

Введение

Работа в Git начинается с установки и запуска самой программы, что представлено на рисунке 1.



Рисунок 1 – Git.cmd

Далее выполняется базовая настройка в виде назначения имени и электронной почты, продемонстрированная на рисунке 2.

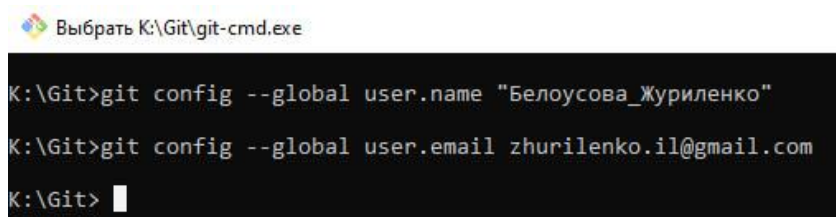


Рисунок 2 – Настройка Git

Для того, чтобы посмотреть все настройки системы, используется команда, представленная на рисунке 3.

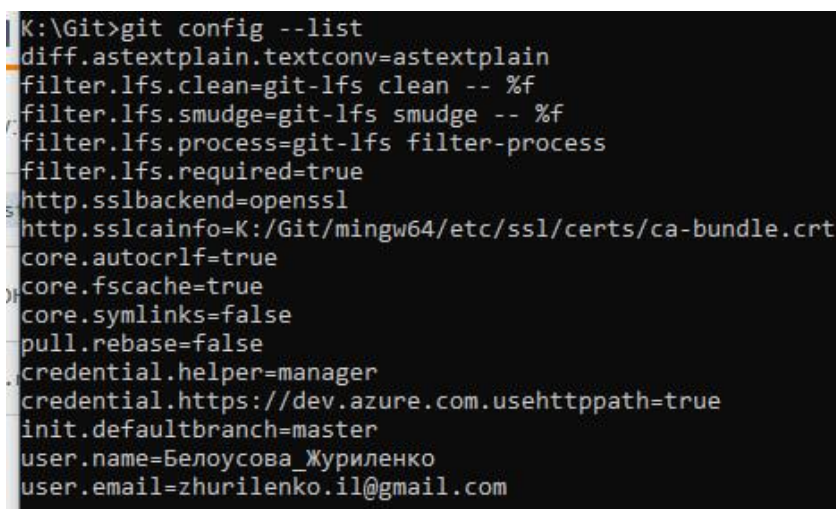


Рисунок 3 – Настройки системы

Для удобства и лёгкости зрительного восприятия, некоторые группы команд в Гит могут выделяться цветом, для этого нужно прописать в консоли команды, изображённые на рисунке 4, однако по неизвестной причине они не повлияли на дальнейший процесс.

```
K:\Git>git config --global color.ui true
K:\Git>git config --global color.status auto
K:\Git>git config --global color.branch auto
```

Рисунок 4 – Дополнительные команды

Также можно воспользоваться командой помощи, представленной на рисунке 5.

```
K:\Git>git --help
usage: git [-v | --version] [-h | --help] [-C <path>] [-c <name>=<value>]
      [--exec-path[=<path>]] [--html-path] [--man-path] [--info-path]
      [-p | --paginate | -P | --no-pager] [--no-replace-objects] [--no-lazy-fetch]
      [--no-optional-locks] [--no-advice] [--bare] [--git-dir=<path>]
      [--work-tree=<path>] [--namespace=<name>] [--config-env=<name>=<envvar>]
      <command> [<args>]

These are common Git commands used in various situations:


start a working area (see also: git help tutorial)
  clone      Clone a repository into a new directory
  init       Create an empty Git repository or reinitialize an existing one


work on the current change (see also: git help everyday)
  add        Add file contents to the index
  mv         Move or rename a file, a directory, or a symlink
  restore    Restore working tree files
  rm         Remove files from the working tree and from the index


examine the history and state (see also: git help revisions)
  bisect     Use binary search to find the commit that introduced a bug
  diff       Show changes between commits, commit and working tree, etc
  grep       Print lines matching a pattern
  log        Show commit logs
  show       Show various types of objects
  status     Show the working tree status
```

Рисунок 5 – Команда помощи

Создания репозитория с помощью Git лучше всего осуществлять на диске где находится система, поэтому была произведена переустановка программы. Создание репозитория, переход в него и создание в нём папки проекта продемонстрировано на рисунке 6.

```
C:\Users\Илья>mkdir C:\Users\Илья\Desktop\git_exercise
C:\Users\Илья>cd C:\Users\Илья\Desktop\git_exercise
C:\Users\Илья\Desktop\git_exercise>git init
Initialized empty Git repository in C:/Users/Илья/Desktop/git_exercise/.git/
C:\Users\Илья\Desktop\git_exercise>
```

Рисунок 6 – Создание папки проекта

После создаётся текстовый файл под названием hello.txt в рабочей папке.

