

**Министерство науки и высшего образования Российской Федерации
Федеральное государственное автономное образовательное учреждение
высшего образования
«СЕВЕРО-КАВКАЗСКИЙ ФЕДЕРАЛЬНЫЙ УНИВЕРСИТЕТ»**

**Институт перспективной инженерии
Департамент цифровых, робототехнических систем и электроники**

**ОТЧЕТ
ПО ЛАБОРАТОРНОЙ РАБОТЕ №1
дисциплины «Программирование на Python»**

Выполнил:
Оразов Тимур
2 курс, группа ИВТ-б-о-24-1,
09.03.01 «Информатика и
вычислительная техника»,
направленность (профиль)
«Программное обеспечение средств
вычислительной техники и
автоматизированных систем», очная
форма обучения

(подпись)

Руководитель практики:
Воронкин Р. А.

(подпись)

Отчет защищен с оценкой _____ Дата защиты _____

Ставрополь, 2025 г.

Тема: Исследование основных возможностей Get и GitHub
Цель: исследовать базовые возможности системы контроля версий Git и веб-сервиса для хостинга IT-проектов GitHub.
Ссылка на репозиторий: <https://github.com/tim12zxc12/Tim>

Ход работы:

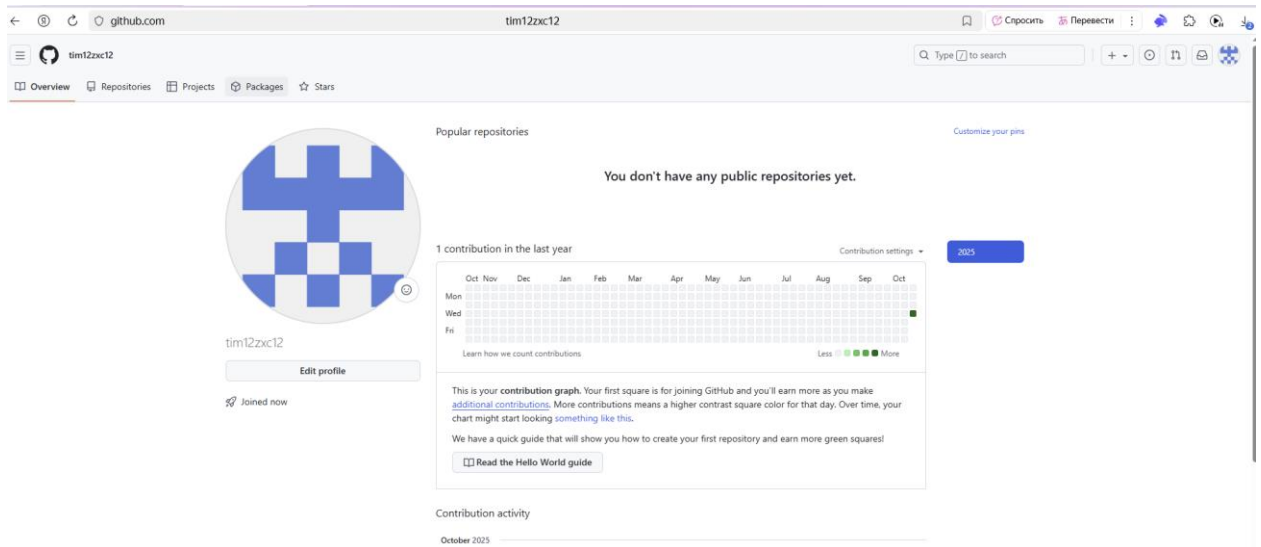


Рис. 1 – создание персонального аккаунта на сайте GitHub.

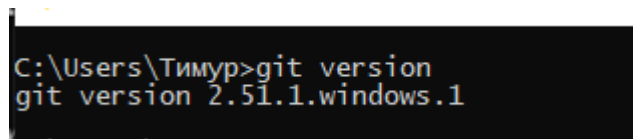


Рис. 2 – установка git, команда для проверки версии.

