**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 Интеллектуальные кибернетические системы

**Лабораторная работа № 2**

**Git и GitHub: изучение базовых понятий и работа с системами**

Выполнил:

студент гр. ИС-М18 \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ Герасимов В.В.

Принял:

профессор ОИКС \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ Сальников Н.Л.

Обнинск, 2018

**Цель работы**: 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 работы**:

Для выполнения работа необходимо установить Git на локальную рабочую машину. Git доступен для Windows, MacOS, Linux **[1]**, поэтому установить его, следуя описанным шагам, не составляет труда.

Далее желательно установить GUI для работы с Git (конечно, можно работать и в консольном режиме). Я использую встроенные возможности работы с системами контроля версии в Intellij IDEA **[2]** – IDE для Java.

Далее необходимо создать учетную запись на GitHub ресурсе **[3]**, где будут храниться репозитории проектов.

Следующим шагом нужно было сделать fork проекта **[4]** для дальнейшей работы. После успешного fork’a проект добавляется в ваш репозитории на GitHub. Он является origin. Альтернативой было создание нового репозитория командой git init.

Проект необходимо скачать локально. Для этого используется команда git clone.

Открываем пустой проект в Intellij IDEA, создаем директорию, где будут храниться исходные коды. Там создаем пустой Java проект. Делаем первый коммит командой git commit. Для отправки коммита в удаленный репозиторий используется git push. Так как используется GUI для работы Git, то фаилы в контроль версии добавляются галочками в пользовательском интерфейсе, где и вводится текст сообщения коммита и данные об его авторе.

Далее была создана ветка – branch (‘warehouse branch’). Ветки используются, чтобы логически отделить разрабатываемые возможности. В данной ветке разрабатываются сущности хранилища (warehouse) и хранимые элементы (items).

Когда работа завершилась и данные были успешно отправлены в удаленный репозиторий, была создана новая ветка ‘client branch’ на основе ветки ‘warehouse branch’. В данной ветке велась разработка клиентской части – работа с хранилищем, разработанном в предыдущей части.

После того, как работа была завершена, основная ветка репозитория ‘master branch’ оказалась пуста. Поскольку необходимо изменения других веток слить с ‘master branch’. Для этого используется команда git merge. В результате работа возможны конфликты. В данном случае их не было, поскольку в разных ветках работа над одними и теми же файлами одновременно не велась.

В результате работы имеем ветку ‘master branch’, в которой отражен этап разработки системы. Мы хотим, чтобы нашими изменениями воспользовались в главном репозитории **[4]** (тот, который мы делали fork). Для этого создается pull request. Остается только ждать решения владельца репозитория **[4]** о принятии / непринятии новых features.

**Список используемых источников**

1. Git. [Электронный ресурс]. URL: <https://git-scm.com> [Доступ от 8.11.2018]
2. Github. [Электронный ресурс]. URL: <https://github.com> [Доступ от 8.11.2018]
3. Intellij IDEA. [Электронный ресурс]. URL: <https://www.jetbrains.com/idea/> [Доступ от 8.11.2018]
4. Рабочий проект. [Электронный ресурс]. URL: <https://github.com/mikhail-turicyn/IS-M18> [Доступ от 8.11.2018]