Далее производится проверка статуса репозитория, что представлено на рисунке 7.

```
C:\Users\Илья\Desktop\git_exercise>git status
On branch master

No commits yet

Untracked files:
  (use "git add <file>..." to include in what will be committed)
    hello.txt

nothing added to commit but untracked files present (use "git add" to track)
```

Рисунок 7 – Проверка статуса

Затем данный файл добавлять в область подготовленных файлов и вновь проверяется статус, действия продемонстрированы на рисунке 8.

```
C:\Users\Илья\Desktop\git_exercise>git add hello.txt
C:\Users\Илья\Desktop\git_exercise>git status
On branch master

No commits yet

Changes to be committed:
  (use "git rm --cached <file>..." to unstage)
    new file:   hello.txt
```

Рисунок 8 – Добавление в область подготовленных файлов

В репозитории создаются файлы, которые также добавляются к подготовленным, что есть на рисунке 9.

```
C:\Users\Илья\Desktop\git_exercise>git add index.html
C:\Users\Илья\Desktop\git_exercise>git add style.css
```

Рисунок 9 – Добавление в новых файлов

После в репозитории создаётся коммит, которые максимально ясно объясняет содержание.

```
C:\Users\Илья\Desktop\git_exercise>git commit -m "Максимально ясно"
[master (root-commit) 90c6c05] Максимально ясно
 3 files changed, 11 insertions(+)
 create mode 100644 hello.txt
 create mode 100644 index.html
 create mode 100644 style.css
```

Рисунок 10 – Создание коммит

Существует команда для просмотра коммитов, представленная на рисунке 11.

```
C:\Users\Илья\Desktop\git_exercise>git log
commit 90c6c056f4ef74ef81386b23e261a3612582a31f (HEAD -> master)
Author: Белоусова Журиленко <zhurilenko.il@gmail.com>
Date: Mon Feb 3 18:12:47 2025 +0300

    Максимально ясно
```

Рисунок 11 – Просмотр коммитов

Чтобы перезаписать комментарий используется команда на рисунке 12.

```
C:\Users\Илья\Desktop\git_exercise>git commit --amend -m "Важный и ясный комментарий"
[master c3f944a] Важный и ясный комментарий
Date: Mon Feb 3 18:12:47 2025 +0300
3 files changed, 11 insertions(+)
create mode 100644 hello.txt
create mode 100644 index.html
create mode 100644 style.css
```

Рисунок 12 – Изменение коммита

Для создания удалённо репозитория необходимо пройти авторизацию на сайте GitHub, и с помощью графического интерфейса создаётся этот самый репозиторий. После производится подключение его к локальному репозиторию, что представлено на рисунке 13.

```
C:\Users\Илья\Desktop\git_exercise>git remote add origin https://github.com/MgCarbonate/MDK.git
C:\Users\Илья\Desktop\git_exercise>
```

Рисунок 13 – Подключение к удалённому репозиторию

Затем проводится отправка изменений с помощью команды, продемонстрированной на рисунке 14.

```
C:\Users\Илья\Desktop\git_exercise>git push -u origin main
info: please complete authentication in your browser...
Enumerating objects: 4, done.
Counting objects: 100% (4/4), done.
Delta compression using up to 12 threads
Compressing objects: 100% (3/3), done.
Writing objects: 100% (4/4), 483 bytes | 483.00 KiB/s, done.
Total 4 (delta 0), reused 0 (delta 0), pack-reused 0 (from 0)
To https://github.com/MgCarbonate/MDK.git
 * [new branch]      main -> main
branch 'main' set up to track 'origin/main'.
```

Рисунок 14 – Отправка изменений

В репозитории также может создаваться новая ветка, что представлено на рисунке 15.

```
C:\Users\Илья\Desktop\git_exercise>git branch novi_vetka
C:\Users\Илья\Desktop\git_exercise>
```

Рисунок 15 – Создание ветви

С помощью команды на рисунке 16 проверяется, какие ветви существуют.

```
C:\Users\Илья\Desktop\git_exercise>git branch
* main
  novi_vetka
```

Рисунок 16 – Проверка имеющихся ветвей

Переключение между ветка производится с помощью команды, представленной на рисунке 17.

```
C:\Users\Илья\Desktop\git_exercise>git checkout novi_vetka
Switched to branch 'novi_vetka'
```

Рисунок 17 – Переключение ветвей

Затем создаётся новый файл в репозитории, добавляется к подготовленным, в ветке создаётся новый коммит, после чего происходит возвращение в основную ветвь. Действия представлены на рисунке 18.

```
C:\Users\Илья\Desktop\git_exercise>git add budu.txt

C:\Users\Илья\Desktop\git_exercise>git commit -m "Новое будущее компете."
[novi_vetka 25f95c2] Новое будущее компете.
1 file changed, 0 insertions(+), 0 deletions(-)
create mode 100644 budu.txt

C:\Users\Илья\Desktop\git_exercise>git checkout master
error: pathspec 'master' did not match any file(s) known to git

C:\Users\Илья\Desktop\git_exercise>git branch
  main
* novi_vetka

C:\Users\Илья\Desktop\git_exercise>git checkout main
Switched to branch 'main'
Your branch is up to date with 'origin/main'.
```

Рисунок 18 – Махинации с ветвями

С помощью команды происходит слияние имеющихся веток и удаление ненужной. Описанные действия продемонстрированы на рисунке 19.

```
C:\Users\Илья\Desktop\git_exercise>git merge novi_vetka
Updating c3f944a..25f95c2
Fast-forward
 budu.txt | 0
 1 file changed, 0 insertions(+), 0 deletions(-)
 create mode 100644 budu.txt

C:\Users\Илья\Desktop\git_exercise>git branch -d novi_vetka
Deleted branch novi_vetka (was 25f95c2).
```

Рисунок 19 – Слияние ветвей

Для отслеживание изменений используется команда на рисунке 20.

```
C:\Users\Илья\Desktop\git_exercise>git log
commit 25f95c22056540db97b6dbc1f06b4a9052794812 (HEAD -> main)
Author: Белоусова Журиленко <zhurilenko.il@gmail.com>
Date: Mon Feb 3 18:36:17 2025 +0300

    Новое будущее компете.

commit c3f944a6d71d5bad142446c6ba4ebe94f02b31ea (origin/main)
Author: Белоусова Журиленко <zhurilenko.il@gmail.com>
Date: Mon Feb 3 18:12:47 2025 +0300

    Важный и ясный комментарий
```

Рисунок 20 – Проверка изменений

