

МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ РОССИЙСКОЙ ФЕДЕРАЦИИ
ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ АВТОНОМНОЕ ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ
«Национальный исследовательский ядерный университет «МИФИ»
Обнинский институт атомной энергетики –
филиал федерального государственного автономного образовательного учреждения высшего
образования «Национальный исследовательский ядерный университет «МИФИ»
(ИАТЭ НИЯУ МИФИ)

Отделение Интеллектуальные кибернетические системы

Отчёт по лабораторной работе №2
По курсу: «Информационные системы и технологии»

Выполнил:
студент гр. ИС-М18
Ермаков В.Е.

Проверил:
д.т.н., профессор
Сальников Н.Л.

Обнинск 2018

1 Задание

Создать учетную запись на сервисе gitHub. Ознакомиться с основами работы системы контроля версий (СКВ) **git**. Сделать форк проекта по ссылке <https://github.com/mikhail-turicyn/IS-M18>. Добавить mikhail-turicyn в collaborators своего проекта. Пройти по типовому «GitHub flow» для своего проекта.

2 Цель работы

Научиться работать и применять СКВ **git** в повседневной работе.

3 Ход работы

Поскольку у меня уже есть аккаунт (connectDIY) и проект(SUNs-game) на GitHub, то аккаунт создавать не пришлось, а проект было решено перенести в «форкнутый» репозиторий IS-M18.

Немного о проекте:

Проект – это игра, космический симулятор, написанный на java, совместно тремя разработчиками. Целью разработки была сдача курсовой работы по предмету «Технология программирования». Проект содержит более 90 коммитов.

Первым делом был произведён форк проекта по ссылке <https://github.com/mikhail-turicyn/IS-M18> (см. рис.1).

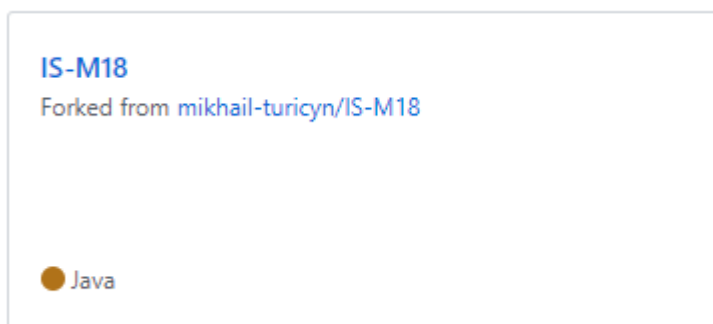


Рисунок 1 – форк проекта IS-M18

Далее добавлен mikhail-turicyn в collaborators (рис.2).

