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

Выполнила:

студентка группы ИВТАПбд-22

Ренжина А.С.

Проверил:

преподаватель кафедры

«Вычислительная техника»

Исхаков И.И.

Ульяновск, 2023

**Задание:**

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

Описание реализации:

Для работы с файлами, создаем общий класс, от которого будут наследоваться остальные.

В нем прописываем основные функции - класс file

Cоздаем дочерние классы - txtFile и csvFile - для работы с текстовыми файлами и csv файлами.

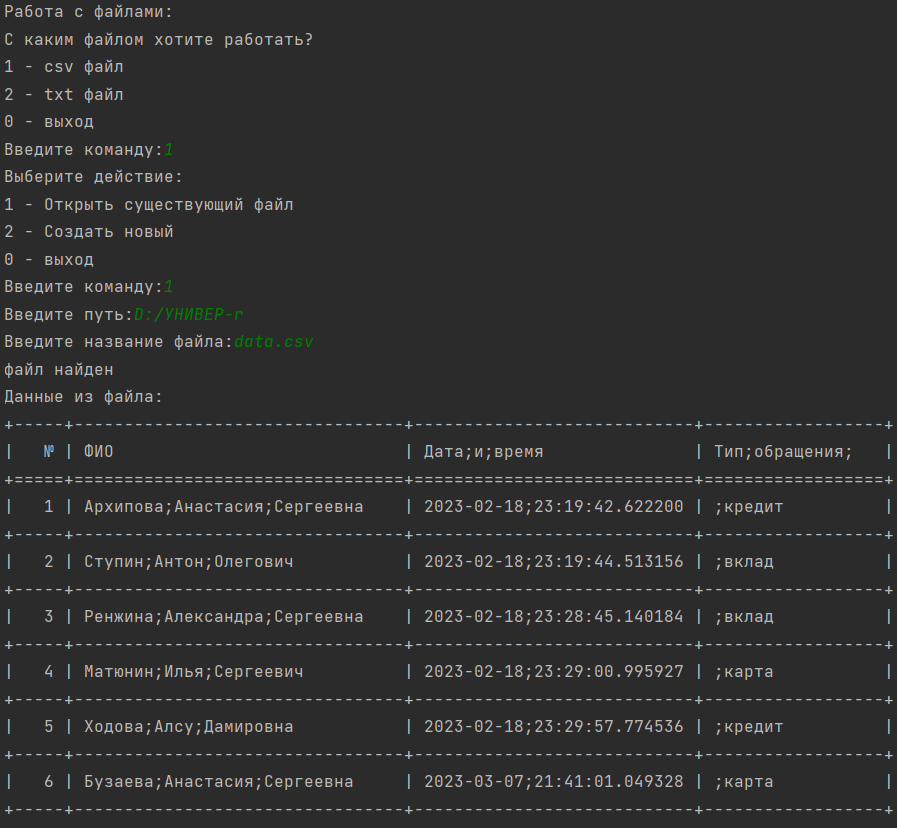
Для работы с csv файлами, все функции из лабораторной номер 3, с добавлением записи ключей, и записи данных в файл любого содержания.

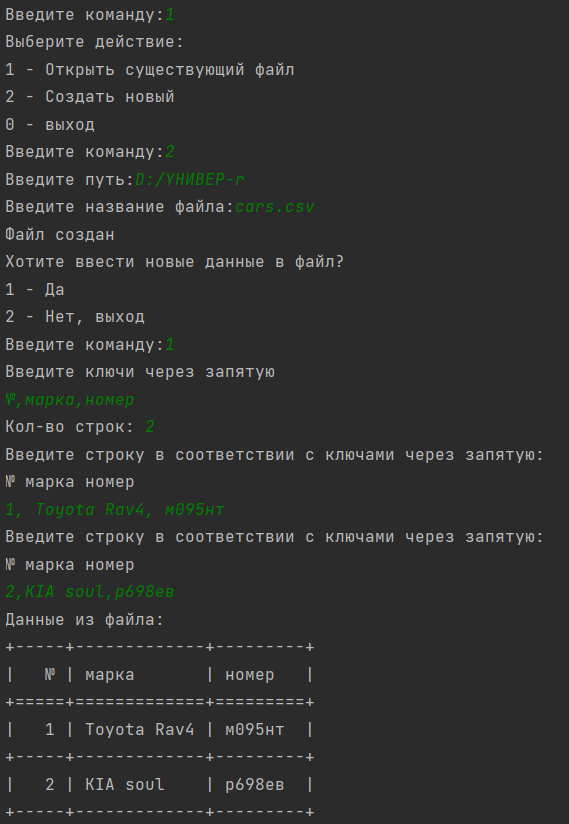
Для работы с текстовыми файлами, создаем функции для чтения из файла, записи в файл и вывода данных. Все функции схожи с функциями из лабораторной номер 3, но с изменением под текстовый документ.

