Федеральное государственное бюджетное образовательное учреждение высшего образования «Национальный исследовательский университет «МЭИ»

Институт информационных и вычислительных технологий

Кафедра Управления и интеллектуальных технологий

**Лабораторная работа №2**

# «Системы контроля версий Git»

Задание выполнила: Жданова Е.Д.

Студент группы: А-02-19

Проверили: Мохов А.С.

Козлюк Д.А.

Москва

2020

**Вход в терминал и создание структуры каталогов**

1. Создаем на рабочем столе каталог lab02 для данной ЛР и запускаем в нем Git Bash.
2. Просматриваем файлы в рабочем каталоге можно командой ls. В каталоге lab 02 пусто, поэтому ls ничего не выведет.

u115-11@PROG-02 MINGW32 /l/I êóðñ/À-2-19/Æäàíîâà/lab02

$ ls

1. Создаем папки alice и bob командой mkdir.

kаtya@LAPTOP-5QAD3F7T MINGW32 ~/Desktop/lab02

$ mkdir alice

katya@LAPTOP-5QAD3F7T MINGW32 ~/Desktop/lab02

$ mkdir bob

Перейдем «на компьютер Алисы»  командой cd. Создаем здесь каталог project (команда mkdir)  и перейдем в него командой cd.

katya@LAPTOP-5QAD3F7T MINGW32 ~/Desktop/lab02

$ cd alice

katya@LAPTOP-5QAD3F7T MINGW32 ~/Desktop/lab02/alice

$ mkdir project

katya@LAPTOP-5QAD3F7T MINGW32 ~/Desktop/lab02/alice

$ cd project

1. Перейдем на уровень выше командой cd .. (окажемся в папке alice). Затем снова в каталог project.

katya@LAPTOP-5QAD3F7T MINGW32 ~/Desktop/lab02/alice/project

$ cd ..

katya@LAPTOP-5QAD3F7T MINGW32 ~/Desktop/lab02/alice

$ cd project

## Инициализация репозитария и настройка Git

1. Инициализируем репозитарий в текущем каталоге ( project) командой git init.

katya@LAPTOP-5QAD3F7T MINGW32 ~/Desktop/lab02/alice/project

$ git init

Initialized empty Git repository in C:/Users/katya/Desktop/lab02/alice/project/.git/

1. Настроим репозитарий Алисы, чтобы коммиты были от ее имени:

katya@LAPTOP-5QAD3F7T MINGW32 ~/Desktop/lab02/alice/project (master)

$ git config user.name 'Alice(ZhdanovaED)'

katya@LAPTOP-5QAD3F7T MINGW32 ~/Desktop/lab02/alice/project (master)

$ git config user.email 'alice@example.com'

## Создание коммитов

1. Запустим CodeBlocks и создадим проект в репозитарии Алисы.
2. Вернувшись в Git Bash, просматриваем состояние рабочей копии командой git status.

katya@LAPTOP-5QAD3F7T MINGW32 ~/Desktop/lab02/alice/project (master)

$ git status

On branch master мы находимся на ветке master

No commits yet в истории еще нет никаких коммитов

Untracked files: есть не отслеживаемые файлы

(use "git add <file>..." to include in what will be committed) следует использовать команду "git add <file>..." , чтобы включить то, что будет закоммичено

bin/

main.cpp

obj/

project.cbp файлы неотслеживаемые

nothing added to commit but untracked files present (use "git add" to track) Ничего не добавлено [в индекс], но присутствуют неотслеживаемые файлы (используйте "git add", чтобы начать их отслеживать)

1. Добавим файл main.cpp в отслеживаемые (в индекс) командой git add. Проверим  состояние рабочей копии (git status).

katya@LAPTOP-5QAD3F7T MINGW32 ~/Desktop/lab02/alice/project (master)

$ git add main.cpp

katya@LAPTOP-5QAD3F7T MINGW32 ~/Desktop/lab02/alice/project (master)

$ git status

On branch master

No commits yet

Changes to be committed:

(use "git rm --cached <file>..." to unstage)

new file: main.cpp файл стал отслеживаемым

Untracked files:

(use "git add <file>..." to include in what will be committed)

bin/

obj/

project.cbp файл остался не отслеживаемым

1. Выполним коммит с файлом main.cpp командой git commit –m и коротким сообщением.

katya@LAPTOP-5QAD3F7T MINGW32 ~/Desktop/lab02/alice/project (master)

$ git commit -m 'code:çàãîòîâêà ïðîãðàììû'

[master (root-commit) c954e70] code:çàãîòîâêà ïðîãðàììû

1 file changed, 9 insertions(+)

create mode 100644 main.cpp

1. Добавим файл project.cbp в индекс (git add).

katya@LAPTOP-5QAD3F7T MINGW32 ~/Desktop/lab02/alice/project (master)

$ git add project.cbp

warning: LF will be replaced by CRLF in project.cbp.

The file will have its original line endings in your working directory

1. Посмотрим состояние рабочей копии в этот раз.

katya@LAPTOP-5QAD3F7T MINGW32 ~/Desktop/lab02/alice/project (master)

$ git status

On branch master

Changes to be committed:

(use "git reset HEAD <file>..." to unstage)

new file: project.cbp файл project.cbp стал отслеживаемым

1. Создаем коммит с файлом project.cbp, который будет относиться к сборке (тема *– build*)

katya@LAPTOP-5QAD3F7T MINGW32 ~/Desktop/lab02/alice/project (master)

$ git commit -m 'build: add project file'

[master afcac68] build: add project file

1 file changed, 40 insertions(+)

create mode 100644 project.cbp

1. Заменим тело функции main() на ввод двух чисел:
2. Просмотрим состояние репозитария (git status). Видим различия между случаем, когда добавлялся новый файл, и когда изменился существующий (появились две новые строки).

katya@LAPTOP-5QAD3F7T MINGW32 ~/Desktop/lab02/alice/project (master)

$ git status

On branch master

Changes not staged for commit:

(use "git add <file>..." to update what will be committed)

(use "git checkout -- <file>..." to discard changes in working directory) чтобы отказаться от изменений в рабочей директории следует использовать команду "git checkout -- <file>..."

modified: main.cpp произошли изменения в файле main.cpp

Untracked files:

(use "git add <file>..." to include in what will be committed)

bin/

obj/

no changes added to commit (use "git add" and/or "git commit -a")-Изменился список не отслеживаемых файлов: появились bin/ и obj/.

1. Сделаем коммит с изменениями в main.cpp (git add main.cpp, git commit -m "..."). Сначала выбираем файлы, изменения которых должны войти в коммит, затем сделаем коммит.

katya@LAPTOP-5QAD3F7T MINGW32 ~/Desktop/lab02/alice/project (master)

$ git add main.cpp

gi

katya@LAPTOP-5QAD3F7T MINGW32 ~/Desktop/lab02/alice/project (master)

$ git commit -m 'code: ввод двух чисел'

[master d01521d] code: ввод двух чисел

1 file changed, 3 insertions(+), 1 deletion(-)

1. Добавим в тело функции main() вывод суммы чисел и выполним коммит . Добавим в индекс все изменения, затем сделаем коммит.

katya@LAPTOP-5QAD3F7T MINGW32 ~/Desktop/lab02/alice/project (master)

$ git add -u

katya@LAPTOP-5QAD3F7T MINGW32 ~/Desktop/lab02/alice/project (master)

$ git commit -m 'code: вывод двух чисел

[master 94d792a] code вывод двух чисел

1 file changed, 3 insertions(+)

1. Добавим в тело функции *main()* вывод разности чисел и выполним коммит (git commit -a -m "..."). Добавим все изменения в индекс и сделаем коммит в один шаг.

katya@LAPTOP-5QAD3F7T MINGW32 ~/Desktop/lab02/alice/project (master)

$ git commit -a -m 'code: вывод разности'

[master dbf3076] code: вывод разности

1 file changed, 2 insertions(+), 1 deletion(-)

