Министерство образования Новосибирской области

ГБПОУ НСО «Новосибирский авиационный технический колледж имени Б.С. Галущака»

Публикация проекта в Git

Методическое пособие

Учебная дисциплина: МДК.01.01 Разработка программных модулей

Специальность: 09.02.07 Информационные системы и программирование

Разработал:

студент группы ПР-20.101

Зиберева Анна

2022

**Содержание**

[1 Установка GIT Bash 3](#_Toc117283598)

[2 Настройка GIT 5](#_Toc117283599)

[3 Создание репозитория 6](#_Toc117283600)

[3.1 Создание нового репозитория 6](#_Toc117283601)

[3.2 Определение состояния репозитория 6](#_Toc117283602)

[4 Подготовка файлов 8](#_Toc117283603)

[4.1 Добавление файлов 8](#_Toc117283604)

[4.2 Создание коммитов 8](#_Toc117283605)

[5 Отправка файлов на сервер 10](#_Toc117283606)

[5.1 Отправка проектов 10](#_Toc117283607)

[5.2 Создание удаленного репозитория на GitHub 10](#_Toc117283608)

[5.3 Работа с удаленным репозиторием на GitHub 12](#_Toc117283609)

[Список источников 15](#_Toc117283610)

# **1 Установка GIT Bash**

Для установки GIT Bash, необходимо перейти на официальный электронный ресурс gitforwindows.org.

Главная страница ресурса представлена на рисунке 1.

