**МИНИСТЕРСТВО ТРАНСПОРТА РОССИЙСКОЙ ФЕДЕРАЦИИ**

**ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ АВТОНОМНОЕ ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ**

**«РОССИЙСКИЙ УНИВЕРСИТЕТ ТРАНСПОРТА»**

**ИНСТИТУТ ТРАНСПОРТНОЙ ТЕХНИКИ И СИСТЕМ УПРАВЛЕНИЯ (ИТТСУ)**

Кафедра «Управление и защита информации»

А.И. САФРОНОВ, А.А. БАРАНОВ, Е.А. КРУГЛИКОВ

**Основы работы в системе контроля версий *Git***

**Учебно-методическое пособие**

**для проведения лабораторных работ и практических занятий по дисциплине «Программирование и основы алгоритмизации»**

**МОСКВА – 2025**

МИНИСТЕРСТВО ТРАНСПОРТА РОССИЙСКОЙ ФЕДЕРАЦИИ

ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ АВТОНОМНОЕ ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ

«РОССИЙСКИЙ УНИВЕРСИТЕТ ТРАНСПОРТА»

ИНСТИТУТ ТРАНСПОРТНОЙ ТЕХНИКИ И СИСТЕМ УПРАВЛЕНИЯ (ИТТСУ)

Кафедра «Управление и защита информации»

А.И. САФРОНОВ, А.А. БАРАНОВ, Е.А. КРУГЛИКОВ

Основы работы в системе контроля версий *Git*

Учебно-методическое пособие

для бакалавров направления

27.03.04 «Управление в технических системах»

**МОСКВА – 2025**

**УДК 004**

**С 21**

Сафронов А.И., Баранов А.А., Кругликов Е.А. Основы работы в системе контроля версий *Git*: Учебно-методическое пособие для проведения лабораторных работ и практических занятий по дисциплине «Программирование и основы алгоритмизации». – М.: РУТ (МИИТ), 2025. – 79 с.

Учебно-методическое пособие разработано для бакалавров направления «Управление в технических системах». Оно нацелено на изложение основ работы в системе контроля версий *Git* с целью повышения эффективности командной разработки программного обеспечения.

Рецензент:

Заведующий кафедрой «Электропоезда и локомотивы» РУТ (МИИТ), д.т.н., проф. Пудовиков О.Е.

© РУТ (МИИТ), 2025

# Введение

Система контроля версий (СКВ) позволяет отслеживать и хранить изменения, внесённые в различные файлы. СКВ имеет важную роль в командной разработке, позволяет разработчикам следить за изменениями, грамотно вносить свои изменения и возвращаться к предыдущим версиям проекта при необходимости. Использование СКВ помогает предотвращать возникновение различных конфликтов при командной разработке.

При разработке проекта с открытым исходным кодом – ядра *Linux*, между разработчиками передавались изменения в виде патчей и архивов. В 2002 команда разработчиков *Linux* начала использовать СКВ *BitKeeper*. Отношения между разработчиками *Linux* и *BitKeeper* прекратились, в связи с чем Линусом Торвальдсом (создателем *Linux*) было принято решение о создании собственной утилиты системы контроля версий – *GIT*, так как *Linux* – проект с открытым исходным кодом, то и *GIT* унаследовал такую же модель распространения.

В мире информационных технологий *GIT* давно стал стандартом, но он пользуется спросом не только в сфере информационных технологий, но и в образовательном процессе – студент может всегда вернуться на рабочую версию решения, а преподаватель отслеживать весь процесс решения.

Существует множество онлайн-сервисов для работы с *GIT*: *GitHub*, *GitLab*, *GitFlic*. Самым популярным из них является *GitHub*. *GitHub* позволяет вести полноценную командную разработку, предоставляя возможность получать удалённо изменения, вносить удалённо изменения, вести обсуждение по мере разработки, создавать отдельные ветки разработки, позволяет делиться продуктами разработки, а также предоставляет множество различных возможностей.

Таблица 1 – Сравнение существующих платформ коллективной разработки ПО и контроля версий

|  |  |  |  |
| --- | --- | --- | --- |
|  | ***GitHub*** | ***GitLab*** | ***GitFlic*** |
| **Популярность** | 16.8 миллионов пользователей в день | 544 тысячи пользователей в день | 10 тысяч пользователей в день |
| **Пропускная способность** | Размер загружаемого файла не может превышать 100 Мб | Администратор может выставить максимальный лимит самостоятельно | Размер загружаемого файла не может превышать 2 Гб |
| **Уникальность** | Большое количество проектов с открытым исходным кодом | Высокая популярность в корпоративной сфере | Отечественная разработка |

На основании данных из Таблицы 1, будет рассмотрена онлайн-платформа *GitHub*.

# 1. Регистрация на *GitHub*

Для начал работы с *GitHub* необходимо пройти регистрацию: [*https://github.com/*](https://github.com/).

Для перехода на страницу регистрации необходимо нажать на кнопку «Sign up» продемонстрированную на Рисунке 1.1.

