

Лабораторная работа №5

«GitHub. Совместная работа»

Цель: научиться работать с веб-сервисом для хостинга проектов и их совместной разработки GitHub. Создать свой публичный репозиторий.

Теоретическая часть:

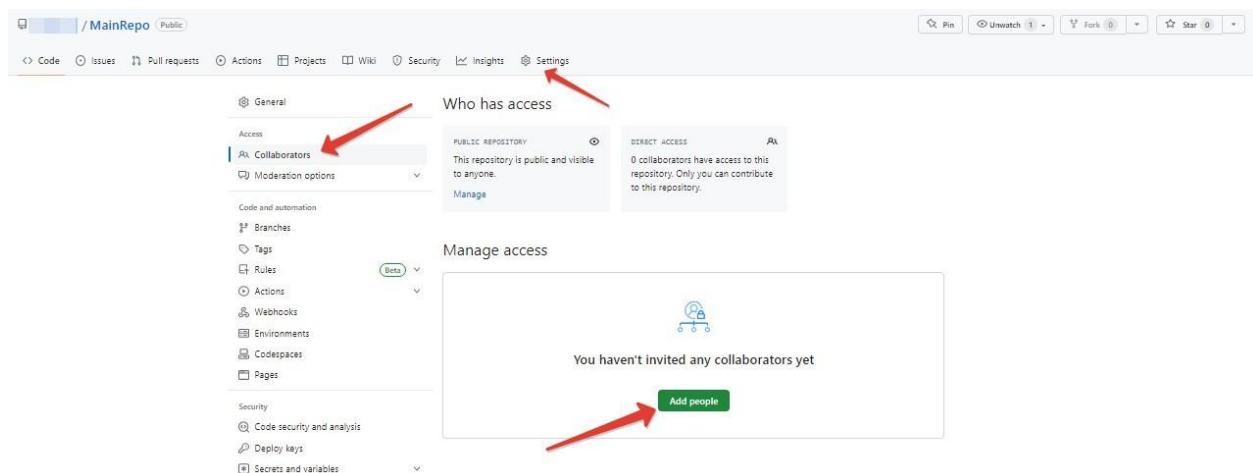
Как правило, существует два способа настройки Github для совместной работы:

Организации. Владелец организации может создавать множество команд с разными уровнями доступа для различных репозиториев.

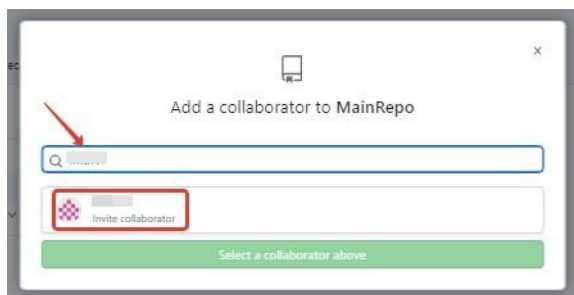
Сотрудники. Владелец репозитория может добавлять коллабораторов (соавторов) с доступом Read + Write для одного репозитория.

Рассмотрим второй вариант совместной работы.

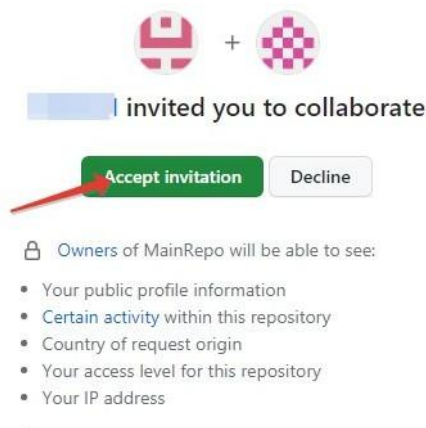
Чтобы добавить Collaborators (другие личные учетные записи Github) к публичному репозиторию, владельцу репозитория необходимо перейти на страницу Settings -> Collaborators и нажать кнопку Add People.



