|  |
| --- |
| Федеральное государственное бюджетное образовательное учреждение высшего образования "Национальный исследовательский университет "МЭИ" |
|  |

Институт информационных и вычислительных технологий

Кафедра Управления и интеллектуальных технологий

**Отчёт по лабораторной работе № 2**

# По дисциплине «Разработка ПО систем управления»

# Выполнил студент группы А-03-19 Симич А.Б.

***Отчет***

Создаем на рабочем столе каталог lab02 и запускаем в нем Git Bash Here.

В терминале открывается приглашение такого вида:

Alexander@DESKTOP-GMVT49I MINGW32 ~/Desktop/lab02

$

Символ $ обозначает начало ввода команд.

Для просмотра файлов в рабочем каталоге можно воспользоваться командой ls:

ls

В каталоге lab02 пусто, поэтому ls ничего не выведет.

В ходе работы будем имитировать проект с двумя участниками: Алисой и Бобом. Мы создаем папки с помощью команд mkdir alice и mkdir bob, которые будут имитировать компьютеры Алиса и Боб соответственно alice и bob :

Alexander@DESKTOP-GMVT49I MINGW32 ~/Desktop/lab02

$ mkdir alice

Alexander@DESKTOP-GMVT49I MINGW32 ~/Desktop/lab02

$ mkdir bob

Для осуществления перехода между каталогами воспользуемся командой cd. Для перехода к компьютеру Алисы воспользуемся командой cd alice:

Alexander@DESKTOP-GMVT49I MINGW32 ~/Desktop/lab02

$ cd alice

Сейчас мы находимся в каталоге alice.

Теперь создаем каталог project и переходим в него:

Alexander@DESKTOP-GMVT49I MINGW32 ~/Desktop/lab02/alice

$ mkdir project

Alexander@DESKTOP-GMVT49I MINGW32 ~/Desktop/lab02/alice

$ cd project

Далее с помощью команды cd .. переходим в папку alice, а затем перемещаемся с помощью команды cd project в папку project :

Alexander@DESKTOP-GMVT49I MINGW32 ~/Desktop/lab02/alice/project

$ cd ..

Alexander@DESKTOP-GMVT49I MINGW32 ~/Desktop/lab02/alice

$ cd project

Для инициализации репозитария в текущем каталоге используем команду git init:

Alexander@DESKTOP-GMVT49I MINGW32 ~/Desktop/lab02/alice/project

$ git init

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

К приглашению командной строки добавилось (master): имя текущий ветви Git (данная ветвь используется по умолчанию).

Настроим репозитарий Алисы, чтобы коммиты были от ее имени:

Alexander@DESKTOP-GMVT49I MINGW32 ~/Desktop/lab02/alice/project (master)

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

Alexander@DESKTOP-GMVT49I MINGW32 ~/Desktop/lab02/alice/project (master)

$ git config user.email 'simi55@mail.ru'

Проверим состояние рабочей копии:

$ git status

On branch master

No commits yet

Untracked files:

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

bin/

main.cpp

obj/

project.cbp

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

**On branch master** – мы находимся в ветви master

**No commits yet** – история не содержит никаких еще коммитов

**Untracked files** – есть не отслеживаемые файлы

**(use “ git add …” to include in what will be committed)** – использовать команду “ git add …”, чтобы включить то, что будет закоммичено.

**bin/** -не отслеживаемый файл **main.cpp** - не отслеживаемый файл

**obj/** -не отслеживаемый файл

**project.cbp** - не отслеживаемый файл

**nothing added to commit but untracked files present (use “git add” to track)** – в рабочей копии нет ничего, что можно было бы добавить

Теперь добавляем файл main.cpp в отслеживаемые:

Alexander@DESKTOP-GMVT49I MINGW32 ~/Desktop/lab02/alice/project (master)

$ git add main.cpp

Alexander@DESKTOP-GMVT49I MINGW32 ~/Desktop/lab02/alice/project (master)

$ git status

On branch master

No commits yet

Changes to be committed:

(use "git rm --cached ..." to unstage)

new file: main.cpp

Untracked files:

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

bin/

obj/

project.cbp

Произошли небольшие изменения: файл main.cpp стал отслеживаемым. Строка nothing added to commit but untracked files present (use “git add” to track) пропала.

На данном этапе делаем коммит с коротким сообщением “заготовка программы”:

Alexander@DESKTOP-GMVT49I MINGW32 ~/Desktop/lab02/alice/project (master)

$ git commit -m 'code: заготовка программы'

[master (root-commit) 83fbfb6] code: заготовка программы

1 file changed, 9 insertions(+)

create mode 100644 main.cpp

Добавляем файл project.cbp:

Alexander@DESKTOP-GMVT49I 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

Посмотрим состояние рабочей копии в этот раз:

Alexander@DESKTOP-GMVT49I MINGW32 ~/Desktop/lab02/alice/project (master)

$ git status

On branch master

Changes to be committed:

(use "git reset HEAD ..." to unstage)

new file: project.cbp

Untracked files:

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

bin/

obj/

Произошли некоторые изменения: файл project.cbp стал отслеживаемым.