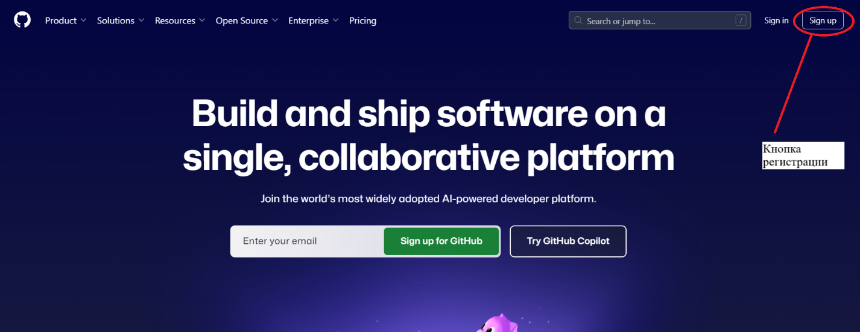


Рисунок 1.1 – Кнопка регистрации

После нажатия кнопки регистрации вы будете перенаправлены на страницу регистрации, где требуется ввести ваши личные данные. Форма регистрации на Рисунке 1.2.

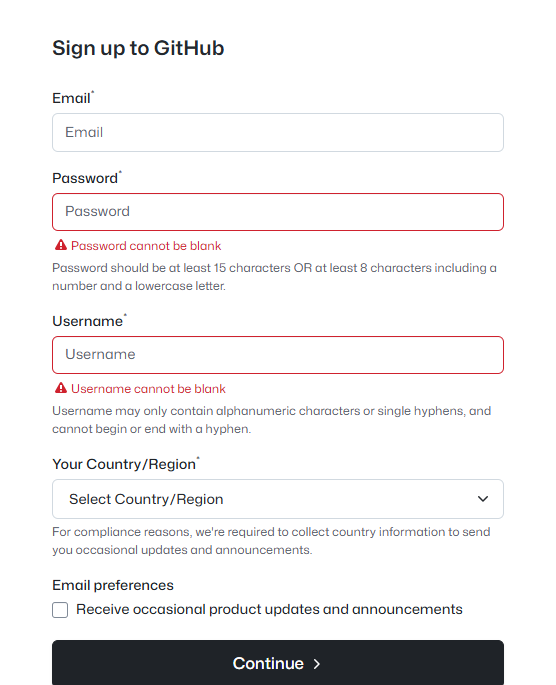


Рисунок 1.2 – Форма регистрации

После заполнения данных регистрации необходимо пройти визуальную головоломку, нажав на соответствующие кнопку. Кнопка визуальной головоломки на Рисунке 1.3.

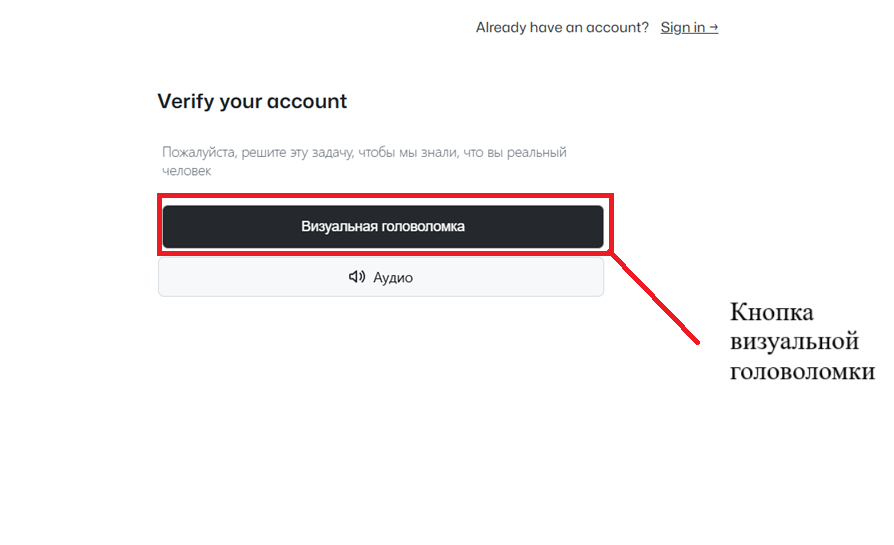


Рисунок 1.3 – Кнопка визуальной головоломки

После прохождения визуальной головоломки необходимо подтвердить электронную почту, введя код из присланного письма. Поле для ввода кода на Рисунке 1.4.

