# МИНИCTEPCTBO НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ

Федеральное государственное автономное образовательное учреждение высшего образования

# «СЕВЕРО-КАВКАЗСКИЙ ФЕДЕРАЛЬНЫЙ УНИВЕРСИТЕТ»

Кафедра инфокоммуникаций

# Отчет по лабораторной работе № 2.14

по дисциплине «Основы программной инженерии»

Выполнил студент группы ПИЖ-б-о-22-1 Душин Александр Владимирович.

Подпись студента

Работа защищена « » 20 г.

Проверил Воронкин Р.А.

(подпись)

Ставрополь 2024

Тема: Модули и пакеты.

Цель работы: приобретение навыков по работе с модулями и пакетами языка программирования Python версии 3.x.

Ход выполнения работы:

1. Создать общедоступный репозиторий на GitHub с использованием лицензии MIT и язык программирования Python:

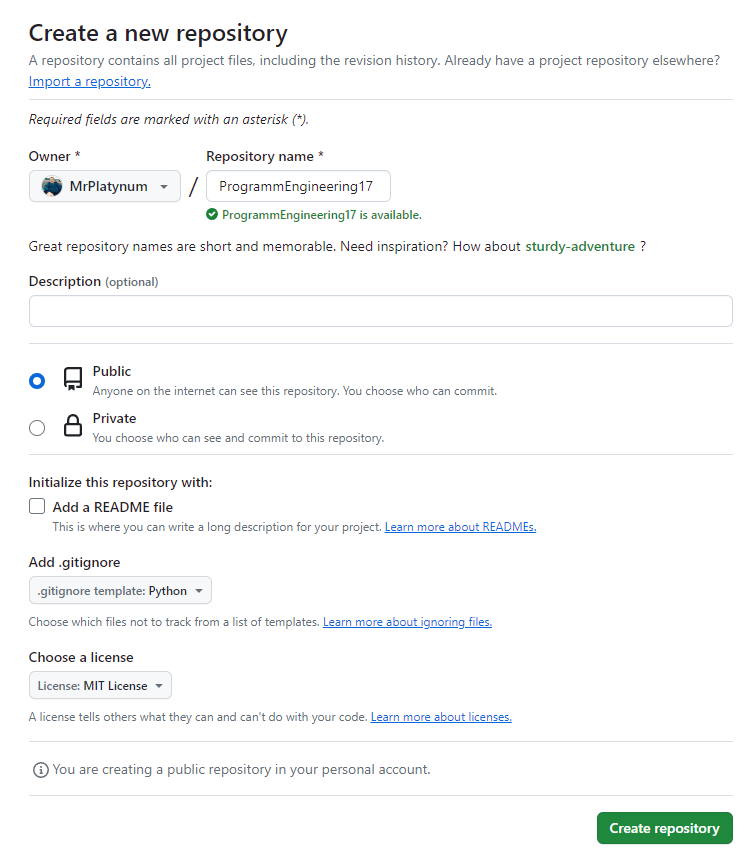


Рисунок 1 – Создание общедоступного репозитория на GitHub с заданными настройками

