



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 ИУК2 "Информационные системы и сети"

ДОМАШНЯЯ РАБОТА №2

«РАБОТА С ФАЙЛАМИ»

ДИСЦИПЛИНА: «Перспективные языки программирования»

Выполнил: студент гр. ИУК4-33Б _____ (____ Сароян Н.А.____)
(Подпись) (Ф.И.О.)

Проверил: _____ (____ Осипова О.В.____)
(Подпись) (Ф.И.О.)

Дата сдачи (защиты):

Результаты сдачи (защиты):

- Балльная оценка:

- Оценка:

Калуга, 2024

ЦЕЛИ: Целью выполнения домашней работы является приобретение практических навыков разработки графического интерфейса с помощью библиотеки PyQt средствами языка Python.

ЗАДАЧИ: Основными задачами выполнения домашней работы являются:

1. Ознакомиться с конструкцией библиотеки PyQt;
2. Изучить способы создания мини-приложений с помощью PyQt;
3. Изучить возможности данной платформы;
4. Закрепить полученные в ходе выполнения домашней работы навыки

Вариант 8

Задача 1.

Выберите любую папку на своем компьютере, имеющую вложенные директории.

Выведите на печать в терминал и сохраните в файл txt названия всех файлов с названием «main». Заархивируйте данную папку средствами python.

Задача 2.

Создать файл в формате xml, содержащий сведения о видеокартах. Структура записи: название, объем видеопамати, тип видеопамати, частота графического процессора, частота памяти и ширина шины.

Предусмотреть возможность корректировки файла по вводимому коду корректировки, например:

- 1 - удалить запись (по названию);
- 2 - добавить новую запись;
- 3 - изменить запись (по названию);
- 4 - получить информацию (по названию)
- 5 – отсортировать видеокарты по:
 - Названию
 - Объему видеопамати
 - Частоте графического процессора
 - Типу видеопамати
- 6 – сохранить в отдельный файл и вывести на консоль видеокарты, у которых объем видеокарты больше заданно.

Листинг программы 1

```
import os

import tarfile

if __name__ == '__main__':

    folder_path = '/Users/noriksaroyan/PycharmProjects/ArchicadAPI'

    main_files = []

    for root, dirs, files in os.walk(folder_path):

        for file in files:

            if 'main' in file:

                main_files.append(os.path.join(root, file))

    with open('main_files.txt', 'w') as f:

        for file in main_files:

            print(file)

            f.write(file + '\n')

    with tarfile.open('archived_folder.tar.gz', 'w:gz') as tar:

        tar.add(folder_path, arcname=os.path.basename(folder_path))
```

Результат выполнения программы:

```
/Users/noriksaroyan/PycharmProjects/PPL_HW1/.venv/bin/python /Users/noriksaroyan/PycharmProjects/PPL_HW1/hw2/Task1.py
/Users/noriksaroyan/PycharmProjects/ArchicadAPI/main.py
/Users/noriksaroyan/PycharmProjects/ArchicadAPI/junks/main2.py
/Users/noriksaroyan/PycharmProjects/ArchicadAPI/venv/lib/python3.8/site-packages/wheel/__main__.py
/Users/noriksaroyan/PycharmProjects/ArchicadAPI/venv/lib/python3.8/site-packages/numpy/f2py/__main__.py
/Users/noriksaroyan/PycharmProjects/ArchicadAPI/venv/lib/python3.8/site-packages/pip/__main__.py
/Users/noriksaroyan/PycharmProjects/ArchicadAPI/venv/lib/python3.8/site-packages/pip/_internal/main.py
/Users/noriksaroyan/PycharmProjects/ArchicadAPI/venv/lib/python3.8/site-packages/pip/_internal/cli/main.py
/Users/noriksaroyan/PycharmProjects/ArchicadAPI/venv/lib/python3.8/site-packages/pip/_vendor/pygments/__main__.py
/Users/noriksaroyan/PycharmProjects/ArchicadAPI/venv/lib/python3.8/site-packages/pip/_vendor/distro/__main__.py
/Users/noriksaroyan/PycharmProjects/ArchicadAPI/venv/lib/python3.8/site-packages/pip/_vendor/certifi/__main__.py
/Users/noriksaroyan/PycharmProjects/ArchicadAPI/venv/lib/python3.8/site-packages/pip/_vendor/rich/__main__.py
/Users/noriksaroyan/PycharmProjects/ArchicadAPI/venv/lib/python3.8/site-packages/pip/_vendor/platformdirs/__main__.py
```