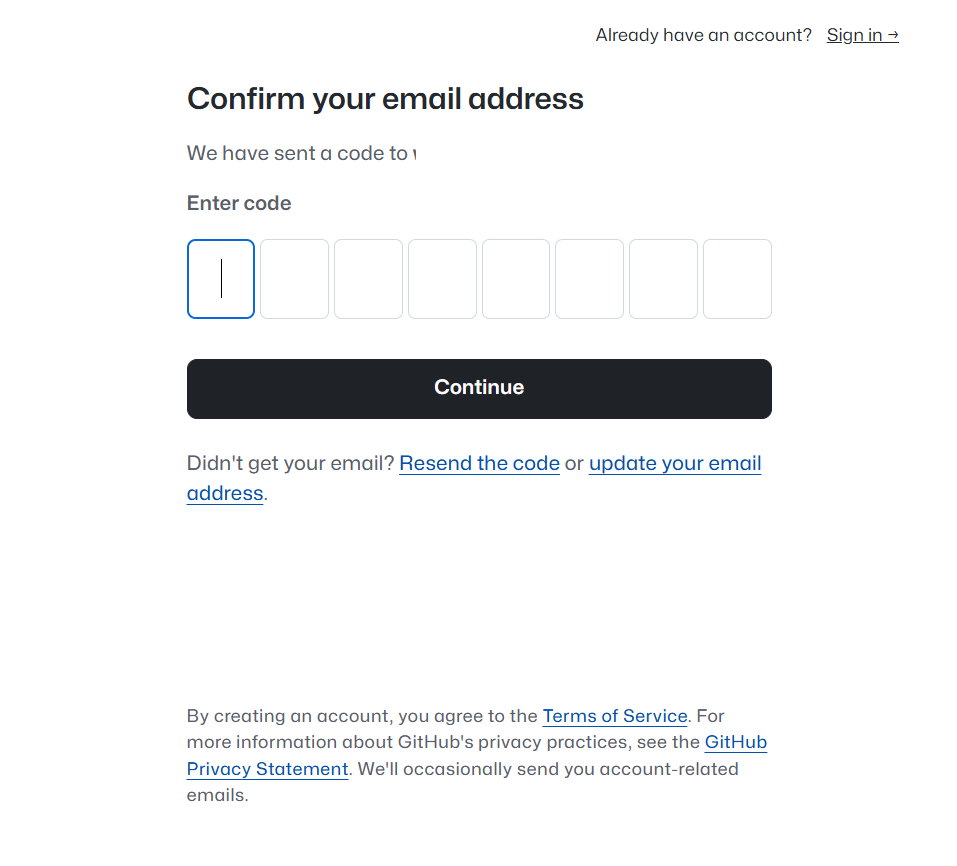


Рисунок 1.4 – Поле для ввода кода

# 2. Обзор *web*-интерфейса *GitHub*

***Dashboard***

В разделе *Dashboard* можно отслеживать можно отслеживать *issues*, запросы на включение изменений, переходить к своим основным репозиториям. Раздел *Dashboard* представлен на Рисунке 2.1.

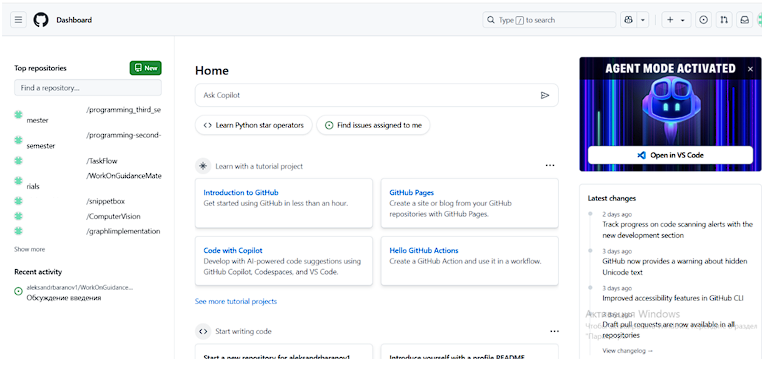


Рисунок 2.1 – Раздел *dashboard*

***Your repositories***

Чтобы попасть в раздел *Your repositories* необходимо совершить нажатие по иконке профиля. Иконка профиля на Рисунке 2.2.

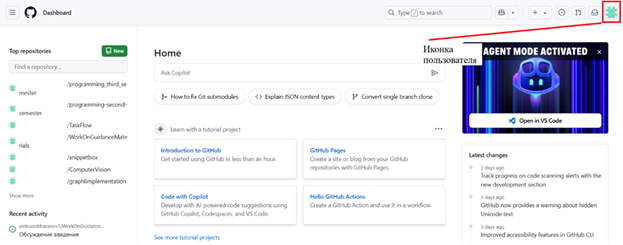


Рисунок 2.2 – Иконка пользователя

После нажатия на иконку пользователя появляется меню, в котором необходимо нажать кнопку «*Your repositories*». Кнопка «*Your repositories*» на Рисунке 3.

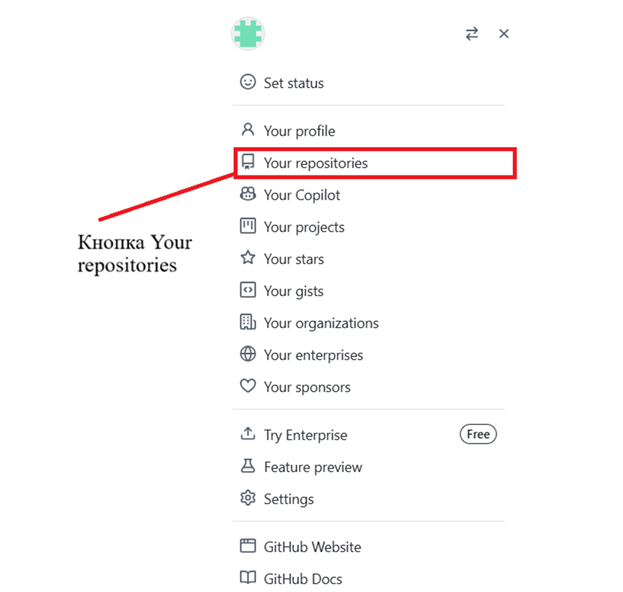


Рисунок 3 – Кнопка «*Your repositories*»

На странице *Your repositories* можно просмотреть все имеющиеся репозитории и перейти в каждый из них. Страница «*Your repositories*» на Рисунке 4.

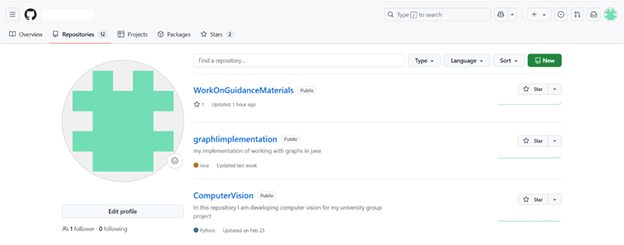


Рисунок 4 – Странице «*Your repositories*»

**Настройка и создание удалённого репозитория**

Для создания удалённого репозитория, необходимо находясь на странице Your repositories, нажать на кнопку New. Кнопка New на Рисунке 5.

