|  |  |
| --- | --- |
| **Gerb-BMSTU_01** | **Министерство науки и высшего образования Российской Федерации**  Калужский филиал  федерального государственного бюджетного  образовательного учреждения высшего образования  ***«Московский государственный технический университет имени Н.Э. Баумана (национальный исследовательский университет)»***  ***(КФ МГТУ им. Н.Э. Баумана)*** |

**ФАКУЛЬТЕТ** \_***ИУК «Информатика и управление»*\_\_**\_\_\_\_\_\_\_\_\_\_\_\_

**КАФЕДРА** \_\_***ИУК4 «Программное обеспечение ЭВМ, информационные технологии»***

**ДОМАШНЯЯ РАБОТА №2**

**«Работа с файлами»**

**ДИСЦИПЛИНА: «Перспективные языки программирования»**

|  |  |  |
| --- | --- | --- |
| Выполнил: студент гр. ИУК4-32Б | | \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ (\_\_\_\_\_\_Зудин Д.В.\_\_\_\_\_)  (Подпись) (Ф.И.О.) |
| Проверил: | | \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ (\_\_\_Пчелинцева Н.И.\_\_\_)  (Подпись) (Ф.И.О.) |
| Дата сдачи (защиты):  Результаты сдачи (защиты): | | |
|  | - Балльная оценка:  - Оценка: | |
| Калуга, 2022 г.  **Цель**: приобретение практических навыков работы с файлами и файловой системой на языке программирования Python.  **Задачи**:  1. Ознакомиться со способами работы с файлами и файловой системой в Python;  2. Изучить способы работы с файлами формата csv, json, xml;  3. Закрепить полученные в ходе выполнения домашней работы навыки.  **Вариант №20**  **Формулировка задания №1**  Выберите любую папку на своём компьютере, имеющую вложенные директории.  Выведите на печать в терминал и сохраните в файл txt названия всех файлов, которые содержат введенное с консоли слово. Заархивируйте данную папку средствами python.  **Листинг программы для задания №1**  *import* os *import* shutil  print("Поиск в папке Test файлов, содержащих в названии введенное слово") string = input("Введите слово: ") path = 'C:/Users/user/OneDrive/Рабочий стол/3-ий семестр/Перспективные языки программирования/Test' res = [] *for* current\_dir, dirs, files *in* os.walk(path):  *for* i *in* files:  *if* i.find(string) != -1:  res.append(i)  *with* open("Task\_1.txt", "w") *as* f:  f.write('\n'.join(res))  print("Искомые файлы:") print('\n'.join(res))  shutil.make\_archive("Test", 'zip', path)  **Результат выполнения программы для задания №1**  1. Исходная папка до выполнения программы    Некоторые папки также содержат файлы, но есть и пустые.  2. Вывод в консоль  Поиск в папке Test файлов, содержащих в названии введенное слово  Введите слово: Python  Искомые файлы:  Python\_32.txt  Python\_1.txt  Python\_2.txt  Python.txt  Python\_1.txt  3. Текстовый файл «Task\_1.txt»  Python\_32.txt Python\_1.txt Python\_2.txt Python.txt Python\_1.txt  4. Исходная папка после выполнения программы    **Формулировка задания №2**  Создать файл в формате csv, содержащий сведения о покупателях в интернет-магазине. Структура: id, ФИО, дата рождения, общая сумма покупок, количество заказов.  Предусмотреть возможность корректировки файла по вводимому коду корректировки, например:  1 – удалить запись (по id);  2 – добавить новую запись;  3 – изменить запись (по id);  4 – получить информацию (по id);  5 – отсортировать покупателей по:   * ФИО * Дате рождения * Общей сумме покупок * Количеству заказов   6 – сохранить в отдельный файл и вывести на консоль покупателя с минимальной суммой покупок и покупателя с минимальным количеством заказов.  **Листинг файла shoppers.py**  *class* Shopper:  *def \_\_init\_\_*(*self*):  *"""Инициализация покупателя"""  self*.\_\_shopper\_id = 0 *# ID  self*.\_\_name = '' *# ФИО  self*.\_\_date\_of\_birthday = '' *# Дата рождения  self*.\_\_total = 0 *# Общая сумма покупок  self*.\_\_orders = 0 *# Количество заказов   def \_\_str\_\_*(*self*):  *"""Форматирование вывода информации о покупателе"""  return* f"ID: {*self*.\_\_shopper\_id}\n" \  f"ФИО: {*self*.\_\_name}\n" \  f"Дата рождения: {*self*.\_\_date\_of\_birthday}\n" \  f"Общая сумма покупок: {*self*.\_\_total} р.\n"\  f"Количество заказов: {*self*.\_\_orders} шт."   @property  *def* shopper\_id(*self*):  *return self*.\_\_shopper\_id   @shopper\_id.setter  *def* shopper\_id(*self*, shopper\_id: int):  *if* shopper\_id > 0:  *self*.\_\_shopper\_id = shopper\_id  *else*:  *raise* ValueError("Incorrect input!")   @property  *def* name(*self*):  *return self*.\_\_name   @name.setter  *def* name(*self*, name: str):  *self*.\_\_name = name   @property  *def* date\_of\_birthday(*self*):  *return self*.\_\_date\_of\_birthday   @date\_of\_birthday.setter  *def* date\_of\_birthday(*self*, date\_of\_birthday: str):  valid = *True  if* len(date\_of\_birthday) != 10 *and* date\_of\_birthday[2] != '.' *and* date\_of\_birthday[5] != '.':  valid = *False  if not* 0 < int(date\_of\_birthday[:2]) < 32:  valid = *False  if not* 0 < int(date\_of\_birthday[3:5]) < 13:  valid = *False  if not* 1900 < int(date\_of\_birthday[6:10]) < 2012:  valid = *False   if* valid:  *self*.\_\_date\_of\_birthday = date\_of\_birthday  *else*:  *raise* ValueError("Incorrect input!")   @property  *def* total(*self*):  *return self*.\_\_total   @total.setter  *def* total(*self*, total: float):  *if* total > 0:  *self*.\_\_total = total  *else*:  *raise* ValueError("Incorrect input!")   @property  *def* orders(*self*):  *return self*.\_\_orders   @orders.setter  *def* orders(*self*, orders: int):  *if* orders > 0:  *self*.\_\_orders = orders  *else*:  *raise* ValueError("Incorrect input!")  **Листинг программы для задания №2**  *import* os *import* csv *from* enum *import* Enum *from* shoppers *import* Shopper  *class* Commands(Enum):  DELETE\_SHOPPER = 1  ADD\_SHOPPER = 2  CHANGE\_SHOPPER = 3  GET\_SHOPPER\_INFO = 4  SORT\_SHOPPERS\_NAME = 5  SORT\_SHOPPERS\_DATE\_OF\_BIRTHDAY = 6  SORT\_SHOPPERS\_TOTAL = 7  SORT\_SHOPPERS\_ORDERS = 8  SAVE\_IN\_CSV\_FILE = 9  LOAD\_FROM\_CSV\_FILE = 10  PRINT\_SHOPPERS = 11  INDIVIDUAL\_TASK = 12  EXIT = 0   *def* input\_int(message: str, start: int, end: int) -> int:  valid = *False* n = 0  *while not* valid:  *try*:  n = int(input(message))  *except* ValueError:  print('Вы ввели не число. Попробуйте снова.')  *else*:  *if* start <= n <= end:  valid = *True  else*:  print('Введённое число вне диапазона!')  *return* n   *def* input\_shopper(data: list[Shopper]) -> Shopper:  *"""Добавление записи о покупателе"""* new\_shopper = Shopper()  new\_id, valid\_id = -1, *False  while not* valid\_id:  new\_id = int(input("Введите ID покупателя: "))  valid\_id = new\_id *not in* list(i.shopper\_id *for* i *in* data)  *if not* valid\_id:  print("ID должен быть уникальным!")  