**МИНИCTEPCTBO НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ**

**РОССИЙСКОЙ ФЕДЕРАЦИИ**

**Федеральное государственное автономное**

**образовательное учреждение высшего образования**

**«СЕВЕРО-КАВКАЗСКИЙ ФЕДЕРАЛЬНЫЙ УНИВЕРСИТЕТ»**

**Кафедра инфокоммуникаций**

**Языки программирования**

**Отчет по лабораторной работе № 2**

**«Работа с кортежами в языке Python».**

Выполнил студент группы

ИТС-б-о-20-1 (1)

Рудаков В.Б. « »\_\_\_\_\_\_\_\_\_20\_\_г.

Подпись студента\_\_\_\_\_\_\_\_\_\_\_

Работа защищена « »\_\_\_\_\_\_\_\_\_\_\_\_20\_\_г.

Проверил доцент

Кафедры инфокоммуникаций, кандидат

технических наук

Воронкин Р.А.

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

(подпись)

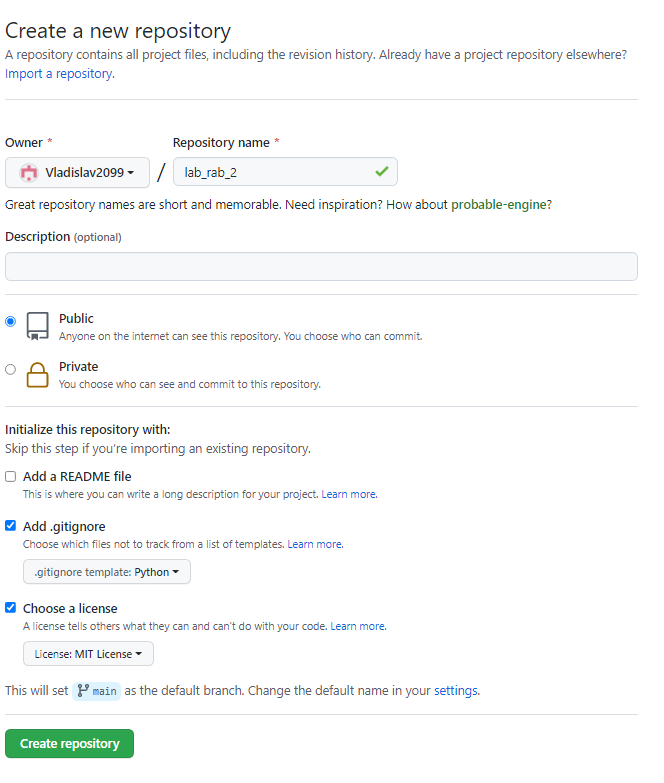
Ставрополь, 2021

**Тема работы:** Работа с кортежами в языке Python.

**Цель работы:** приобретение навыков по работе с кортежами при написании программ с помощью языка программирования Python версии 3.x.

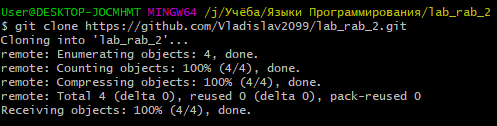
**Ссылка на репозиторий:** <https://github.com/Vladislav2099/lab_rab_2/tree/main>

1. Создать общедоступный репозиторий на GitHub, в котором будет использована лицензия MIT и язык программирования Python.



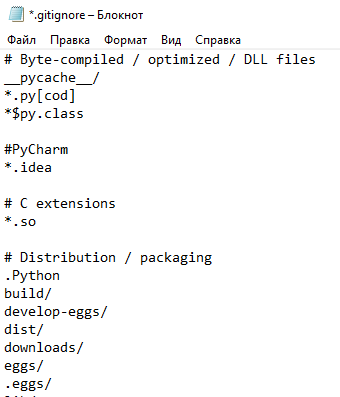
**Рисунок 1. Создание репозитория.**

1. Выполните клонирование созданного репозитория.



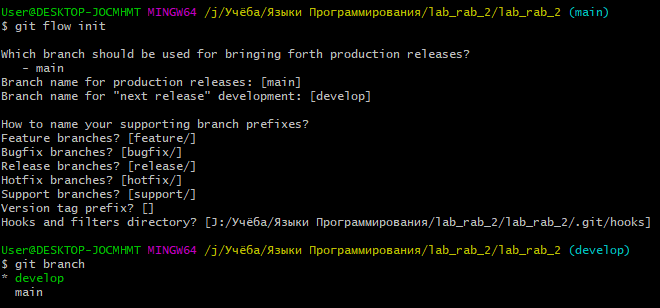
**Рисунок 2. Клонирование репозитория.**

1. Дополните файл .gitignore необходимыми правилами для работы с IDE PyCharm.