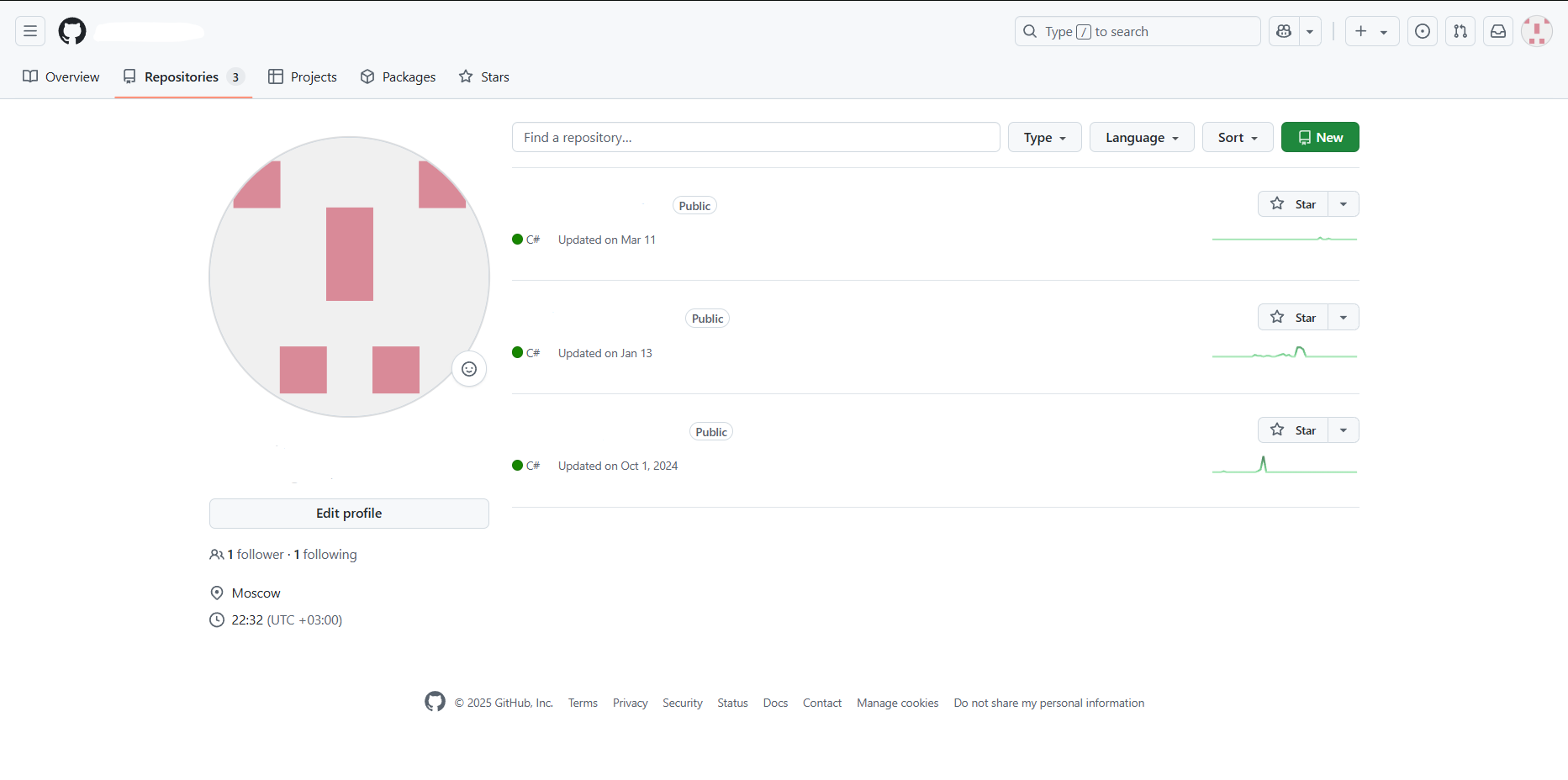


Рисунок 5 – Кнопка *New*

После нажатия на кнопку New пользователь попадает в меню создания репозитория. В меню создания репозитория можно указать создателя репозитория, его название, описание (опционально), задать статус репозитория (публичный или приватный), создать репозиторий сразу с README-файлом (в нём, обычно, описывается предназначение репозитория). Кнопки функций gitignore и выбора лицензии не рекомендуются к использованию в учебных целях. Меню создания репозитория на Рисунке 6.