Создаем коммит, который будет относиться к сборке:

Alexander@DESKTOP-GMVT49I MINGW32 ~/Desktop/lab02/alice/project (master)

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

[master 970c4ac] build: add project file

1 file changed, 42 insertions(+)

create mode 100644 project.cbp

После внесенных изменений в main.cpp мы проверили состояние репозитария.

Alexander@DESKTOP-GMVT49I MINGW32 ~/Desktop/lab02/alice/project (master)

$ git status

On branch master

Changes not staged for commit:

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

(use "git checkout -- ..." to discard changes in working directory)

modified: main.cpp

Untracked files:

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

bin/

obj/

no changes added to commit (use "git add" and/or "git commit -a")

Видим различия между случаем, когда добавлялся новый файл, и когда изменился существующий. Появились новые строки:

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

**(use "git checkout -- ..." to discard changes in working directory)** **-** используйте команду **“**git checkpoint -- …” , чтобы отказаться от изменений в рабочем детекторе.

Для того, чтобы создать коммит, необходимо выбрать файл, который в дальнейшем будет закоммичен:

Alexander@DESKTOP-GMVT49I MINGW32 ~/Desktop/lab02/alice/project (master)

$ git add main.cpp

Alexander@DESKTOP-GMVT49I MINGW32 ~/Desktop/lab02/alice/project (master)

$ git commit -m "изменение кода"

[master 89bf736] изменение кода

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

Мы выбираем снова файл, в котором должен произойти коммит «вывод суммы» и создаем его:

Alexander@DESKTOP-GMVT49I MINGW32 ~/Desktop/lab02/alice/project (master)

$ git add main.cpp

Alexander@DESKTOP-GMVT49I MINGW32 ~/Desktop/lab02/alice/project (master)

$ git commit -m "вывод суммы"

[master 2ed6587] вывод суммы

1 file changed, 1 insertion(+)

Мы выбираем файл, в котором должен произойти коммит «вывод разности» и создаем его:

Alexander@DESKTOP-GMVT49I MINGW32 ~/Desktop/lab02/alice/project (master)

$ git add main.cpp

Alexander@DESKTOP-GMVT49I MINGW32 ~/Desktop/lab02/alice/project (master)

$ git commit -m "вывод разности"

[master b13491c] вывод разности

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

Мы проверили состояние репозитария и видим, что у нас пропал файл bin/.

Alexander@DESKTOP-GMVT49I MINGW32 ~/Desktop/lab02/alice/project (master)

$ git status

On branch master

Untracked files:

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

.gitignore

obj/

project.depend

no changes added to commit (use "git add" and/or "git commit -a")

После добавления в пустом файле /obj мы заметили, что и он больше не отображается после проверки репозитария:

Alexander@DESKTOP-GMVT49I MINGW32 ~/Desktop/lab02/alice/project (master)

$ git status

On branch master

Untracked files:

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

.gitignore

project.depend

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

Закоммитил файл .gitignore и добавил его в репозитарий.

Alexander@DESKTOP-GMVT49I MINGW32 ~/Desktop/lab02/alice/project (master)

$ git add .gitignore

Alexander@DESKTOP-GMVT49I MINGW32 ~/Desktop/lab02/alice/project (master)

$ git add -u

Alexander@DESKTOP-GMVT49I MINGW32 ~/Desktop/lab02/alice/project (master)

$ git commit -m 'build: добавил .gitignore'

[master 6ce02f1] build: добавил .gitignore

1 file changed, 2 insertions(+)

create mode 100644 .gitignore

Alexander@DESKTOP-GMVT49I MINGW32 ~/Desktop/lab02/alice/project (master)

$ git push

Enumerating objects: 10, done.

Counting objects: 100% (10/10), done.

Delta compression using up to 4 threads

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

Writing objects: 100% (6/6), 697 bytes | 697.00 KiB/s, done.

Total 6 (delta 2), reused 0 (delta 0)

remote: Resolving deltas: 100% (2/2), completed with 1 local object.