## Игнорирование файлов

1. Укажем Git игнорировать присутствие каталога bin. Для этого создадим в CodeBlocks новый файл и запишем в него строку: /bin. Косая черта в начале означает путь от корня репозитария (каталога project), без нее игнорировался бы файл или каталог bin в любой подпапке. Сохраним файл в корне репозитария под именем .gitignore. Выполним git status, увидим, что каталог bin не отображается.

katya@LAPTOP-5QAD3F7T MINGW32 ~/Desktop/lab02/alice/project (master)

$ git status

On branch master

Untracked files:

(use "git add <file>..." to include in what will be committed)

.gitignore

obj/

project.depend

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

1. Занесем каталог obj в список игнорируемых и выполним git status.

katya@LAPTOP-5QAD3F7T MINGW32 ~/Desktop/lab02/alice/project (master)

$ git status

On branch master

Untracked files:

(use "git add <file>..." to include in what will be committed)

.gitignore

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

1. Создадим коммит с *.*gitignore (тема — *git*).

katya@LAPTOP-5QAD3F7T MINGW32 ~/Desktop/lab02/alice/project (master)

$ git add .gitignore

katya@LAPTOP-5QAD3F7T MINGW32 ~/Desktop/lab02/alice/project (master)

$ git commit -m 'git: игнорирование файлов'

[master 9c6dd41] git: игнорирование файлов

1 file changed, 2 insertions(+)

create mode 100644 .gitignore

## Просмотр истории

1. Используем команду git log --stat (показываются файлы, измененные в коммитах).

u115-11@PROG-02 MINGW32 ~/Desktop/lab02/alice/project (master)

$ git log --stat

commit 0e9ae9fa5375c1706bf89b325b11b6f149bcc7e8 (HEAD -> master) -хэш коммита, указатель на ветвь

Author: Alice (ZhdanovaED) <alice@example.com>- -автор

Date: Fri Mar 13 10:46:36 2020 +0300. - дата

git: ôàéëû obj bin -комментарий

.gitignore | 2 ++ -имя файла, два изменения

1 file changed, 2 insertions(+) -1 файл изменен, 2 вставки (+)

commit 49e1699f8b128680c9b7a9d7631a743f413736f0

Author: Alice (ZhdanovaED) <alice@example.com>

Date: Fri Mar 13 10:35:04 2020 +0300

code: вывод разности

main.cpp | 3 ++-

1 file changed, 2 insertions(+), 1 deletion(-)

commit f1e7805df5513658bc3f4403a2a87d381b0e26ce

Author: Alice (ZhdanovaED) <alice@example.com>

Date: Fri Mar 13 10:33:35 2020 +0300

code:вывод суммы

main.cpp | 3 +++

1 file changed, 3 insertions(+)

commit 2a0668087c5daeadef19006a7d3fd6b5702c0585

Author: Alice (ZhdanovaED) <alice@example.com>

Date: Fri Mar 13 10:15:24 2020 +0300

code: ввод двух чисел

main.cpp | 4 +++-

1 file changed, 3 insertions(+), 1 deletion(-)

commit ce78b2459c8b71c802e6863b215e3370194c5624

Author: Alice (ZhdanovaED) <alice@example.com>

Date: Fri Mar 13 10:08:53 2020 +0300

1. Используем команду git log --oneline --decorate (показываются коммиты компактно).

u115-11@PROG-02 MINGW32 ~/Desktop/lab02/alice/project (master)

$ git log --oneline --decorate

0e9ae9f (HEAD -> master) git: ôàéëû obj bin

49e1699 code:вывод разности чисел

f1e7805 code:вывод суммы чисел

2a06680 code: ввод двух чисел

ce78b24 build

7811239 code:заготовка программы

1. Используем команду git log --oneline --decorate --all --graph (делает то же для всех веток).