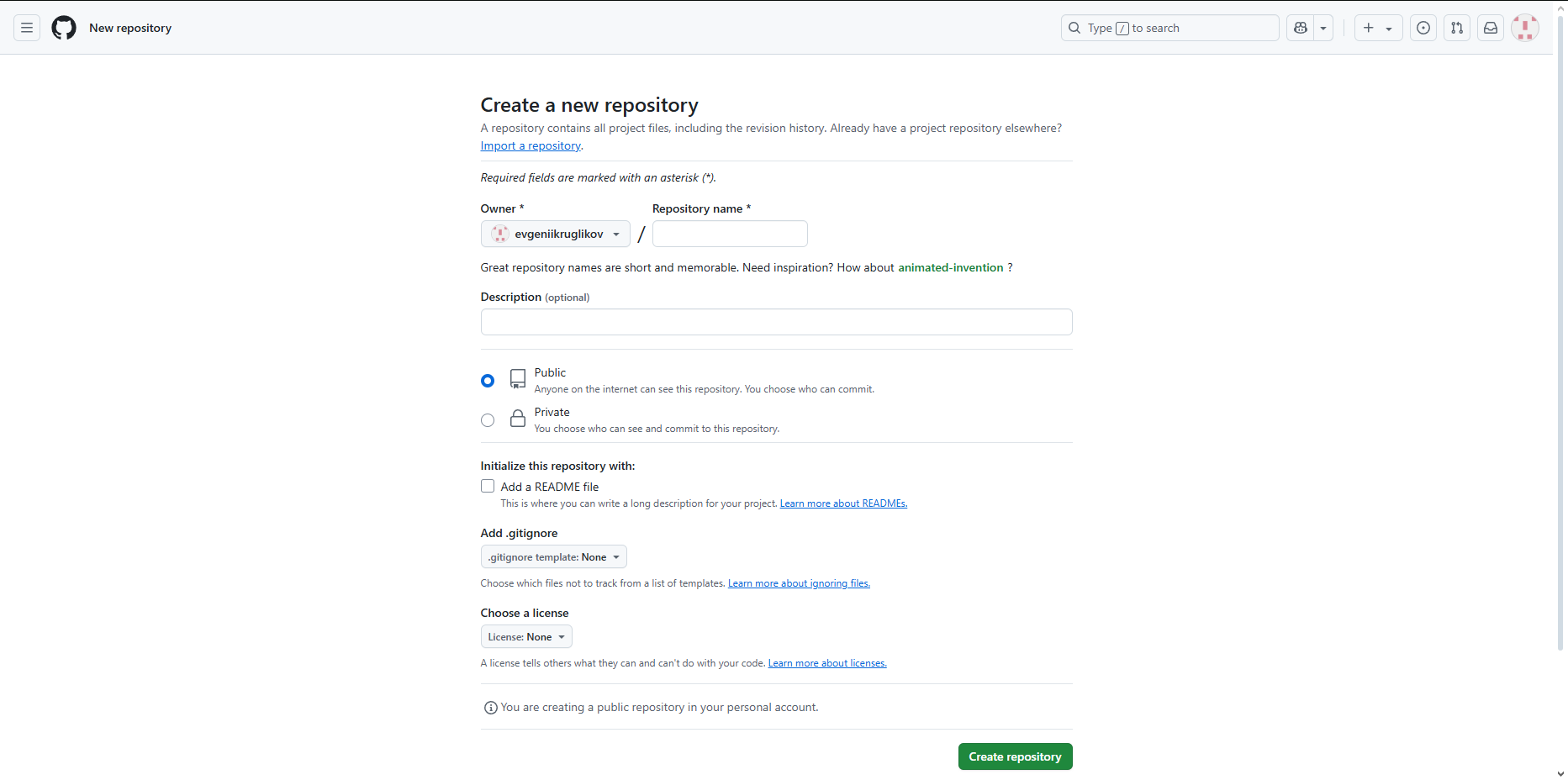


Рисунок 6 – Меню создания и настроек репозитория

После настройки удалённого репозитория необходимо нажать кнопку *Create repository*. Нажав на кнопку *Create repository*, вы создадите новый репозиторий.

***Issues***

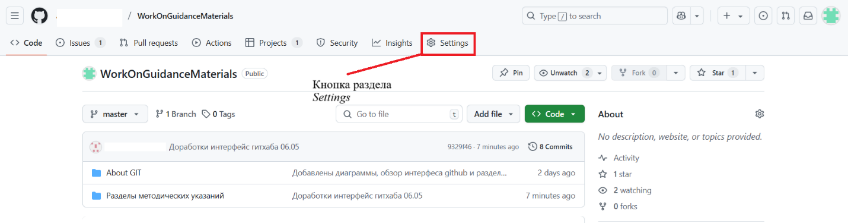
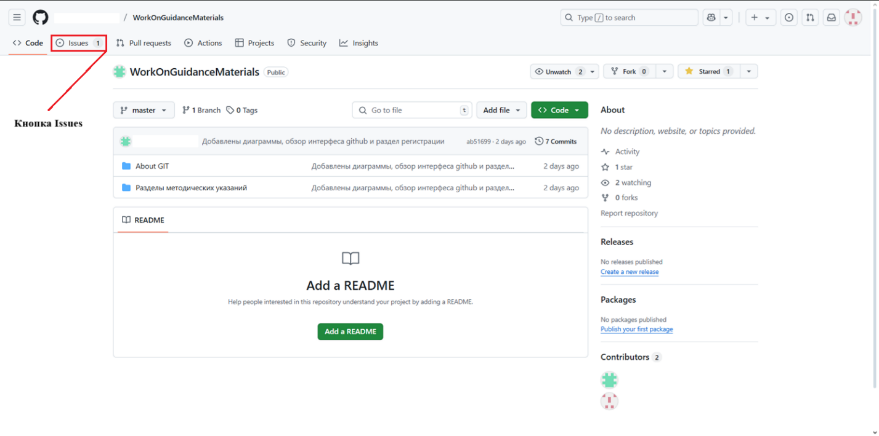
Для того, чтобы попасть в раздел *Issues* сначала необходимо перейти в определенный репозиторий. Кнопка раздела *Issues* на рисунке 7 :