valid = *False  while not* valid:  *try*:  new\_shopper.shopper\_id = new\_id  new\_shopper.name = input("Введите ФИО: ")  new\_shopper.date\_of\_birthday = input("Введите дату рождения: ")  new\_shopper.total = int(input("Введите общую сумму покупок: "))  new\_shopper.orders = int(input("Введите количество заказов: "))  *except* ValueError:  print("Неккоректный ввод! Попробуйте снова.")  *except* IndexError:  print("Неккоректный ввод! Попробуйте снова.")  *else*:  valid = *True  return* new\_shopper   *def* m\_print\_shoppers(data: list[Shopper]):  *"""Вывод списка покупателей"""  if not* data:  print("Список пустой!")  *else*:  print("Покупатели:")  *for* i *in* data:  print(i)  print()   *def* m\_add\_shopper(data: list[Shopper]):  *"""Добавление покупателя"""* data.append(input\_shopper(data))   *def* m\_delete\_shopper(data: list[Shopper]):  *"""Удаление покупателя"""  if not* data:  print("Список пустой!")  *else*:  shopper\_id = int(input("Введите ID удаляемого покупателя: "))  index = -1  *for* i *in* range(len(data)):  *if* data[i].shopper\_id == shopper\_id:  index = i  *break  if* index != -1:  data.pop(index)  print(f"Покупатель с ID {shopper\_id} удалён")  *else*:  print("Покупателя с таким ID не существует!")   *def* m\_change\_shopper(data: list[Shopper]):  *"""Изменение информации о покупателе"""  if not* data:  print("Список пустой!")  *else*:  shopper\_id = int(input("Введите ID изменяемого покупателя "))  index = -1  *for* i *in* range(len(data)):  *if* data[i].shopper\_id == shopper\_id:  index = i  *break  if* index != -1:  data[index] = input\_shopper(data)  *else*:  print("Покупателя с таким ID не существует!")   *def* m\_get\_shopper\_info(data: list[Shopper]):  *"""Получить информацию о покупателе"""  if not* data:  print("Список пустой!")  *else*:  shopper\_id = int(input("Введите ID просматриваемого покупателя: "))  find\_shopper = *False  for* i *in* data:  *if* i.shopper\_id == shopper\_id:  find\_shopper = *True* print(i)  *if not* find\_shopper:  print("Покупателя с таким ID не существует!")   *def* m\_sort\_shoppers\_name(data: list[Shopper]):  *"""Сортировка по имени"""* data.sort(key=*lambda* x: x.name)  print("Покупатели отсортированы по ФИО")   *def* m\_sort\_shoppers\_date\_of\_birthday(data: list[Shopper]):  *"""Сортировка по дате рождения"""* data.sort(key=*lambda* x: x.date\_of\_birthday)  print("Покупатели отсортированы по дате рождения")   *def* m\_sort\_shoppers\_total(data: list[Shopper]):  *"""Сортировка по общей сумме покупок"""* data.sort(key=*lambda* x: x.total)  print("Покупатели отсортированы по общей сумме покупок")   *def* m\_sort\_shoppers\_orders(data: list[Shopper]):  *"""Сортировка по количеству заказов"""* data.sort(key=*lambda* x: x.orders)  print("Покупатели отсортированы по количеству заказов")   *def* m\_save\_in\_csv\_file(data: list[Shopper], data\_file: str):  *"""Сохранение данных в csv-файл"""  with* open(data\_file, 'w', encoding='utf8', newline='') *as* file:  writer = csv.writer(file)  *for* i *in* data:  *if* i:  writer.writerow([i.shopper\_id, i.name, i.date\_of\_birthday, i.total, i.orders])  