$ git log --oneline --decorate --all --graph

\* 0e9ae9f (HEAD -> master) git: ôàéëû obj bin

\* 49e1699 code:вывод разности чисел

\* f1e7805 code:вывод суммы чисел

\* 2a06680 code: ввод двух чисел

\* ce78b24 build

\* 7811239 code:заготовка программы

1. Найдем коммиты по теме *build* с помощью команды git log --grep " build:".

u115-05@PROG-07 MINGW32 ~/Desktop/laba02/alice/project (master)

$ git log --grep "build:"

сommit …

Author: Alice (ZhdanovaED) <alice@example.com>

Date: Fri Mar 13 10:32:16 2020 +0300

build: заготовка программы

1. Найдем коммиты затрагивающие project.cbp с помощью команды git log -- project.cbp.

u115-11@PROG-02 MINGW32 ~/Desktop/lab02/alice/project (master)

$ git log -- project.cbp

сommit …

Author: Alice (ZhdanovaED) <alice@example.com>

Date: Fri Mar 13 10:32:16 2020 +0300

build: заготовка программы

1. Просмотрим предпоследний коммит командой git show HEAD~1.

u115-11@PROG-02 MINGW32 ~/Desktop/lab02/alice/project (master)

$ git show 49e1699

commit 49e1699f8b128680c9b7a9d7631a743f413736f0

Author: Alice (ZhdanovaED) <alice@example.com>

Date: Fri Mar 13 10:35:04 2020 +0300

code:ðàçíîñòü ÷èñåë

diff --git a/main.cpp b/main.cpp

index b459f0d..af78703 100644

--- a/main.cpp

+++ b/main.cpp

@@ -8,7 +8,8 @@ int main()

int a,b;

cin>>a>>b;

- cout<<"A+B="<<a+b<<'\n';

+ cout<<"A+B="<<a+b<<'\n'

+ <<"A-B="<<a-b<<'\n';

return 0;

}

1. Внесем изменения в main.cpp: добавим печать произведения чисел, но не станем пока делать коммит. Просмотрим изменения в рабочей копии командой git diff, которая позволяет нам увидеть разницу между исходным файлом и измененным. «+» - данный значок показывает, что была добавлена строка cout << "A \* B = " << a \* b << '\n';

u115-11@PROG-02 MINGW32 ~/Desktop/lab02/alice/project (master)

$ git diff

diff --git a/main.cpp b/main.cpp

index af78703..a3f402f 100644

--- a/main.cpp

+++ b/main.cpp

@@ -9,7 +9,8 @@ int main()

cin>>a>>b;

cout<<"A+B="<<a+b<<'\n'

- <<"A-B="<<a-b<<'\n';

+ <<"A-B="<<a-b<<'\n'

+ <<"A\*B="<<a\*b<<'\n';

return 0;

}

1. Команда git diff HEAD~2 HEAD показывает разницу между указанными коммитами.

u115-11@PROG-02 MINGW32 ~/Desktop/lab02/alice/project (master)

$ git diff HEAD~2 HEAD

diff --git a/.gitignore b/.gitignore

new file mode 100644

index 0000000..4c7473d

--- /dev/null

+++ b/.gitignore

@@ -0,0 +1,2 @@

+/bin

+/obj

diff --git a/main.cpp b/main.cpp

index b459f0d..af78703 100644

--- a/main.cpp

+++ b/main.cpp

@@ -8,7 +8,8 @@ int main()

int a,b;

cin>>a>>b;

- cout<<"A+B="<<a+b<<'\n';

+ cout<<"A+B="<<a+b<<'\n'

+ <<"A-B="<<a-b<<'\n';

return 0;

}

1. Просмотрим изменения между самым первым коммитом и коммитом, добавляющим вывод разности. Сделаем это с помощью указания хэша каждого коммита ($ git diff ce78b f1e78 ). Где ce78b – первый коммит «заготовка программы», f1e78 – коммит «вывод разности».

u115-11@PROG-02 MINGW32 ~/Desktop/lab02/alice/project (master)