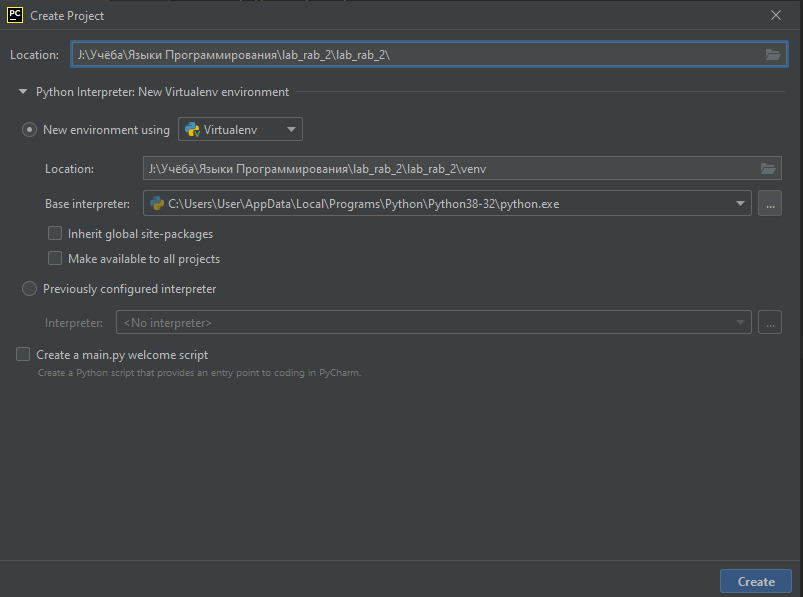
**Рисунок 3. Редактирование файла .gitignore.**

1. Организуйте свой репозиторий в соответствие с моделью ветвления git-flow.



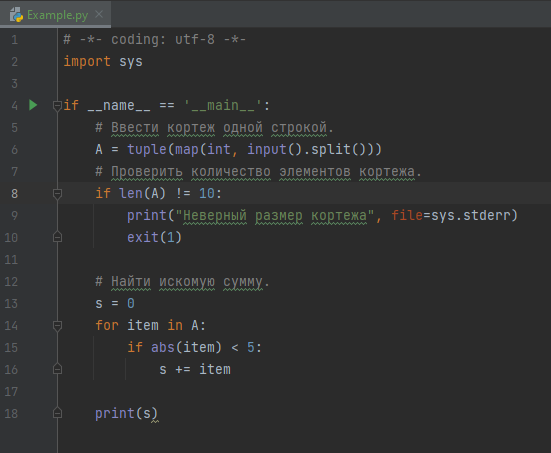
**Рисунок 4. Использование ветвления git-flow.**

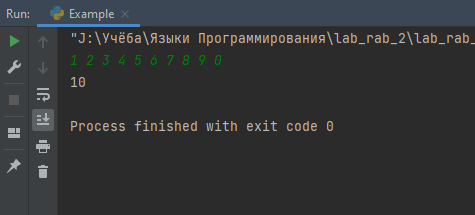
1. Создайте проект PyCharm в папке репозитория.

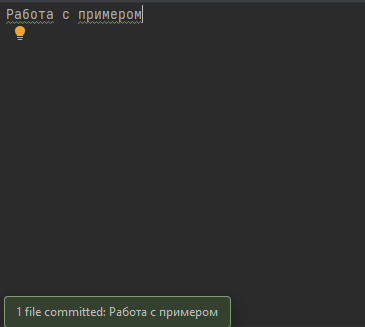


**Рисунок 5. Создание проекта PyCharm.**

1. Проработайте пример лабораторной работы. Создайте для него отдельный модуль языка Python. Зафиксируйте изменения в репозитории.

