МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ АВТОНОМНОЕ ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ

«СЕВЕРО-КАВКАЗСКИЙ ФЕДЕРАЛЬНЫЙ УНИВЕРСИТЕТ» ИНСТИТУТ ЦИФРОВОГО РАЗВИТИЯ

Отчет о лабораторной работе №9

по дисциплине основы программной инженерии

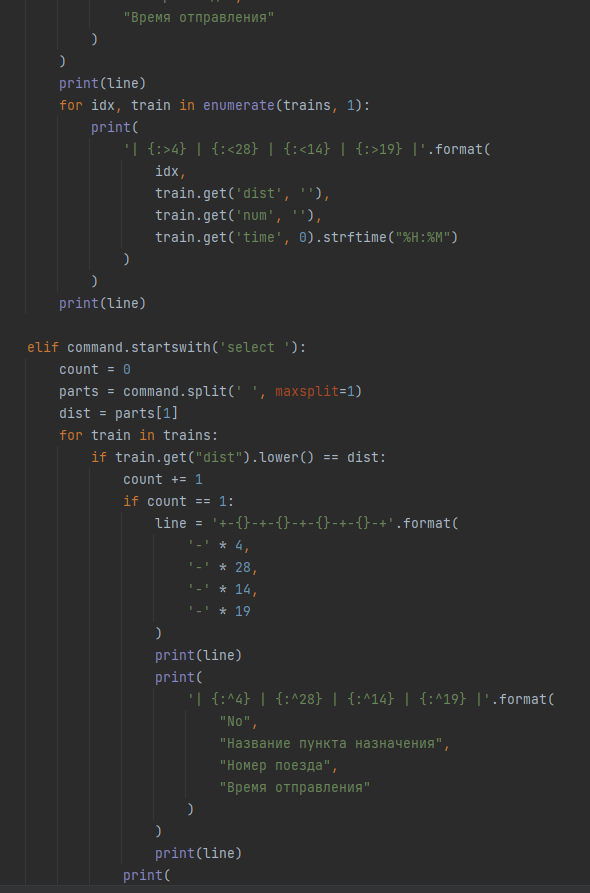
Выполнила: Гробова Софья Кирилловна,

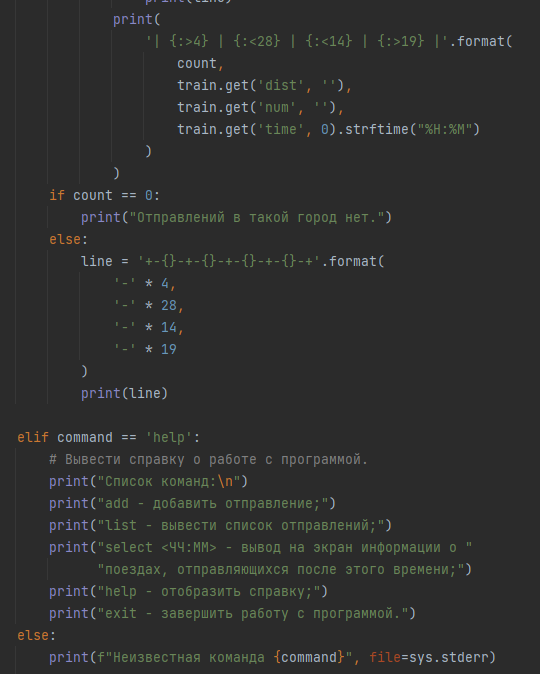
2 курс, группа ПИЖ-б-о-20-1,

Проверил: Доцент кафедры

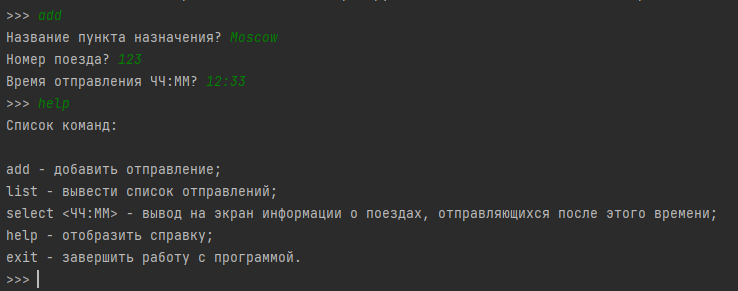
инфокоммуникаций, Воронкин Р.А.

Ставрополь, 2021 г





Индивидуальное задание



Результат работы программы

**Контрольные вопросы:**

# Что такое словари в языке Python?

**Словари** в **Python** - неупорядоченные коллекции произвольных объектов с доступом по ключу. Их иногда ещё называют ассоциативными массивами или хеш-таблицами.