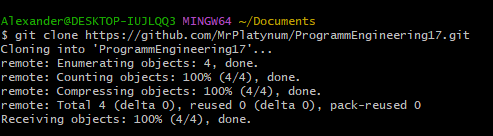


Рисунок 2 – Клонирование созданного репозитория на локальный компьютер

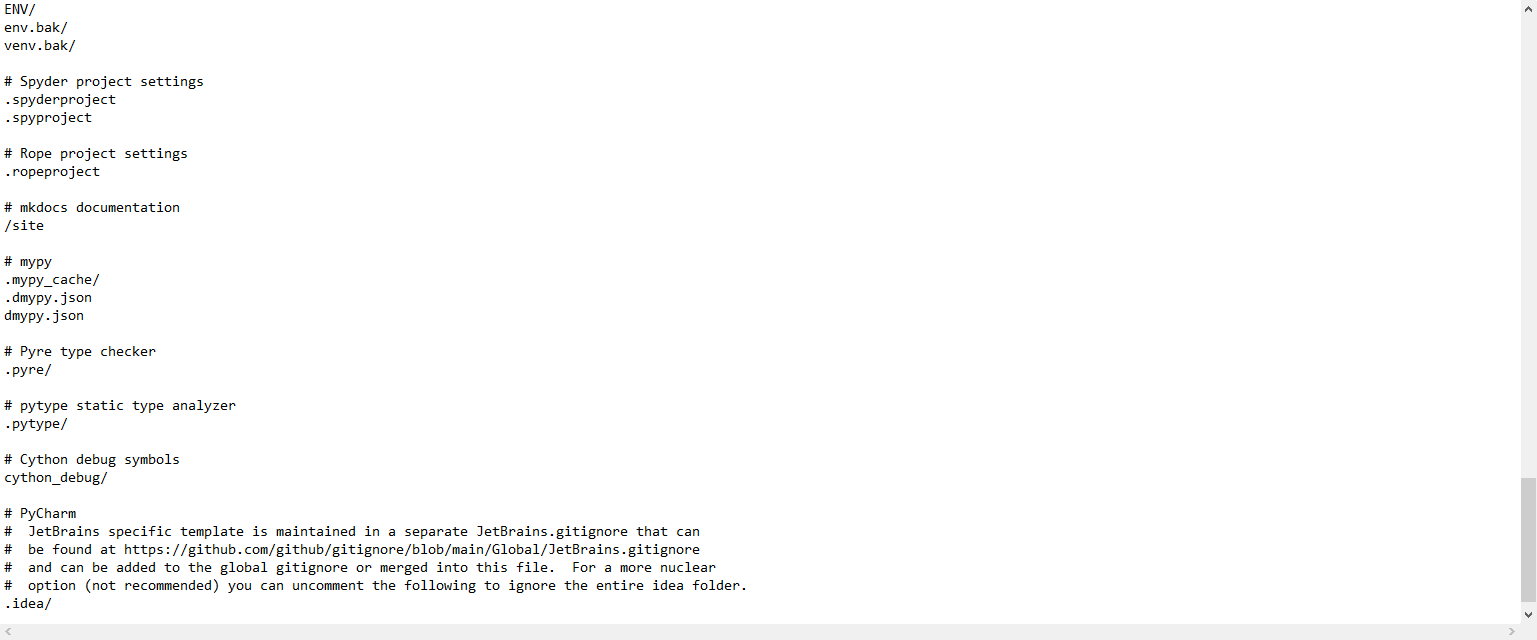


Рисунок 3 – файл .gitignore

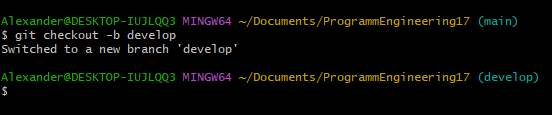


Рисунок 4 – организация репозитория в соответствии с моделью ветвления git flow

1. Выполнить индивидуальное задание лабораторной работы 2.11, оформив все функции программы в виде отдельного модуля. Разработанный модуль должен быть подключен в основную программу с помощью одного из вариантов команды import . Номер варианта уточнить у преподавателя:

Листинг индивидуального задания 1:

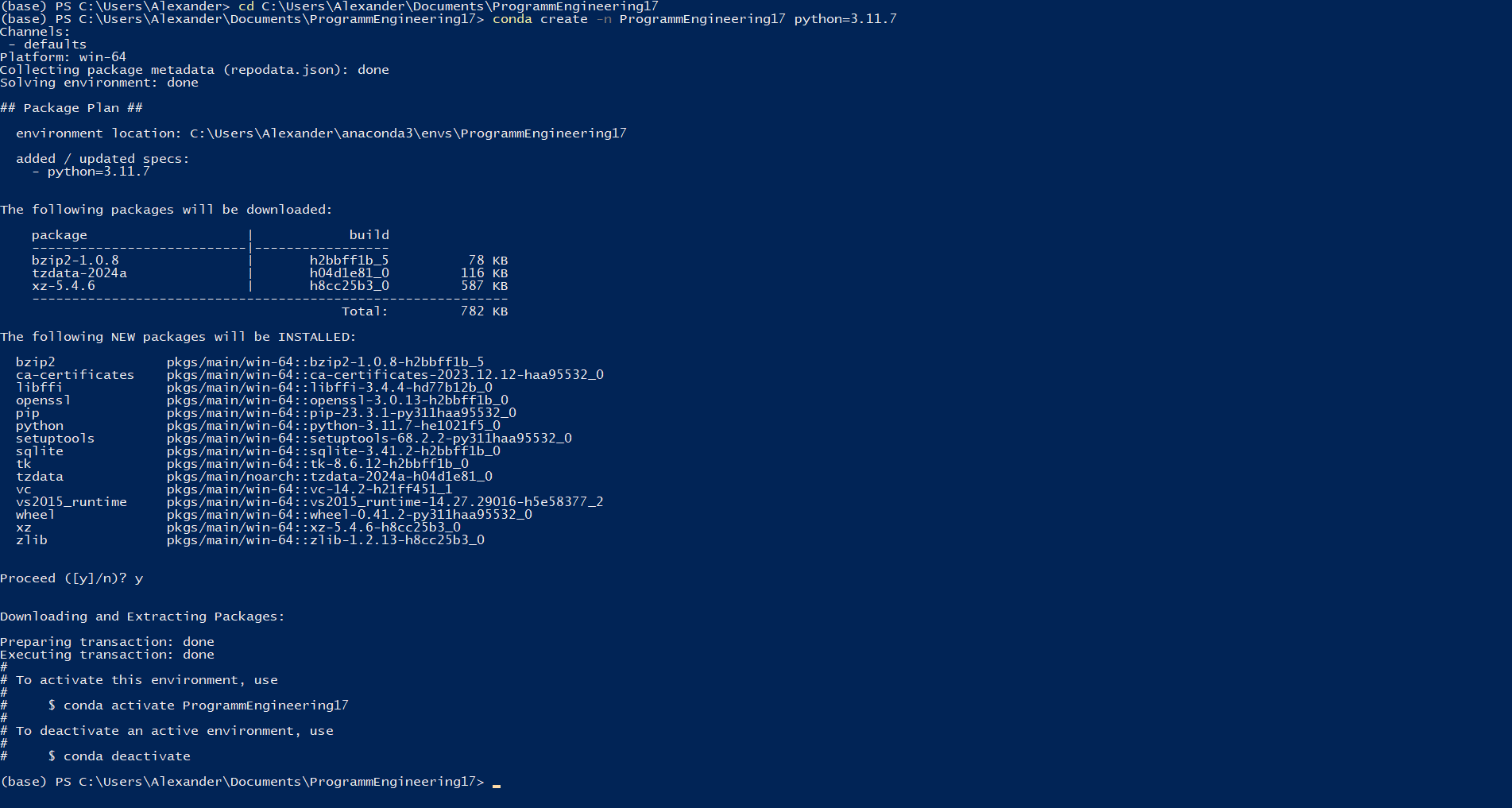


Рисунок 5 – Индивидуальное задание №1

1. Выполнить индивидуальное задание лабораторной работы 2.8, оформив все классы программы в виде отдельного пакета. Разработанный пакет должен быть подключен в основную программу с помощью одного из вариантов команды import . Настроить соответствующим образом переменную \_\_all\_\_ в файле \_\_init\_\_.py пакета. Номер варианта уточнить у преподавателя.

