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 № 20

Выполнил:

студент группы ИВТАПбд-21

Рябов И. А.

Проверил:

преподаватель кафедры

«Вычислительная техника»

Исхаков И.И.

Ульяновск, 2023

**Задание по варианту**

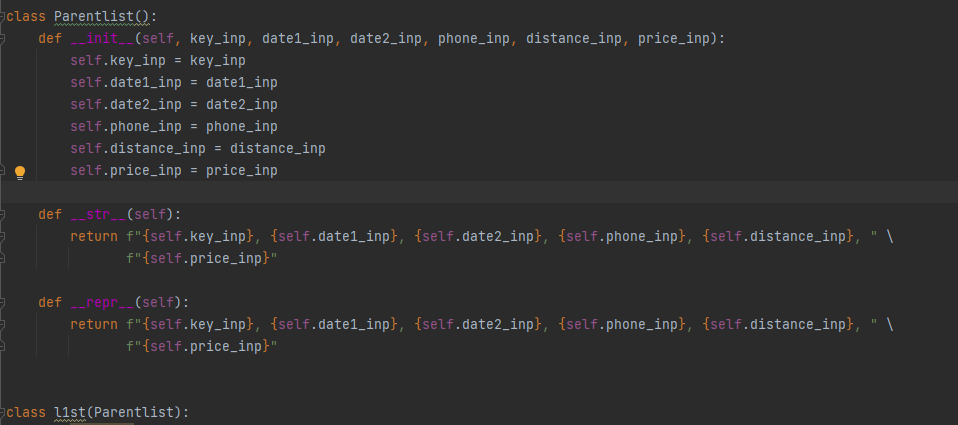
Необходимо переделать лабораторную работу №3 с использованием классов, описывающих предметную область, заданную вариантом, с реализацией следующих особенностей (вполне возможно, что предлагаемое в 3 лабе задание для этого нужно будет расширить):

**Требования к программе:**

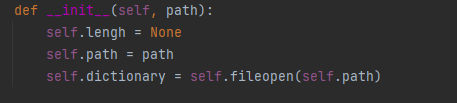
1. Класс должен содержать итератор
2. Должна быть реализована перегрузка стандартных операций (repr, например)
3. Должно быть реализовано наследование
4. Запись значений в свойства - только через \_\_setattr\_\_
5. Возможность доступа к элементам коллекции по индексу (\_\_getitem\_\_)
6. Должны быть реализованы статические методы
7. Должны быть реализованы генераторы

**Описание реализации**

Класс Parentlist с перегрузкой методов \_\_str\_\_ и \_\_repr\_\_:



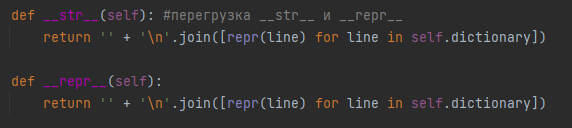
Инициализация данных, конструктор в котором указан путь до файла



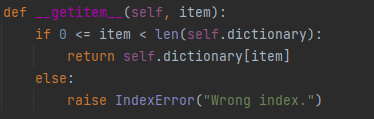
Итератор для вывода всего словаря:



Перегрузка методов:



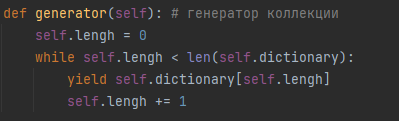
\_\_getitem\_\_ , благодаря которому возможно получить любую строку словаря по ее индексу:



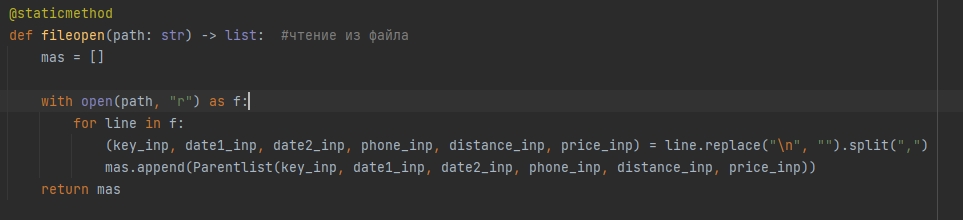
\_\_setattr\_ для записи значений в словарь:



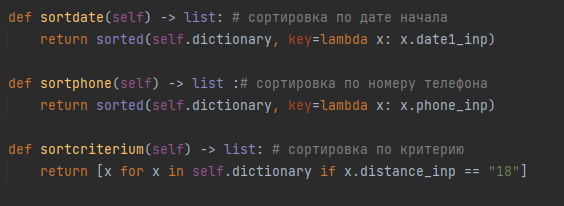
Генератор коллекции:



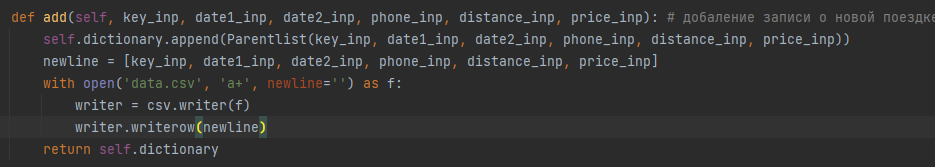
Чтение из файла:



Сортировка словаря по дате начала, номеру телефона и критерию:



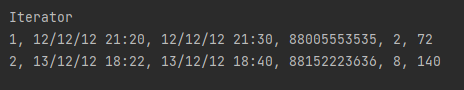
Добавление новых данных в словарь:



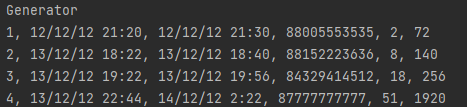
**Описание возникших затруднений**

Работа с ООП, классами, методами, перегрузками методов, а также с наследованием была применена впервые на языке Python. ООП в этом языке достаточно сильно отличается от ООП других языков, на работу ушло много времени, тем не менее с выполнением поставленной задачи справился.

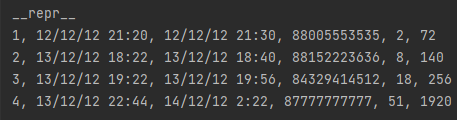
**Результаты работы программы**

****

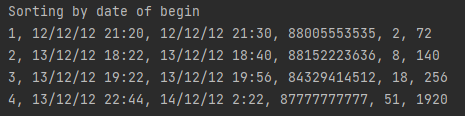
**1 рис. Вывод данных итератором**

****

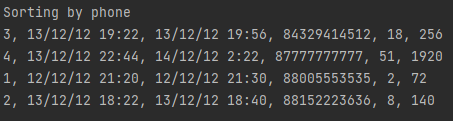
**2 рис. Вывод данных генератором**

****

**3 рис. Вывод данных через перегрузку (\_\_repr\_\_)**

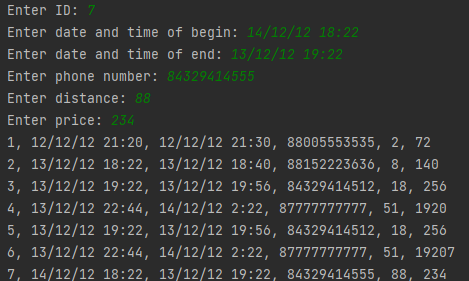
****

**4 рис. Сортировка по Дате начала поездки**

****

**5 рис. Сортировка по номеру телефона**

****

**6 рис. Сортировка по критерию **

**7 рис. Добавление новой записи в словарь**

****

**8 рис. Получение доступа к элементу коллекции через \_\_getitem\_\_**

**Код программы**

import csv  
  
  
class Parentlist():  
 def \_\_init\_\_(self, key\_inp, date1\_inp, date2\_inp, phone\_inp, distance\_inp, price\_inp):  
 self.key\_inp = key\_inp  
 self.date1\_inp = date1\_inp  
 self.date2\_inp = date2\_inp  
 self.phone\_inp = phone\_inp  
 self.distance\_inp = distance\_inp  
 self.price\_inp = price\_inp  
  
 def \_\_str\_\_(self):  
 return f"{self.key\_inp}, {self.date1\_inp}, {self.date2\_inp}, {self.phone\_inp}, {self.distance\_inp}, " \  
 f"{self.price\_inp}"  
  
