



9-11 классы

## Программирование на C++

Презентация занятия

### QT. Работа с GIT.

32 занятие



Минцифры  
России



**20.35**  
УНИВЕРСИТЕТ

## Тема: QT. Работа с GIT

Вы пишете код, запускаете его, и всё работает как надо. Вы добавляете новую фичу, и всё перестаёт работать. Каждому разработчику знакома ситуация, когда технология не хочет ему подчиняться. Ничто не идеально, и порой что-нибудь ломается. Иногда на поиски небольшой ошибки могут потребоваться часы работы. В таких случаях на помощь приходят системы контроля версий.



## Тема: QT. Работа с GIT

Git — это набор консольных утилит, которые отслеживают и фиксируют изменения в файлах (чаще всего речь идет об исходном коде программ, но вы можете использовать его для любых файлов на ваш вкус). С его помощью вы можете откатиться на более старую версию вашего проекта, сравнивать, анализировать, сливать изменения и многое другое. Этот процесс называется контролем версий.



## Тема: QT. Работа с GIT

Многие люди в качестве метода контроля версий применяют копирование файлов в отдельную директорию, возможно даже в директорию с отметкой по времени для большего контроля. Данный подход всё ещё очень популярен и распространён. Изменения сохраняются в виде наборов патчей, где каждый патч датируется и получает отметку времени. Таким образом, если код перестаёт работать, наборы патчей можно совместить, чтобы получить исходное состояние файла.

### Локальный компьютер



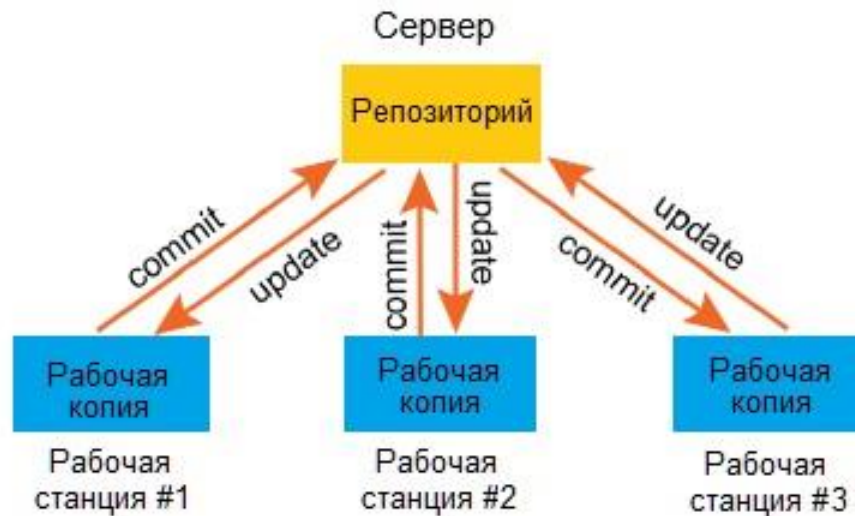
## Тема: QT. Работа с GIT

### Централизованные системы контроля версий

ЦСКВ были созданы для решения проблемы взаимодействия с другими разработчиками. Такие системы имеют единственный сервер, содержащий все версии файлов, и некоторое количество клиентов, которые получают файлы из этого централизованного хранилища и там же их сохраняют. Тем не менее, такой подход имеет существенный недостаток — выход сервера из строя обернётся потерей всех данных.

