Государственный Университет Молдовы

Факультет Математики и Информатики

Департамент Информатики

“Inițiere în Cloud Computing și Sisteme de Operare Distribuite”

Лабораторная работа 2   
“Система контроля версий Git. ”

Проверил: Препелица Аурелия

Выполнил: Чобану Артём

Кишинев 2021

# Требования к лабораторной работе:

* Установка Git. Создание учётной записи и репозитория на GitHub.

Проработать следующие команды:

* git config
* git init
* git clone
* git add
* git commit
* git diff
* git log
* git reset
* git status
* git rm
* git branch
* git checkout
* git merge --squash
* git remote
* git push
* git pull
* git stash
* git fetch
* git rebase

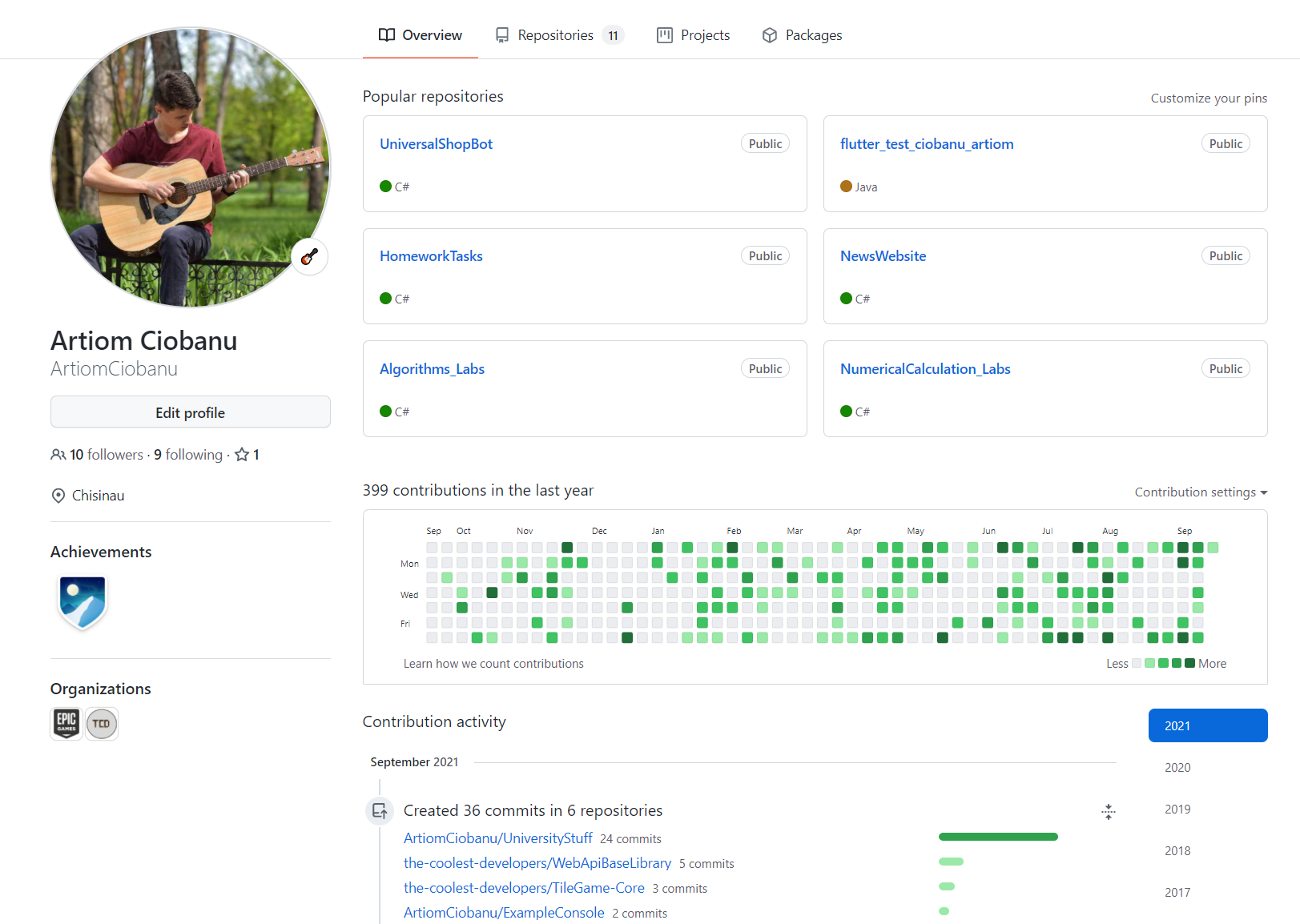
# Выполнение лабораторной работы

Моя учётная запись на GitHub существует уже давно:

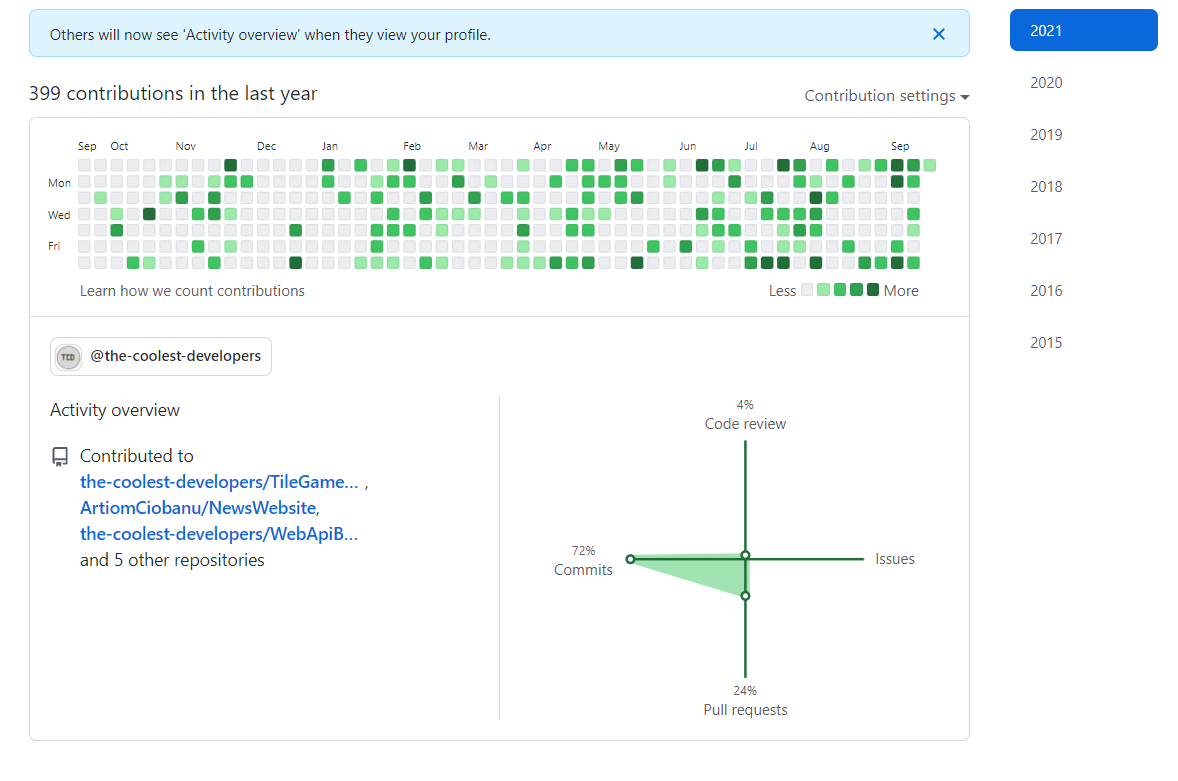
<https://github.com/ArtiomCiobanu>

Также есть созданная организация для совместных проектов с друзьями:

<https://github.com/the-coolest-developers>



Моя активность на GitHub за 2021 год:



Я создал папку, в которой и будет находится репозиторий, и файл, с изменениями которого я буду работать:

Graphical user interface, application

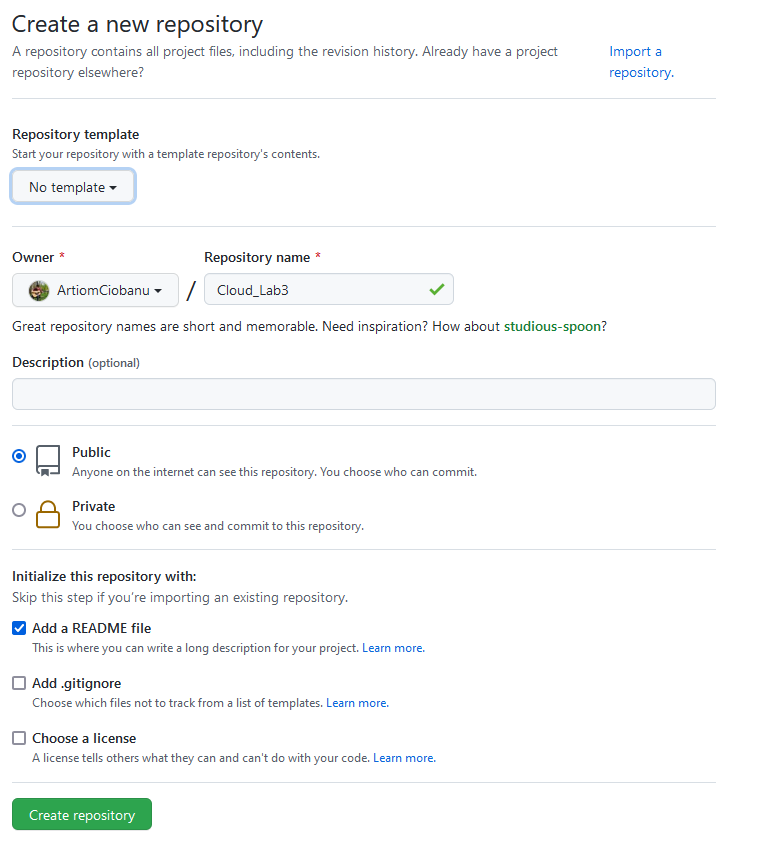
Description automatically generated

Инициализируем Git репозиторий:

Graphical user interface, text, application

Description automatically generated

Создадим репозиторий на GitHub:



Graphical user interface, text, application, Teams

Description automatically generated

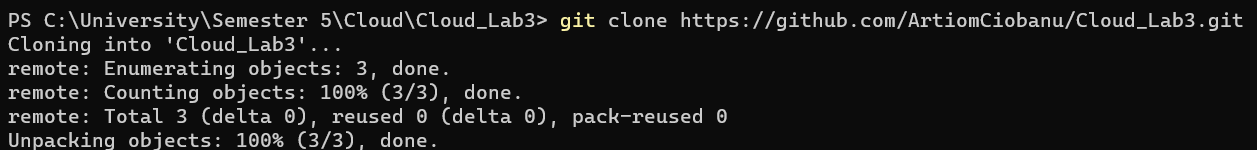
Скопируем ссылку на репозиторий:

<https://github.com/ArtiomCiobanu/Cloud_Lab3/blob/master/Lab3.txt>

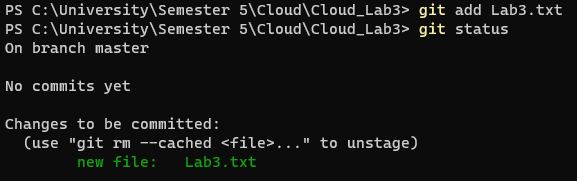
Graphical user interface, application

Description automatically generated

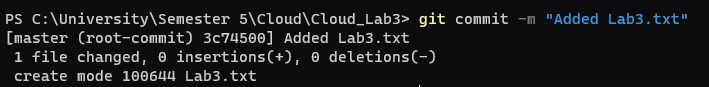
Удалим папку и скачаем репозиторий командой git clone:



Добавим файл Lab3.txt для коммита:



Создадим сам коммит:



Отправим на удалённый репозиторий командой git push:

Text

Description automatically generated

Коммиты на GitHub:

Graphical user interface, application

Description automatically generated

Получим историю коммитов командой git log:

Text

Description automatically generated

Сделаем коммит на GitHub:

Graphical user interface, text, application, email

Description automatically generated

Получим изменения командой git fetch:

Text

Description automatically generated

Получим изменения в нашу локальную ветку:

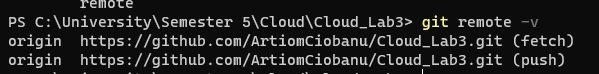
Text

Description automatically generated

Добавим ссылку на удалённый GitHub репозиторий:



Получим описание удалённых репозиториев:



Получим состояние репозитория:

Text

Description automatically generated

Получим изменения командой git diff:

Text

Description automatically generated

Отправим коммиты на удалённый репозиторий и сделаем другие изменения (в этот раз с удалением данных)

Text

Description automatically generated

Отложим изменения командой git stash:



Получим их обратно:

Text

Description automatically generated

Создадим новую ветку и получим их список:

Text

Description automatically generated

Перейдём на новую ветку командой git checkout:

Text

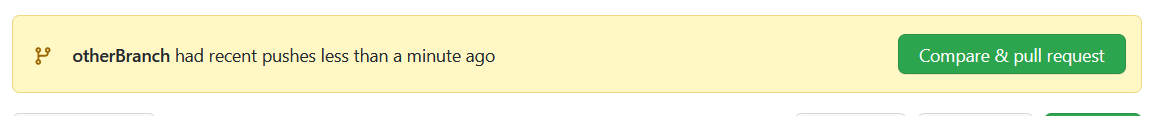
Description automatically generated

Сделаем коммит в другой ветке:

Text

Description automatically generated

GitHub предлагает создать pull request:



Коммиты в ветке master:

Graphical user interface, text, application, email

Description automatically generated

Коммиты в ветке otherBranch (создал ещё один):

Graphical user interface, text, application, email

Description automatically generated

Сделаем squash. Для этого переместим HEAD на 2 коммита:



Отмена rebase:



-i (interactive) открывает rebase в интерактивном режиме:

Graphical user interface, text, application

Description automatically generated

Один переименуем, а с другим сделаем squash – т.е. включим его в состав другого коммита:

Graphical user interface, text, application

Description automatically generated

Переименование:

Graphical user interface, text, application

Description automatically generated

Выход с сохранением:

Graphical user interface, application

Description automatically generated

Необходимо установить имя для финального коммита:

Text

Description automatically generated

Text

Description automatically generated

Отправим с флагом -f (force) для принудительной перезаписи:

Text

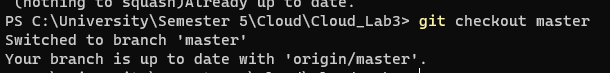
Description automatically generated

Результат:

Graphical user interface, text, application, chat or text message

Description automatically generated

Вернёмся на ветку master:



Как видим, здесь старые изменения, оставшиеся в данной ветке:

Text

Description automatically generated

Сделаем merge –squash:

Text

Description automatically generated

Увидим, что изменения сохранились, но они не являются частью ни одного коммита:

Text

Description automatically generated

После коммита отправляем на удалённый репозиторий.

Результат:

Graphical user interface, text, application

Description automatically generated

Сделаем изменения и отменим их командой git restore:

Text

Description automatically generated

Переместим HEAD на 4 коммита назад:

Text

Description automatically generated

Text

Description automatically generated

Вернём назад:

Text

Description automatically generated

Отменим все изменения командой git reset

Text

Description automatically generated

# Вывод

Во время работы над данной лабораторной работой я работал с системой контроля версий Git используя консоль. Я использовал команды commit, push, pull, fetch, checkout и другие.

# Библиография

<https://docs.github.com/en/github/importing-your-projects-to-github/importing-source-code-to-github/adding-an-existing-project-to-github-using-the-command-line>

<https://git-scm.com/book/ru/v2/%D0%98%D0%BD%D1%81%D1%82%D1%80%D1%83%D0%BC%D0%B5%D0%BD%D1%82%D1%8B-Git-%D0%A0%D0%B0%D1%81%D0%BA%D1%80%D1%8B%D1%82%D0%B8%D0%B5-%D1%82%D0%B0%D0%B9%D0%BD-reset>

<https://git-scm.com/docs/git-reset>