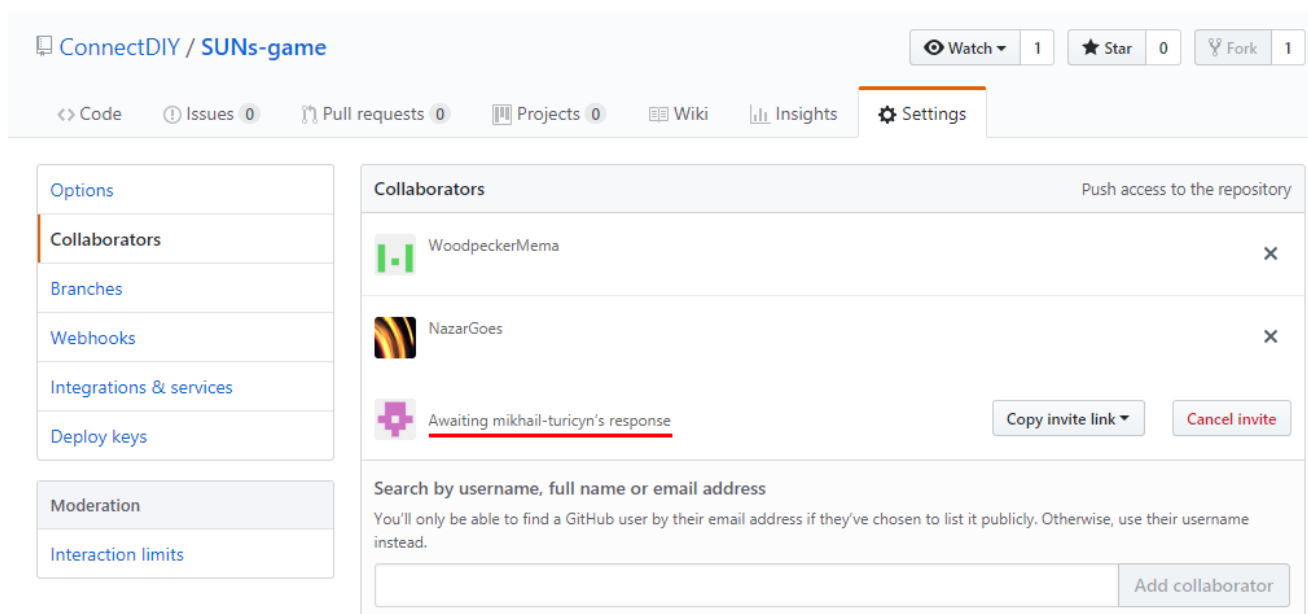


Рисунок 2 – добавление mikhail-turicyn в collaborators

После чего я добавил проект в «форкнутый» репозиторий (рис.3) и сделал *pull request* (рис.4).