## Тема: QT. Работа с GIT

### Централизованная система контроля версий



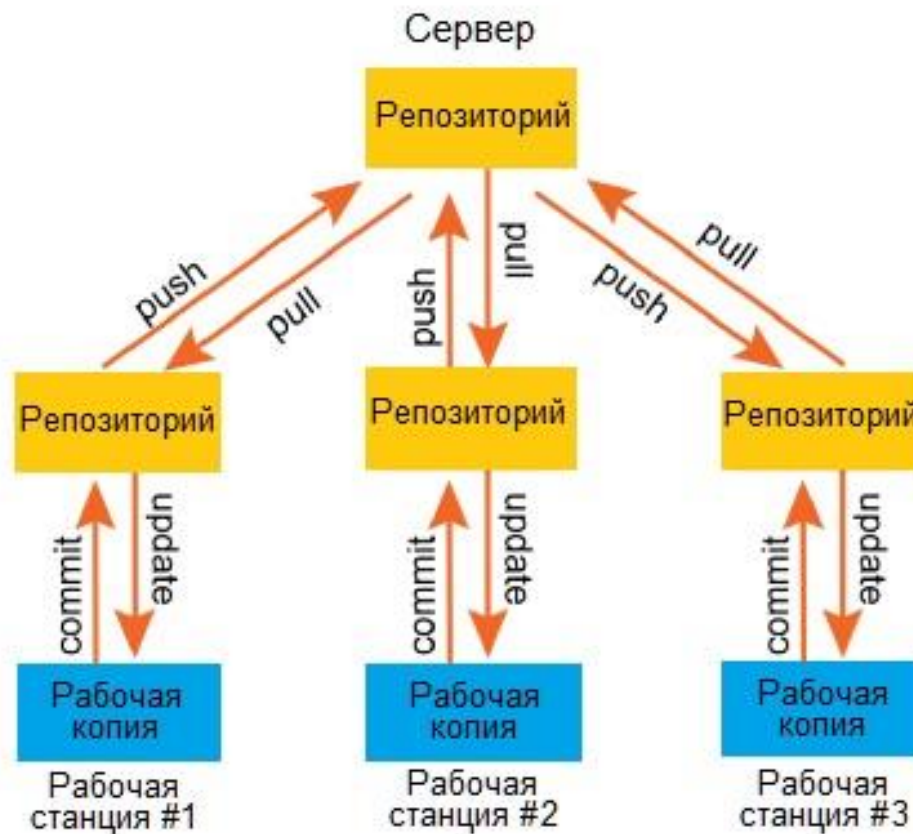
## Тема: QT. Работа с GIT

### Распределённые системы контроля версий

Недостаток ЦСКВ был исправлен в РСКВ, клиенты которых не просто скачивают снимок всех файлов (состояние файлов на определённый момент времени), а полностью копируют репозиторий. Это значит, что у каждого клиента есть копия всего исходного кода и внесённых изменений. В этом случае, если один из серверов выйдет из строя, любой клиентский репозиторий может быть скопирован на другой сервер для продолжения работы. Ещё одним преимуществом РСКВ является то, что они могут одновременно взаимодействовать с несколькими удалёнными репозиториями, что означает, что вы можете параллельно работать над несколькими проектами.

## Тема: QT. Работа с GIT

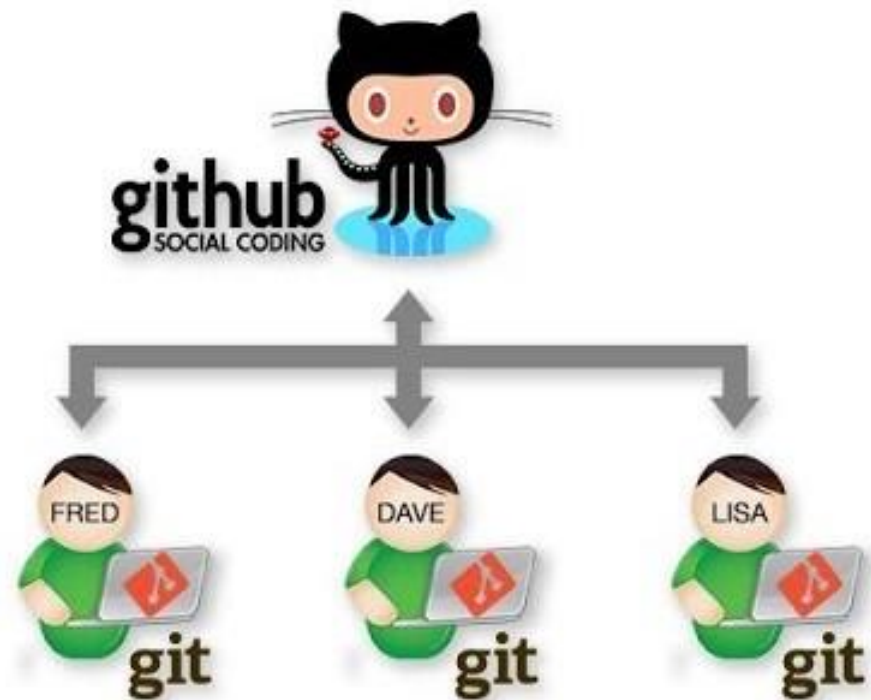
### Распределённая система контроля версий





## Тема: QT. Работа с GIT

Git является распределенным, то есть не зависит от одного центрального сервера, на котором хранятся файлы. Вместо этого он работает полностью локально, сохраняя данные в папках на жестком диске, которые называются репозиторием. Тем не менее, вы можете хранить копию репозитория онлайн, это сильно облегчает работу над одним проектом для нескольких людей. Для этого используются сайты вроде github, gitlab и bitbucket.