To https://github.com/Alexandr-Simich/cs-lab02.git

fe8ae37..6ce02f1 master -> master

С помощью команды git log –stat смотрим все созданные нами коммиты. Эта команда показывает файлы, измененные в коммитах. Она показывает хэш каждой из ниже перечисленных команд, далее автора, потом дату создания коммита, затем сам коммит, а в конце показывает затронутые коммитом файлы:

Alexander@DESKTOP-GMVT49I MINGW32 ~/Desktop/lab02/alice/project (master)

$ git log --stat

commit 0f5e7e1ce7065b4c53c6d87442c87866f5c02ba4 (HEAD -> master)

Author: Alice (SimichAB)

Date: Mon Mar 23 00:12:48 2020 +0300

игнор git

project.cbp | 1 +

1 file changed, 1 insertion(+)

commit b13491c987be75e1e4bca7e1dcf9bfb8d4261dea

Author: Alice (SimichAB)

Date: Sun Mar 22 23:26:38 2020 +0300

вывод разности

main.cpp | 2 +-

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

commit 2ed6587ea1470b33ca7e73b2e979b81c61e5c5c7

Author: Alice (SimichAB)

Date: Sun Mar 22 23:23:17 2020 +0300

вывод суммы

main.cpp | 1 +

1 file changed, 1 insertion(+)

commit 89bf7365a6747cbedf1fec71b3bfe1713498cee3

Author: Alice (SimichAB)

Date: Sun Mar 22 23:16:25 2020 +0300

изменение кода

main.cpp | 4 +++-

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

commit 5c1a801ee1b008e398b0a183849de1fa393a3826

Author: Alice (SimichAB)

Date: Sun Mar 22 23:01:49 2020 +0300

build: add project file

project.cbp | 43 +++++++++++++++++++++++++++++++++++++++++++

1 file changed, 43 insertions(+)

commit 01da5358c14091ca037bf25d91ec07c957abdd4d

Author: Alice (SimichAB)

Date: Sun Mar 22 22:58:33 2020 +0300

code: заготовка программы

main.cpp | 9 +++++++++

1 file changed, 9 insertions(+)

ESCOC

ESCOD

Date: Sun Mar 22 23:23:17 2020 +0300

вывод суммы

main.cpp | 1 +

1 file changed, 1 insertion(+)

commit 89bf7365a6747cbedf1fec71b3bfe1713498cee3

Author: Alice (SimichAB)

Date: Sun Mar 22 23:16:25 2020 +0300

изменение кода

main.cpp | 4 +++-

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

commit 5c1a801ee1b008e398b0a183849de1fa393a3826

Author: Alice (SimichAB)

Date: Sun Mar 22 23:01:49 2020 +0300

build: add project file

project.cbp | 43 +++++++++++++++++++++++++++++++++++++++++++

1 file changed, 43 insertions(+)

commit 01da5358c14091ca037bf25d91ec07c957abdd4d

Author: Alice (SimichAB)

Date: Sun Mar 22 22:58:33 2020 +0300

code: заготовка программы

main.cpp | 9 +++++++++

1 file changed, 9 insertions(+)

Сейчас мы используем команду git log –oneline –decore для наиболее удобного отображения всех коммитов:

Alexander@DESKTOP-GMVT49I MINGW32 ~/Desktop/lab02/alice/project (master)

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

\* 0f5e7e1 (HEAD -> master) игнор git

\* b13491c вывод разности

\* 2ed6587 вывод суммы

\* 89bf736 изменение кода

\* 5c1a801 build: add project file

\* 01da535 code: заготовка программы

Теперь мы использовали команду git log --oneline –decorate –all –graph, которая позволяет вывести все строчки компактно, для наиболее удобного просмотра коммитов:

Alexander@DESKTOP-GMVT49I MINGW32 ~/Desktop/lab02/alice/project (master)

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

\* 0f5e7e1 (HEAD -> master) игнор git

\* b13491c вывод разности

\* 2ed6587 вывод суммы

\* 89bf736 изменение кода

\* 5c1a801 build: add project file

\* 01da535 code: заготовка программы

Используем команду git log -- main.cpp для просмотра изменений с этим файлом:

Alexander@DESKTOP-GMVT49I MINGW32 ~/Desktop/lab02/alice/project (master)

$ git log -- main.cpp

commit b13491c987be75e1e4bca7e1dcf9bfb8d4261dea

Author: Alice (SimichAB)

Date: Sun Mar 22 23:26:38 2020 +0300

вывод разности

commit 2ed6587ea1470b33ca7e73b2e979b81c61e5c5c7

Author: Alice (SimichAB)

Date: Sun Mar 22 23:23:17 2020 +0300

вывод суммы

commit 89bf7365a6747cbedf1fec71b3bfe1713498cee3

Author: Alice (SimichAB)

Date: Sun Mar 22 23:16:25 2020 +0300

изменение кода

commit 01da5358c14091ca037bf25d91ec07c957abdd4d

Author: Alice (SimichAB)