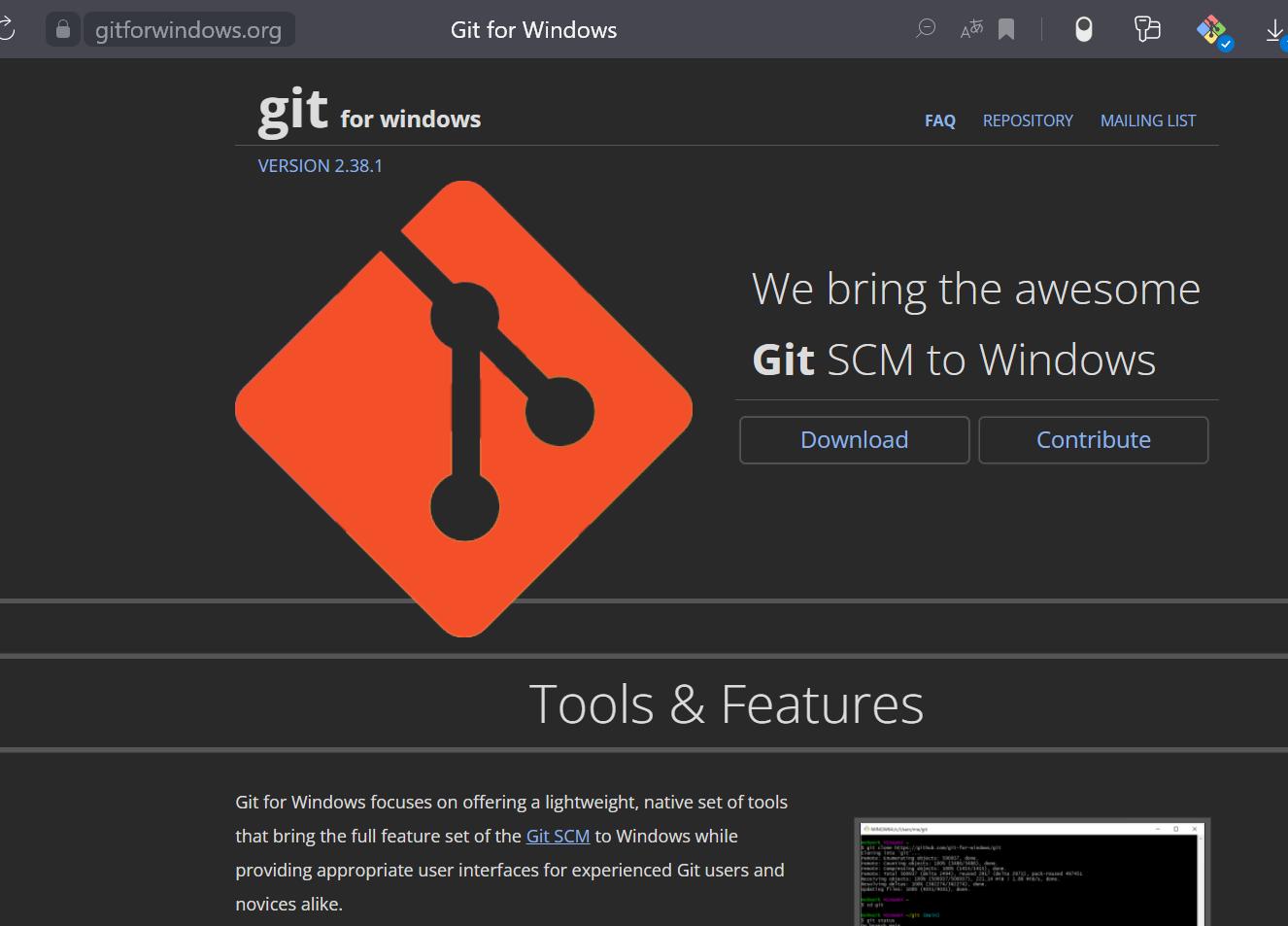


Рисунок 1 — Главная страница ресурса

На странице нужно найти кнопку «Download» и нажат на нее. После этого автоматически начнется скачивание .exe файла. Ниже расположен рисунок 2 c внешним видом файла.



Рисунок 2 — Иконка установщика

После этого необходимо нажать на ярлык скачанного установщика для запуска установки программы. На рисунке 3 описывается последовательность действий для установки Git Bash.

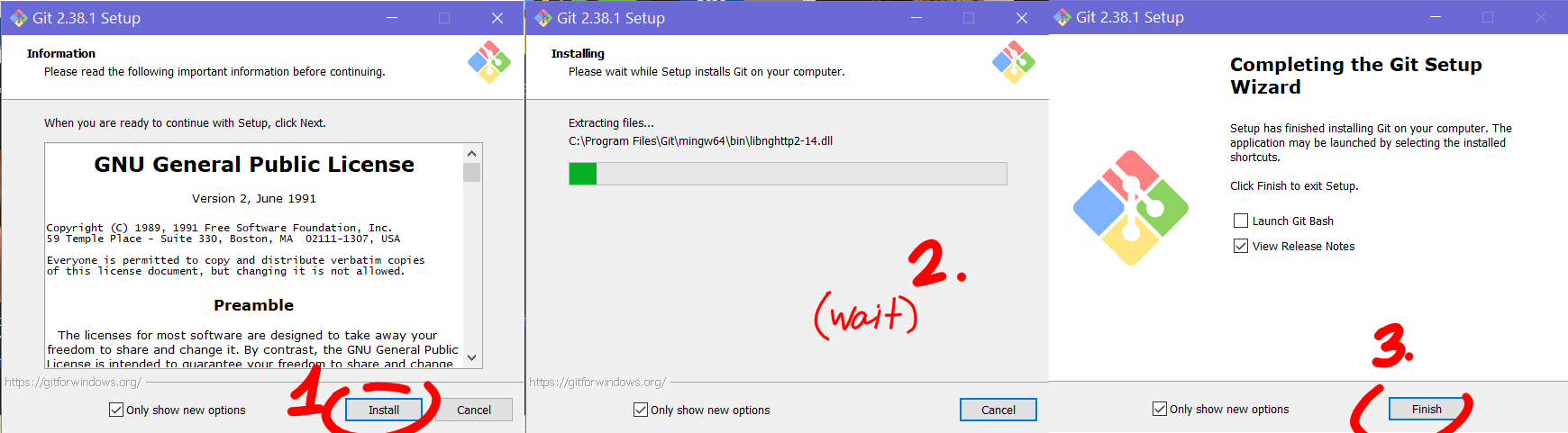


Рисунок 3 — Последовательность установки

# **2 Настройка GIT**

Первое, что вам следует сделать после установки Git — указать ваше имя и адрес электронной почты. Это важно, потому что каждый коммит в Git содержит эту информацию, и она включена в коммиты, передаваемые вами, и не может быть далее изменена:

$ git config --global user.name "Jane Doe"

$ git config --global user.email janedoe@example.com

# **3 Создание репозитория**

# **3.1 Создание нового репозитория**

Чтобы создать новый репозиторий, нам нужно зайти в папку нашего проекта, с помощью ПКМ найти пункт «Git Bash here» и выполнить команду «git init». Это включит приложение в этой конкретной папке и создаст скрытую директорию .git, где будет храниться история репозитория и настройки.

Для этого в окне терминала введите команду:

* git init

Выполнение и результат команды показаны на рисунке 4.

