**Информационные системы управления корпоративным контентом**

**Практическая работа №10**

**ФИО:** Гусев Тимофей Андреевич

**Группа:** ИКБО-08-21

**Дата предоставления в СДО:**

**Задание 1. Дополнить скрипт для генерации накладных из таблицы с данными из предыдущей работы:**

**➔ сгенерировать файл со штрих-кодом следующего формата: [СКЛАД][НОМЕР\_НАКЛАДНОЙ][ДАТА] (пример на рисунке выше);**

**➔ вставить изображение со штрих-кодом в левый верхний угол накладной;**

**➔ удалить файл со штрих-кодом.**

На рисунке 1 представлено содержимое исходного файла с даннымиdata.xlsx.

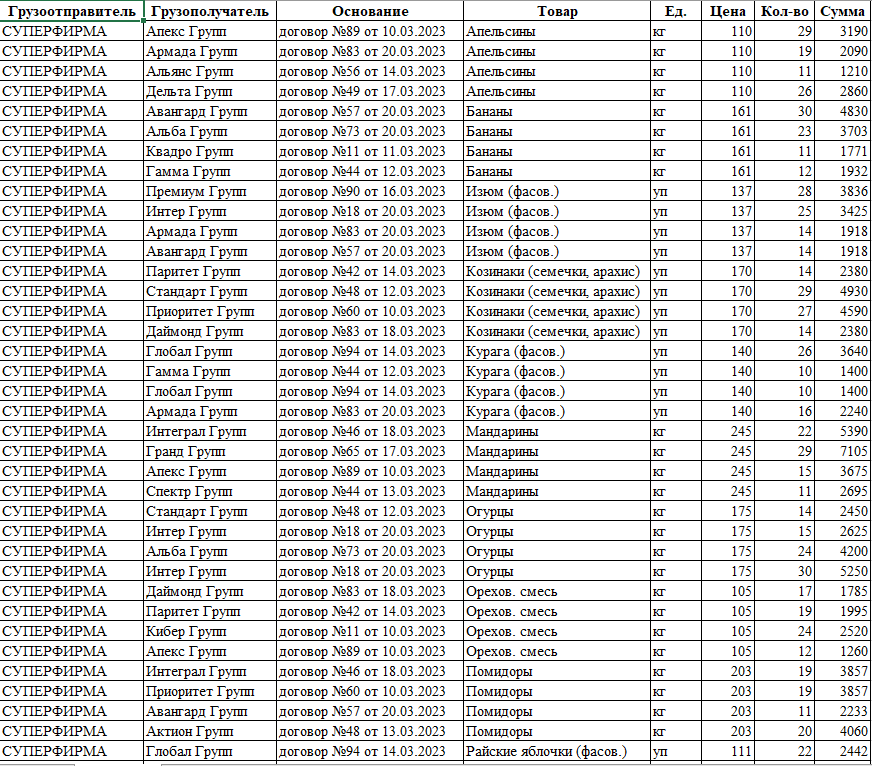


Рисунок 1 – Содержимое файла data.xlsx

На листинге 1 представлен код функции по чтению данных из файла data.xlsx в словарь.

Листинг 1 – Код функции read\_data

def read\_data():  
 "Читает данные из файла excel в dict"  
 data = {}  
 df = pd.read\_excel("data.xlsx")  
 df = df.sort\_values("Грузополучатель")  
 for i, row in df.iterrows():  
 data[row["Грузополучатель"]] = {}  
 data[row["Грузополучатель"]]["Позиции"] = []  
 data[row["Грузополучатель"]]["{{NUMBER}}"] = len(list(data.keys()))  
 for i, row in df.iterrows():  
 data[row["Грузополучатель"]]["{{TO}}"] = row["Грузополучатель"]  
 data[row["Грузополучатель"]]["{{FROM}}"] = row["Грузоотправитель"]  
 data[row["Грузополучатель"]]["{{DOCUMENT}}"] = row["Основание"]  
 data[row["Грузополучатель"]]["{{DATE}}"] = datetime.today().date().strftime("%d.%m.%y")  
 data[row["Грузополучатель"]]["Позиции"].append([len(data[row["Грузополучатель"]]["Позиции"]) + 1, row["Товар"], row["Ед."], row["Цена"], row["Кол-во"], row["Сумма"]])  
 data[row["Грузополучатель"]]["{{GLOBAL\_PRICE}}"] = sum(el[5] for el in data[row["Грузополучатель"]]["Позиции"])  
 data[row["Грузополучатель"]]["{{GLOBAL\_COUNT}}"] = len(data[row["Грузополучатель"]]["Позиции"])  
 data[row["Грузополучатель"]]["{{DOCUMENT}}"] = row["Основание"]  
 return data