Листинг программы 2

```
import xml.etree.ElementTree as ET

import os

filename = "videocards.xml"

if not os.path.exists(filename):

    root = ET.Element("videocards")

    tree = ET.ElementTree(root)

    tree.write(filename, encoding="utf-8", xml_declaration=True)

tree = ET.parse(filename)

root = tree.getroot()

def add_videocard(name, memory_size, memory_type, gpu_frequency,
memory_frequency, bus_width):

    card = ET.SubElement(root, "videocard")

    ET.SubElement(card, "name").text = name

    ET.SubElement(card, "memory_size").text = str(memory_size)
```

```

ET.SubElement(card, "memory_type").text = memory_type

ET.SubElement(card, "gpu_frequency").text = str(gpu_frequency)

ET.SubElement(card, "memory_frequency").text = str(memory_frequency)

ET.SubElement(card, "bus_width").text = str(bus_width)

tree.write(filename, encoding="utf-8", xml_declaration=True)

print(f"Видеокарта '{name}' добавлена.")

def delete_videocard(name):

    for card in root.findall("videocard"):

        if card.find("name").text == name:

            root.remove(card)

            tree.write(filename, encoding="utf-8", xml_declaration=True)

            print(f"Видеокарта '{name}' удалена.")

            return

    print(f"Видеокарта с названием '{name}' не найдена.")

def edit_videocard(name):

    for card in root.findall("videocard"):

        if card.find("name").text == name:

            card.find("memory_size").text = input("Введите новый объем
видеопамяти: ")

            card.find("memory_type").text = input("Введите новый тип
видеопамяти: ")

            card.find("gpu_frequency").text = input("Введите новую частоту
графического процессора: ")

            card.find("memory_frequency").text = input("Введите новую частоту
памяти: ")

            card.find("bus_width").text = input("Введите новую ширину шины: ")

```

```

        tree.write(filename, encoding="utf-8", xml_declaration=True)

        print(f"Видеокарта '{name}' изменена.")

        return

    print(f"Видеокарта с названием '{name}' не найдена.")

def get_videocard_info(name):

    for card in root.findall("videocard"):

        if card.find("name").text == name:

            for detail in card:

                print(f"{detail.tag}: {detail.text}")

            return

    print(f"Видеокарта с названием '{name}' не найдена.")

def sort_videocards(criteria):

    valid_criteria = ["name", "memory_size", "memory_type", "gpu_frequency",
"memory_frequency", "bus_width"]

    if criteria not in valid_criteria:

        print("Недопустимый критерий сортировки.")

        return

    videocards = list(root)

    videocards.sort(key=lambda x: (

        int(x.find(criteria).text) if criteria in ["memory_size",
"gpu_frequency", "memory_frequency",

"bus_width"] else
x.find(criteria).text))

```

```

root.clear()

root.extend(videocards)

tree.write(filename, encoding="utf-8", xml_declaration=True)

print(f"Видеокарты отсортированы по '{criteria}'.")

def save_filtered_videocards(threshold):
    threshold = int(threshold)

    cards_to_save = ET.Element("videocards")

    for card in root.findall("videocard"):
        if int(card.find("memory_size").text) > threshold:
            cards_to_save.append(card)

    filtered_tree = ET.ElementTree(cards_to_save)

    filtered_tree.write("filtered_videocards.xml", encoding="utf-8",
xml_declaration=True)

    print(f"Видеокарты с объемом памяти больше {threshold} сохранены в
'filtered_videocards.xml'.")

# Главное меню

def edit_videocards():
    while True:
        print("\n1 - удалить запись (по названию)")

        print("2 - добавить новую запись")

        print("3 - изменить запись (по названию)")

        print("4 - получить информацию (по названию)")

        print("5 - отсортировать видеокарты")

