Лабораторная работа №3 часть 2.

Основы работы с ветвями в GIT. Основы слияния. Разрешение конфликтов.

Часть 6. Создание ветвей и слияние.

1. Создайте новую ветвь greet\_name (где name – Ваша фамилия), используя команду

git checkout -b <branchname>

1. Добавьте новый каталог (пример lib\_name) и файл *greeter\_name.rb* c содержимым:

class Greeter

def initialize(who)

@who = who

end

def greet

"Hello, #{@who}"

end

end

1. Сделайте 3 commits в ветку greet\_name
2. Продемонстрируйте переключение на ветку master.
3. Создайте файл README и произведите commit в master.
4. Произведите слияние ветвей greet и master – продемонстрируйте историю commits. (используйте опцию Fast forward).
5. Изобразите в удобном для Вас графическом редакторе граф коммитов, где вершинами графа будут коды коммитов.

Часть 7.

Смоделируйте искусственно конфликт, при слиянии описанном в предыдущей части.

Часть 8.

Используя команду reset верните указатель на состояние ветвей до их слияния.

Часть 9. Проделайте шаги 3-5 в части 6. Выполните слияние изменений командой rebase. Изобразите в удобном для Вас графическом редакторе граф коммитов, где вершинами графа будут коды коммитов.

Объясните разницу между merge и rebase. На примере Ваших графов.

Для начала работы создайте/зарегистрируйтесь (в случае отсутствия на текущий момент) на удаленном сервисе (типа *github*) *,* т.е. инициализируйте удаленный репозиторий.

Часть 10.

1. Разбейтесь на пары.
2. Выберите в качестве общего репозитория один из двух удаленных.
3. Student1 заливает все содержимое своих папок в удаленный репозиторий.

Student 2 «забирает» проект к себе в локальный репозиторий.

1. Выполните команду git status.
2. Проведите каждый по три коммита в локальный репозиторий и «залейте изменения» в удаленный. (разрешается использование и merge и rebase).