**МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ РОССИЙСКОЙ ФЕДЕРАЦИИ**

ФЕДЕРАЛЬНОЕ ГОСУДАСТВЕННОЕ 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 интеллектуальных кибернетических систем

**Лабораторная работа № 2**

**”Основы работы с git”**

Выполнил:

студент гр. ИС-М18 \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ Кастыря Н. А.

Принял:

Д.т.н., профессор \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ Сальников Н.Л.

Обнинск, 2018 г

**Задание:** Получить базовые понятия о системе контроля версий git, научиться работать с веб-сервисом для хостинга IT-проектов и совместной разработки GitHub, познакомиться с программой GitHub Desktop.

**Выполнение:**

Первым шагом было скачивание и установка программы GitHub Desktop. Был изучен базовый функционал системы и принято решение, что выполнение лабораторной работы будет происходить через web-интерфейс сервиса GitHub.

Далее был сделан fork требуемого проекта <https://github.com/mikhail-turicyn/IS-M18>. Fork – создание копии проекта в своём репозитории. Следующим шагом было создание личной папки студента, в которую необходимо было отправлять отчёты и лабораторные работы. Был залит проект на Python, содержащий несколько классов. В web-интерфейсе это делается простым переносом файлов или непосредственным написание кода в самом интерфейсе. При работе через приложение сначала необходимо создать файл, сделать commit (запись изменений в репозитории), а потом отправить на сервер с помощью команды pull.

После того, как проект и отчёт были загружены, был сделан pull reguest, то есть запрос владельцу на объединение моего репозитория с основным.

В ходе выполнения работы был поставлен следующий вопрос: «Конфликт в git – создать и разрешить». Рассмотрим такую ситуацию: вы работаете в одной ветке проекта и редактируете файл main.py. Ваш коллега работает в другой ветке этого же проекта и редактирует этот же файл main.py. При объедении этих веток возникнет merge conflict (конфликт слияния). Тогда будет доступно два варианта:

* Вручную разрешить конфликт. Вам будет даны куски из обеих сливаемых веток, и нужно будет вручную редактировать файл, удаляя метки оставленные git (метка выглядит так: >>>>>>> otherbranch).
* Выбрать одну из версий файла

**Вывод:**

С системой git работаю не в первый раз. Ранее использовал возможности, предлагаемые IDE, в которых разрабатывал проекты. В данной лабораторной работе познакомился с таким понятиями как fork и pull reguest, а также с десктопной версией GitHub.

**Список использованных источников:**

1. Github. [Электронный ресурс]. URL: <https://github.com> [Доступ от 1.12.2018]
2. Документация по git. [Электронный ресурс]. URL: <https://git-scm.com/book/ru/v1> [Доступ от 1.12.2018]