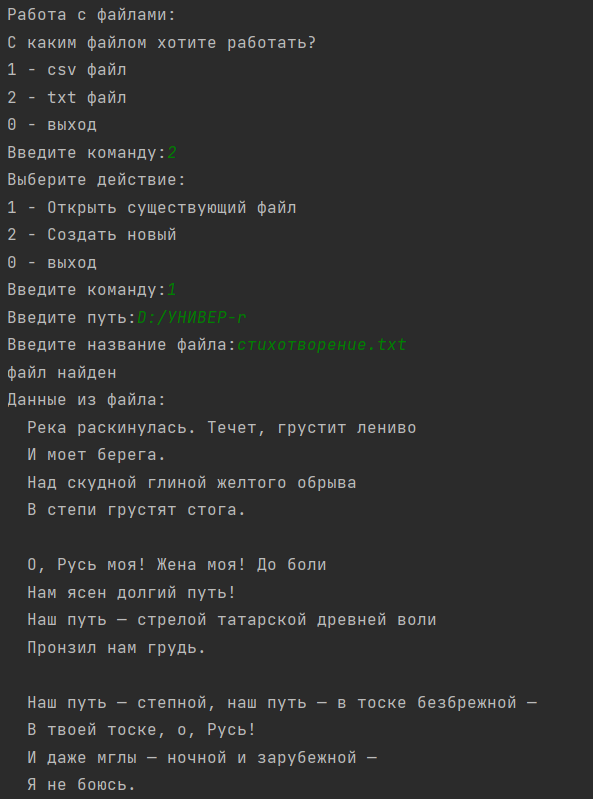
В функции main(), также реализовано мини консольное приложение, с выбором работы над каким файлом - чтение или запись.

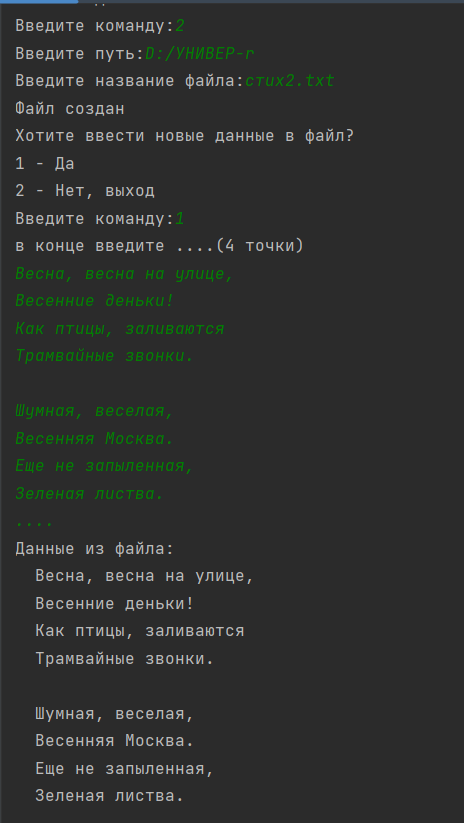
Тестовые примеры.

1. Работа с csv файлом.





1. Работа с текстовыми файлами:



Исходный код:

файл File.py

import os

class File:

def \_\_init\_\_(self):

self.\_path = ""

self.\_expansions = [".txt", ".csv"]

self.\_exist = False

self.\_isOpen = False

def get\_expansion(self, name):

for expansion in self.\_expansions:

if expansion in name:

return expansion

def get\_path(self):

return self.\_path

def set\_path(self, path):

self.\_path = path

def is\_exist(self, fpath, name):

if os.path.exists(fpath + f"/{name}"):

self.\_exist = True

self.\_path = fpath + f"/{name}"

return True

return False

def open\_file(self):

self.\_isOpen = True

def is\_open(self):

return self.\_isOpen

def right\_exp(self, name):

for expansion in self.\_expansions:

if expansion in name:

return True

return False

def get\_exp(self, file\_name):

for expansion in self.\_expansions:

if expansion in file\_name:

return expansion

def create\_file(self, fpath, name):

if self.right\_exp(name):

self.\_path = f"{fpath}/{name}"

f = open(self.\_path, 'w+')

print("Файл создан")

f.close()

файл txtFile.py

import file

class TxtFile(file.File):

data = []

def get\_data(self):

with open(self.get\_path(), "r",encoding="utf-8") as f:

for line in f:

self.data.append(line)

f.close()

def set\_data(self, data):

with open(self.get\_path(), "w") as f:

f.writelines("%s\n" % row for row in data)

f.close()

def print\_data(self):

print("Данные из файла:")

s = ""

for row in self.data:

if row.\_\_contains\_\_("\n"):

print(f"{s:>2}{row}", end='')

else:

print(f"{s:>2}{row}")

print()

@staticmethod

def add\_new\_data\_to\_txt\_file(data):

print("в конце введите ....(4 точки)")

flag = True

while flag:

s = input()

if s == "....":

flag = False

else:

data.append(s)

файл csvFile.py

import csv

import pandas as pd

from tabulate import tabulate

import file

class CsvFile(file.File):

data = []

keys = []

def get\_data(self):

self.data, keys = [], []

with open(self.get\_path(), "r", encoding="cp1251") as f:

reader = csv.DictReader(f)

self.keys = reader.fieldnames

for row in reader:

self.data.append(row)

f.close()

def set\_data(self, data, keys):

with open(self.get\_path(), "w", encoding="cp1251", newline="\n") as f:

writer = csv.DictWriter(f, fieldnames=keys)

writer.writeheader()

writer.writerows(data)

f.close()