$ git diff ce78b f1e78

diff --git a/main.cpp b/main.cpp

index b4392ec..d207941 100644

--- a/main.cpp

+++ b/main.cpp

@@ -4,6 +4,10 @@ using namespace std;

int main()

{

- cout << "Hello world!" << endl;

+ cout << "Enter A and B: ";

+ int a, b;

+ cin >> a >> b;

+ cout << "A + B = " << a + b << '\n'

+ << "A - B = " << a - b << '\n';

return 0;

}

diff --git a/project.cbp b/project.cbp

new file mode 100644

index 0000000..e066312

--- /dev/null

+++ b/project.cbp

@@ -0,0 +1,43 @@

+<?xml version="1.0" encoding="UTF-8" standalone="yes" ?>

+<CodeBlocks\_project\_file>

+ <FileVersion major="1" minor="6" />

+ <Project>

+ <Option title="project" />

+ <Option pch\_mode="2" />

+ <Option compiler="gcc" />

+ <Build>

+ <Target title="Debug">

+ <Option output="bin/Debug/project" prefix\_auto="1" extension\_auto="1" />

+ <Option object\_output="obj/Debug/" />

+ <Option type="1" />

+ <Option compiler="gcc" />

+ <Compiler>

## Откат изменений

1. Закоммитим изменения в рабочей копии (вывод произведения).

katya@LAPTOP-5QAD3F7T MINGW32 ~/Desktop/lab02/alice/project (master)

$ git add -u

katya@LAPTOP-5QAD3F7T MINGW32 ~/Desktop/lab02/alice/project (master)

$ git commit -m 'code:вывод произведения'

[master 8ee853f] code:вывд произведения

1 file changed, 2 insertions(+), 1 deletion(-)

1. Необходимо отменить (откатить) этот коммит, то есть вернуться к предыдущему. Для этого воспользуемся командной git reset.

katya@LAPTOP-5QAD3F7T MINGW32 ~/Desktop/lab02/alice/project (master)

$ git reset --hard HEAD~1

HEAD is now at 9c6dd41 git: èãíîðèðîâàíèå ôàéëîâ

1. Добавим над функцией main() комментарий*: // you may type whatever you want.*
2. Уберем изменения в main.cpp другим способом — откатив этот файл к состоянию в последнем коммите (HEAD).

katya@LAPTOP-5QAD3F7T MINGW32 ~/Desktop/lab02/alice/project (master)

$ git checkout HEAD -- main.cpp

## Обмен кодом через удаленное хранилище

1. Зарегистрируемся на [GitHub](https://github.com/). Создаем репозитарий под названием cs-lab02.

katya@LAPTOP-5QAD3F7T MINGW32 ~/Desktop/lab02/alice/project (master)

$ git remote add origin https://github.com/ZhdanovaED/cs-lab02.git

katya@LAPTOP-5QAD3F7T MINGW32 ~/Desktop/lab02/alice/project (master)

$ git push -u origin master

Enumerating objects: 18, done.

Counting objects: 100% (18/18), done.

Delta compression using up to 4 threads

Compressing objects: 100% (16/16), done.

Writing objects: 100% (18/18), 2.27 KiB | 232.00 KiB/s, done.

Total 18 (delta 2), reused 0 (delta 0)

remote: Resolving deltas: 100% (2/2), done.

To https://github.com/ZhdanovaED/cs-lab02.git

\* [new branch] master -> master

Branch 'master' set up to track remote branch 'master' from 'origin'.

1. Предположим, к разработке проекта присоединяется Боб. Откроем новый терминал Git Bash в каталоге *bob*. Клонируем проект командой git clone <адрес> <каталог>.

katya@LAPTOP-5QAD3F7T MINGW32 ~/Desktop/lab02/bob

$ git clone https://github.com/ZhdanovaED/cs-lab02.git project

Cloning into 'project'...

remote: Enumerating objects: 18, done.

remote: Counting objects: 100% (18/18), done.

remote: Compressing objects: 100% (14/14), done.

remote: Total 18 (delta 2), reused 18 (delta 2), pack-reused 0

Unpacking objects: 100% (18/18), done.