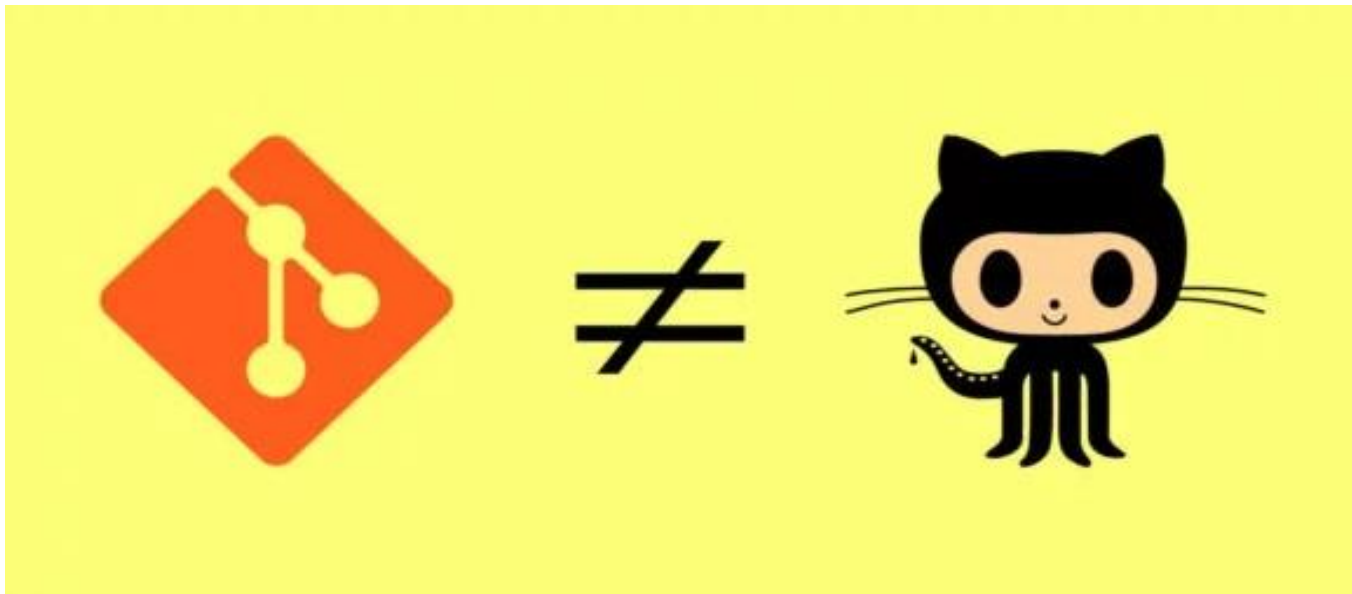
## Тема: QT. Работа с GIT

GitHub — сервис онлайн-хостинга репозиторий, обладающий всеми функциями распределённого контроля версий и функциональностью управления исходным кодом — всё, что поддерживает Git и даже больше. Обычно он используется вместе с Git и даёт разработчикам возможность сохранять их код онлайн, а затем взаимодействовать с другими разработчиками в разных проектах. Также GitHub может похвастаться контролем доступа, багтрекингом, управлением задачами и вики для каждого проекта. Цель GitHub — содействовать взаимодействию разработчиков.

