Министерство науки и высшего образования Российской Федерации

Федеральное государственное бюджетное образовательное учреждение высшего образования «Ульяновский государственный технический университет»

Кафедра «Вычислительная техника»

**Отчет по лабораторной работе №4**

Дисциплина: «Разработка профессиональных приложений»

Классы

Вариант № 1

Выполнил:

студент группы ИВТАСбд-21

Абросимов К. С.

Проверил:

преподаватель кафедры

«Вычислительная техника»

Исхаков И.И.

Ульяновск, 2023

1. **Задание по варианту**

Необходимо переделать лабораторную работу №3 с использованием классов, описывающих предметную область, заданную вариантом, с реализацией следующих особенностей (вполне возможно, что предлагаемое в 3 лабе задание для этого нужно будет расширить):

1. Класс должен содержать итератор

2. Должна быть реализована перегрузка стандартных операций (repr, например)

3. Должно быть реализовано наследование

4. Запись значений в свойства - только через \_\_setattr\_\_

5. Возможность доступа к элементам коллекции по индексу (\_\_getitem\_\_)

6. Должны быть реализованы статические методы

7. Должны быть реализованы генераторы

1. **Описание реализации**

Программа содержит 3 класса: Data, которая хранит в себе список студентов со значениями. Этот класс использует RowModel, которая взаимодействует непосредственно с самими значениями в списке, с конкретными значениями. RowModel наследуется от класс Row, который взаимодействует лишь с одним атрибутом – номером студента, и работает с ним.

В начале создаётся переменная d как новая Data, на входе есть путь к файлу. Вызывается функция \_\_init\_\_, при которой вызывается функция parse. В этой функции проходит заполнение списка элементами. Элементы заполняются с помощью RowModel, в которой каждый раз при присваивании значения (ФИО, email и группы) вызывается функция \_\_setattr\_\_, которая записывает значения в свойства. Так и составляется список.

Далее при выводе этого списка вызывается функция \_\_str\_\_, которая отвечает за непосредственно вывод списка, удобный для человека. Проходимся по списку с помощью итератора сперва (он проходит список от начала до конца, но никогда не возвращается назад), а затем с помощью генератора (функция as\_generator с указателями, функцией \_\_next\_\_ (возвращает текущий элемент) и \_\_iter\_\_ (получает итератор для перебора))

Далее идут сортировки (по ФИО, по номеру, по номеру, и по принадлежности к группе; функции примерно такие же, только находятся в классе data, а также в командах используются анонимные функции, использующиеся единожды (lambda)).

Добавление элемента также понятно, только здесь в одной функции элемент записывается в список, а потом в другом же классе этот элемент дописывается в файл. (функции add\_new и save).

Выборка по индексу выполняется с помощью \_\_getitem\_\_.

1. **Описание возникших затруднений**

Затруднений было много. Было непонятно, как вообще реализовать саму задачу. Нужно было забыть про словарь и использовать как раз классы. Нужно было изучить классы на языке Python. Также нужно было разбираться в большом количестве методов. Чем, к примеру отличалось \_\_str\_\_ от \_\_repr\_\_? Дело в том, что \_\_str\_\_ явно для человека выводит информацию, когда же \_\_repr\_\_ требуется для разработчика, в формальной и подробной форме.

Также было тяжело отличить генератор от итератора, ведь концептуально они похожи.

1. **Описание альтернативных способов решения**

Альтернативы здесь не было найдено.