



Professional Software Engineering

Lecture 3: Exercises

Exercise 1 Your First Repository

In this exercise, we will set up our first git repository. Start by going to <https://gitlab.lrz.de/> and logging in using your TUM username and password (your username will be similar to *ge12abc*). Then follow the steps presented below.

1. Create a new project: *New Project* → *Create Blank Project*. Set the visibility to *internal* and initialize the repository with a README file.
2. Once your repository is created, click on “clone” and copy the link under “Clone with HTTPS”
3. Open Git Bash on your PC and type in “*git --version*”. You should get an output similar to “*git version 2.27.0.*”, otherwise, git is not installed correctly on your PC.
4. Configure git with your credentials by using the following commands.

```
git config --global user.name "your_username"  
git config --global user.email "your_email@mytum.de"
```

You can check your credentials by typing in the following command.

```
git config --global --list
```

5. To clone your repository, type in the following command.

```
git clone https://gitlab.com/your_repositorys_name.git
```

6. Type in “*cd your_repositorys_name*” followed by “*start .*” to open up your repository in file explorer.

Once your repository is set up, change your README.md file and commit the changes to your repository.

Note: Since you set your repo’s visibility to internal, anyone with an LRZ account can view your repository. You can change your repository’s visibility to private, however, you will need to input your password while cloning it.

Exercise 2 Interfaces

In a new branch of your git repository, create a C# interface “*IVector*” to represent vector objects. Implement this interface in two new classes: “*Vector2D*” and “*Vector3D*” that should represent two and three-dimensional vectors of type *double*, respectively.

The interface should include the following:

1. A method to calculate the Euclidean (L_2) norm of the vector. The Euclidean norm of a vector \underline{v} ($||\underline{v}||_2$) is given by eq. (1).

$$||\underline{v}||_2 = \sqrt{v_i v_i} = \sqrt{v_x^2 + v_y^2 + v_z^2 \dots} \quad (1)$$

2. A method to print the vector object to the console.

Once you are done, compare your solution to last week's Exercise 5.

Bonus: Create a complex vector class that implements *IVector*. Use the complex class you created during last week's exercise session (Exercise 1).

Exercise 3 Unit-Tests

Create a new branch from your Interface branch (Exercise 2) and use it to add unit-tests (in a separate file) to your project.

Hint: Start with small tests and work your way up. Try to think of extreme cases and be creative; try to break your code *i.e.* find scenarios that can cause your tests to fail.

Important Note: Make sure that the following packages are installed in your Visual Studio before using NUnit.

1. Microsoft.NET.Test.Sdk
2. NUnit
3. NUnit3TestAdapter

Bonus Question

Unit test coverage is a metric used to evaluate the percentage of the program that was tested by a set of unit-tests. The metric can include *function coverage*, the number of functions tested, *statement coverage*, the number of tested code statements, *code coverage* etc.

Do you think this is a good metric? Can you trust a program based solely on this metric (assuming all the unit tests, with a 100% coverage, pass)?