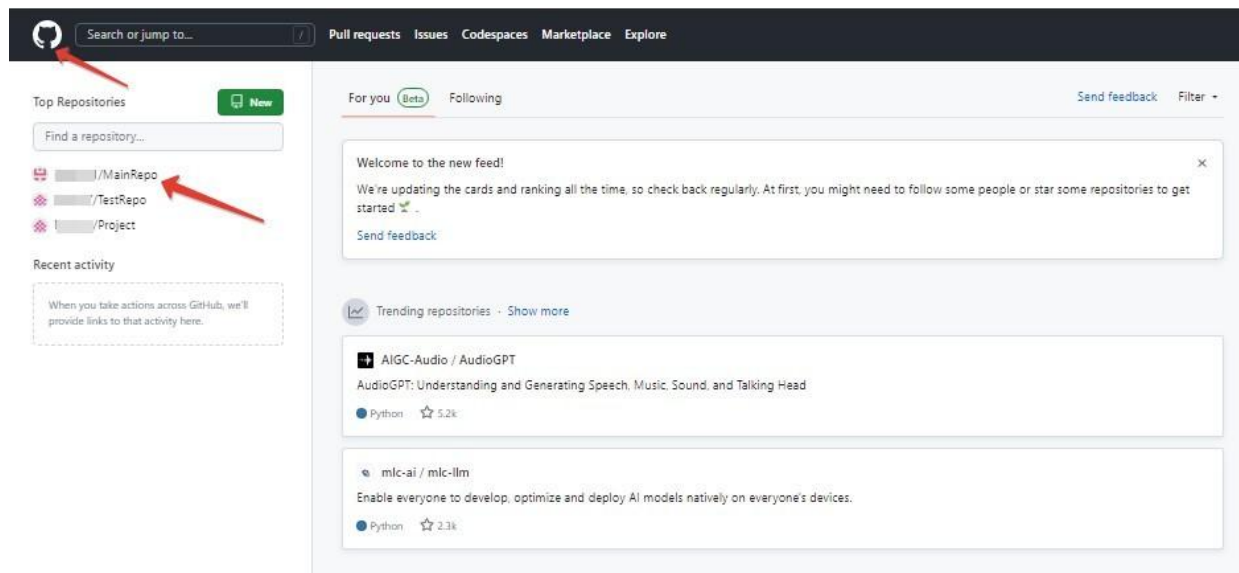
В поисковой строке вводится username добавляемого соавтора, выбирается его профиль и подтверждается выбор.



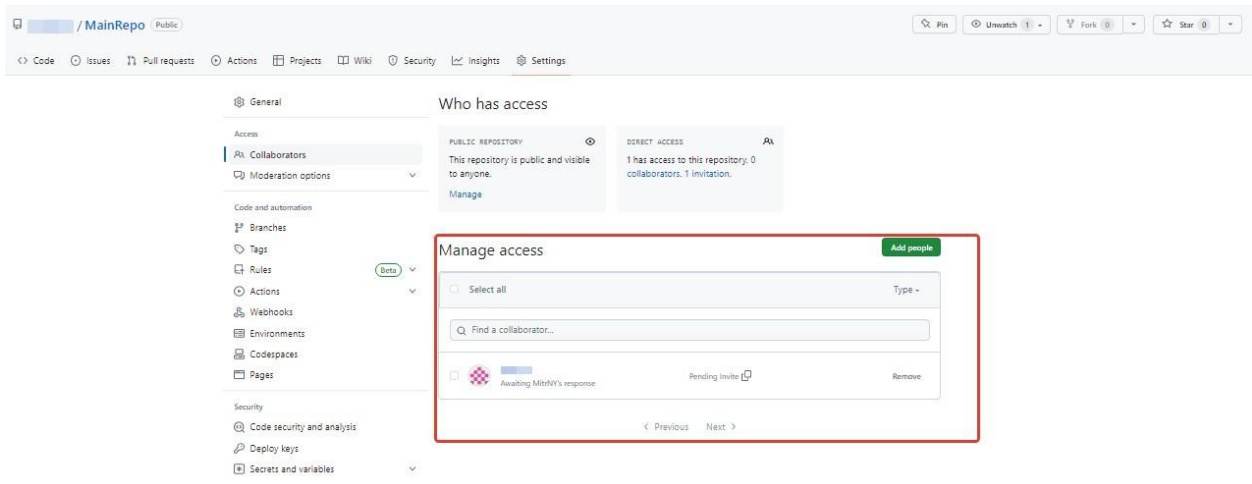
После отправки приглашения соавтор получает приглашение по электронной почте, а также в своем профиле GitHub. Соавтор должен принять его, чтобы получить разрешение на совместную работу над одним и тем же проектом.