Date: Sun Mar 22 22:58:33 2020 +0300

code: заготовка программы

Теперь воспользуемся командой git log --grep “code:”, которая позволяет отобразить коммиты с code: в сообщении:

Alexander@DESKTOP-GMVT49I MINGW32 ~/Desktop/lab02/alice/project (master)

$ git log --grep "code:"

commit 01da5358c14091ca037bf25d91ec07c957abdd4d

Author: Alice (SimichAB)

Date: Sun Mar 22 22:58:33 2020 +0300

code: заготовка программы

Используем команду git log --grep “build:” , которая отображает коммиты с build: в сообщении:

Alexander@DESKTOP-GMVT49I MINGW32 ~/Desktop/lab02/alice/project (master)

$ git log --grep "build"

commit 5c1a801ee1b008e398b0a183849de1fa393a3826

Author: Alice (SimichAB)

Date: Sun Mar 22 23:01:49 2020 +0300

build: add project file

Используем команду git log -- project.cbp , которая показывает коммиты, затрагивающие project.cbp:

Alexander@DESKTOP-GMVT49I MINGW32 ~/Desktop/lab02/alice/project (master)

$ git log -- project.cbp

commit 0f5e7e1ce7065b4c53c6d87442c87866f5c02ba4 (HEAD -> master)

Author: Alice (SimichAB)

Date: Mon Mar 23 00:12:48 2020 +0300

игнор git

commit 5c1a801ee1b008e398b0a183849de1fa393a3826

Author: Alice (SimichAB)

Date: Sun Mar 22 23:01:49 2020 +0300

build: add project file

Для просмотра предыдущего коммита используем команду git show HEAD~1:

Alexander@DESKTOP-GMVT49I MINGW32 ~/Desktop/lab02/alice/project (master)

$ git show HEAD~1

commit b13491c987be75e1e4bca7e1dcf9bfb8d4261dea

Author: Alice (SimichAB)

Date: Sun Mar 22 23:26:38 2020 +0300

вывод разности

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

index fda380f..3ddfabd 100644

--- a/main.cpp

+++ b/main.cpp

@@ -7,6 +7,6 @@ int main()

cout << "Enter A and B: ";

int a, b;

cin >> a >> b;

-cout << a+b;

+cout << "A - B = " << a - b;

return 0;

}

Есть еще два способа просмотра предыдущего коммита:

1)

Alexander@DESKTOP-GMVT49I MINGW32 ~/Desktop/lab02/alice/project (master)

$ git show master~1

commit b13491c987be75e1e4bca7e1dcf9bfb8d4261dea

Author: Alice (SimichAB)

Date: Sun Mar 22 23:26:38 2020 +0300

вывод разности

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

index fda380f..3ddfabd 100644

--- a/main.cpp

+++ b/main.cpp

@@ -7,6 +7,6 @@ int main()

cout << "Enter A and B: ";

int a, b;

cin >> a >> b;

-cout << a+b;

+cout << "A - B = " << a - b;

return 0;

}

2)

Alexander@DESKTOP-GMVT49I MINGW32 ~/Desktop/lab02/alice/project (master)

$ git show b13491c987be75e1e4bca7e1dcf9bfb8d4261dea

commit b13491c987be75e1e4bca7e1dcf9bfb8d4261dea

Author: Alice (SimichAB)

Date: Sun Mar 22 23:26:38 2020 +0300

вывод разности

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

index fda380f..3ddfabd 100644

--- a/main.cpp

+++ b/main.cpp

@@ -7,6 +7,6 @@ int main()

cout << "Enter A and B: ";

int a, b;

cin >> a >> b;

-cout << a+b;

+cout << "A - B = " << a - b;

return 0;

}

Воспользовались 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';

Alexander@DESKTOP-GMVT49I MINGW32 ~/Desktop/lab02/alice/project (master)

$ git diff

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

index 3ddfabd..25fca0b 100644

--- a/main.cpp

+++ b/main.cpp

@@ -8,5 +8,6 @@ int main()

int a, b;

cin >> a >> b;

cout << "A - B = " << a - b;

+cout << "A \* B = " << a \* b;

return 0;

}

Команда git diff HEAD~2 показывает изменения от указанного коммита до последнего, включая изменения в рабочей копии :

Alexander@DESKTOP-GMVT49I MINGW32 ~/Desktop/lab02/alice/project (master)

$ git diff HEAD~2

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

index fda380f..25fca0b 100644

--- a/main.cpp

+++ b/main.cpp

@@ -7,6 +7,7 @@ int main()

cout << "Enter A and B: ";

int a, b;

cin >> a >> b;

-cout << a+b;

+cout << "A - B = " << a - b;

+cout << "A \* B = " << a \* b;

return 0;