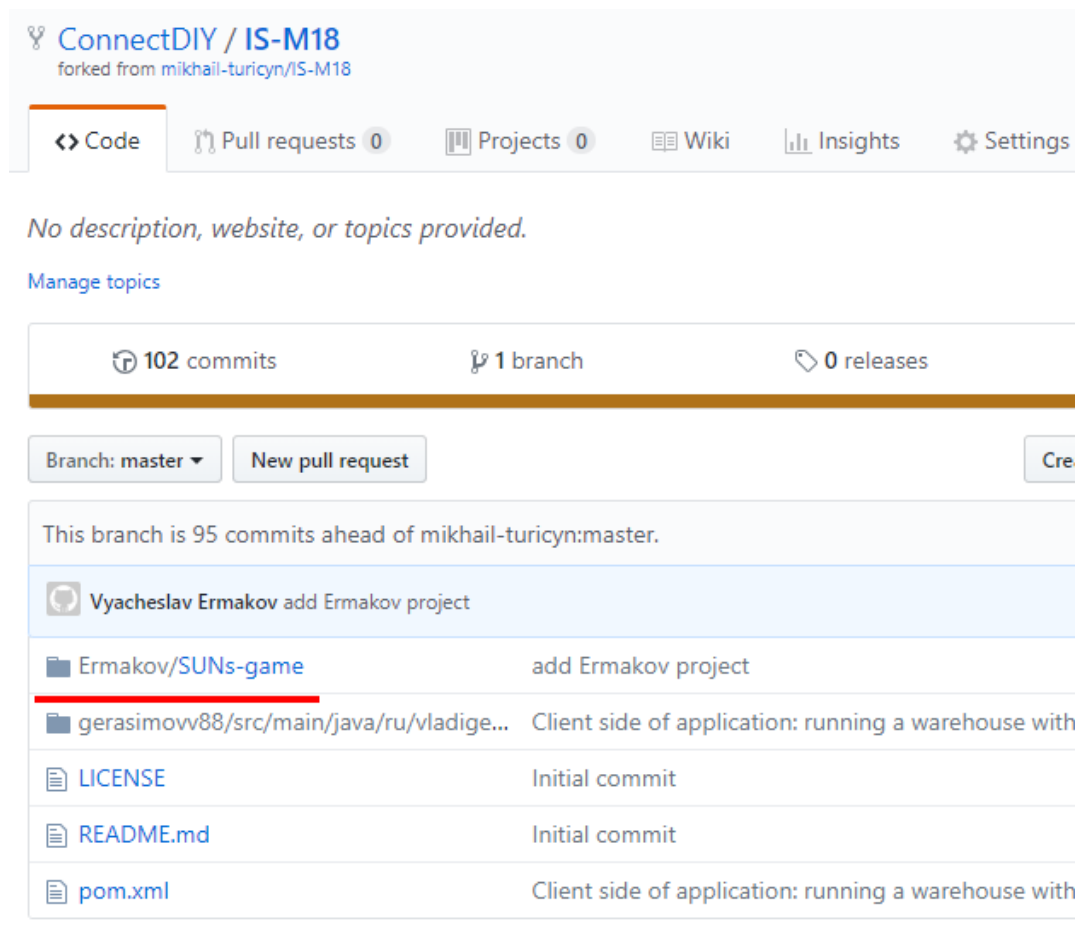


Рисунок 3 – добавление проекта в «форкнутый» репозиторий

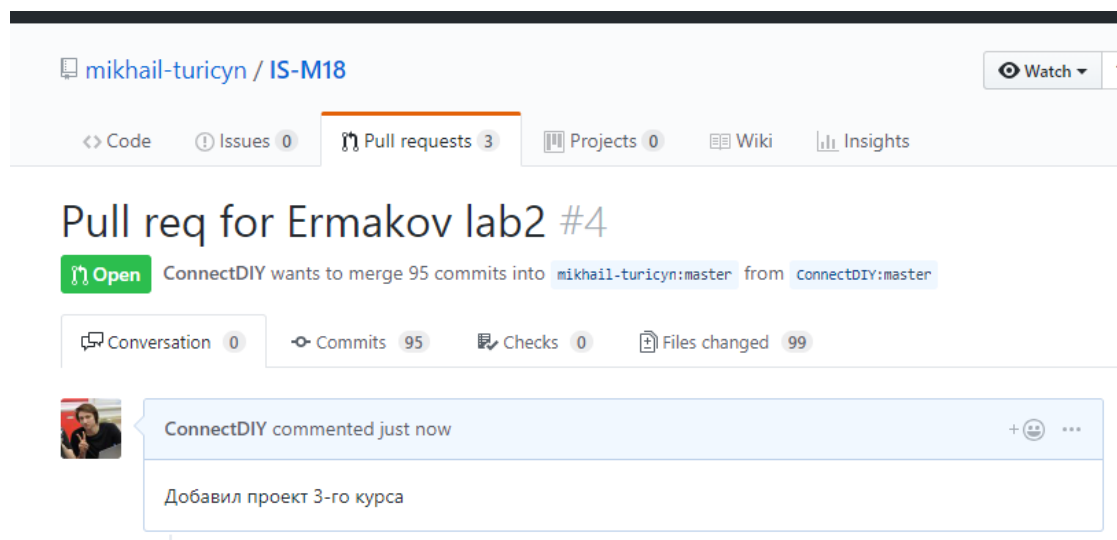


Рисунок 4 – *pull request*

Задача, отражённая на рисунке 3, оказалась крайне не тривиальной. По двум причинам:

1. Мой репозиторий находится в папке с названием проекта, а перенести проект нужно в папку со своей фамилией. Если просто выполнить *git merge* двух проектов, то все файлы из папки моего проекта перейдут в корень «форкнутго» проекта;
2. Просто перенести все файлы в папку не получится, т.к. потеряется история гит коммитов.

Решение данной задачи:

1. Скачиваем оба репозитория (форкнутый и свой проект) к себе на локальную машину;
2. Заходим в папку со своим проектом и переносим все файлы и папки, кроме *.git*, в созданную здесь же папку со своей фамилией. Выполняется это не просто так, а командой *git mv* т.к. иначе гит подумает, что вы удалили все файлы;
3. Переходим в форкнутый проект и устанавливаем отслеживание удалённого репозитория (на самом деле он локальный и находится у нас на ПК) – нашего проекта. Делаем это с помощью команды *git remote add some_name путь/к/папке/с/нашим/проектом*;
4. Теперь можно мёржить два проекта (мы мёржим, а не просто переносим файлы из одной папки в другую, чтобы не потерять историю коммитов). Команда *git merge some_name/master*;
5. Радуетесь результату.

Данный проект удовлетворяет типовому «GitHub flow», а именно содержит ветви, коммиты, мёржи и т.д.

ВЫВОДЫ

Были освоены навыки работы с **GitHub** и GitHub flow. Изучены базовые команды системы контроля версий **git** такие, как *add*, *commit*, *mv*, *merge*, *checkout*, *status*, *branch*. Узнал про такую вещь как *pull request* в GitHub, научился объединять два разных проекта, узнал про тонкости переноса проектов с сохранением истории.

Навыки, полученные в результате работы, считаю, чрезвычайно полезными в повседневной рабочей деятельности, а так же будут полезны для взаимодействия с другими участниками команды, работая над одним проектом.

СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ

1. Scott Chacon and Ben Straub. Pro Git – 2nd Edition (2014)