На листинге 2 представлен код функции по перезаписи данных в документ.

Листинг 2 – Код функции replace\_text

if key in paragraph.text:  
 paragraph.text = paragraph.text.replace(key, value)

На листинге 3 представлен код основной функции, выполняющей подстановку данных в шаблон.

Листинг 3 – Код функции main

def main():  
 "Основная функция"  
 data = read\_data()  
 pattern = r"{{\w{1,}}}"  
 template = 'template.docx'  
 n = 1  
 for key, value in data.items():  
 result = f'накладная\_{n:02}\_{key}.docx'  
 template\_doc = Document(template)  
 barcode\_1 = Code128(f'AB{data[key]["{{NUMBER}}"]}\_{data[key]["{{DATE}}"]}', writer=ImageWriter(format='PNG'))  
 filename = barcode\_1.save('example', options=dict(module\_height=3))  
 par = template\_doc.paragraphs[0].insert\_paragraph\_before()  
 lr = par.add\_run()  
 lr.add\_picture('example.PNG')  
 os.remove('example.PNG')  
 for k, v in value.items():  
 if re.fullmatch(pattern, k.strip()):  
 for paragraph in template\_doc.paragraphs:  
 replace\_text(paragraph, k, str(v))  
 my\_table = template\_doc.tables[0]  
 for line in value["Позиции"]:  
 row = my\_table.add\_row()  
 i = 0  
 for cell in row.cells:  
 cell.text = str(line[i])  
 i += 1  
 template\_doc.save(f"result/{result}")  
 n += 1

На рисунке 2 представлен список сгенерированных в ходе выполнения программы файлов.

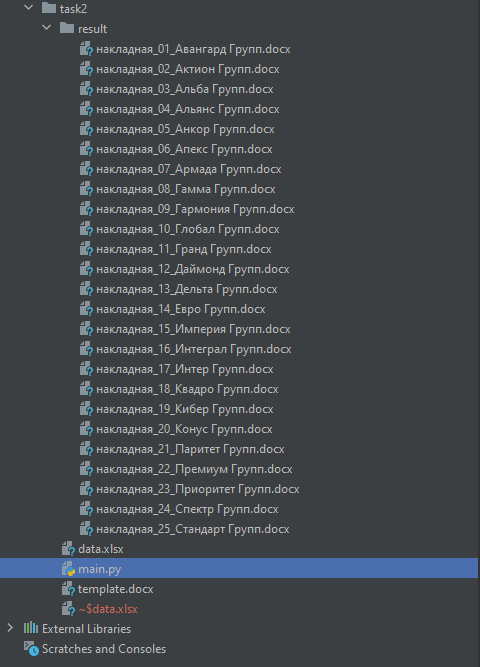


Рисунок 2 – Сгенерированные файлы

На рисунке 3 представлено содержимое одного из сгенерированных файлов.