Листинг индивидуального задания 2:

*#!/usr/bin/env python3*

*# -\*- coding: utf-8 -\*-*

from train\_system import add\_train, list\_trains, select\_trains, display\_help

def main():

"""Основная функция управления программой."""

trains = []

while True:

command = input(">>> ").lower()

match command:

case 'exit':

break

case 'add':

add\_train(trains)

case 'list':

list\_trains(trains)

case 'select':

selected = select\_trains(trains, input("Введите время для поиска поездов (в формате ЧЧ:ММ): "))

if isinstance(selected, list):

list\_trains(selected)

else:

print(selected)

case 'help':

display\_help()

case \_:

print(f"Неизвестная команда {command}")

if \_\_name\_\_ == '\_\_main\_\_':

main()

 Листинг модуля train\_function:

from datetime import time

def add\_train(trains):

"""Добавляет информацию о поезде."""

destination = input("Название пункта назначения: ")

train\_number = input("Номер поезда: ")

departure\_time\_str = input("Время отправления (в формате ЧЧ:ММ): ")

hours, minutes = map(int, departure\_time\_str.split(':'))

departure\_time = time(hours, minutes)

train = {

'название пункта назначения': destination,

'номер поезда': train\_number,

'время отправления': departure\_time,

}

trains.append(train)

trains.sort(key=lambda x: x['название пункта назначения'])

def list\_trains(trains):

"""Выводит список всех поездов."""

line = f'+-{"-" \* 35}-+-{"-" \* 15}-+-{"-" \* 25}-+'

print(line)

print(f"| {'Название пункта назначения':^35} | {'Номер поезда':^15} | {'Время отправления':^25} |")

for train in trains:

print(line)

departure\_time = train['время отправления'].strftime('%H:%M')

print(

f"| {train['название пункта назначения']:^35} | {train['номер поезда']:^15} | {departure\_time:^25} |")

print(line)

def select\_trains(trains, search\_time\_str):

"""Выводит поезда, отправляющиеся после указанного времени."""

found = False

result = []

hours, minutes = map(int, search\_time\_str.split(':'))

search\_time = time(hours, minutes)

print(f"Поезда, отправляющиеся после {search\_time}:")

for train in trains:

train\_time = train['время отправления']

if train\_time > search\_time:

result.append(train)

found = True

if found:

return result

if not found:

return "Нет поездов, отправляющихся после указанного времени."

def display\_help():

"""Выводит справку о доступных командах."""

print("Список команд:\n")

print("add - добавить информацию о поезде;")

print("list - вывести список всех поездов;")

print("select <время> - вывести поезда, отправляющиеся после указанного времени;")

print("exit - завершить работу с программой.")

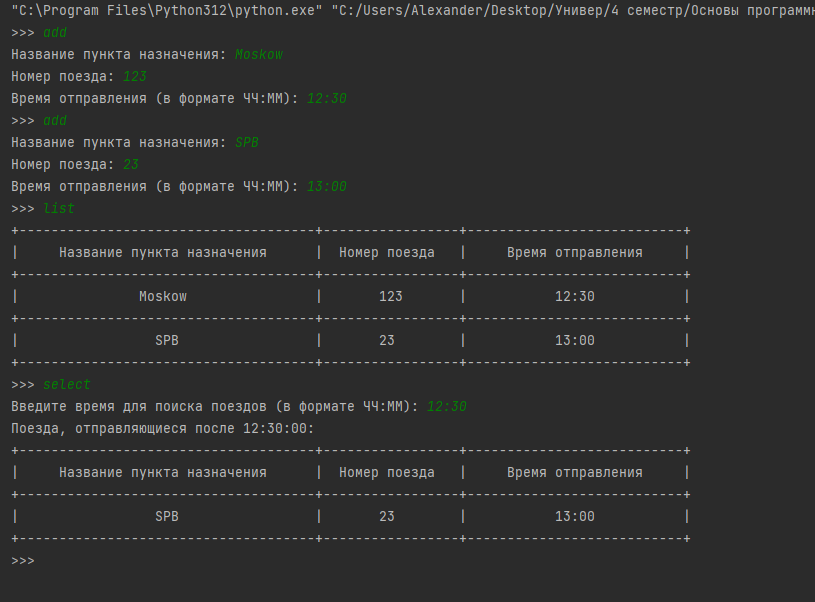


Рисунок 6 – Индивидуальное задание №2

1. Зафиксируем проделанные изменения, сольем ветки и отправим на удаленный репозиторий:

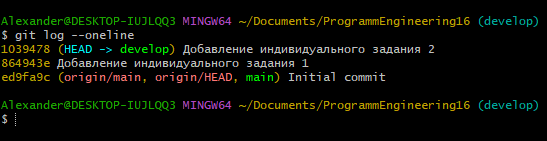


Рисунок 7 – Коммиты ветки develop во время выполнения лабораторной работы

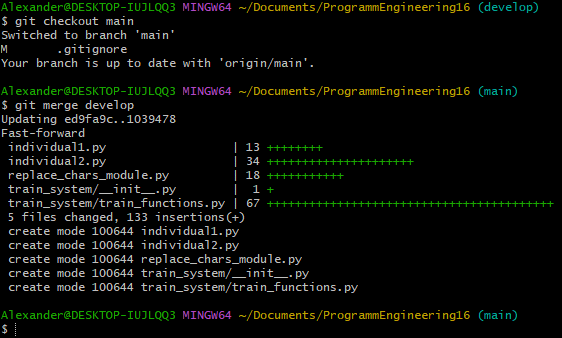


Рисунок 8 – Слияние ветки develop в ветку main

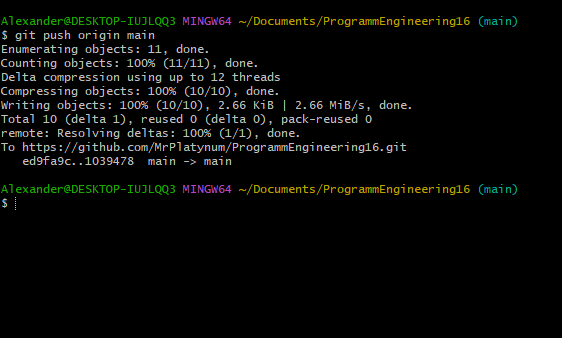


Рисунок 9 – Отправка на удаленный репозиторий

Ответы на контрольные вопросы:

1. Что является модулем языка в Python?

В Python модуль - это файл с расширением .py, содержащий код на языке Python. Модуль может содержать функции, классы и переменные, и его можно использовать для организации кода и повторного использования.

1. Какие существуют способы подключения модулей в языке Python?

Есть несколько способов подключения модулей:

Использование ключевого слова import (например, import module\_name).

Использование ключевого слова from для импорта конкретных элементов из модуля (например, from module\_name import function\_name).

Использование псевдонимов с ключевым словом as (например, import module\_name as alias).

1. Что является пакетом языка Python?

Пакет в Python - это способ организации модулей в иерархическую структуру внутри директории. Пакеты позволяют логически группировать связанный код для более удобного управления проектами.

1. Каково назначение файла \_\_init\_\_.py?

Файл \_\_init\_\_.py в директории Python указывает, что эта директория должна рассматриваться как пакет, а не просто коллекция модулей. Этот файл может быть пустым, но его наличие делает директорию пригодной для использования в качестве пакета.

1. Каково назначение переменной \_\_all\_\_ в файле \_\_init\_\_.py?

Переменная \_\_all\_\_ в файле \_\_init\_\_.py используется для определения списка модулей, которые будут импортированы при использовании оператора from package\_name import \*. Если \_\_all\_\_ не определен, будут импортированы все модули, кроме тех, чьи имена начинаются с подчеркивания. Если \_\_all\_\_ определен, будут импортированы только модули, указанные в этом списке.