@staticmethod

def add\_new\_data(data, keys):

n = int(input("Кол-во строк: "))

for v in range(len(data), len(data) + n):

print("Введите строку в соответствии с ключами через запятую:")

for key in keys:

print(key, end=" ")

print()

input\_row = input().split(",")

if len(input\_row) == len(keys):

new\_row = {}

for i in range(len(keys)):

new\_row[keys[i]] = input\_row[i]

data.append(new\_row)

else:

print("Ошибка!")

@staticmethod

def print\_data(data, keys):

print("Данные из файла:")

flag = True

while flag:

if len(data) != 0:

print(tabulate(pd.DataFrame(data), headers=keys, tablefmt='grid', showindex=False))

flag = False

else:

print("Пустой файл")

def write\_keys(self):

key = []

print("Введите ключи через запятую")

key = input().split(",")

if len(key) == 0:

print("Вы ничего не ввели!")

self.keys = key

файл main.py

import os

import csvFile

import txtFile

@staticmethod

def director(path):

if os.path.exists(path):

print("папка найдена")

return path

else:

print("Ошибка: такой папки не существует.\n"

"1 - попробовать ещё раз\n"

"0 - выход")

op = int(input("введите команду: "))

if op == 1:

director()

if op == 0:

return

@staticmethod

def file\_set(path, file\_name):

file = f"{path}/{file\_name}"

if os.path.exists(file):

print("файл найден")

return file

else:

print("Ошибка: такой файл не существует.\n"

"1 - попробовать ещё раз\n"

"0 - выход")

op = int(input("введите команду: "))

if op == 1:

file = file\_set(path, file\_name)

return file

if op == 0:

return

def main():

print("Работа с файлами:\n"

"С каким файлом хотите работать?\n"

"1 - csv файл\n"

"2 - txt файл\n"

"0 - выход")

oper = int(input("Введите команду:"))

match(oper):

case 1:

print("Выберите действие:\n"

"1 - Открыть существующий файл\n"

"2 - Создать новый\n"

"0 - выход")

oper = int(input("Введите команду:"))

match(oper):

case 1:

file1 = csvFile.CsvFile()

path = input("Введите путь:")

file\_name = input("Введите название файла:")

full\_path = file\_set(path, file\_name)

file1.set\_path(full\_path)

file1.get\_data()

file1.print\_data(file1.data, file1.keys)

print("Хотите ввести новые данные в файл?\n"

"1 - Да\n"

"2 - Нет, выход")

oper = int(input("Введите команду:"))

match(oper):

case 1:

file1.add\_new\_data(file1.data, file1.keys)

file1.set\_data(file1.data, file1.keys)

file1.print\_data(file1.data, file1.keys)

case 2:

exit(1)

case 2:

file2 = csvFile.CsvFile()

path = input("Введите путь:")

file\_name = input("Введите название файла:")

file2.create\_file(path, file\_name)

print("Хотите ввести новые данные в файл?\n"

"1 - Да\n"

"2 - Нет, выход")

oper = int(input("Введите команду:"))

match (oper):

case 1:

file2.write\_keys()

file2.add\_new\_data(file2.data, file2.keys)

file2.set\_data(file2.data, file2.keys)

file2.print\_data(file2.data, file2.keys)

case 2:

exit(1)

case 0:

exit(1)

case 2:

print("Выберите действие:\n"

"1 - Открыть существующий файл\n"

"2 - Создать новый\n"

"0 - выход")

oper = int(input("Введите команду:"))

match (oper):

case 1:

file\_txt = txtFile.TxtFile()

path1 = input("Введите путь:")

file\_name1 = input("Введите название файла:")

full\_path1 = file\_set(path1, file\_name1)

file\_txt.set\_path(full\_path1)

file\_txt.get\_data()

file\_txt.print\_data()

case 2:

file\_txt2 = txtFile.TxtFile()

path1 = input("Введите путь:")

file\_name1 = input("Введите название файла:")

file\_txt2.create\_file(path1, file\_name1)

print("Хотите ввести новые данные в файл?\n"

"1 - Да\n"

"2 - Нет, выход")

oper = int(input("Введите команду:"))

match (oper):

case 1:

file\_txt2.add\_new\_data\_to\_txt\_file(file\_txt2.data)

file\_txt2.set\_data(file\_txt2.data)

file\_txt2.print\_data()

case 2:

exit(1)

case 0:

exit(1)

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

main()

**Описание возникших затруднений:**

При выполнении лабораторной работы, затруднений не возникло.

**Описание альтернативных способ решения:**

Альтернативных способ решения при выполнении лабораторной работы не возникло.

**Вывод о проделанной работе:**

Во время выполнения лабораторной работы, познакомились с классами, наследованием и статическими методами, библиотеками, которые работают с файлами.