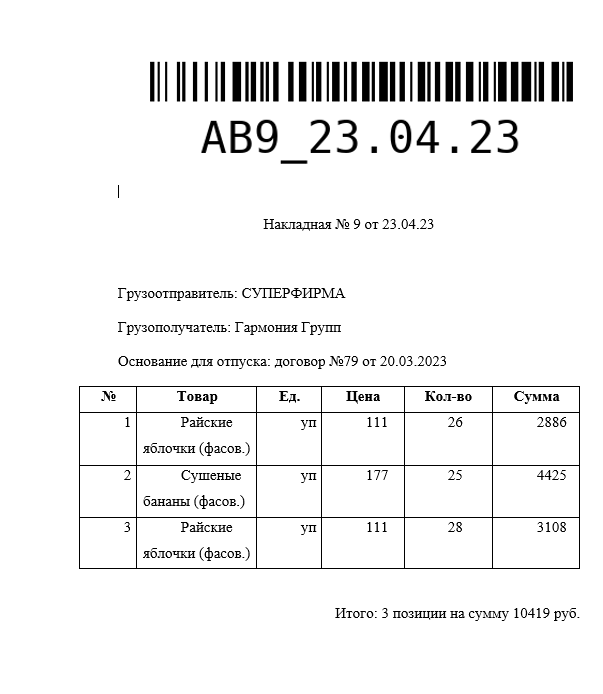


Рисунок 3 – Содержимое файла накладная\_09\_Гармония Групп.docx

**Задание 2. Дополнить скрипт для генерации упаковочных листов: ➔ сгенерировать файл по шаблону;**

**➔ вставить изображение со штрих-кодом в левый верхний угол накладной;**

**➔ преобразовать файл в pdf с названием [СКЛАД][НОМЕР][ДАТА].pdf (например, AB27\_17.04.23.pdf);**

**➔ удалить файл docx.**

На рисунке 1 представлено содержимое исходного файла с даннымиdata.xlsx.

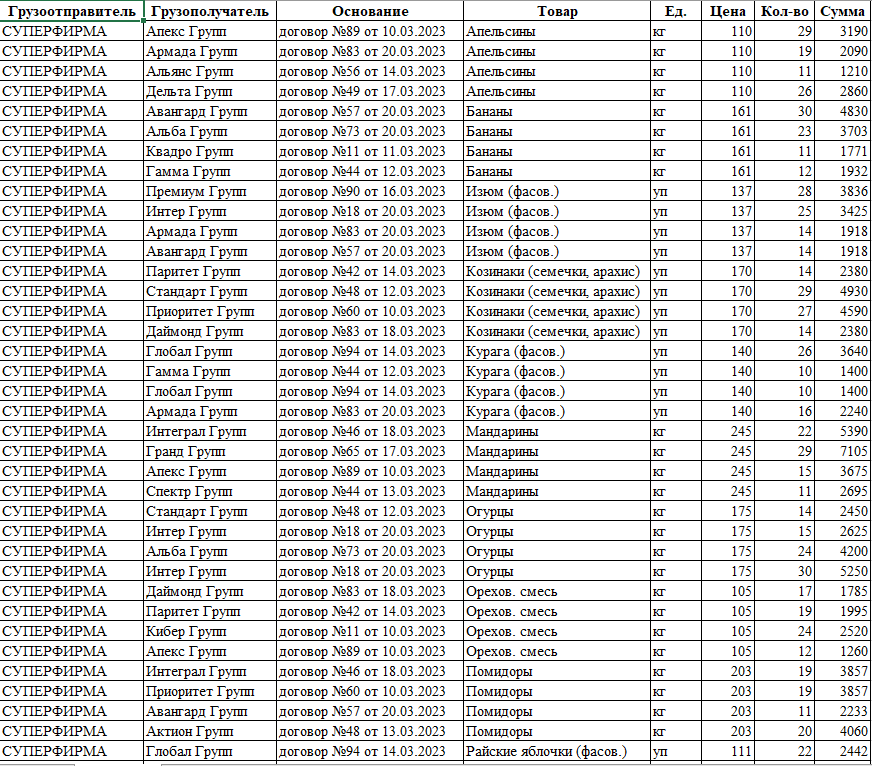


Рисунок 4 – Содержимое файла data.xlsx

На листинге 1 представлен код функции по чтению данных из файла data.xlsx в словарь.

