**Testing in a Continuous Delivery world**

**Practical Part**

**Before you start**

1. Sign up to [Github](https://github.com/)
2. Confirm your e-mail address.
3. Send me a direct message with your **e-mail and** **login** (I will add your user to our work repository).
4. Confirm invitation from Github to enable access to the repository.
5. Make sure you have had already installed following program on your computer for practical part:

* Git - <https://gitforwindows.org/>

**Class task 1.** [**Recording Changes to the Repository**](https://git-scm.com/book/en/v2/Git-Basics-Recording-Changes-to-the-Repository)

1. Open any Command Line Interface on your computer:

1st way (recommended) - Open git bash

2nd way - Run cmd.exe

3rd way - open PowerShell

1. Go to the directory where you are going to make your code:

cd <path>

1. Set up your name and e-mail address (use the same e-mail as for github):

git config --global user.name "*FirstName LastName*"

git config --global user.email *address@example.com*

1. Clone the existing Git repository:

git clone https://github.com/Stani-G/fk2019autumn.git

1. Restart PowerShell/cmd/gitbash. Check the directory in your local machine - make sure that fk2019autumn repository is copied. Go to directory where you store repository:

cd fk2019autumn

1. Get the full list of remote references:

git remote show origin

1. Create your txt-file in your local fk2019autumn directory with any text (you can use any text editor):

notepad *YourFullName*.txt

1. Check the state of your working directory and the staging area:

git status

1. Begin tracking your file. Check the state of your working directory and the staging area:

git add *YourFullName*.txt

git status

1. Commit your changes with message like ‘Add my own file to the branch - *YourFullName’*:

git commit -m “Add my own file to the branch - *YourFullName*”

1. Check the state of your working directory and the staging area:

git status

1. Fetch and integrate your local branch with remote repository

git pull

1. Update remote refs along with associated objects

git push

TIP: if you fall into the situation when Git Bash opens a text file, you can:

1. Press ESC
2. Type the following command “:x!”
3. press ENTER

**Class task 2. Branching and Merging**

1. Switch to the branch ‘Master*’*:

git checkout master

1. Fetch and merge the changes from the remote branch to your local branch:

git pull

1. Create a new branch and switch to it:

git checkout -b *test-<YourName>*

1. Edit existing html-file in your local directory:

notepad *YourFile.html*

*(e.g. atom imosiichuk.html)*

List of files:

boliynyk.html

fvasiliuk.html

adzhuryak.html

ayushchuk.html

ogospodarska.html

vtymchyshyn.html

rlevytskyi.html

sstruchok.html

imanachyn.html

obezhnar.html

vostapchuk.html

vyanets.html

ivovk.html

mnechyporuk.html

1. Save file with following changes:

* set any text you want instead of ‘*Put your code here!*’ in the file:

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="UTF-8">

<meta name="viewport" content="width=device-width, initial-scale=1">

<title>Git Training</title>

<link rel="stylesheet" href="css/bootstrap.min.css">

<link rel="stylesheet" href="css/myStyles.css">

<link rel="stylesheet" href="css/hover.css" media="all">

</head>

<body>

<div class="container text-center border-bottom">

<h1>Your Full Name</h1>

</div>

<div class="container-fluid text-center">

<div class="row">

<div class="just-text">Put your code here!</div>

</div>

</div>

<script src="js/jquery-1.12.0.min.js"></script>

<script src="js/bootstrap.min.js"></script>

</body>

</html>

1. Begin tracking your file. Check the state of your working directory and the staging area:

git add *YourFile.html*

git status

1. Commit your changes with message like ‘Editing of my html-file to the branch - *YourFullName’*:

git commit -m “Editing of my html-file to the branch - *YourFullName*”

1. Switch to the branch named “Master”:

git checkout master

1. Merge your own branch with a branch named “Master”:

git merge *test-YourName*

1. Delete your own branch from local directory:

git branch -d *test-YourName*

1. Fetch and merge the changes from the remote branch to your local branch:

git pull

1. Update remote refs along with associated objects

git push

**Home task**

1. Pass course for reviewing the basic concepts of Git version control - <http://learngitbranching.js.org/>
2. Push all the test cases you’ve created during this course to the “master” branch.
   1. open fk2019 folder in the file browser
   2. create a folder named after your surname like “vklischko”
   3. drop your test files into it
   4. push them all
3. Ask for support if needed