

МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ   
РОССИЙСКОЙ ФЕДЕРАЦИИ

**ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ БЮДЖЕТНОЕ**

**ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ  
«ДОНСКОЙ ГОСУДАРСТВЕННЫЙ ТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ» (ДГТУ)**

Факультет: Информатика и вычислительная техника

Кафедра: Кибербезопасность информационных систем

**Лабораторная работ №5**

Выполнил обучающийся гр.ВКБ21

Юровского Никиты Владимировича

(Фамилия, Имя, Отчество)

Проверила:

Ст.Пр. Жданова Марина Михайловна

(должность, Фамилия, Имя, Отчество)

Ростов-на-Дону

2024

**Лабораторная работа №5**

**Тема: «ФАЙЛЫ И ФАЙЛОВАЯ СИСТЕМА.**

**ПОЛЬЗОВАТЕЛЬСКИЕ ФУНКЦИИ»**

Вариант 2

1. Пусть дана некоторая директория (папка). Посчитайте количество файлов в данной директории (папке) и выведите на экран.

Решение:

import os  
  
dir\_path = r'C:/Users/user/Documents'  
files = []  
  
for f in os.listdir(dir\_path):  
 if os.path.isfile(os.path.join(dir\_path, f)):  
 files.append(f)  
  
print(f"Кол-во файлов: {len(files)}")  
  
for f in files:  
 print(f'- {f}')

1. Пусть дан файл students.csv, в котором содержится информация о студентах в виде:

№;ФИО;Возраст;Группа

1;Иванов Иван Иванович;23;БО-111111

2;Сидоров Семен Семенович;23;БО-111111

3;Яшков Илья Петрович;24;БО-222222

...

Считайте информацию из файла в структуру: [[No, ФИО, Возраст, Группа],[No, ФИО, Возраст, Группа],[No, ФИО, Возраст, Группа]] (список списков). Выведите информацию о студентах, отсортировав их по возрасту.

Решение:

# Запись в файл  
content = ('№;ФИО;Возраст;Группа\n'  
 '1;Иванов Иван Иванович;23;БО-111111\n'  
 '2;Сидоров Семен Семенович;23;БО-111111\n'  
 '3;Яшков Илья Петрович;24;БО-222222\n')  
  
with open('students.csv', 'w') as file\_object:  
 file\_object.write(content)  
  
# Чтение файла  
with open('students.csv') as a:  
 v = a.read()  
  
print(v)  
  
# Считаем информацию в опр. структуру  
with open('students.csv') as file:  
 students = []  
 titles = file.readlines()[1:]  
 for line in titles:  
 students.append(line.replace('\n', '').split(';'))  
  
  
print(students)  
  
# Сортировка по возрасту студентов  
print("Сортировка по возрасту:")  
student\_by\_age = sorted(students, key=lambda x: int(x[2]))  
for s in student\_by\_age:  
 print(s)

1. Добавьте к задаче №2 пользовательский интерфейс по уменьшению возраста всех студентов на 1.
2. Добавьте к пользовательскому интерфейсу из задачи №3 возможность сохранения новых данных обратно в файл.

Решение:

import tkinter as tk  
from tkinter import ttk, messagebox, simpledialog  
import csv  
  
with open('students.csv', 'w', encoding='utf-8') as file:  
 file.write('№;ФИО;Возраст;Группа\n')  
 file.write('1;Иванов Иван Иванович;23;БО-111111\n')  
 file.write('2;Сидоров Семен Семенович;23;БО-111111\n')  
 file.write('3;Яшков Илья Петрович;24;БО-222222\n')  
  
# функция для чтения студентов из файла  
  
  
def read\_students():  
 students = []  
 with open('students.csv', mode='r', encoding='utf-8') as files:  
 reader = csv.reader(files, delimiter=';')  
 next(reader)  
 for row in reader:  
 if len(row) == 4:  
 students.append(row)  
 return students  
  
  
# функция для отображения студентов в таблице  
def display\_students(students):  
 for row in tree.get\_children():  
 tree.delete(row)  
  
 for student in students:  
 tree.insert('', tk.END, values=student)  
  
  
# Функция для сортировки студентов по возрасту и отображения результата  
def sort\_students\_by\_age():  
 students = read\_students()  
 sorted\_students = sorted(students, key=lambda x: int(x[2]))  
 display\_students(sorted\_students)  
  
  
# Функция для увеличения возраста всех студентов на 1 и обновления таблицы  
  
  
def decrease\_age():  
 students = read\_students()  
 for student in students:  
 age = int(student[2])  
 student[2] = str(age - 1)  
  
 display\_students(students)  
  
   
def add\_student():  
 # Открываем диалоговое окно для ввода данных студента  
 fio = simpledialog.askstring("Ввод ФИО", "Введите ФИО студента:")  
 age = simpledialog.askinteger("Ввод возраста", "Введите возраст студента:")  
 group = simpledialog.askstring("Ввод группы", "Введите группу студента:")  
  
 if fio and age is not None and group:  
 # Генерируем номер студента (количество строк в файле + 1)  
 students = read\_students()  
 student\_number = len(students) + 1  
  
 # Записываем нового студента в файл  
 with open('students.csv', 'a', encoding='utf-8') as files:  
 writer = csv.writer(files, delimiter=';')  
 writer.writerow([student\_number, fio, age, group])  
  
 # Обновляем таблицу  
 display\_students(read\_students())  
 else:  
 messagebox.showwarning("Ошибка", "Все поля должны быть заполнены!")  
  
  
# Функция для сохранения данных в файле  
def save\_to\_new\_file():  
 new\_file\_name = simpledialog.askstring("Сохранение", "Введите имя нового файла (с расширением .csv):")  
  
 if new\_file\_name:  
 students = read\_students()  
 with open(new\_file\_name, 'w', encoding='utf-8') as file:  
 writer = csv.writer(file, delimiter=';')  
 writer.writerow(['№', 'ФИО', 'Возраст', 'Группа']) # Записываем заголовок  
 writer.writerows(students) # Записываем данные студентов  
  
 messagebox.showinfo("Сохранение", f"Данные успешно сохранены в файл {new\_file\_name}")  
 else:  
 messagebox.showwarning("Ошибка", "Имя файла не может быть пустым!")  
  
  
# Создание основного окна Tkinter  
root = tk.Tk()  
root.title("Таблица студентов")  
  
# Создание виджета Treeview  
tree = ttk.Treeview(root, columns=('№', 'ФИО', 'Возраст', 'Группа'), show='headings')  
tree.heading('№', text='№')  
tree.heading('ФИО', text='ФИО')  
tree.heading('Возраст', text='Возраст')  
tree.heading('Группа', text='Группа')  
  
tree.column('№', width=50)  
tree.column('ФИО', width=200)  
tree.column('Возраст', width=100)  
tree.column('Группа', width=100)  
  
tree.pack(fill=tk.BOTH, expand=True)  
  
# Создание кнопок  
btn\_sort\_age = tk.Button(root, text="Сортировать по возрасту", command=sort\_students\_by\_age)  
btn\_sort\_age.pack(pady=5)  
  
btn\_decrease\_age = tk.Button(root, text="Уменьшить возраст на 1", command=decrease\_age)  
btn\_decrease\_age.pack(pady=5)  
  
btn\_add\_student = tk.Button(root, text="Добавить студента", command=add\_student)  
btn\_add\_student.pack(pady=5)  
  
save\_button = tk.Button(root, text="Сохранить данные", command=save\_to\_new\_file)  
save\_button.pack(pady=10)  
  
# Изначально отображаем студентов в таблице  
display\_students(read\_students())  
  
  
root.mainloop()