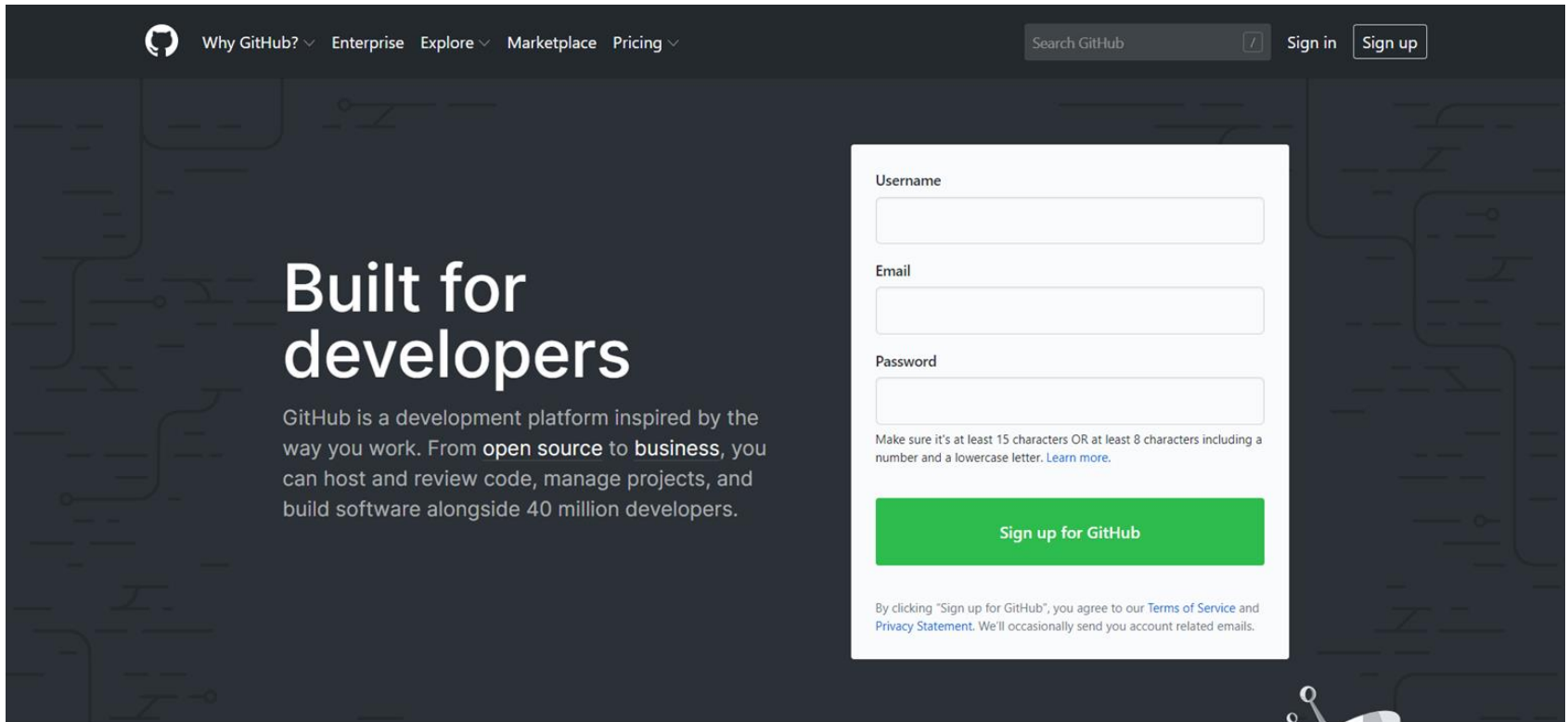


## Тема: QT. Работа с GIT

Git — это инструмент, позволяющий реализовать распределённую систему контроля версий, а GitHub — это сервис для проектов, использующих Git.



## Тема: QT. Работа с GIT



The screenshot shows the GitHub homepage with a dark background and a white sign-up form on the right. The form includes fields for Username, Email, and Password, along with a green 'Sign up for GitHub' button. The background text reads 'Built for developers' and describes GitHub as a development platform.

Why GitHub? ▾ Enterprise Explore ▾ Marketplace Pricing ▾

Search GitHub / Sign in Sign up

# Built for developers

GitHub is a development platform inspired by the way you work. From **open source** to **business**, you can host and review code, manage projects, and build software alongside 40 million developers.