Рисунок 7 – Кнопка раздела *Issues*

В разделе *Issues* будет показан список актуальных обсуждений проекта. Пример раздела *Issues* продемонстрирован на рисунке 8:

Рисунок 8 – Раздел *Issues*

***Collaborators***



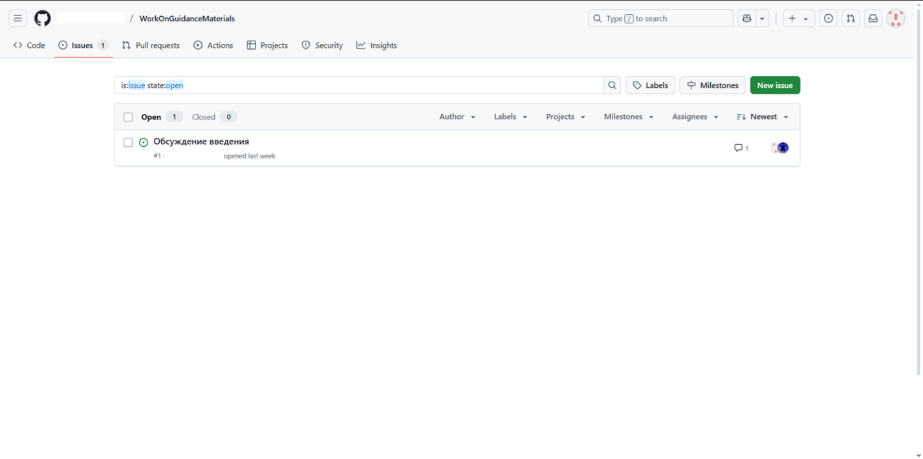
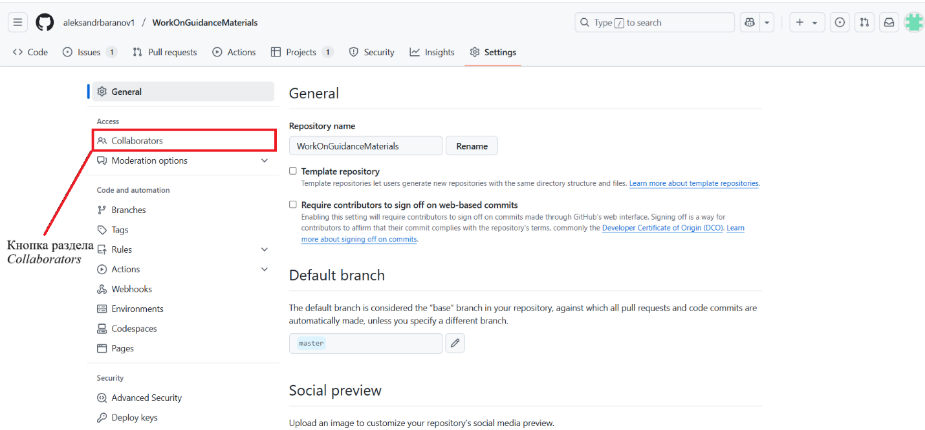
Чтобы попасть в раздел *Collaborators* нужно перейти в раздел *Settings* со страницы конкретного репозитория. Кнопка раздела *Setting* на Рисунке 9:

Рисунок 9 – Кнопка раздела *Settings*

В разделе *Settings* необходимо нажать кнопку *Collaborators.* Кнопка *Collaborators* на рисунке 10:

Рисунок 10 – Кнопка раздела *Collaborators*

В разделе *Collaborators* можно добавлять или удалять людей, которые будут иметь доступ к работе с репозиторием. Раздел *Collaborators* на Рисунке 11:

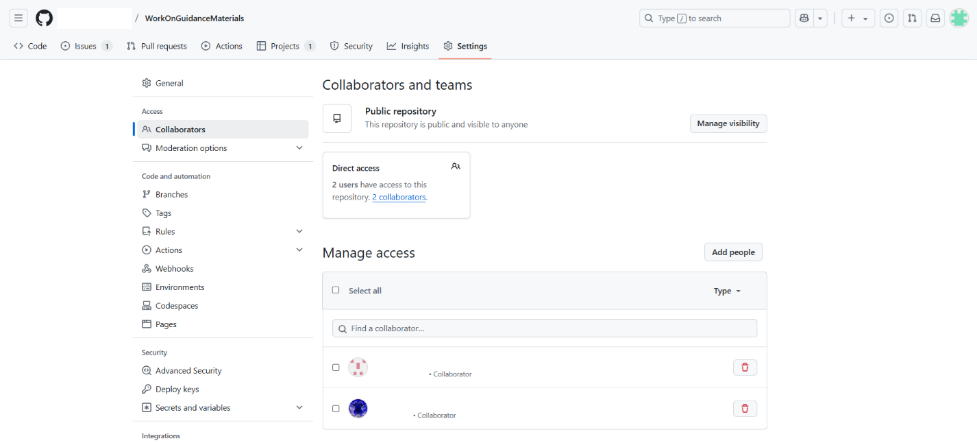
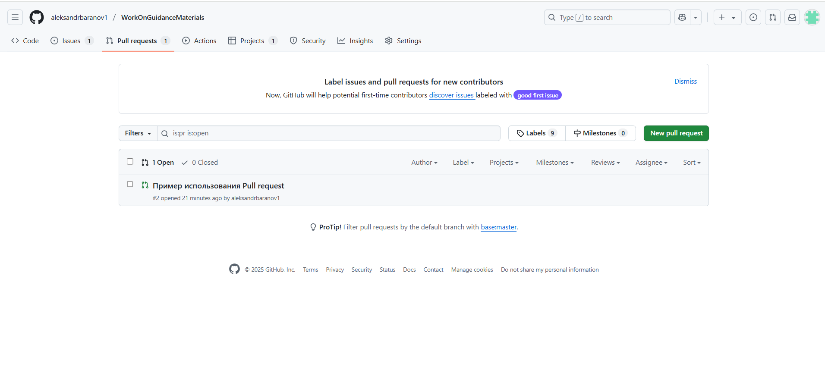