}

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

index e066312..e9838bd 100644

--- a/project.cbp

+++ b/project.cbp

@@ -32,6 +32,7 @@

+

Команда git diff HEAD~2 HEAD показывает разницу между указанными коммитами:

Alexander@DESKTOP-GMVT49I MINGW32 ~/Desktop/lab02/alice/project (master)

$ git diff HEAD~2 HEAD

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

index fda380f..3ddfabd 100644

--- a/main.cpp

+++ b/main.cpp

@@ -7,6 +7,6 @@ int main()

cout << "Enter A and B: ";

int a, b;

cin >> a >> b;

-cout << a+b;

+cout << "A - B = " << a - b;

return 0;

}

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

index e066312..e9838bd 100644

--- a/project.cbp

+++ b/project.cbp

@@ -32,6 +32,7 @@

+

Мы использовали git diff HEAD~4 HEAD. Эта команда выводит изменение в файле main.cpp, а именно ввод значений и разность с комментариями:

Alexander@DESKTOP-GMVT49I MINGW32 ~/Desktop/lab02/alice/project (master)

$ git diff HEAD~4 HEAD

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

index b4392ec..3ddfabd 100644

--- a/main.cpp

+++ b/main.cpp

@@ -4,6 +4,9 @@ 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;

return 0;

}

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

index e066312..e9838bd 100644

--- a/project.cbp

+++ b/project.cbp

@@ -32,6 +32,7 @@

+

На данном этапе закоммитим вывод произведения:

Alexander@DESKTOP-GMVT49I MINGW32 ~/Desktop/lab02/alice/project (master)

$ git add main.cpp

Alexander@DESKTOP-GMVT49I MINGW32 ~/Desktop/lab02/alice/project (master)

$ git commit -m "code: вывод произведения"

[master 2a9b776] code: вывод произведения

1 file changed, 1 insertion(+)

Используем команду git reset --hard HEAD~1, которая помогает вернуться к предыдущему коммиту. Здесь HEAD~1 указывает на коммит, к которому нужно откатить состояние рабочей копии, а ключ --hard означает, что нужно привести рабочую копию точно к состоянию выбранного коммита.

Alexander@DESKTOP-GMVT49I MINGW32 ~/Desktop/lab02/alice/project (master)

$ git reset --hard HEAD~1

HEAD is now at 0f5e7e1 игнор git

Создаём репозитaрий на GitHub и загружаем туда наш проект с таким образом:

Alexander@DESKTOP-GMVT49I MINGW32 ~/Desktop/lab02/alice/project (master)

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

Alexander@DESKTOP-GMVT49I 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% (17/17), done.

Writing objects: 100% (18/18), 2.22 KiB | 568.00 KiB/s, done.

Total 18 (delta 3), reused 0 (delta 0)

remote: Resolving deltas: 100% (3/3), done.

To https://github.com/Alexandr-Simich/cs-lab02.git

\* [new branch] master -> master

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

Далее клонируем наш проект в папку bob:

Alexander@DESKTOP-GMVT49I MINGW32 ~/Desktop/lab02/bob

$ git clone https://github.com/Alexandr-Simich/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 3), reused 18 (delta 3), pack-reused 0

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

Проинициализируем репозитарий:

Alexander@DESKTOP-GMVT49I MINGW32 ~/Desktop/lab02/bob/project (master)

$ git init

Reinitialized existing Git repository in C:/Users/Alexander/Desktop/lab02/bob/project/.git/

Alexander@DESKTOP-GMVT49I MINGW32 ~/Desktop/lab02/bob/project (master)

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

Alexander@DESKTOP-GMVT49I MINGW32 ~/Desktop/lab02/bob/project (master)

$ git config user.email 'simi55@mail.ru'

Теперь добавим в программу main.cpp произведение чисел и создадим коммит:

Alexander@DESKTOP-GMVT49I MINGW32 ~/Desktop/lab02/bob/project (master)

$ git add main.cpp

Alexander@DESKTOP-GMVT49I MINGW32 ~/Desktop/lab02/bob/project (master)

$ git commit -m "code: вывод произведения"

[master 233c2c1] code: вывод произведения

1 file changed, 1 insertion(+)

Необходимо проверить, что коммит был сделан от имени Боба:

Alexander@DESKTOP-GMVT49I MINGW32 ~/Desktop/lab02/bob/project (master)

$ git show HEAD

commit 233c2c1f36bdd066a60ad0e01e79ef9548dd9705 (HEAD -> master)

Author: Bob (SimichAB)

Date: Mon Mar 23 01:58:01 2020 +0300

code: вывод произведения

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

index 3ddfabd..25fca0b 100644

--- a/main.cpp

+++ b/main.cpp

@@ -8,5 +8,6 @@ int main()

int a, b;

cin >> a >> b;

cout << "A - B = " << a - b;

+cout << "A \* B = " << a \* b;

return 0;

}

)

С помощью данной команды мы увидели, что коммит был действительно создан от имени боба и что было добавлено произведение в файле main.cpp.