У соавторов проекта общий репозиторий виден в списке всех репозиториях:

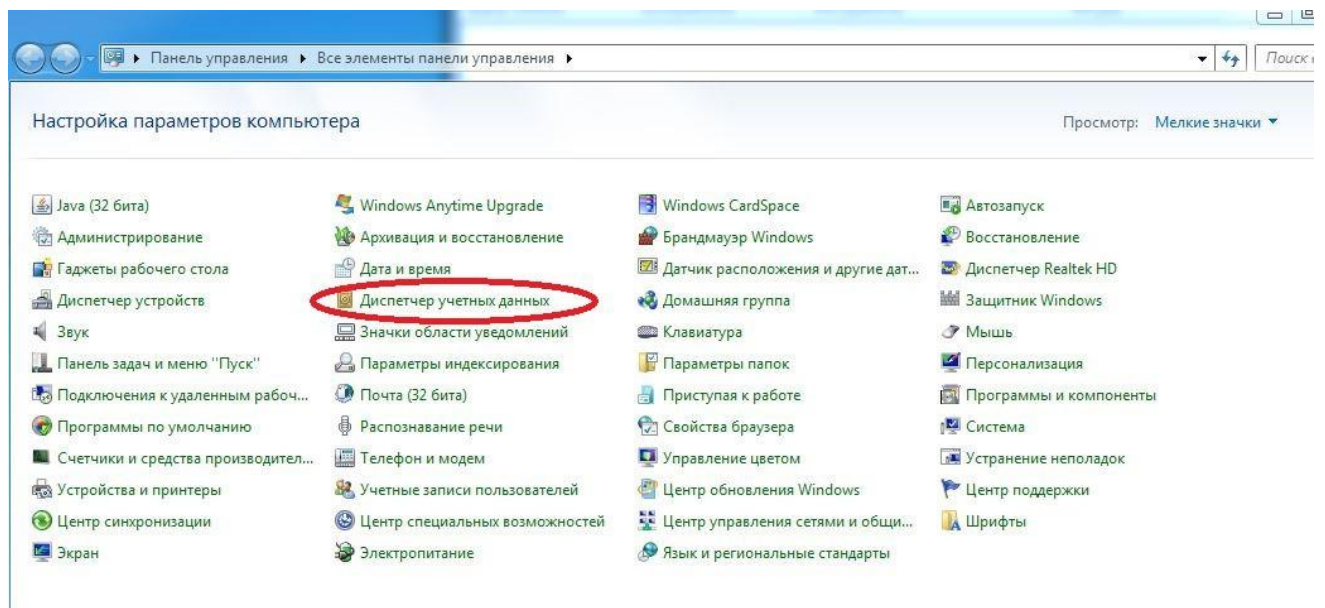


Владелец репозитория (координатор проекта) может просматривать приглашения, которые находятся на рассмотрении и пока еще не приняты.