```

```
        print("6 - сохранить в отдельный файл видеокарты с объемом памяти больше заданного")

    print("0 - выход")

    choice = input("Введите код корректировки: ")

    if choice == "1":

        name = input("Введите название видеокарты для удаления: ")

        delete_videocard(name)

    elif choice == "2":

        name = input("Введите название: ")

        memory_size = int(input("Введите объем видеопамяти: "))

        memory_type = input("Введите тип видеопамяти: ")

        gpu_frequency = int(input("Введите частоту графического процессора: "))

        memory_frequency = int(input("Введите частоту памяти: "))

        bus_width = int(input("Введите ширину шины: "))

        add_videocard(name, memory_size, memory_type, gpu_frequency, memory_frequency, bus_width)

    elif choice == "3":

        name = input("Введите название видеокарты для изменения: ")

        edit_videocard(name)

    elif choice == "4":

        name = input("Введите название видеокарты для получения информации: ")

        get_videocard_info(name)
```



```

elif choice == "5":

    criteria = input(
                                "Отсортировать по
(name/memory_size/gpu_frequency/memory_type/memory_frequency/bus_width): ")

    sort_videocards(criteria)

elif choice == "6":

    threshold = input("Введите минимальный объем видеопамяти: ")

    save_filtered_videocards(threshold)

elif choice == "0":

    break

def main():

    add_videocard("GeForce RTX 3080", 10240, "GDDR6X", 1440, 1188, 320)

    add_videocard("Radeon RX 6800 XT", 16384, "GDDR6", 1825, 2000, 256)

    edit_videocards()

if __name__ == "__main__":

    main()