Далее мы должны отправить коммит на GitHub с помощью команды git push:

Alexander@DESKTOP-GMVT49I MINGW32 ~/Desktop/lab02/bob/project (master)

$ git push -u origin master

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), 383 bytes | 383.00 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/Alexandr-Simich/cs-lab02.git

0f5e7e1..233c2c1 master -> master

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

После обновления страницы GitHub я убедился, что коммит попал в удаленный репозитарий.

«На машине Алисы» выполнил загрузку изменений с помощью команды git fetch:

Alexander@DESKTOP-GMVT49I 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/Alexandr-Simich/cs-lab02

0f5e7e1..233c2c1 master -> origin/master

Теперь посмотрим историю всех веток с помощью команды git log --oneline --decorate --all --graph:

Alexander@DESKTOP-GMVT49I MINGW32 ~/Desktop/lab02/alice/project (master)

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

\* 233c2c1 (origin/master) code: вывод произведения

\* 0f5e7e1 (HEAD -> master) игнор git

\* b13491c вывод разности

\* 2ed6587 вывод суммы

\* 89bf736 изменение кода

\* 5c1a801 build: add project file

\* 01da535 code: заготовка программы

Для того, чтобы продвинуть ветку master к скачанной версии, воспользуемся командой git pull --ff-only:

Alexander@DESKTOP-GMVT49I MINGW32 ~/Desktop/lab02/alice/project (master)

$ git pull --ff-only

Updating 0f5e7e1..233c2c1

Fast-forward

main.cpp | 1 +

1 file changed, 1 insertion(+)

На данном этапе мы проделываем все тоже самое, только меняем alice и bob:

В main.cpp добавляем деление чисел, создаем коммит:

Alexander@DESKTOP-GMVT49I MINGW32 ~/Desktop/lab02/alice/project (master)

$ git commit -m "code: вывод деления"

[master e8a9083] code: вывод деления

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

Для того, чтобы убедиться, что коммит действительно был создан от имени alice, мы воспользовались командой git show HEAD:

Alexander@DESKTOP-GMVT49I MINGW32 ~/Desktop/lab02/alice/project (master)

$ git show HEAD

commit e8a9083cc9dfbdd0a77a10bff10c27b354f0f4b7 (HEAD -> master)

Author: Alice (SimichAB)

Date: Mon Mar 23 02:23:33 2020 +0300

code: вывод деления

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

index 25fca0b..dcfc57e 100644

--- a/main.cpp

+++ b/main.cpp

@@ -8,6 +8,6 @@ int main()

int a, b;

cin >> a >> b;

cout << "A - B = " << a - b;

-cout << "A \* B = " << a \* b;

+cout << "A / B = " << a / b;

return 0;

}

Отправляем коммит на GitHub:

|  |
| --- |
| Alexander@DESKTOP-GMVT49I MINGW32 ~/Desktop/lab02/alice/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), 372 bytes | 372.00 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/Alexandr-Simich/cs-lab02.git  233c2c1..e8a9083 master -> master |

Обновляем страницу и убеждаемся в том, что коммит отправился.

Выполняем загрузку изменений «на машине Боба» с помощью команды git fetch:

Alexander@DESKTOP-GMVT49I MINGW32 ~/Desktop/lab02/bob/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/Alexandr-Simich/cs-lab02

233c2c1..e8a9083 master -> origin/master

Просмотрим историю всех веток с помощью команды git log --oneline --decorate --all --graph:

Alexander@DESKTOP-GMVT49I MINGW32 ~/Desktop/lab02/bob/project (master)

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

\* e8a9083 (origin/master, origin/HEAD) code: вывод деления

\* 233c2c1 (HEAD -> master) code: вывод произведения

\* 0f5e7e1 игнор git

\* b13491c вывод разности

\* 2ed6587 вывод суммы

\* 89bf736 изменение кода

\* 5c1a801 build: add project file

\* 01da535 code: заготовка программы

Для того, чтобы продвинуть ветку master к скачанной версии, воспользуемся командой git pull --ff-only:

Alexander@DESKTOP-GMVT49I MINGW32 ~/Desktop/lab02/bob/project (master)

$ git pull --ff-only

Updating 233c2c1..e8a9083

Fast-forward

main.cpp | 2 +-

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

«На машине Алисы» дополняем программу печатью максимума, затем создаем коммит и отправляем его на GitHub:

Alexander@DESKTOP-GMVT49I MINGW32 ~/Desktop/lab02/alice/project (master)

$ git add main.cpp

Alexander@DESKTOP-GMVT49I MINGW32 ~/Desktop/lab02/alice/project (master)

$ git commit -m "code:максимум"

[master ae6d784] code:максимум

1 file changed, 6 insertions(+)

Alexander@DESKTOP-GMVT49I MINGW32 ~/Desktop/lab02/alice/project (master)

$ git push

Enumerating objects: 8, done.