1. Перейдем в каталог проекта «на машине Боба» (cd project). «На машине Боба» настроим Git (git config) аналогично тому, как это делалось для Алисы в начале лабораторной работы.

katya@LAPTOP-5QAD3F7T MINGW32 ~/Desktop/lab02/bob

$ cd project

katya@LAPTOP-5QAD3F7T MINGW32 ~/Desktop/lab02/bob/project (master)

$ git config user.name 'Bob(ZhdanovaED)'

katya@LAPTOP-5QAD3F7T MINGW32 ~/Desktop/lab02/bob/project (master)

$ git config user.email 'bob@example.com'

1. «На машине Боба» добавим в программу печать произведения чисел и сделаем коммит. Просмотрим последний коммит и убедимся, что он сделан от имени Боба. Отправьте коммит на GitHub.

katya@LAPTOP-5QAD3F7T MINGW32 ~/Desktop/lab02/bob/project (master)

$ git add main.cpp

katya@LAPTOP-5QAD3F7T MINGW32 ~/Desktop/lab02/bob/project (master)

$ git commit -m 'code:вывод произведения'

[master e56210d] code:вывод произведения

1 file changed, 2 insertions(+), 1 deletion(-)

katya@LAPTOP-5QAD3F7T MINGW32 ~/Desktop/lab02/bob/project (master)

$ git push

Enumerating objects: 5, done.

Counting objects: 100% (5/5), done.

Delta compression using up to 4 threads

Compressing objects: 100% (3/3), done.

Writing objects: 100% (3/3), 401 bytes | 401.00 KiB/s, done.

Total 3 (delta 1), reused 0 (delta 0)

remote: Resolving deltas: 100% (1/1), completed with 1 local object.

To https://github.com/ZhdanovaED/cs-lab02.git

9c6dd41..e56210d master -> master

1. «На машине Алисы» выполните загрузку изменений командой git fetch.

katya@LAPTOP-5QAD3F7T MINGW32 ~/Desktop/lab02/alice/project (master)

$ git fetch

remote: Enumerating objects: 5, done.

remote: Counting objects: 100% (5/5), done.

remote: Compressing objects: 100% (2/2), done.

remote: Total 3 (delta 1), reused 3 (delta 1), pack-reused 0

Unpacking objects: 100% (3/3), done.

From https://github.com/ZhdanovaED/cs-lab02

9c6dd41..e56210d master -> origin/master

1. Посмотрим историю всех веток с помощью команды git log --oneline --decorate --all –graph.

katya@LAPTOP-5QAD3F7T MINGW32 ~/Desktop/lab02/alice/project (master)

$ git log --oneline --decorate --all --graph

\* e56210d (origin/master) code:âûâîä ïðîèçâåäåíèÿ

\* 9c6dd41 (HEAD -> master) git: èãíîðèðîâàíèå ôàéëîâ

\* dbf3076 code: âûâîä ðàçíîñòè

\* 94d792a code: âûâîä ñóììû

\* d01521d code: ââîä äâóõ ÷èñåë

\* afcac68 build: add project file

\* c954e70 code:çàãîòîâêà ïðîãðàììû

1. Для того, чтобы продвинуть ветку *master* к скачанной версии, воспользуемся командой git pull.

katya@LAPTOP-5QAD3F7T MINGW32 ~/Desktop/lab02/alice/project (master)

$ git pull --ff-only

Updating 9c6dd41..e56210d

Fast-forward

main.cpp | 3 ++-

1 file changed, 2 insertions(+), 1 deletion(-)

## Разрешение конфликтов правок при совместной работе

1. Предположим, Алиса решает добавить в программу печать максимума из чисел, а Боб — минимума.
2. «На машине Алисы» дополним программу печатью максимума, сделаем коммит и отправим его на GitHub.
3. «На машине Боба» дополните программу печатью минимума, сделайте коммит и попытайтесь отправить его на GitHub.

katya@LAPTOP-5QAD3F7T MINGW32 ~/Desktop/lab02/bob/project (master)