```

Результат выполнения программы:

```
/Users/noriksaroyan/PycharmProjects/PPL_HW1/.venv/bin/python /Users/noriksaroyan/PycharmProjects/PPL_HW1/hw2/Task2.py
Видеокарта 'GeForce RTX 3080' добавлена.
Видеокарта 'Radeon RX 6800 XT' добавлена.

1 - удалить запись (по названию)
2 - добавить новую запись
3 - изменить запись (по названию)
4 - получить информацию (по названию)
5 - отсортировать видеокарты
6 - сохранить в отдельный файл видеокарты с объемом памяти больше заданного
0 - выход
Введите код корректировки: 2
Введите название: ddd
Введите объем видеопамяти: 2333
Введите тип видеопамяти: 2334
Введите частоту графического процессора: 234565
Введите частоту памяти: 0945739078039
Введите ширину шины: 45080945097345
Видеокарта 'ddd' добавлена.

1 - удалить запись (по названию)
2 - добавить новую запись
3 - изменить запись (по названию)
4 - получить информацию (по названию)
5 - отсортировать видеокарты
6 - сохранить в отдельный файл видеокарты с объемом памяти больше заданного
0 - выход
Введите код корректировки: |
```

```
<?xml version='1.0' encoding='utf-8'?>

<videocards>

  <videocard>

    <name>Radeon RX 6800 XT</name>

    <memory_size>16384</memory_size>

    <memory_type>GDDR6</memory_type>

    <gpu_frequency>1825</gpu_frequency>

    <memory_frequency>2000</memory_frequency>

    <bus_width>256</bus_width>

  </videocard>

  <videocard>

    <name>GeForce RTX 3080</name>

    <memory_size>10240</memory_size>

    <memory_type>GDDR6X</memory_type>

    <gpu_frequency>1440</gpu_frequency>
```

```
<memory_frequency>1188</memory_frequency>

<bus_width>320</bus_width>

</videocard>

<videocard>

  <name>Radeon RX 6800 XT</name>

  <memory_size>16384</memory_size>

  <memory_type>GDDR6</memory_type>

  <gpu_frequency>1825</gpu_frequency>

  <memory_frequency>2000</memory_frequency>

  <bus_width>256</bus_width>

</videocard>

<videocard>

  <name>GeForce RTX 3080</name>

  <memory_size>10240</memory_size>

  <memory_type>GDDR6X</memory_type>

  <gpu_frequency>1440</gpu_frequency>

  <memory_frequency>1188</memory_frequency>

  <bus_width>320</bus_width>

</videocard>

<videocard>

  <name>Radeon RX 6800 XT</name>

  <memory_size>16384</memory_size>

  <memory_type>GDDR6</memory_type>

  <gpu_frequency>1825</gpu_frequency>

  <memory_frequency>2000</memory_frequency>

  <bus_width>256</bus_width>

</videocard>

<videocard>
```

```
<name>GeForce RTX 3080</name>

<memory_size>10240</memory_size>

<memory_type>GDDR6X</memory_type>

<gpu_frequency>1440</gpu_frequency>

<memory_frequency>1188</memory_frequency>

<bus_width>320</bus_width>
```

```
</videocard>
```

```
<videocard>
```

```
<name>Radeon RX 6800 XT</name>

<memory_size>16384</memory_size>

<memory_type>GDDR6</memory_type>

<gpu_frequency>1825</gpu_frequency>

<memory_frequency>2000</memory_frequency>

<bus_width>256</bus_width>
```

```
</videocard>
```

```
<videocard>
```

```
<name>1234</name>

<memory_size>1234</memory_size>

<memory_type>1234</memory_type>

<gpu_frequency>1234</gpu_frequency>

<memory_frequency>1234</memory_frequency>

<bus_width>1234</bus_width>
```

```
</videocard>
```

```
<videocard>
```

```
<name>1234</name>

<memory_size>1234</memory_size>

<memory_type>12344</memory_type>

<gpu_frequency>1221</gpu_frequency>
```

```
<memory_frequency>111</memory_frequency>

<bus_width>11</bus_width>

</videocard>

<videocard>

  <name>GeForce RTX 3080</name>

  <memory_size>10240</memory_size>

  <memory_type>GDDR6X</memory_type>

  <gpu_frequency>1440</gpu_frequency>

  <memory_frequency>1188</memory_frequency>

  <bus_width>320</bus_width>

</videocard>

<videocard>

  <name>Radeon RX 6800 XT</name>

  <memory_size>16384</memory_size>

  <memory_type>GDDR6</memory_type>

  <gpu_frequency>1825</gpu_frequency>

  <memory_frequency>2000</memory_frequency>

  <bus_width>256</bus_width>

</videocard>

<videocard>

  <name>ghghghg</name>

  <memory_size>123455</memory_size>

  <memory_type>12324545</memory_type>

  <gpu_frequency>223384</gpu_frequency>

  <memory_frequency>47474</memory_frequency>

  <bus_width>384845</bus_width>

</videocard>

<videocard>
```

```
<name>GeForce RTX 3080</name>

<memory_size>10240</memory_size>

<memory_type>GDDR6X</memory_type>

<gpu_frequency>1440</gpu_frequency>

<memory_frequency>1188</memory_frequency>

<bus_width>320</bus_width>

</videocard>

<videocard>

  <name>Radeon RX 6800 XT</name>

  <memory_size>16384</memory_size>

  <memory_type>GDDR6</memory_type>

  <gpu_frequency>1825</gpu_frequency>

  <memory_frequency>2000</memory_frequency>

  <bus_width>256</bus_width>

</videocard>

<videocard>

  <name>ddd</name>

  <memory_size>2333</memory_size>

  <memory_type>2334</memory_type>

  <gpu_frequency>234565</gpu_frequency>

  <memory_frequency>945739078039</memory_frequency>

  <bus_width>45080945097345</bus_width>

</videocard>

</videocards>
```

Вывод: в ходе выполнения работы были приобретены практические навыки необходимые для работы с файлами и типами файлов в Python.