# Может ли функция len() быть использована при работе со словарями?

Да может, она возвращает количество пар ключ значение

# Какие методы обхода словарей Вам известны?

При помощи цикла фор

# Какими способами можно получить значения из словаря по ключу?

dct[<key>]

# Какими способами можно установить значение в словаре по ключу?

dtc[<key>] = <meaning>

{x: x \* x for x in (1, 2, 3, 4)} – Словарь включения

1. **Что такое словарь включений?**

Словарь включений аналогичен списковым включениям, за исключением того, что он создаёт объект словаря вместо списка.

# Самостоятельно изучите возможности функции zip() приведите примеры ее использования.

Функция zip() в Python создает итератор, который объединяет элементы из нескольких источников данных. Эта функция работает со списками, кортежами, множествами и словарями для создания списков или кортежей, включающих все эти данные.

employee\_numbers = [2, 9, 18, 28]

employee\_names = ["Дима", "Марина", "Андрей", "Никита"]

zipped\_values = zip(employee\_names, employee\_numbers) zipped\_list = list(zipped\_values)

[('Дима', 2), ('Марина', 9), ('Андрей', 18), ('Никита', 28)]

# Самостоятельно изучите возможности модуля datetime. Каким функционалом по работе с

**датой и временем обладает этот модуль?**

Модуль datetime предоставляет классы для обработки времени и даты разными способами.

Класс **datetime.date**(year, month, day) - стандартная дата. Атрибуты: year, month, day. Неизменяемый объект.

Класс **datetime.time**(hour=0, minute=0, second=0, microsecond=0, tzinfo=None) - стандартное время, не зависит от даты. Атрибуты: hour, minute, second, microsecond, tzinfo.

Класс **datetime.timedelta** - разница между двумя моментами времени, с точностью до микросекунд.

## Класс **datetime.tzinfo** - абстрактный базовый класс для информации о временной зоне (например, для учета часового пояса и / или летнего времени).

Класс **datetime.datetime**(year, month, day, hour=0, minute=0, second=0, microsecond=0, tzinfo=None) - комбинация даты и времени.

## Обязательные аргументы:

* datetime.MINYEAR (1) ≤ year ≤ datetime.MAXYEAR (9999)

## 1 ≤ month ≤ 12

* 1 ≤ day ≤ количество дней в данном месяце и году Необязательные:

## 0 ≤ minute < 60

* 0 ≤ second < 60

## 0 ≤ microsecond < 1000000 Методы класса datetime:

**datetime.today()** - объект datetime из текущей даты и времени. Работает также,

## как и datetime.now() со значением tz=None.

**datetime.fromtimestamp(timestamp)** - дата из стандартного представления времени.

**datetime.fromordinal(ordinal)** - дата из числа, представляющего собой количество дней, прошедших с 01.01.1970.

## **datetime.now**(tz=None) - объект datetime из текущей даты и времени.

**datetime.combine**(date, time) - объект datetime из комбинации объектов date и time.

**datetime.strptime**(date\_string, format) - преобразует строку в datetime (так же, как и функция strptime из [модуля time](https://pythonworld.ru/moduli/modul-time.html)).

**datetime.strftime**(format) - см. функцию strftime из модуля time. **datetime.date**() - объект даты (с отсечением времени). **datetime.time**() - объект времени (с отсечением даты).

## **datetime.replace**([year[, month[, day[, hour[, minute[, second[, microsecond[, tzinfo]]]]]]]]) - возвращает новый объект datetime с изменёнными атрибутами.

**datetime.timetuple**() - возвращает struct\_time из datetime. **datetime.toordinal**() - количество дней, прошедших с 01.01.1970. **datetime.timestamp**() - возвращает время в секундах с начала эпохи.

## **datetime.weekday**() - день недели в виде числа, понедельник - 0, воскресенье - 6.

**datetime.isoweekday**() - день недели в виде числа, понедельник - 1, воскресенье - 7.

**datetime.isocalendar**() - кортеж (год в формате ISO, ISO номер недели, ISO день недели).

## **datetime.isoformat**(sep='T') - красивая строка вида "YYYY-MM- DDTHH:MM:SS.mmmmmm" или, если microsecond == 0, "YYYY-MM-DDTHH:MM:SS"

**datetime.ctime**() - см. ctime() из [модуля time](https://pythonworld.ru/moduli/modul-time.html).