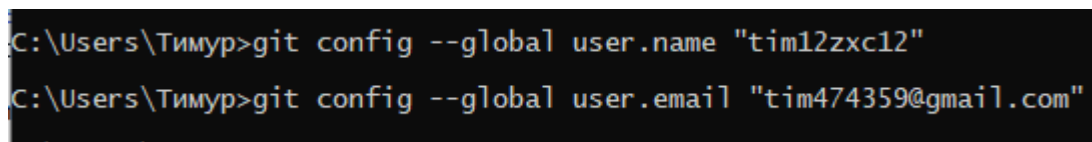


Рис. 3 – добавление в настройки имени, почты.

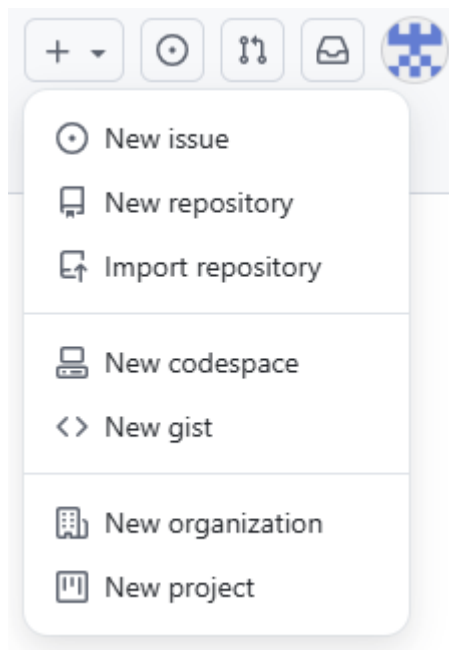



Рис. 4 – переход на страницу создания репозитория.

Create a new repository

Repositories contain a project's files and version history. Have a project elsewhere? [Import a repository](#).
Required fields are marked with an asterisk (*).

- General**

Owner *  tim12zxc12 / **Repository name *** IWT-TIMUR
✔ IWT-TIMUR is available.

Great repository names are short and memorable. How about [scaling-tribble](#)?

Description

0 / 350 characters
- Configuration**

Choose visibility *
Choose who can see and commit to this repository Public

Add README On ☒
READMEs can be used as longer descriptions. [About READMEs](#)

Add .gitignore Python
.gitignore tells git which files not to track. [About ignoring files](#)

Add license MIT License
Licenses explain how others can use your code. [About licenses](#)

[Create repository](#)

Рис. 5 – создание, настройка репозитория.

