

Практическая работа №1

Введение в программирование на языке C/C++

Мухаметов Данил Илгизович

seemsclever@mail.ru

2025

Содержание

Задание 1. Установка Git и создание GitHub-репозитория	3
1. Установка Git.....	3
2. Первичная настройка Git	3
3. Создание аккаунта и репозитория на GitHub	3
4. Подключение локального проекта к GitHub.....	3
Ожидаемый результат	4
Примечание	4
Задание 2. Написание программы на основе пройденной темы	4
Вариант 1. Приветствие пользователя.....	4
Итог выполнения практической работы	5

Задание 1. Установка Git и создание GitHub-репозитория

Цель - освоить установку и базовую настройку системы контроля версий Git, а также научиться работать с удалённым репозиторием на GitHub.

1. Установка Git

1. Перейти на официальный сайт: <https://git-scm.com/downloads>
2. Выбрать версию для своей операционной системы (Windows, macOS или Linux).
3. Установить Git, следуя шагам мастера установки. Для Windows рекомендуется оставить параметры по умолчанию.
4. Проверить установку: открыть терминал или командную строку и ввести:

```
git --version
```

Если Git установлен, вы увидите номер версии.

2. Первичная настройка Git

После установки нужно указать имя пользователя и адрес электронной почты, которые будут использоваться в коммитах:

```
git config --global user.name "Ваше Имя"  
git config --global user.email "ваш.email@example.com"
```

3. Создание аккаунта и репозитория на GitHub

1. Зарегистрироваться или войти на сайт <https://github.com>
2. Нажать New repository (создать новый репозиторий).
3. Указать название: <Фамилия>-cpp-homeworks.
4. Добавить описание (по желанию).
5. Видимость - Public (открытый).
6. Поставить галочку Add README.md, чтобы сразу создать файл описания.
7. Нажать Create repository.

4. Подключение локального проекта к GitHub

1. На компьютере создать папку для проекта (например, CppHomeworksProject).
2. Открыть папку в терминале/консоли и ввести команды:

```
git init  
git remote add origin <ссылка_на_репозиторий>
```

(ссылку можно скопировать на странице GitHub-репозитория — HTTPS или SSH)

3. Создать файл программы homework1.cpp (например, с простой программой «Hello, World!»).
4. Добавить файл в Git и зафиксировать изменения:

```
git add .  
git commit -m "First commit"
```

5. Отправить изменения в GitHub:

```
git push -u origin master
```

6. Проверить, что файлы появились в репозитории на GitHub.

Ожидаемый результат

- Git установлен и настроен.
- У студента есть рабочий аккаунт на GitHub.
- Создан удалённый репозиторий, содержащий в названии фамилию студента.
- В репозитории находится первый загруженный файл программы на C++.

Примечание

Если не получилось – прочтите статью <https://gb.ru/blog/github-nastrojka/>.

При использовании онлайн-компилятора onlinegdb.com допускается скачать созданный файл программы и загрузить его в репозиторий вручную.

В случае работы в среде Microsoft Visual Studio рекомендуется загружать в репозиторий весь проект целиком.

Не забывайте соблюдать правила именования файлов:

- для первого практического занятия - homework1,
- для второго практического занятия - homework2,
- и далее по аналогии.

Задание 2. Написание программы на основе пройденной темы

Код должен быть оформлен в соответствии с общими требованиями:

- имена переменных в camelCase,
- фигурные скобки - в стиле K&R или Allman,
- использовать комментарии (//),
- программа должна выполняться без ошибок и решать поставленную задачу.

Вариант 1. Приветствие пользователя

Программа запрашивает имя пользователя (строку) и выводит персональное приветствие.

Итог выполнения практической работы

У каждого студента должен быть создан GitHub-репозиторий с загруженным проектом или файлом. В репозитории должен находиться файл с расширением `spp`, содержащий решение одного из вариантов задания.