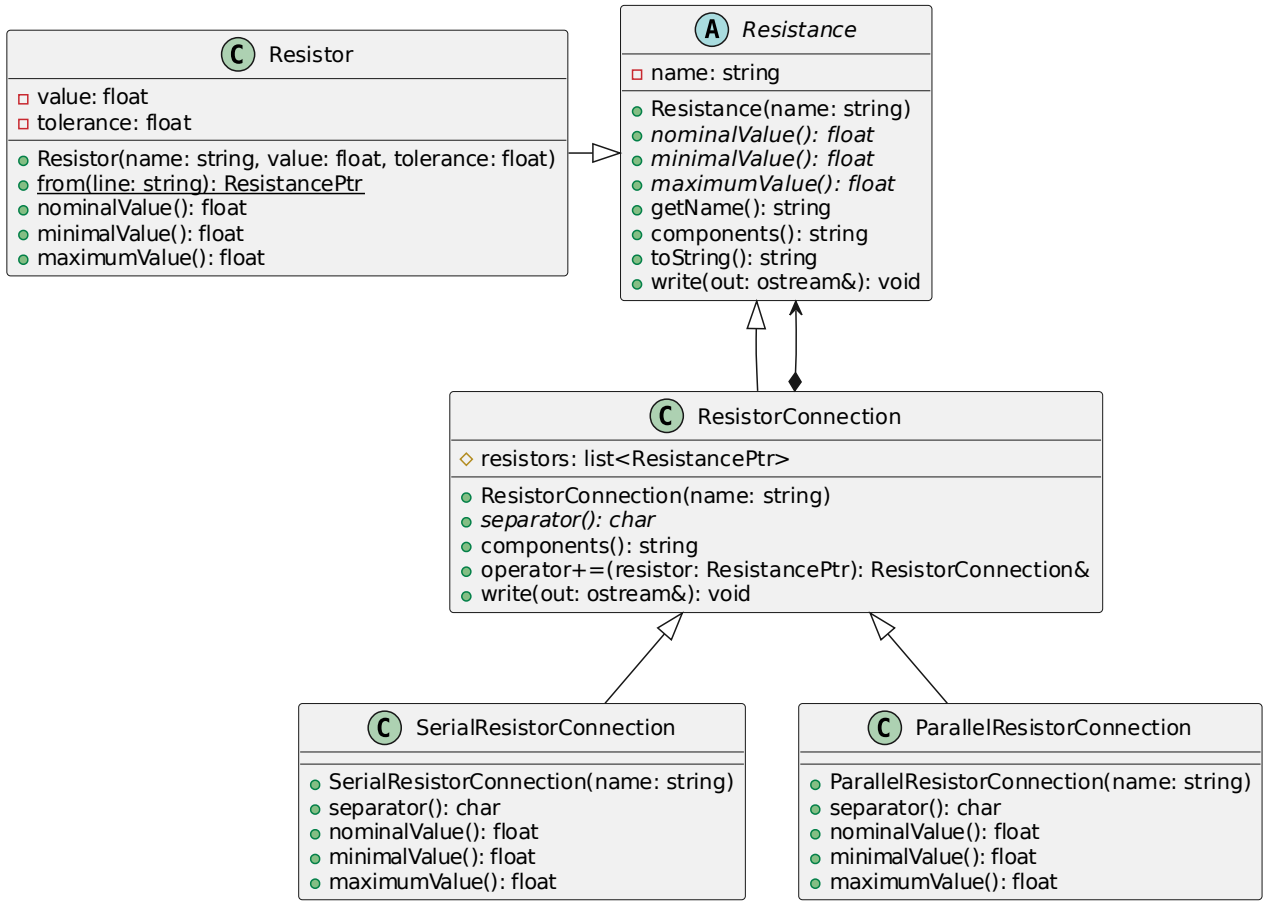
- Writing unit tests for classes can be done before the implementation and is a possibility to get acquainted with the classes and the problem domain. The effort required to familiarize yourself with the task of this exam is therefore taken into account in the task of writing unit tests where you get comparatively many points for little code.

Allowed auxiliary means:

- The slides from the lecture (including print-outs of the uploaded code samples)
- Up to three books about C/C++ programming
- A dictionary.

Communication with other participants or using unauthorized auxiliary means, including but not limited to electronic devices such as smart phones, smart watches calculators, will result in immediate termination of the exam and a 0 point grading.

Topic of this exam is the modelling of resistor networks (see picture below). The general interface to any resistor network is specified by class `Resistance`. An instance may simply be a physical resistor (class `Resistor`). Another possibility is to have a connection of two or more resistors, either a serial connection or a parallel connection.



Reminder: pure virtual (abstract) methods are shown in italics. Prefix "#" indicates a "protected" member. These members (declared in a "protected:" section) can only be accessed by the class itself and derived classes. (If you have problems implementing this, you can also put the member declarations in the public section.)

Exercise

Implement the missing methods according to their description in the header files. Adding to or modifying the classes' attributes or methods is forbidden, except where stated in the comments.

Implement the tests outlined in file "tests.cpp". Note that "assert that …" means that you have to write one or more `assertTrue(...)`-statements. Unless there is an error, running your tests must not produce any output on the console.

Note

The points that you get for implementing a particular method or test can be found in the methods' comments. The total number of points that you can achieve is 100.