Counting objects: 100% (8/8), done.

Delta compression using up to 4 threads

Compressing objects: 100% (6/6), done.

Writing objects: 100% (6/6), 796 bytes | 796.00 KiB/s, done.

Total 6 (delta 1), reused 0 (delta 0)

remote: Resolving deltas: 100% (1/1), done.

To https://github.com/Alexandr-Simich/cs-lab02.git

e8a9083..ae6d784 master -> master

«На машине Боба» дополняем программу печатью минимума, создаем коммит и отправляем его на GitHub. Можно заметить, удаленный репозитарий не принимает изменений: коммит Боба основан не на последнем существующем коммите:

Alexander@DESKTOP-GMVT49I MINGW32 ~/Desktop/lab02/bob/project (master)

$ git add main.cpp

Alexander@DESKTOP-GMVT49I MINGW32 ~/Desktop/lab02/bob/project (master)

$ git commit -m "code: минимум"

[master fb57bf1] code: минимум

1 file changed, 6 insertions(+)

Alexander@DESKTOP-GMVT49I MINGW32 ~/Desktop/lab02/bob/project (master)

$ git push

To https://github.com/Alexandr-Simich/cs-lab02.git

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

error: failed to push some refs to 'https://github.com/Alexandr-Simich/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.

«От лица Боба» необходимо загрузить коммиты из удаленного хранилища и отобразить историю всех веток:

Alexander@DESKTOP-GMVT49I MINGW32 ~/Desktop/lab02/bob/project (master)

$ git fetch

remote: Enumerating objects: 8, done.

remote: Counting objects: 100% (8/8), done.

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

remote: Total 6 (delta 1), reused 6 (delta 1), pack-reused 0

Unpacking objects: 100% (6/6), done.

From https://github.com/Alexandr-Simich/cs-lab02

e8a9083..ae6d784 master -> origin/master

Alexander@DESKTOP-GMVT49I MINGW32 ~/Desktop/lab02/bob/project (master)

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

\* fb57bf1 (HEAD -> master) code: минимум

\* ee4c0c2 сложение,вычитание,умножение,деление

| \* ae6d784 (origin/master, origin/HEAD) code:максимум

| \* a39ec32 сложение,вычитание,умножение,деление

|/

\* e8a9083 code: вывод деления

\* 233c2c1 code: вывод произведения

\* 0f5e7e1 игнор git

\* b13491c вывод разности

\* 2ed6587 вывод суммы

\* 89bf736 изменение кода

\* 5c1a801 build: add project file

\* 01da535 code: заготовка программы

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

Alexander@DESKTOP-GMVT49I MINGW32 ~/Desktop/lab02/bob/project (master)

$ git rebase orogin/master

fatal: invalid upstream 'orogin/master'

Alexander@DESKTOP-GMVT49I MINGW32 ~/Desktop/lab02/bob/project (master)

$ git rebase origin/master

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

Applying: code: минимум

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: минимум

Resolve all conflicts manually, mark them as resolved with

"git add/rm ", 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".

Alexander@DESKTOP-GMVT49I MINGW32 ~/Desktop/lab02/bob/project (master|REBASE 1/1)

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

\* fb57bf1 (master) code: минимум

\* ee4c0c2 сложение,вычитание,умножение,деление

| \* ae6d784 (HEAD, origin/master, origin/HEAD) code:максимум

| \* a39ec32 сложение,вычитание,умножение,деление

|/

\* e8a9083 code: вывод деления

\* 233c2c1 code: вывод произведения

\* 0f5e7e1 игнор git

\* b13491c вывод разности

\* 2ed6587 вывод суммы

\* 89bf736 изменение кода

\* 5c1a801 build: add project file

\* 01da535 code: заготовка программы

Однако эта команда завершается с ошибкой, сообщающей о конфликте в main.cpp. Для просмотра хранилища воспользуемся командой git status:

Alexander@DESKTOP-GMVT49I MINGW32 ~/Desktop/lab02/bob/project (master|REBASE 1/1)

$ git status

rebase in progress; onto ae6d784

You are currently rebasing branch 'master' on 'ae6d784'.