 def \_\_repr\_\_(self):  
 return f"{self.key\_inp}, {self.date1\_inp}, {self.date2\_inp}, {self.phone\_inp}, {self.distance\_inp}, " \  
 f"{self.price\_inp}"  
  
  
class l1st(Parentlist):  
 def \_\_init\_\_(self, path):  
 self.lengh = None  
 self.path = path  
 self.dictionary = self.fileopen(self.path)  
  
 def \_\_iter\_\_(self):  
 return iter(self.dictionary)  
  
 def \_\_next\_\_(self):  
 if self.lengh <= len(self.dictionary):  
 x = self.lengh  
 self.lengh += 1  
 return x  
 else:  
 raise StopIteration  
  
 def \_\_str\_\_(self): #перегрузка \_\_str\_\_ и \_\_repr\_\_  
 return '' + '\n'.join([repr(line) for line in self.dictionary])  
  
 def \_\_repr\_\_(self):  
 return '' + '\n'.join([repr(line) for line in self.dictionary])  
  
 def \_\_setattr\_\_(self, key, value):  
 self.\_\_dict\_\_[key] = value  
  
 def \_\_getitem\_\_(self, item):  
 if 0 <= item < len(self.dictionary):  
 return self.dictionary[item]  
 else:  
 raise IndexError("Wrong index")  
  
 def generator(self): # генератор коллекции  
 self.lengh = 0  
 while self.lengh < len(self.dictionary):  
 yield self.dictionary[self.lengh]  
 self.lengh += 1  
  
 @staticmethod  
 def fileopen(path: str) -> list: #чтение из файла  
 mas = []  
  
 with open(path, "r") as f:  
 for line in f:  
 (key\_inp, date1\_inp, date2\_inp, phone\_inp, distance\_inp, price\_inp) = line.replace("\n", "").split(",")  
 mas.append(Parentlist(key\_inp, date1\_inp, date2\_inp, phone\_inp, distance\_inp, price\_inp))  
 return mas  
  
 def add(self, key\_inp, date1\_inp, date2\_inp, phone\_inp, distance\_inp, price\_inp): # добаление записи о новой поездке  
 self.dictionary.append(Parentlist(key\_inp, date1\_inp, date2\_inp, phone\_inp, distance\_inp, price\_inp))  
 newline = [key\_inp, date1\_inp, date2\_inp, phone\_inp, distance\_inp, price\_inp]  
 with open('data.csv', 'a+', newline='') as f:  
 writer = csv.writer(f)  
 writer.writerow(newline)  
 return self.dictionary  
  
 def sortdate(self) -> list: # сортировка по дате начала  
 return sorted(self.dictionary, key=lambda x: x.date1\_inp)  
  
 def sortphone(self) -> list :# сортировка по номеру телефона  
 return sorted(self.dictionary, key=lambda x: x.phone\_inp)  
  
 def sortcriterium(self) -> list: # сортировка по критерию  
 return [x for x in self.dictionary if x.distance\_inp == "18"]  
  
  
dictionary = l1st('data.csv')  
  
print("Default mas")  
for x in dictionary:  
 print(x)  
print('')  
  
print("Iterator")  
iterator = iter(dictionary)  
print(next(iterator))  
print(next(iterator))  
  
print("\nGenerator")  
for x in dictionary.generator():  
 print(x)  
  
  
print("\n\_\_str\_\_ ", dictionary, sep='\n')  
  
print("\n\_\_repr\_\_ ", repr(dictionary), sep='\n')  
  
print("\nSorting by date of begin")  
for x in dictionary.sortdate():  
 print(x)  
  
print("\nSorting by phone")  
for x in dictionary.sortphone():  
 print(x)  
  
print("\nSorting by criterium")  
for x in dictionary.sortcriterium():  
 print(x)  
  
print(dictionary.\_\_getitem\_\_(int(input("Enter line pos: "))))  
  
dictionary.add(input('Enter ID: '), input('Enter date and time of begin: '), input('Enter date and time of end: '),  
 input('Enter phone number: '), input('Enter distance: '), input('Enter price: '))  
for x in dictionary:  
 print(x)  
print('')