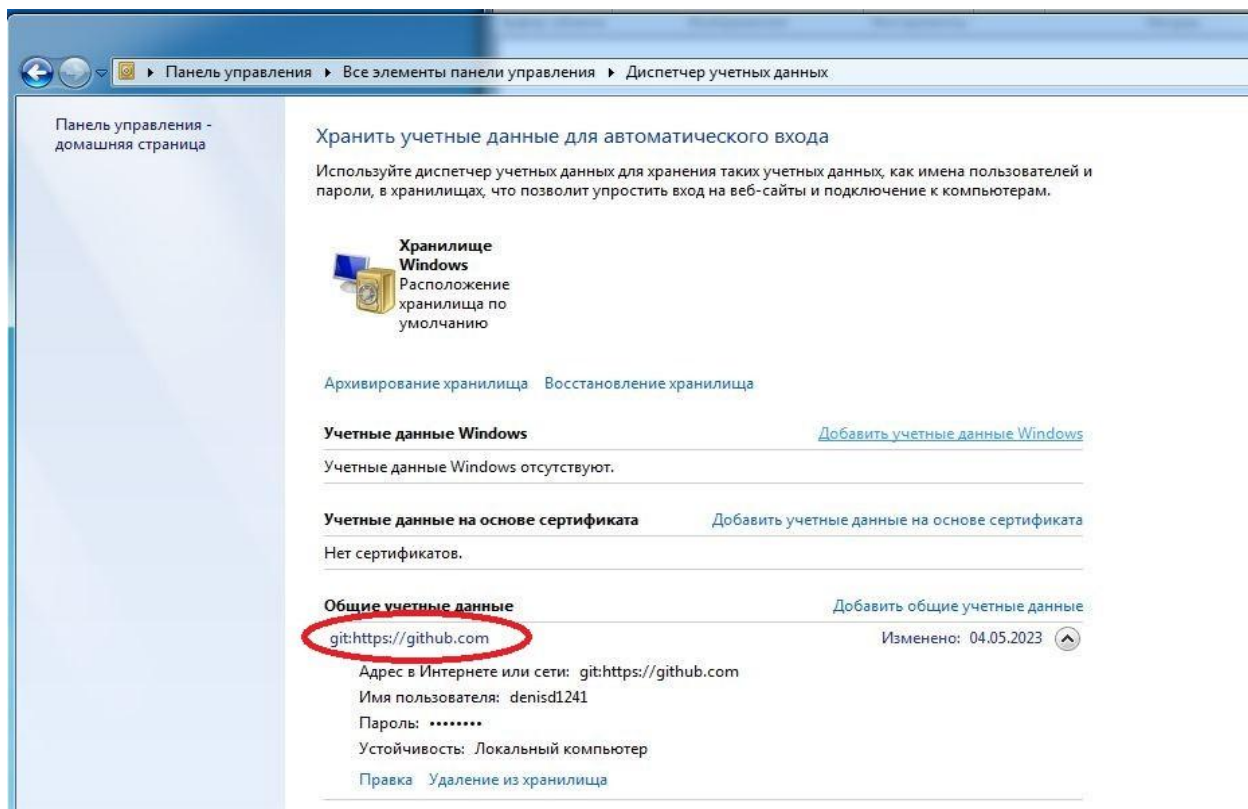


Практическая часть.

Внимание!!! Для корректной работы в учебной аудитории необходимо проверить остались ли данные учетных записей других пользователей. Для этого войдите в Настройки параметров компьютера, а затем перейдите в Диспетчер учетных данных:



После чего найдите в общих учетных данных записи с данными GitHub и удалите их:



После работы в аудитории не забудьте таким же способом удалить свои учетные данные.

1) Регистрация всех членов команды

Для совместной работы над одним проектом необходимо, чтобы все участники команды (студенты вашей бригады) имели профили в Github. Зарегистрируйтесь, как было описано в лабораторной работе №4. Сделайте скрины всех профилей участников на GitHub и добавьте их в отчет.

2) Распределение ролей на проекте

Перед началом совместной работы определитесь с ролями в вашей команде. Выберите студента (далее будет называться «студент №1»), который будет являться координатором проекта с полномочиями администратора репозитория. Остальные студенты команды («студент №2», «студент №3» и т.д.) должны иметь доступ к репозиторию для того чтобы иметь возможность получать файлы проекта. Кроме общего репозитория каждый из участников

команды должен иметь свой локальный репозиторий, в котором будут храниться версии проекта, над которым он работал.

3) *Создание совместного репозитория* (выполняет студент №1)

Создайте в своей учетной записи новый публичный репозиторий **FinalWork**. Добавьте к этому репозиторию соавторов (профили студентов вашей бригады). Продемонстрируйте результаты выполнения задания, сделав скрины, добавьте их в отчет.

4) *Выполнение задания*

Выполнение задания заключается в описании действий и демонстрации полученных результатов в виде скринов каждым студентом бригады. Результатом выполнения индивидуального задания должны стать файлы .docx, содержащий описание работы над проектом, ссылку на репозиторий и файлы самого проекта.

Этап 1 (подготовительный).