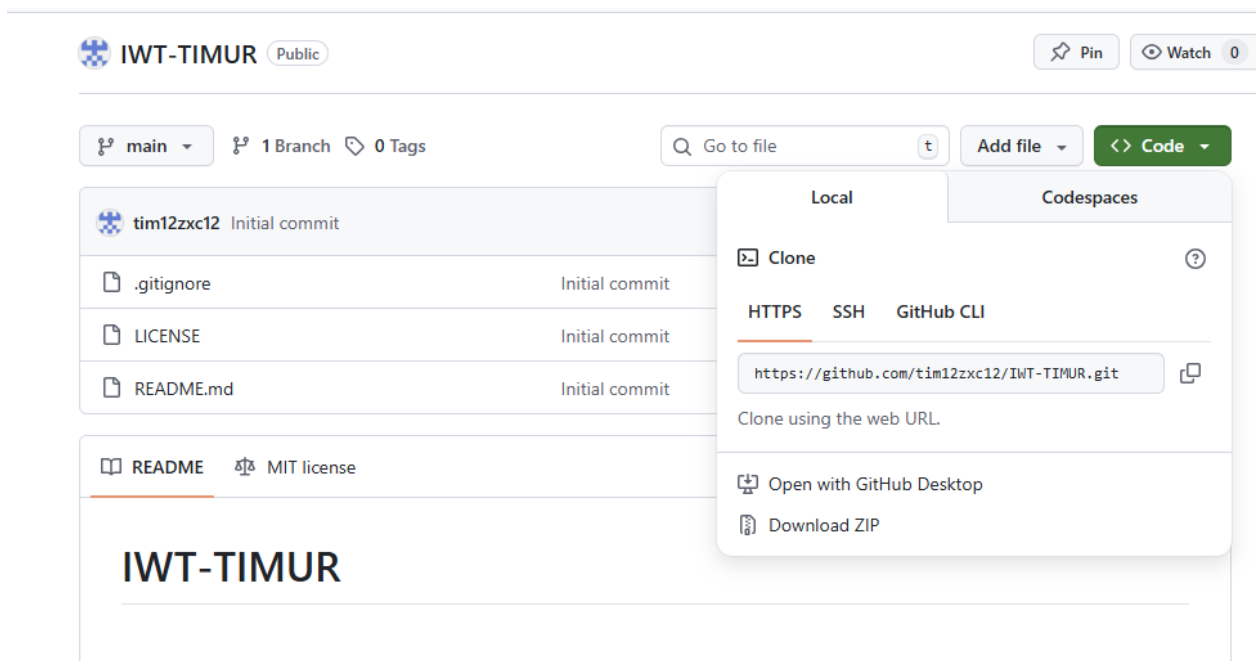


Рис. 6 – копирование адреса репозитория.

```
C:\Users\Тимур\IWT-TIMUR>git clone https://github.com/tim12zxc12/IWT-TIMUR.git
Cloning into 'IWT-TIMUR'...
remote: Enumerating objects: 5, done.
remote: Counting objects: 100% (5/5), done.
remote: Compressing objects: 100% (4/4), done.
remote: Total 5 (delta 0), reused 0 (delta 0), pack-reused 0 (from 0)
Receiving objects: 100% (5/5), done.
```

Рис. 7 – копирование репозитория на устройство.

```
receiving objects: 100% (1/1), done.
C:\Users\Тимур>cd IWT-TIMUR
```

Рис. 8 – переход к файлу проекта.

```
C:\Users\Тимур\IWT-TIMUR>git status
On branch main
Your branch is up to date with 'origin/main'.

nothing to commit, working tree clean
C:\Users\Тимур\IWT-TIMUR>
```

Рис. 9 - проверка состояния репозитория

```
README – Блокнот
Файл  Правка  Формат  Вид  Справка
# IWT-TIMUR
Оразов Тимур ИВТ-6-о-24-1
```

Рис. 10 – переход в файл README.md– ручное изменени

```

1 print ("Программа")
2 name = input ("Введите фамилию: ")
3 name = input ("Введите имя: ")
4
5 a = int(input("Введите a: "))
6 z = int(input("Введите z: "))
7 f = int(input("Введите f: "))
8 s = a + z + f
9 i = a - z - f
10 print(s,i,t)

```

Рис. 11 – создание простой программы.

```

C:\Users\Тимур\Tim>git add README.md
C:\Users\Тимур\Tim>git commit -m "Ввод a"
[main 0bb949b] Ввод a
 1 file changed, 3 insertions(+), 1 deletion(-)
C:\Users\Тимур\Tim>git add README.md
C:\Users\Тимур\Tim>git commit -m "Ввод z"
On branch main
Your branch is ahead of 'origin/main' by 5 commits.
  (use "git push" to publish your local commits)

nothing to commit, working tree clean
C:\Users\Тимур\Tim>git add README.md
C:\Users\Тимур\Tim>git commit -m "Ввод f"
[main 2ac531c] Ввод f
 1 file changed, 3 insertions(+), 1 deletion(-)
C:\Users\Тимур\Tim>git add README.md
C:\Users\Тимур\Tim>git commit -m "сумма"
[main 04ef983] сумма
 1 file changed, 2 insertions(+), 1 deletion(-)
C:\Users\Тимур\Tim>git add README.md
C:\Users\Тимур\Tim>git commit -m "разность"
[main f66d9b9] разность
 1 file changed, 2 insertions(+), 1 deletion(-)
C:\Users\Тимур\Tim>git add README.md
C:\Users\Тимур\Tim>git commit -m "квадрат"
[main ef49243] квадрат
 1 file changed, 2 insertions(+), 1 deletion(-)
C:\Users\Тимур\Tim>git add README.md
C:\Users\Тимур\Tim>git commit -m "вывод значений"
[main eelfc95] вывод значений
 1 file changed, 2 insertions(+), 1 deletion(-)
C:\Users\Тимур\Tim>git add.
git: 'add.' is not a git command. see 'git --help'.