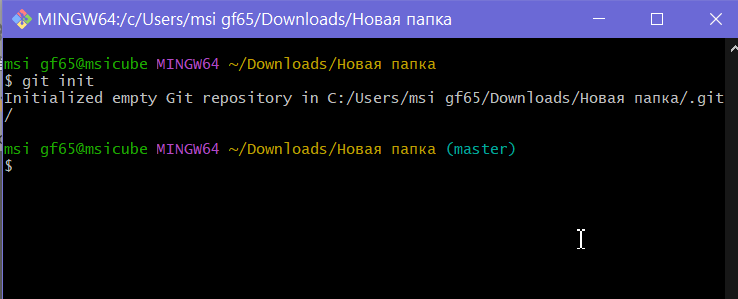


Рисунок 4 — Результат выполнения команды «git init» в терминале

# **3.2 Определение состояния репозитория**

status — это еще одна важнейшая команда, которая показывает информацию о текущем состоянии репозитория: актуальна ли информация на нём, нет ли чего-то нового, что поменялось, и так далее.

Для этого в окне терминала введите команду:

* git status

Результат выполнения команды показан на рисунке 5.

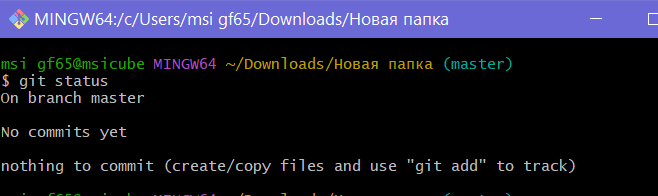


Рисунок 5 — Результат выполнения команды «git status» в терминале

# **4 Подготовка файлов**

# **4.1 Добавление файлов**

Добавить файл можно командами:

* git add "Имя файла"
* git add –A
* git add .

Пример выполнения команды представлен на рисунке 6.

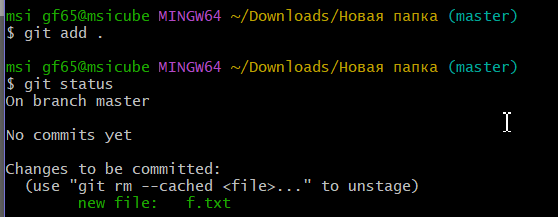


Рисунок 6 — Результат выполнения команды «git status» в терминале

В папке «Новая папка» есть файл «f.txt». Мы добавляем данный файл с помощью команды «add», а после проверяем результат: «new file: f.txt» говорит об успешном выполнении команды.

# **4.2 Создание коммитов**

Для сохранения изменений, их необходимо закоммитить с помощью команды:

* git commit -m ' Сommit message '

Флажок -m задаст commit message - комментарий разработчика. Он необходим для описания изменений. Для просмотра все выполненных фиксаций можно воспользоваться историей коммитов. Она содержит сведения о каждом проведенном коммите проекта. Запросить ее можно при помощи команды:

* git log

Пример выполнения данных команд представлен на рисунке 7.

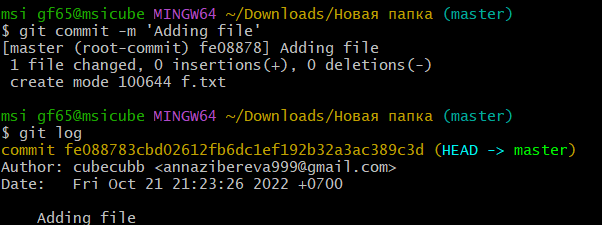


Рисунок 7 — Результат выполнения команды «git commit» и «git log» в терминале

# **5 Отправка файлов на сервер**

# **5.1 Отправка проектов**

Можно переслать наш локальный коммит на сервер Этот процесс происходит каждый раз, когда мы хотим обновить данные в удаленном репозитории.

Команда, предназначенная для этого:

* git push origin master

Команда принимает два параметра: имя удаленного репозитория (например origin) и ветку, в которую необходимо внести изменения (master — это ветка по умолчанию для всех репозиториев).

# **5.2 Создание удаленного репозитория на GitHub**

Также опубликовать ваш проект можно на удаленный репозиторий на ресурсе github.com (система контроля версий).

На рисунке 8 показана страница, на которой необходимо нажать на кнопку «New», чтобы создать новый репозиторий.