Выполняет студент №1:

- создайте локальный репозиторий и инициализируйте его;
- свяжите созданный репозиторий с удаленным (FinalWork);
- убедитесь в том, что привязка прошла успешно, выполнив соответствующую команду `git remote -v`;
- извлеките и загрузите в локальный репозиторий содержимое из удаленного репозитория;
- создайте в своем локальном репозитории новый файл .docx, содержащий отчет по данной лабораторной работе;
- зафиксируйте изменения, выполнив соответствующие команды;
- добавьте в локальный репозиторий файлы проекта по дисциплине «Алгоритмизация и программирование» за 1 семестр (порядковый номер лабораторной, которую нужно взять, соответствует номеру бригады).

Например: бригада №3 берет лабораторную работу №3.

- зафиксируйте изменения, выполнив соответствующие команды;
- отправьте зафиксированные изменения в удаленный репозиторий в главную ветку.

В результате описанных действий совместный репозиторий на GitHub должен содержать файл отчета (.docx) и файлы проекта.

Этап 2 (основной).

Для студента №1:

- создайте в своем локальном репозитории новую ветку (имя ветки должно содержать вашу фамилию, например «Ivanov») и перейдите в эту ветку;
 - выполните индивидуальное задание:
 - в коде проекта измените имена переменной (например: была переменная `int a`, стала `int a1`);
 - зафиксируйте изменения, выполнив соответствующие команды;
 - создайте в своей ветке файл 1.docx, в котором опишите ваши действия, добавив соответствующие скрины, в том числе скрины из консоли git.
 - зафиксируйте добавление файла;
 - отправьте зафиксированные изменения в удаленный репозиторий в вашу ветку (в параметре команды укажите имя вашей ветки, на GitHub она создается автоматически, это действие нужно для сохранности данных);
 - после того как ваши изменения будут просмотрены координатором проекта (студентом №1), выполняется слияние с веткой master.
- Внимание!!!** Перед тем, как выполнить слияние, необходимо скопировать с GitHub из ветки master в свой локальный репозиторий, в ветку master все произведенные изменения (если таковые есть). Затем,

также в локальном репозитории, выполнить слияние своей ветки с веткой master. Только после этого отправить изменения в GitHub на ветку master.

Для студента №2, 3:

- зайдите на свою почту, которую указывали при регистрации профиля на GitHub, и подтвердите свое участие в совместной работе;
- создайте свой локальный репозиторий, клонировав себе общий репозиторий FinalWork;
- создайте новую ветку в локальном репозитории (имя ветки должно содержать вашу фамилию, например «Petrov») и перейдите в эту ветку;
- выполните индивидуальное задание:

- в код проекта внесите изменения:

студент №2:

- измените объявление всех используемых переменных так, чтобы каждая переменная объявлялась с новой строки;
- добавьте хотя бы один комментарий, поясняющий назначение переменных.

студент №3:

- добавьте проверку на положительность первой объявленной переменной с выводом сообщения об этом.
- добавьте комментарий, поясняющий строку с условным выражением.
- зафиксируйте каждое изменение, выполнив соответствующие команды;
- создайте в своей ветке файл 2.docx для студента №2 (для студента №3 – 3.docx), в котором опишите ваши действия, добавив соответствующие скрины, в том числе скрины из консоли git.
- зафиксируйте добавление файла;
- отправьте зафиксированные изменения в удаленный репозиторий в вашу ветку (в параметре команды укажите имя вашей ветки, на GitHub

она создастся автоматически, это действие нужно для сохранности данных);

- после того как ваши изменения будут просмотрены координатором проекта (студентом №1), выполняется слияние с веткой master.

Внимание!!! Перед тем, как выполнить слияние, необходимо скопировать с GitHub из ветки master в свой локальный репозиторий (в ветку master) все произведенные изменения (если таковые есть). Затем, также в локальном репозитории, выполнить слияние своей ветки с веткой master. Только после этого отправить изменения в GitHub на ветку master.

Этап 3 (заключительный).

Выполняет студент №1:

- получите все добавленные изменения в свой локальный репозиторий;
- оформите отчет (добавьте в него описание ваших действий и действий студентов вашей бригады, используя информацию из их файлов);
- зафиксируйте изменения;
- добавьте ссылку на ваш совместный репозиторий в конце отчета;
- зафиксируйте изменения;
- отправьте зафиксированные изменения в удаленный репозиторий в главную ветку.