$ git add main.cpp

katya@LAPTOP-5QAD3F7T MINGW32 ~/Desktop/lab02/bob/project (master)

$ git commit -m 'code: min'

[master 7ac3805] code: min

1 file changed, 8 insertions(+)

katya@LAPTOP-5QAD3F7T MINGW32 ~/Desktop/lab02/bob/project (master)

$ git push

To https://github.com/ZhdanovaED/cs-lab02.git

! [rejected] master -> master (fetch first)

error: failed to push some refs to 'https://github.com/ZhdanovaED/cs-lab02.git'

hint: Updates were rejected because the remote contains work that you do

hint: not have locally. This is usually caused by another repository pushing

hint: to the same ref. You may want to first integrate the remote changes

hint: (e.g., 'git pull ...') before pushing again.

hint: See the 'Note about fast-forwards' in 'git push --help' for details.

Как можно видеть, удаленный репозитарий не принимает изменений: коммит Боба основан не на последнем существующем коммите. «От лица Боба» загрузим коммиты из удаленного хранилища и отобразим историю всех веток.

katya@LAPTOP-5QAD3F7T MINGW32 ~/Desktop/lab02/bob/project (master)

$ git fetch

remote: Enumerating objects: 7, done.

remote: Counting objects: 100% (7/7), done.

remote: Compressing objects: 100% (2/2), done.

remote: Total 4 (delta 2), reused 4 (delta 2), pack-reused 0

Unpacking objects: 100% (4/4), done.

From https://github.com/ZhdanovaED/cs-lab02

5e434ee..eddf148 master -> origin/master

katya@LAPTOP-5QAD3F7T MINGW32 ~/Desktop/lab02/bob/project (master)

$ git log --oneline --decorate --all --graph

\* 7ac3805 (HEAD -> master) code: min

| \* eddf148 (origin/master, origin/HEAD) code: max

|/

\* 5e434ee code:ïå÷àòü äåëåíèÿ

\* e56210d code:âûâîä ïðîèçâåäåíèÿ

\* 9c6dd41 git: èãíîðèðîâàíèå ôàéëîâ

\* dbf3076 code: âûâîä ðàçíîñòè

\* 94d792a code: âûâîä ñóììû

\* d01521d code: ââîä äâóõ ÷èñåë

\* afcac68 build: add project file

\* c954e70 code:çàãîòîâêà ïðîãðàììû

1. Чтобы переместить Бобу свой коммит поверх коммита Алисы, то есть поверх *origin/master*, нужно воспользоваться командой git rebase origin/master.

katya@LAPTOP-5QAD3F7T MINGW32 ~/Desktop/lab02/bob/project (master)

$ git rebase origin/master

First, rewinding head to replay your work on top of it...

Applying: code: min

Using index info to reconstruct a base tree...

M main.cpp

Falling back to patching base and 3-way merge...

Auto-merging main.cpp

CONFLICT (content): Merge conflict in main.cpp

error: Failed to merge in the changes.

hint: Use 'git am --show-current-patch' to see the failed patch

Patch failed at 0001 code: min

Resolve all conflicts manually, mark them as resolved with

"git add/rm <conflicted\_files>", then run "git rebase --continue".

You can instead skip this commit: run "git rebase --skip".

To abort and get back to the state before "git rebase", run "git rebase --abort".

Однако эта команда завершается с ошибкой, сообщающей о конфликте в main.cpp

«На машине Боба» в CodeBlocks на месте конфликта исправляем все необходимое. Затем добавляем файл в индекс и продолжаем прерванную операцию *rebase*. Это выполняется с помощью команды git add main.cpp и команды git rebase –continue. Смотрим историю, и отправляем изменения на GitHub.

1. Проверяем, как выглядит история хранилища.