Username

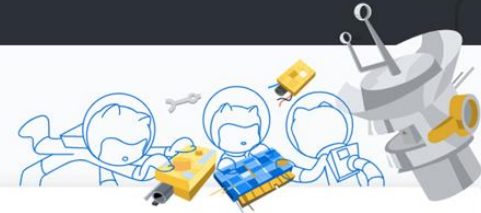
Email

Password

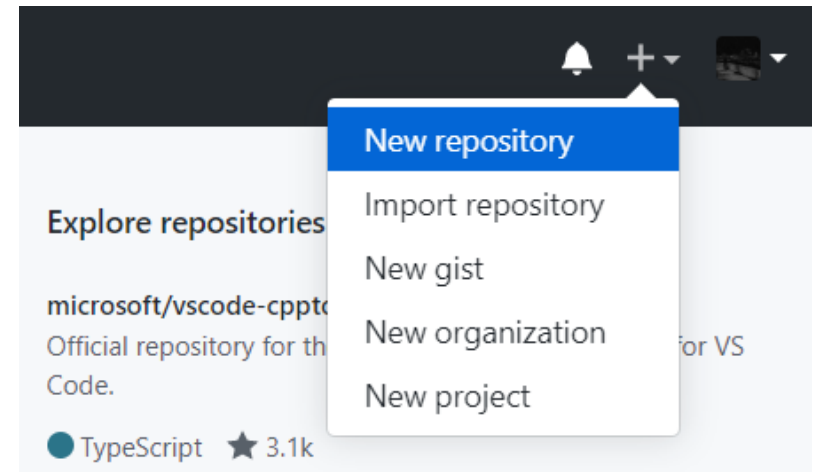
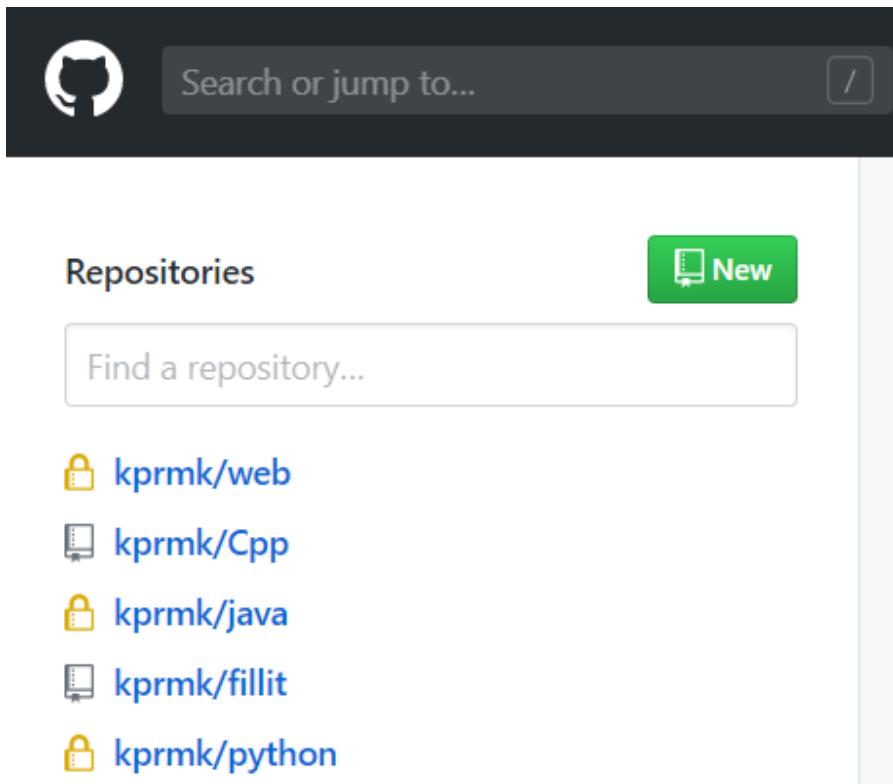
Make sure it's at least 15 characters OR at least 8 characters including a number and a lowercase letter. [Learn more.](#)

**Sign up for GitHub**

By clicking "Sign up for GitHub", you agree to our [Terms of Service](#) and [Privacy Statement](#). We'll occasionally send you account related emails.



## Тема: QT. Работа с GIT



## Тема: QT. Работа с GIT



# GitHub

### Create a new repository

A repository contains all project files, including the revision history. Already have a project repository elsewhere? [Import a repository.](#)

Owner

Repository name \*



kprmk ▾

/

inginirium\_code



Great repository names are short and memorable. Need inspiration? How about **ubiquitous-meme**?

Description (optional)

Project for practicing and HW



**Public**

Anyone can see this repository. You choose who can commit.



**Private**

You choose who can see and commit to this repository.

Skip this step if you're importing an existing repository.



**Initialize this repository with a README**

This will let you immediately clone the repository to your computer.