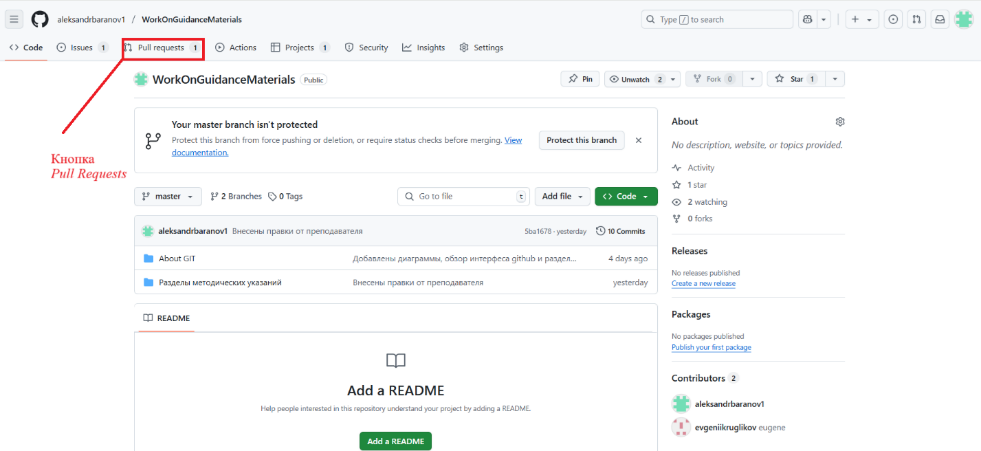
Рисунок 11 – Раздела *Collaborators*

***Pull requests***

Чтобы попасть в раздел *Pull requests* необходимо находясь в нужном репозитории нажать на кнопку *Pull requests*. Кнопка *Pull requests* на Рисунке 12:

Рисунок 12 – Кнопка *Pull requests*

В разделе *Pull requests* можно создавать запросы на слияния, либо просматривать актуальные запросы. Раздел *Pull requests* на Рисунке 13:

****Рисунок 13 – Раздел *Pull requests*

***Tokens***

Чтобы получить токен, сначала необходимо перейти в раздел *Settings*. Кнопка раздела *Settings* на Рисунке 14:

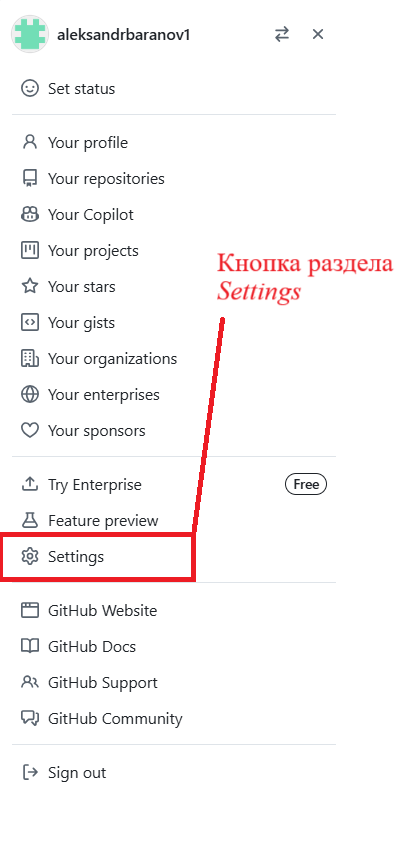
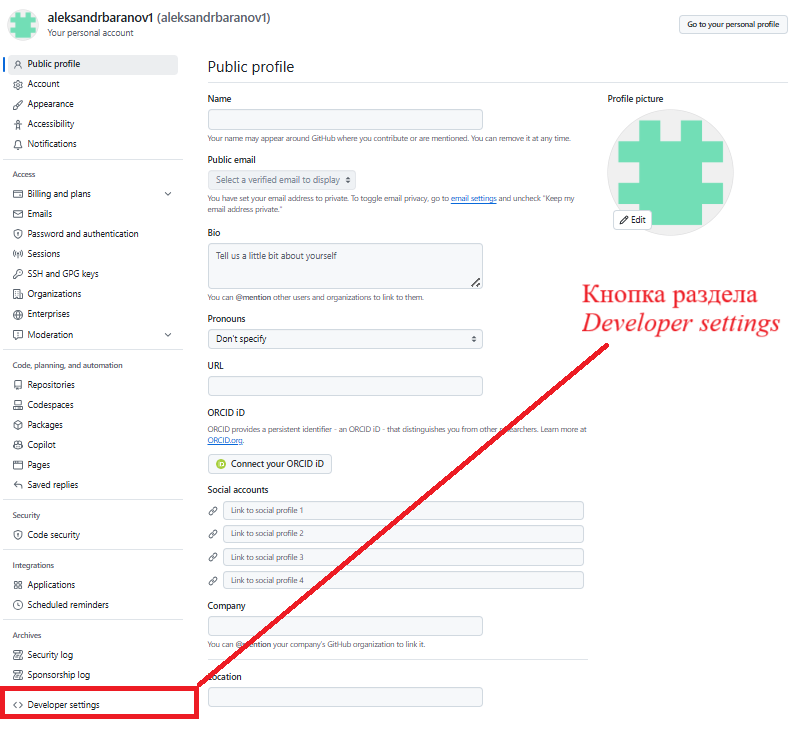