print("Данные успешно сохранены в файл")   *def* m\_load\_from\_csv\_file(data: list[Shopper], data\_file: str):  *"""Загрузка данных из csv-файла"""  with* open(data\_file, 'r', encoding='utf8') *as* file:  reader = csv.reader(file)  *for* curr\_shopper *in* reader:  *if* curr\_shopper:  shopper = Shopper()  print(curr\_shopper)  shopper.shopper\_id = int(curr\_shopper[0])  shopper.name = curr\_shopper[1]  shopper.date\_of\_birthday = curr\_shopper[2]  shopper.total = int(curr\_shopper[3])  shopper.orders = int(curr\_shopper[4])  data.append(shopper)  print("Данные успешно загружены из файла")   *def* m\_individual\_task(data: list[Shopper], data\_file: str):  *"""Индивидуальное задание"""* m\_sort\_shoppers\_total(data)  min\_total = data[0]  m\_sort\_shoppers\_orders(data)  min\_orders = data[0]   *with* open(data\_file, 'w', encoding='utf8', newline='') *as* file:  writer = csv.writer(file)  writer.writerow([min\_total.shopper\_id, min\_total.name, min\_total.date\_of\_birthday,  min\_total.total, min\_total.orders])  writer.writerow([min\_orders.shopper\_id, min\_orders.name, min\_orders.date\_of\_birthday,  min\_orders.total, min\_orders.orders])  print("Информация о покупателе с минимальной общей суммой заказа загружена в файл")  print("Информация о покупателе с минимальным количеством заказов загружена в файл")   *def* print\_menu():  *"""Меню"""* print(f" 1. Удалить запись по id\n"  f" 2. Добавить новую запись\n"  f" 3. Изменить запись по id\n"  f" 4. Получить информацию по id\n"  f" 5. Сортировка по ФИО\n"  f" 6. Сортировка по дате рождения\n"  f" 7. Сортировка по общей сумме покупок\n"  f" 8. Сортировка по количеству заказов\n"  f" 9. Сохранить данные в файл\n"  f"10. Загрузить данные из файла\n"  f"11. Вывести данные\n"  f"12. Индивидуальное задание\n"  f" 0. Выход")   *def* main():  *"""Основная логика программы"""* \_exit = *False* data\_file = 'data.csv'  individual\_task\_file = 'task.csv'  data = []  *while not* \_exit:  os.system('cls')  print\_menu()  command = input\_int("Введите команду: ", 0, 12)  *if* command == Commands.DELETE\_SHOPPER.value:  m\_delete\_shopper(data)  *elif* command == Commands.ADD\_SHOPPER.value:  m\_add\_shopper(data)  *elif* command == Commands.CHANGE\_SHOPPER.value:  m\_change\_shopper(data)  *elif* command == Commands.GET\_SHOPPER\_INFO.value:  m\_get\_shopper\_info(data)  *elif* command == Commands.SORT\_SHOPPERS\_NAME.value:  m\_sort\_shoppers\_name(data)  *elif* command == Commands.SORT\_SHOPPERS\_DATE\_OF\_BIRTHDAY.value:  m\_sort\_shoppers\_date\_of\_birthday(data)  *elif* command == Commands.SORT\_SHOPPERS\_TOTAL.value:  m\_sort\_shoppers\_total(data)  *elif* command == Commands.SORT\_SHOPPERS\_ORDERS.value:  m\_sort\_shoppers\_orders(data)  *elif* command == Commands.SAVE\_IN\_CSV\_FILE.value:  m\_save\_in\_csv\_file(data, data\_file)  *elif* command == Commands.LOAD\_FROM\_CSV\_FILE.value:  m\_load\_from\_csv\_file(data, data\_file)  *elif* command == Commands.PRINT\_SHOPPERS.value:  m\_print\_shoppers(data)  *elif* command == Commands.INDIVIDUAL\_TASK.value:  m\_individual\_task(data, individual\_task\_file)  *elif* command == Commands.EXIT.value:  \_exit = *True  if not* \_exit:  os.system('pause') *if* \_\_name\_\_ == '\_\_main\_\_':  main()  **Результат выполнения программы для задания №2**  **Меню**  1. Удалить запись по id  2. Добавить новую запись  3. Изменить запись по id  4. Получить информацию по id  5. Сортировка по ФИО  6. Сортировка по дате рождения  7. Сортировка по общей сумме покупок  8. Сортировка по количеству заказов  9. Сохранить данные в файл  10. Загрузить данные из файла  11. Вывести данные  12. Индивидуальное задание  0. Выход  **Добавление покупателя**  Введите команду: 2  Введите ID покупателя: 1  Введите ФИО: Зудин Даниил Васильевич  Введите дату рождения: 25.01.2003  Введите общую сумму покупок: 5000  Введите количество заказов: 5  Для продолжения нажмите любую клавишу . . .  Введите команду: 2  Введите ID покупателя: 2  Введите ФИО: Исаев Станислав Алексеевич  Введите дату рождения: 25.06.2003  Введите общую сумму покупок: 1200  Введите количество заказов: 7  Для продолжения нажмите любую клавишу . . .  Введите команду: 2  Введите ID покупателя: 3  Введите ФИО: Хрущев Никита Сергеевич  Введите дату рождения: 12.01.2003  Введите общую сумму покупок: 7000  Введите количество заказов: 1  Для продолжения нажмите любую клавишу . . .  Введите команду: 2  Введите ID покупателя: 4  Введите ФИО: Иову Дмитрий Олегович  Введите дату рождения: 23.08.2005  Введите общую сумму покупок: 12000  Введите количество заказов: 3  Для продолжения нажмите любую клавишу . . .  **Вывод информации о покупателях**  Введите команду: 11  Покупатели:  ID: 1  ФИО: Зудин Даниил Васильевич  Дата рождения: 25.01.2003  Общая сумма покупок: 5000 р.  Количество заказов: 5 шт.  ID: 2  ФИО: Исаев Станислав Алексеевич  Дата рождения: 25.06.2003  Общая сумма покупок: 1200 р.  Количество заказов: 7 шт.  ID: 3  ФИО: Хрущев Никита Сергеевич  Дата рождения: 12.01.2003  Общая сумма покупок: 7000 р.  Количество заказов: 1 шт.  ID: 4  ФИО: Иову Дмитрий Олегович  Дата рождения: 23.08.2005  Общая сумма покупок: 12000 р.  Количество заказов: 3 шт.  **Удаление покупателя**  Введите команду: 1  Введите ID удаляемого покупателя: 4  Покупатель с ID 4 удалён  Для продолжения нажмите любую клавишу . . .  Введите команду: 11  Покупатели:  ID: 1  ФИО: Зудин Даниил Васильевич  Дата рождения: 25.01.2003  Общая сумма покупок: 5000 р.  Количество заказов: 5 шт.  ID: 2  ФИО: Исаев Станислав Алексеевич  Дата рождения: 25.06.2003  Общая сумма покупок: 1200 р.  Количество заказов: 7 шт.  ID: 3  ФИО: Хрущев Никита Сергеевич  Дата рождения: 12.01.2003  Общая сумма покупок: 7000 р.  Количество заказов: 1 шт.  Для продолжения нажмите любую клавишу . . .  **Получить информацию о покупателе по id**  Введите команду: 4  Введите ID просматриваемого покупателя: 1  ID: 1  ФИО: Зудин Даниил Васильевич  Дата рождения: 25.01.2003  Общая сумма покупок: 5000 р.  Количество заказов: 5 шт.  Для продолжения нажмите любую клавишу . . .  **Сортировка покупателей по дате рождения**  Введите команду: 6  Покупатели отсортированы по дате рождения  Для продолжения нажмите любую клавишу . . .  ID: 3  ФИО: Хрущев Никита Сергеевич  Дата рождения: 12.01.2003  Общая сумма покупок: 7000 р.  Количество заказов: 1 шт.  ID: 1  ФИО: Зудин Даниил Васильевич  Дата рождения: 25.01.2003  Общая сумма покупок: 5000 р.  Количество заказов: 5 шт.  