Add .gitignore: **None** ▾


Add a license: **None** ▾















Create repository




## Тема: QT. Работа с GIT






 [kprmk](#) / [inginirium\\_code](#)

 Unwatch ▾ 1  Star 0  Fork 0


 Code  Issues 0  Pull requests 0  Actions  Projects 0  Wiki  Security  Insights  Settings


Project for practicing and HW 



[Manage topics](#)

 1 commit  1 branch  0 packages  0 releases  1 contributor

Branch: master ▾ [New pull request](#) [Create new file](#) [Upload files](#) [Find file](#) [Clone or download ▾](#)

 kprmk Initial commit Latest commit 207a93c now

 [README.md](#) Initial commit now

 README.md 

# inginirium\_code

---

Project for practicing and HW

## Тема: QT. Работа с GIT



Drag files here to add them to your repository

Or [choose your files](#)



### Commit changes

Add files via upload

Add an optional extended description...

- ☒ Commit directly to the `master` branch.
- ☐ Create a new branch for this commit and start a pull request. [Learn more about pull requests.](#)

Commit changes

Cancel

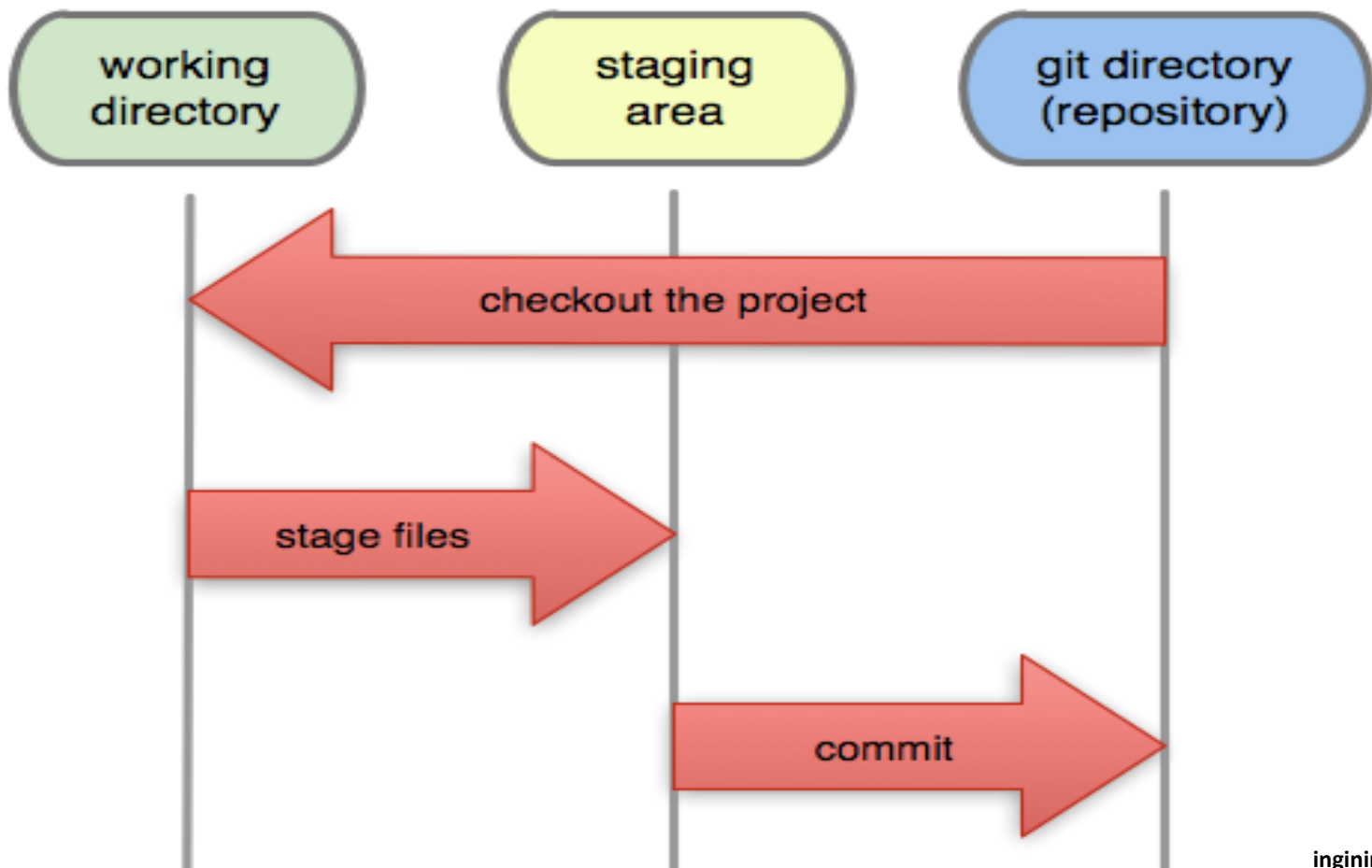






## Тема: QT. Работа с GIT

### Local Operations



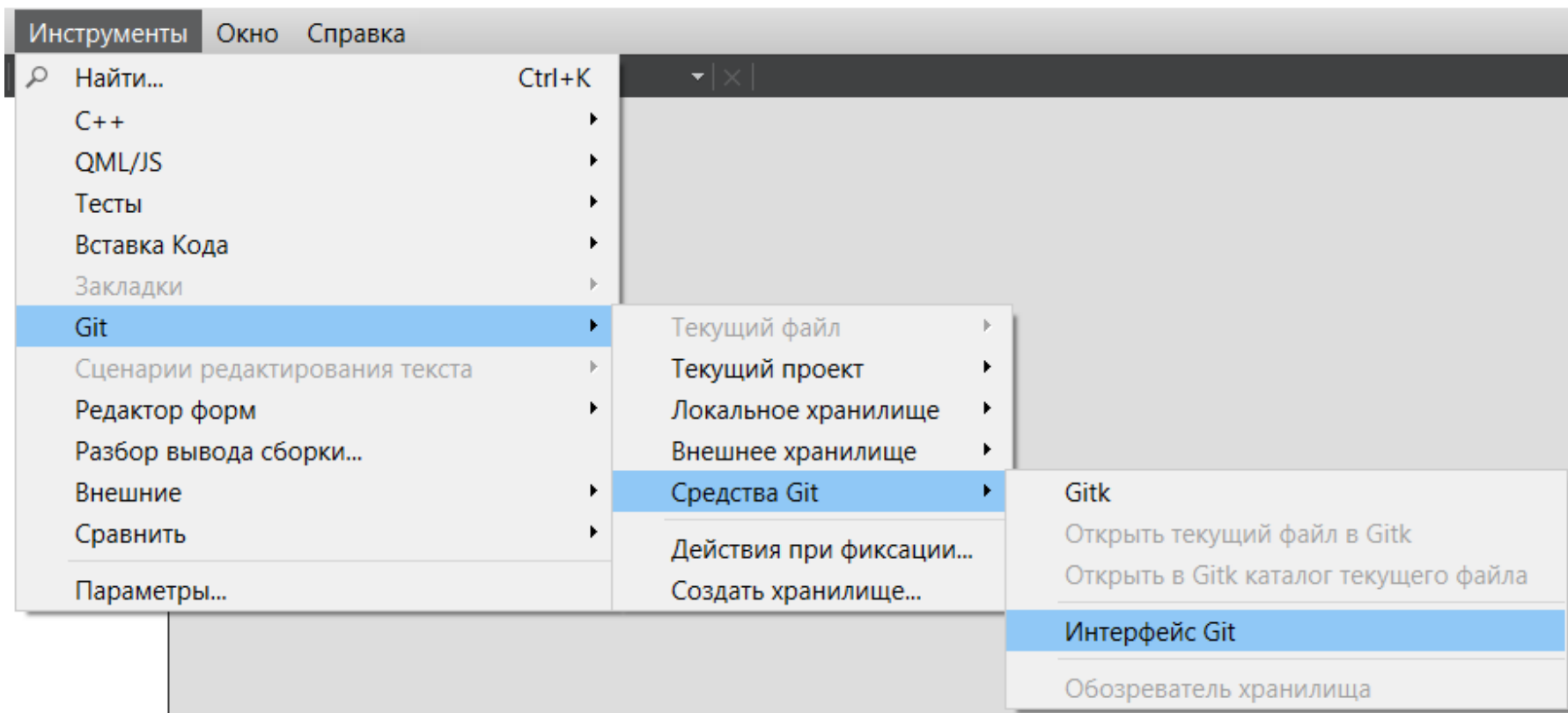


## Тема: QT. Работа с GIT

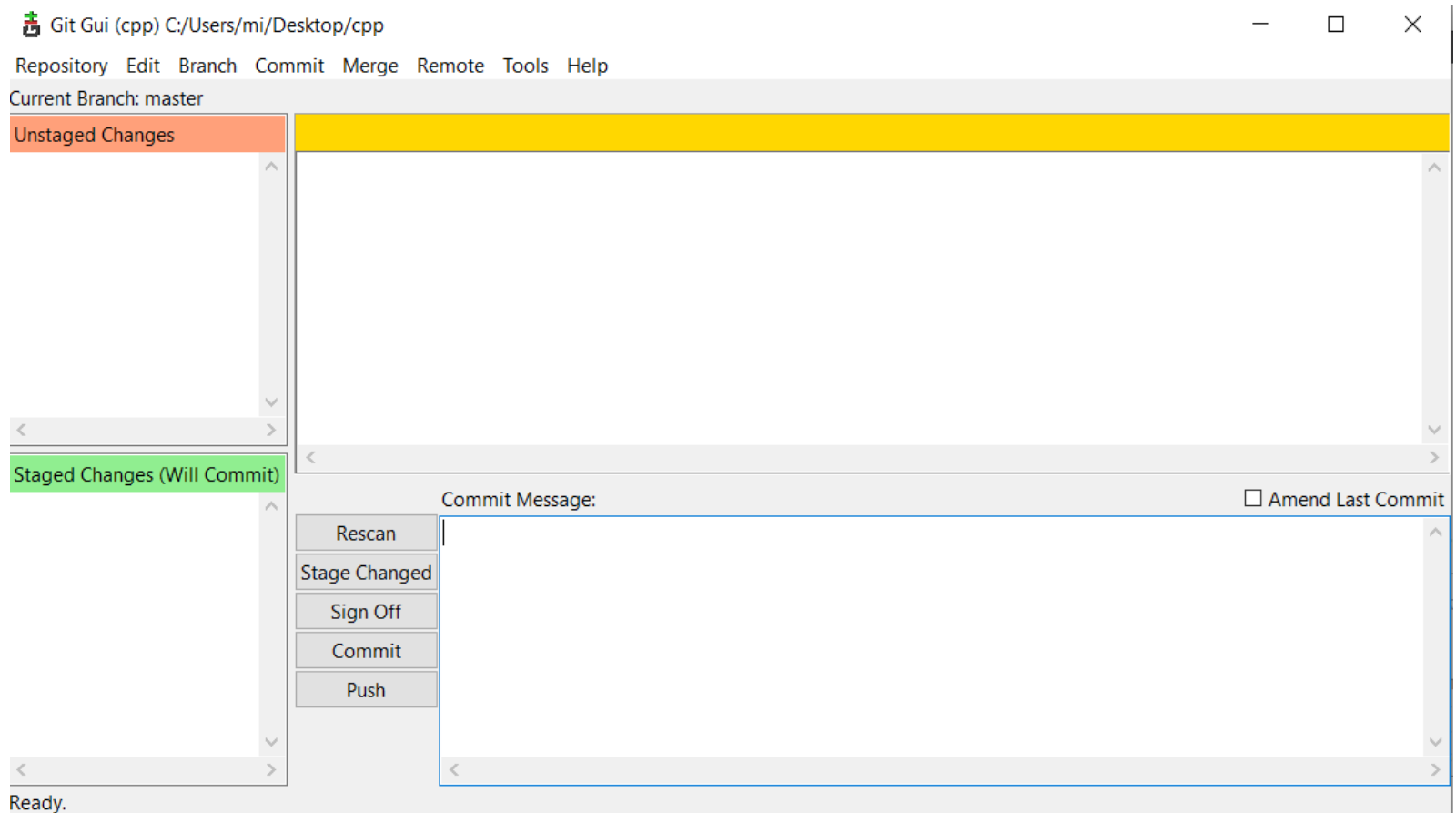
9 qt_/spiral/spiral_matrix.h	
@@ -10,16 +10,15 @@ using std::endl;	
10 class spiral_matrix	10 class spiral_matrix
11 {	11 {
12 private:	12 private:
13 - int **pointer;	13 + int **matrix;
14 - int n;	14 + int size;
15 - public:	
16	15
17 - spiral_matrix(int n = 5);	16 + public:
	17 + spiral_matrix();
18 ~spiral_matrix();	18 ~spiral_matrix();
19	19
20 void matrix_print();	20 void matrix_print();
21 void matrix_circle_around();	21 void matrix_circle_around();
22 -	
23 };	22 };
24	23
25 #endif // SPIRAL_MATRIX_H	24 #endif // SPIRAL_MATRIX_H



## Тема: QT. Работа с GIT



## Тема: QT. Работа с GIT





## Тема: QT. Работа с GIT

### Задание

Содать учётную запись на GitHub. Создать репозиторий и внести туда все файлы проектов. Сделать изменения в проектах и запустить новые версии.