Боб:

katya@LAPTOP-5QAD3F7T MINGW32 ~/Desktop/lab02/bob/project (master|REBASE 1/1)

$ git log --oneline --decorate --all --graph

\* 7ac3805 (master) code: min

| \* eddf148 (HEAD, origin/master, origin/HEAD) code: max

|/

\* 5e434ee code:ïå÷àòü äåëåíèÿ

\* e56210d code:âûâîä ïðîèçâåäåíèÿ

\* 9c6dd41 git: èãíîðèðîâàíèå ôàéëîâ

\* dbf3076 code: âûâîä ðàçíîñòè

\* 94d792a code: âûâîä ñóììû

\* d01521d code: ââîä äâóõ ÷èñåë

\* afcac68 build: add project file

\* c954e70 code:çàãîòîâêà ïðîãðàììû

Алиса:

katya@LAPTOP-5QAD3F7T MINGW32 ~/Desktop/lab02/alice/project (master)

$ git log --oneline --decorate --all --graph

\* eddf148 (HEAD -> master, origin/master) code: max

\* 5e434ee code:ïå÷àòü äåëåíèÿ

\* e56210d code:âûâîä ïðîèçâåäåíèÿ

\* 9c6dd41 git: èãíîðèðîâàíèå ôàéëîâ

\* dbf3076 code: âûâîä ðàçíîñòè

\* 94d792a code: âûâîä ñóììû

\* d01521d code: ââîä äâóõ ÷èñåë

\* afcac68 build: add project file

\* c954e70 code:çàãîòîâêà ïðîãðàììû

## Использование веток

1. «На машине Алисы» создаем ветку (git branch) и переключаемся на нее (git checkout).

katya@LAPTOP-5QAD3F7T MINGW32 ~/Desktop/lab02/alice/project (master)

$ git branch double

katya@LAPTOP-5QAD3F7T MINGW32 ~/Desktop/lab02/alice/project (master)

$ git checkout double

Switched to branch 'double'

1. Заменим тип переменных *a* и *b* на *double* и сделаем коммит.

katya@LAPTOP-5QAD3F7T MINGW32 ~/Desktop/lab02/alice/project (double)

$ git add main.cpp

katya@LAPTOP-5QAD3F7T MINGW32 ~/Desktop/lab02/alice/project (double)

$ git commit -m 'code: double'

[double b0c1c49] code: double

1 file changed, 3 insertions(+), 3 deletions(-)

1. Переключимся на ветку *master*.
2. На данном этапе синхронизируем ветку *master* «на машине Алисы» с GitHub.
3. Сливаем ветку *double* в master с помощью git merge double и смотрим историю всех веток.

katya@LAPTOP-5QAD3F7T MINGW32 ~/Desktop/lab02/alice/project (master)

$ git merge double

Auto-merging main.cpp

Merge made by the 'recursive' strategy.

main.cpp | 2 +-

1 file changed, 1 insertion(+), 1 deletion(-)

Алиса:

katya@LAPTOP-5QAD3F7T MINGW32 ~/Desktop/lab02/alice/project (master)

$ git log --oneline --decorate --all --graph

\* e3e0286 (HEAD -> master, origin/master) Merge branch 'double'

|\

| \* b0c1c49 (double) code: double

\* | be4540d code: min

|/

\* eddf148 code: max

\* 5e434ee code:печать деления

\* e56210d code:ввод произведения

\* 9c6dd41 git: игнорирование файлов

\* dbf3076 code: вывод разности

\* 94d792a code: вывод суммы

\* d01521d code: ввод двух чисел

\* afcac68 build: add project file

\* c954e70 code:заготовка программы

Боб:

katya@LAPTOP-5QAD3F7T MINGW32 ~/Desktop/lab02/bob/project (master)

$ git log --oneline --decorate --all --graph

\* be4540d (HEAD -> master, origin/master, origin/HEAD) code: min

\* eddf148 code: max

\* 5e434ee code:печать деления

\* e56210d code:вывод произведения

\* 9c6dd41 git: игнорирование файлов

\* dbf3076 code: вывод разности

\* 94d792a code: вывод суммы

\* d01521d code: ввод двух чисел

\* afcac68 build: add project file

\* c954e70 code:заготовка программы