Рисунок 14 – Кнопка раздела *Settings*

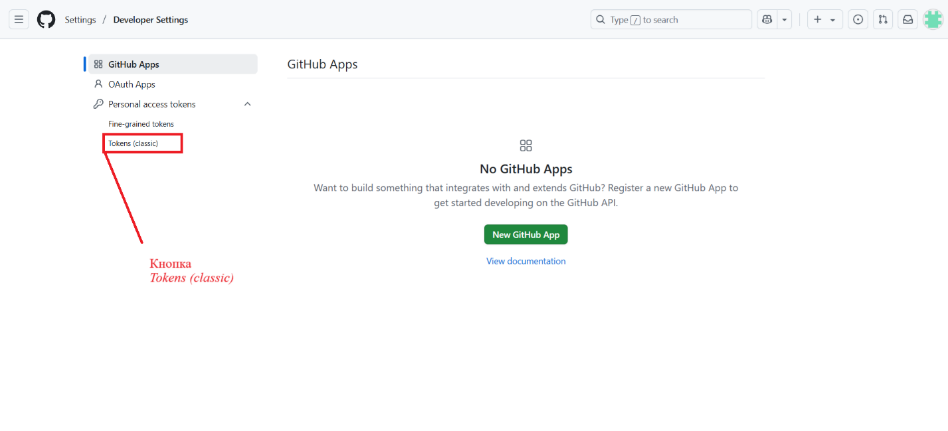
Далее необходимо перейти в раздел *Developer settings*. Кнопка раздела *Developer settings* на Рисунке 15:

Рисунок 15 – Кнопка раздела *Developer* *settings*



Далее необходимо развернуть раскрыть выпадающий список *Personal access tokens*, после чего перейти в раздел *Tokens (classic)*. Кнопка *Tokens (classic)* на Рисунке 16:

Рисунок 16 – Кнопка раздела *Tokens (classic)*

Раздел *Tokens (classic)* на Рисунке 17:

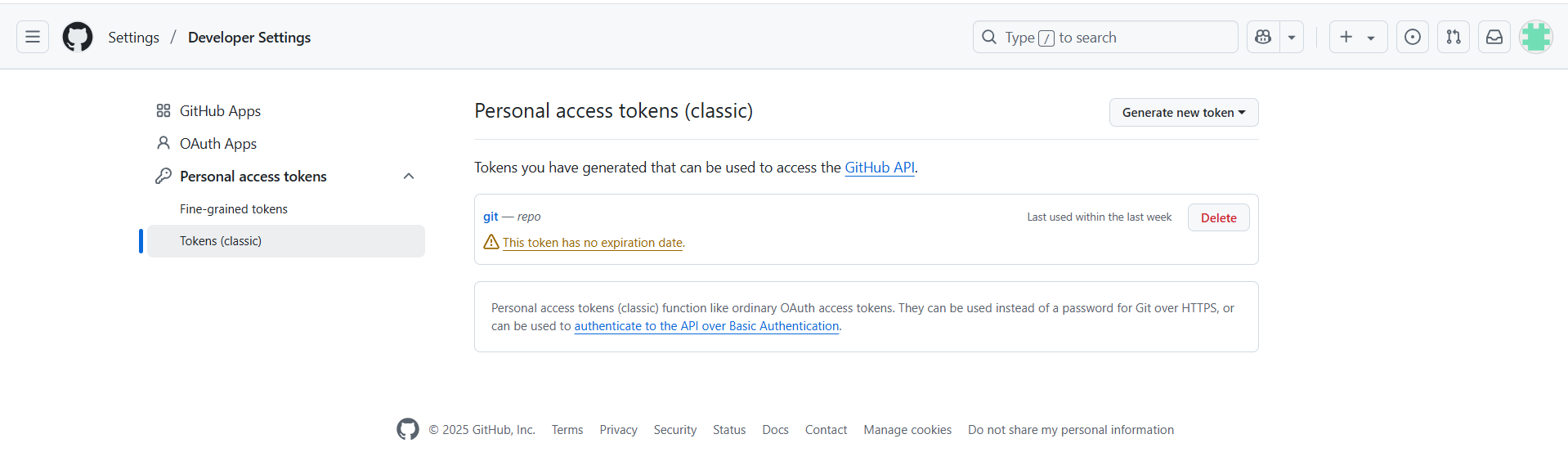


Рисунок 17 – Раздел *Tokens (classic)*

# Задачи для самостоятельного решения

Оглавление

[Введение 3](#_Toc203006201)

[Задачи для самостоятельного решения 3](#_Toc203006202)

УЧЕБНО-МЕТОДИЧЕСКОЕ ИЗДАНИЕ

Сафронов Антон Игоревич

Баранов Александр Алексеевич

Кругликов Евгений Александрович

Основы работы в системе контроля версий *Git*

Учебно-методическое пособие

для проведения лабораторных работ и практических занятий по дисциплине «Программирование и основы алгоритмизации»

Изд. № 99-20