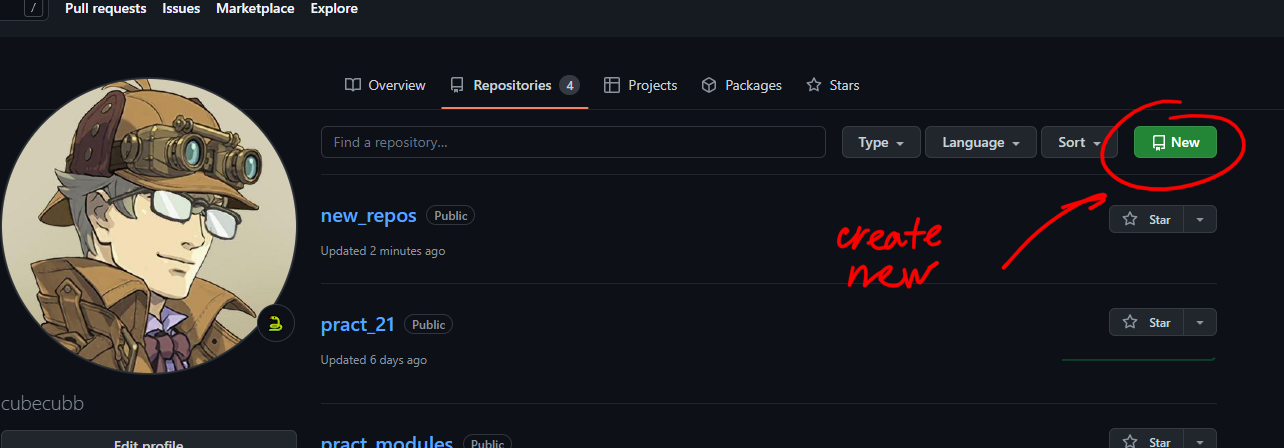


Рисунок 8 — Страница с репозиториями вашего аккаунта на GitHub

Далее необходимо настроить свой новый репозиторий. На рисунке 9 представлена последовательность для настройки нового репозитория: можно добавить файл «README», настроить доступ к репозиторию и т.д.

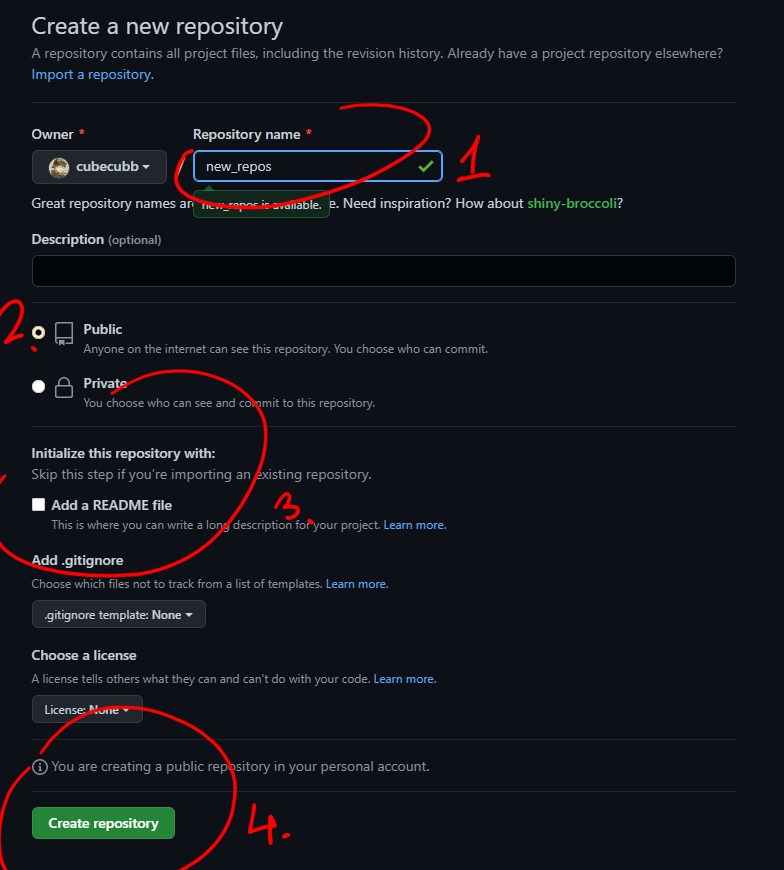


Рисунок 9 — Страница настройки удаленного репозитория

# **5.3 Работа с удаленным репозиторием на GitHub**

Далее мы можем приступить к работе с нашим удаленным репозиторием. Для начала, чтобы подготовить данный файл к работе нам необходимо выполнить все предыдущие пункты. Вся последовательность выполнения шагов показана на рисунке 10.