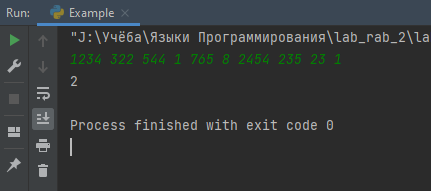


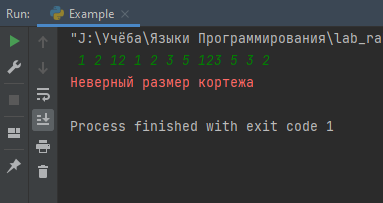




**Рисунок 6. Пример лабораторной работы.**

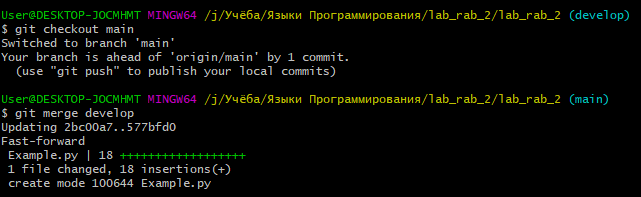
1. Приведите в отчете скриншоты результатов выполнения примера при различных исходных данных вводимых с клавиатуры.





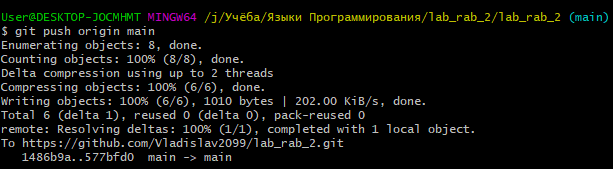
**Рисунок 7. Пример с различными исходными данными.**

1. Выполните слияние ветки для разработки с веткой main.



**Рисунок 8. Слияние веток.**

1. Отправьте сделанные изменения на сервер GitHub.



**Рисунок 9. Отправка изменений.**

КОНТРОЛЬНЫЕ ВОПРОСЫ:

1. Что такое списки в языке Python?

Список – это нумерованный набор объектов.

1. Каково назначение кортежей в языке Python?

Одна из причин – это безопасность данные от случайного изменения.

1. Как осуществляется создание кортежей?

*>>> a = ()*

*>>> print(type(a))*

*>>> a = tuple([1, 2, 3, 4])*

*>>> print(a)*

1. Как осуществляется доступ к элементам кортежа?

Доступ к элементам кортежа осуществляется также как к элементам списка – через указание индекса

1. Зачем нужна распаковка (деструктуризация) кортежа?

Для упрощённого обращения к данным кортежа, не только используя индекс.

1. Какую роль играют кортежи в множественном присваивании?

Благодаря тому, что кортежи легко собирать и разбирать, в Python удобно делать такие вещи, как множественное присваивание

*(a, b, c) = (1, 2, 3)*

*a # 1*

*b # 2*

*c # 3*

1. Как выбрать элементы кортежа с помощью среза?

С помощью операции взятия среза можно получить другой кортеж. Общая форма операции взятия среза для кортежа:

*T2 = T1[i:j]*

здесь

T2 – новый кортеж, который получается из кортежа T1;

T1 – исходный кортеж, для которого происходит срез;

i, j – соответственно нижняя и верхняя границы среза. Фактически берутся ко вниманию элементы, лежащие на позициях i, i+1, …, j-1. Значение j определяет позицию за последним элементом среза.

Операция взятия среза для кортежа может иметь модификации такие же как и для списков.

1. Как выполняется конкатенация и повторение кортежей?

Для кортежей можно выполнять операцию конкатенации, которая обозначается символом +.

*T3 = T1 + T2*

где

T1, T2 – кортежи, для которых нужно выполнить операцию конкатенации. Операнды T1, T2 обязательно должны быть кортежами. При выполнении операции конкатенации для кортежей, использовать в качестве операндов любые другие типы (строки, списки) запрещено;

T3 – кортеж, который есть результатом.

1. Как выполняется обход элементов кортежа?

Элементы кортежа можно последовательно просмотреть с помощью операторов цикла *while* или *for*.

1. Как проверить принадлежность элемента кортежу?

*# Проверка вхождения элемента в кортеж*

*# Оператор in*

*# Заданный кортеж, который содержит строки*

*A = ("abc", "abcd", "bcd", "cde")*

*# Ввести элемент*

*item = str(input("s = "))*

*if (item in A):*

*print(item, " in ", A, " = True")*

*else:*

*print(item, " in ", A, " = False")*

1. Какие методы работы с кортежами Вам известны?

**Метод index(). Поиск позиции элемента в кортеже:**

*pos = T.index(item)*

**Метод count(). Количество вхождений элемента в кортеж:**

*k = T.count(item)*

1. Допустимо ли использование функций агрегации таких как *len(), sum()* и т. д. при работе с кортежами?

Допустимо, если условия использования той или иной функции не нарушены.

1. Как создать кортеж с помощью спискового включения.

*(a for a in A ...)* дает на выходе специальный объект генератора, а не кортеж. Для

преобразования генератора в кортеж необходимо воспользоваться вызовом *tuple()* .