ID: 2  ФИО: Исаев Станислав Алексеевич  Дата рождения: 25.06.2003  Общая сумма покупок: 1200 р.  Количество заказов: 7 шт.  Для продолжения нажмите любую клавишу . . .  **Сортировка покупателей по общей сумме покупок**  Введите команду: 7  Покупатели отсортированы по общей сумме покупок  Для продолжения нажмите любую клавишу . . .  Введите команду: 11  Покупатели:  ID: 2  ФИО: Исаев Станислав Алексеевич  Дата рождения: 25.06.2003  Общая сумма покупок: 1200 р.  Количество заказов: 7 шт.  ID: 1  ФИО: Зудин Даниил Васильевич  Дата рождения: 25.01.2003  Общая сумма покупок: 5000 р.  Количество заказов: 5 шт.  ID: 3  ФИО: Хрущев Никита Сергеевич  Дата рождения: 12.01.2003  Общая сумма покупок: 7000 р.  Количество заказов: 1 шт.  Для продолжения нажмите любую клавишу . . .  **Сортировка покупателей по количеству заказов**  Введите команду: 8  Покупатели отсортированы по количеству заказов  Для продолжения нажмите любую клавишу . . .  Введите команду: 11  Покупатели:  ID: 3  ФИО: Хрущев Никита Сергеевич  Дата рождения: 12.01.2003  Общая сумма покупок: 7000 р.  Количество заказов: 1 шт.  ID: 1  ФИО: Зудин Даниил Васильевич  Дата рождения: 25.01.2003  Общая сумма покупок: 5000 р.  Количество заказов: 5 шт.  ID: 2  ФИО: Исаев Станислав Алексеевич  Дата рождения: 25.06.2003  Общая сумма покупок: 1200 р.  Количество заказов: 7 шт.  Для продолжения нажмите любую клавишу . . .  **Сортировка покупателей по ФИО**  Введите команду: 5  Покупатели отсортированы по ФИО  Для продолжения нажмите любую клавишу . . .  Введите команду: 11  Покупатели:  ID: 1  ФИО: Зудин Даниил Васильевич  Дата рождения: 25.01.2003  Общая сумма покупок: 5000 р.  Количество заказов: 5 шт.  ID: 2  ФИО: Исаев Станислав Алексеевич  Дата рождения: 25.06.2003  Общая сумма покупок: 1200 р.  Количество заказов: 7 шт.  ID: 3  ФИО: Хрущев Никита Сергеевич  Дата рождения: 12.01.2003  Общая сумма покупок: 7000 р.  Количество заказов: 1 шт.  Для продолжения нажмите любую клавишу . . .  **Сохранение данных в csv-файл**  Введите команду: 9  Данные успешно сохранены в файл  Для продолжения нажмите любую клавишу . . .  **Загрузка данных из csv-файла**  Введите команду: 10  ['1', 'Зудин Даниил Васильевич', '25.01.2003', '5000', '5']  ['2', 'Исаев Станислав Алексеевич', '25.06.2003', '1200', '7']  ['3', 'Хрущев Никита Сергеевич', '12.01.2003', '7000', '1']  Данные успешно загружены из файла  Для продолжения нажмите любую клавишу . . .  **Индивидуальное задание**  Введите команду: 12  Покупатели отсортированы по общей сумме покупок  Покупатели отсортированы по количеству заказов  Информация о покупателе с минимальной общей суммой заказа загружена в файл  Информация о покупателе с минимальным количеством заказов загружена в файл  Для продолжения нажмите любую клавишу . . .  **Основной csv-файл после выполнения программы**  1,Зудин Даниил Васильевич,25.01.2003,5000,5 2,Исаев Станислав Алексеевич,25.06.2003,1200,7 3,Хрущев Никита Сергеевич,12.01.2003,7000,1  **Csv-файл индивидуального задания**  2,Исаев Станислав Алексеевич,25.06.2003,1200,7 3,Хрущев Никита Сергеевич,12.01.2003,7000,1  **Выводы:**  В ходе работы были приобретены практические навыки работы с файлами и файловой системой на языке программирования Python. | | |