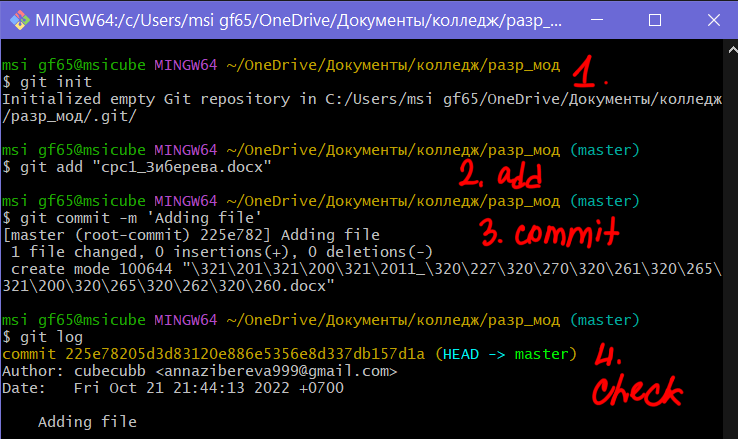


Рисунок 10 — Подготовка файла к публикации

Далее нам нужно выполнить ряд команд в GitBash. Сперва воспользуемся командой «git remote add line <URL>». Команда «git remote» служит для управления списком удалённых репозиториев. Она позволяет сохранять длинные URL репозиториев в виде понятных коротких строк, например "origin". Результат выполнения представлен на рисунке 11.



Рисунок 11 — Команда в терминале

Далее используем команду «git branch -M main». Команда git branch -M main переименовывает ветку master на локальном репозитории в main. Таким образом, изменения, произошедшие на удаленном репозитории, больше не конфликтуют с локальным хранилищем, в котором главная ветка стала также называться main. Выполнение команды в терминале представлено на рисунке 12.

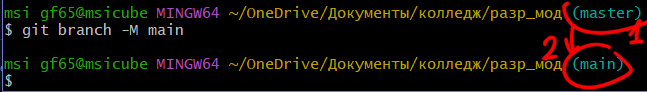


Рисунок 11 — Результат выполнения команды в терминале

Далее нам необходимо «вытолкнуть» наш проект на удаленный репозиторий на ресурсе github.com: для этого мы используем команду «git push -u origin main». Push в программировании означает заталкивание элемента в массив/стек, или git-ветки на удаленный репозиторий. - origin - имя удаленного репозитория и main — это ветка удаленного репозитория. -u — это ключ, который устанавливает связь с веткой main. Этот ключ указывается единожды. Выполнение данной команды представлено на рисунке 12.

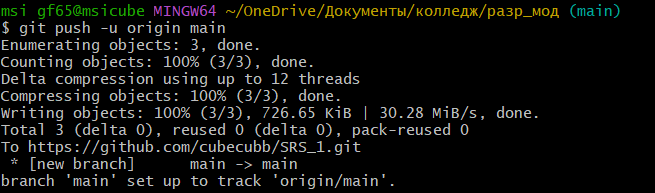


Рисунок 12 — Результат выполнения команды в терминале

Далее мы можем проверить страницу удаленного репозитория на GitHub: наш проект должен был появиться там.

На рисунке 13 представлен окончательный вид репозитория после публикации нашего проекта.

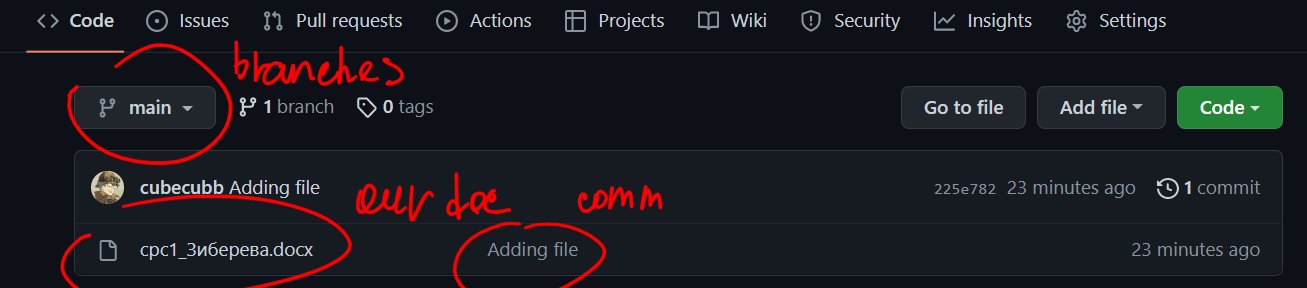


Рисунок 13 — Файл (проект) в удаленном репозитории

# **Список источников**

gitforwindows [Электронный ресурс]. — Загрузка Git Bash — URL: https://gitforwindows.org/ (дата обращения: 21.10.2022)

gitforwindows [Электронный ресурс]. — Работа с системой контроля версий GIT и сервисом GITHub — URL: https://www.webpupil.ru/javascript\_view.php?id=219#461/ (дата обращения: 21.10.2022)