(fix conflicts and then run "git rebase --continue")

(use "git rebase --skip" to skip this patch)

(use "git rebase --abort" to check out the original branch)

Unmerged paths:

(use "git reset HEAD ..." to unstage)

(use "git add ..." to mark resolution)

both modified: main.cpp

no changes added to commit (use "git add" and/or "git commit -a")

«На машине Боба» в CodeBlocks на месте конфликта исправляем все необходимое. После того, как конфликт разрешил, необходимо добавить файл в индекс и продолжить прерванную операцию rebase. Это выполняется с помощью команды git add main.cpp и команды git rebase --continue:

Alexander@DESKTOP-GMVT49I MINGW32 ~/Desktop/lab02/bob/project (master|REBASE 1/1)

$ git add main.cpp

Alexander@DESKTOP-GMVT49I MINGW32 ~/Desktop/lab02/bob/project (master|REBASE 1/1)

$ git rebase --continue

Applying: code: минимум

Просмотрим историю и убедимся, что история соответствует требованиям (осуществляем с помощью команды git log –oneline –decorate –all --graph) и отправляем изменения на GitHub:

Alexander@DESKTOP-GMVT49I MINGW32 ~/Desktop/lab02/bob/project (master)

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

\* a02fc55 (HEAD -> master) code: минимум

\* ae6d784 (origin/master, origin/HEAD) code:максимум

\* a39ec32 сложение,вычитание,умножение,деление

\* e8a9083 code: вывод деления

\* 233c2c1 code: вывод произведения

\* 0f5e7e1 игнор git

\* b13491c вывод разности

\* 2ed6587 вывод суммы

\* 89bf736 изменение кода

\* 5c1a801 build: add project file

\* 01da535 code: заготовка программы

Alexander@DESKTOP-GMVT49I 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), 356 bytes | 356.00 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/Alexandr-Simich/cs-lab02.git

ae6d784..a02fc55 master -> master

Предположим, пока Боб синхронизировал изменения, Алиса решила изменить тип чисел с целых на действительные. Предполагая, что это займет время, Алиса ведет работу в отдельной ветке. На момент начала работы репозитарий Алисы *не* синхронизирован с GitHub, то есть последний коммит добавляет печать максимума. Все действия ведутся «на машине Алисы».

Создайте ветку double, но на момент работы репозитарий Алисы не синхронизирован с GitHub. Воспользуемся командой git branch double:

Alexander@DESKTOP-GMVT49I MINGW32 ~/Desktop/lab02/alice/project (master) $ git branch double

Чтобы переключиться на нее используем команду git checkout double:

Alexander@DESKTOP-GMVT49I MINGW32 ~/Desktop/lab02/alice/project (master)

$ git checkout double

Switched to branch 'double'

Заменяем тип переменных на double и создаем коммит:

Alexander@DESKTOP-GMVT49I MINGW32 ~/Desktop/lab02/alice/project (double)

$ git commit -m "code: изменение типа переменных"

[double 1bf68b1] code: изменение типа переменных

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

Переключаемся на ветку master:

Alexander@DESKTOP-GMVT49I MINGW32 ~/Desktop/lab02/alice/project (double)

$ git checkout master

Switched to branch 'master'

Your branch is up to date with 'origin/master'.

На данном этапе синхронизируем ветку master «на машине Алисы» с GitHub:

Alexander@DESKTOP-GMVT49I MINGW32 ~/Desktop/lab02/alice/project (master)

$ git push --set-upstream origin double

Enumerating objects: 33, done.

Counting objects: 100% (33/33), done.

Delta compression using up to 4 threads

Compressing objects: 100% (32/32), done.

Writing objects: 100% (33/33), 3.82 KiB | 781.00 KiB/s, done.

Total 33 (delta 8), reused 0 (delta 0)

remote: Resolving deltas: 100% (8/8), done.

remote:

remote: Create a pull request for 'double' on GitHub by visiting:

remote: https://github.com/Alexandr-Simich/cs-lab02/pull/new/double

remote:

To https://github.com/Alexandr-Simich/cs-lab02.git

\* [new branch] double -> double

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

Сливаем ветку double в master с помощью git merge double:

Alexander@DESKTOP-GMVT49I MINGW32 ~/Desktop/lab02/alice/project (master)

$ git merge double

Updating ae6d784..1bf68b1

Fast-forward

main.cpp | 2 +-

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

Alexander@DESKTOP-GMVT49I MINGW32 ~/Desktop/lab02/alice/project (master)

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

\* 1bf68b1 (HEAD -> master, origin/double, double) code: изменение типа переменных

\* ae6d784 (origin/master) code:максимум

\* a39ec32 сложение,вычитание,умножение,деление

\* e8a9083 code: вывод деления

\* 233c2c1 code: вывод произведения

\* 0f5e7e1 игнор git

\* b13491c вывод разности

\* 2ed6587 вывод суммы

\* 89bf736 изменение кода

\* 5c1a801 build: add project file

\* 01da535 code: заготовка программы