```

Рис. 12 – построчное добавление коммитов параллельно с кодом.

```

C:\Users\Тимур\Tim>git status
On branch main
Your branch is ahead of 'origin/main' by 10 commits.
(use "git push" to publish your local commits)

nothing to commit, working tree clean

C:\Users\Тимур\Tim>git push
fatal: unable to access 'https://github.com/tim12zxc12/Tim.git/': could not resolve host: github.com

C:\Users\Тимур\Tim>git push
Enumerating objects: 32, done.
Counting objects: 100% (32/32), done.
Delta compression using up to 8 threads
Compressing objects: 100% (29/29), done.
Writing objects: 100% (30/30), 2.61 KiB | 668.00 KiB/s, done.
Total 30 (delta 18), reused 0 (delta 0), pack-reused 0 (from 0)
remote: Resolving deltas: 100% (18/18), completed with 1 local object.
To https://github.com/tim12zxc12/Tim.git
  58aace9..ee1fc95  main -> main

```

Рис. 13 - локальный и удаленный репозитории полностью синхронизированы.

git pull - проверка обновления на GitHub (все актуально)
git push - отправка свои коммиты на GitHub (все отправлено)
git push --set-upstream origin main - настройка отслеживания ветки.

Ответы на контрольные вопросы:

1. Что такое СКВ (git) и каково ее назначение?
Система контроля версий (СКВ) — это программа для отслеживания изменений в файлах. Позволяет сохранять историю, возвращаться к предыдущим версиям и работать в команде.
2. В чем недостатки локальных и централизованных СКВ?
Локальные: нет совместной работы, риск потери данных.
Централизованные: единый сервер — если он упадет, работа остановится.
3. К какой СКВ относится Git?
К распределенной (каждый разработчик имеет полную копию репозитория).
4. В чем концептуальное отличие Git от других СКВ?
Git хранит не изменения файлов, а снимки всего проекта на каждый коммит.
5. Как обеспечивается целостность хранимых данных в Git?
Через хеш-суммы (SHA-1). Каждый коммит и файл имеют уникальный хеш.
6. В каких состояниях могут находиться файлы в Git?
modified (изменен),
staged (подготовлен к коммиту),
committed (сохранен в репозитории).
7. Что такое профиль пользователя в GitHub?
Это ваша учетная запись на github.com с репозиториями, настройками и историей действий.
8. Какие бывают репозитории в GitHub?
Public (открытые для всех),
Private (только для вас и collaborators).

9. Основные этапы модели работы с GitHub:
Клонировать(`git clone`) → Изменить(`modify`) → Добавить (`git add`) →
Закоммитить (`git commit`) → Запустить (`git push`).

10. Как осуществляется первоначальная настройка Git после установки?

`git config --global user.name "Ваше имя"`

`git config --global user.email "ваша@почта.com"`

11. Этапы создания репозитория в GitHub:
Зарегистрироваться → Нажать «New repository» → Ввести имя → Выбрать лицензию → Create.

12. Какие типы лицензий поддерживаются GitHub?
MIT, GPL, Apache — они определяют правила использования вашего кода.

13. Как осуществляется клонирование репозитория?

`git clone https://github.com/username/repo.git`

Нужно, чтобы скопировать проект с GitHub на компьютер.

14. Как проверить состояние локального репозитория?

`git status`

15. Как изменяется состояние после операций?

`git add` — файлы из `modified` переходят в `staged`.

`git commit` — файлы из `staged` переходят в `committed`.

`git push` — изменения отправляются на сервер.

16. Как синхронизировать два компьютера с GitHub?
На каждом компьютере:

`git clone https://github.com/username/repo.git`

После изменений:

`git pull` - получить обновления

`git add . && git commit -m "update" && git push` отправить свои

17. Другие сервисы, кроме GitHub?

GitLab: больше функций для CI/CD, можно hosting свой. Веб-платформа для управления репозиториями, проектами.

Bitbucket: бесплатные private-репозитории.

18. Программы с графическим интерфейсом для Git?

GitKraken, Sourcetree, GitHub Desktop.

Вместо команд `add/commit` — выделяете файлы и жмете кнопки «Stage» и «Commit».

Вывод: в ходе работы были исследованы базовые возможности системы контроля версий Git и веб-сервиса для хостинга IT-проектов GitHub.