Листинг 4 – Код функции read\_data

def read\_data():  
 "Читает данные из файла excel в dict"  
 data = {}  
 df = pd.read\_excel("data.xlsx")  
 df = df.sort\_values("Грузополучатель")  
 for i, row in df.iterrows():  
 data[row["Грузополучатель"]] = {}  
 data[row["Грузополучатель"]]["Позиции"] = []  
 data[row["Грузополучатель"]]["{{NUMBER}}"] = len(list(data.keys()))  
 for i, row in df.iterrows():  
 data[row["Грузополучатель"]]["{{TO}}"] = row["Грузополучатель"]  
 data[row["Грузополучатель"]]["{{FROM}}"] = row["Грузоотправитель"]  
 data[row["Грузополучатель"]]["{{DOCUMENT}}"] = row["Основание"]  
 data[row["Грузополучатель"]]["{{DATE}}"] = datetime.today().date().strftime("%d.%m.%y")  
 data[row["Грузополучатель"]]["Позиции"].append([len(data[row["Грузополучатель"]]["Позиции"]) + 1, row["Товар"], row["Ед."], row["Кол-во"]])  
 data[row["Грузополучатель"]]["{{DOCUMENT}}"] = row["Основание"]  
 return data

На листинге 2 представлен код функции по перезаписи данных в документ.

Листинг 5 – Код функции replace\_text

if key in paragraph.text:  
 paragraph.text = paragraph.text.replace(key, value)

На листинге 3 представлен код основной функции, выполняющей подстановку данных в шаблон.

Листинг 6 – Код функции main

def main():  
 "Основная функция"  
 data = read\_data()  
 pattern = r"{{\w{1,}}}"  
 template = 'template.docx'  
 n = 1  
 for key, value in data.items():  
 result = f'накладная\_{n:02}\_{key}'  
 template\_doc = Document(template)  
 barcode\_1 = Code128(f'AB{data[key]["{{NUMBER}}"]}\_{data[key]["{{DATE}}"]}', writer=ImageWriter(format='PNG'))  
 filename = barcode\_1.save('example', options=dict(module\_height=3))  
 par = template\_doc.paragraphs[0].insert\_paragraph\_before()  
 lr = par.add\_run()  
 lr.add\_picture('example.PNG')  
 os.remove('example.PNG')  
 for k, v in value.items():  
 if re.fullmatch(pattern, k.strip()):  
 for paragraph in template\_doc.paragraphs:  
 replace\_text(paragraph, k, str(v))  
 my\_table = template\_doc.tables[0]  
 for line in value["Позиции"]:  
 row = my\_table.add\_row()  
 i = 0  
 for cell in row.cells:  
 cell.text = str(line[i])  
 i += 1  
 template\_doc.save(f"result/{result}.docx")  
 convert(f"result/{result}.docx", f"result/{result}.pdf")  
 os.remove(f"result/{result}.docx")  
 n += 1

На рисунке 5 представлен список сгенерированных в ходе выполнения программы файлов.

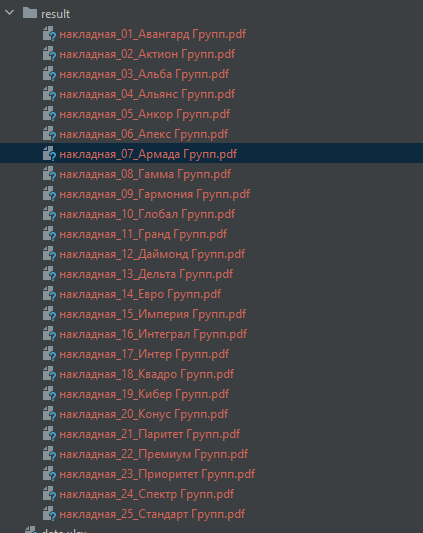


Рисунок 5 – Сгенерированные файлы

На рисунке 6 представлено содержимое одного из сгенерированных файлов.

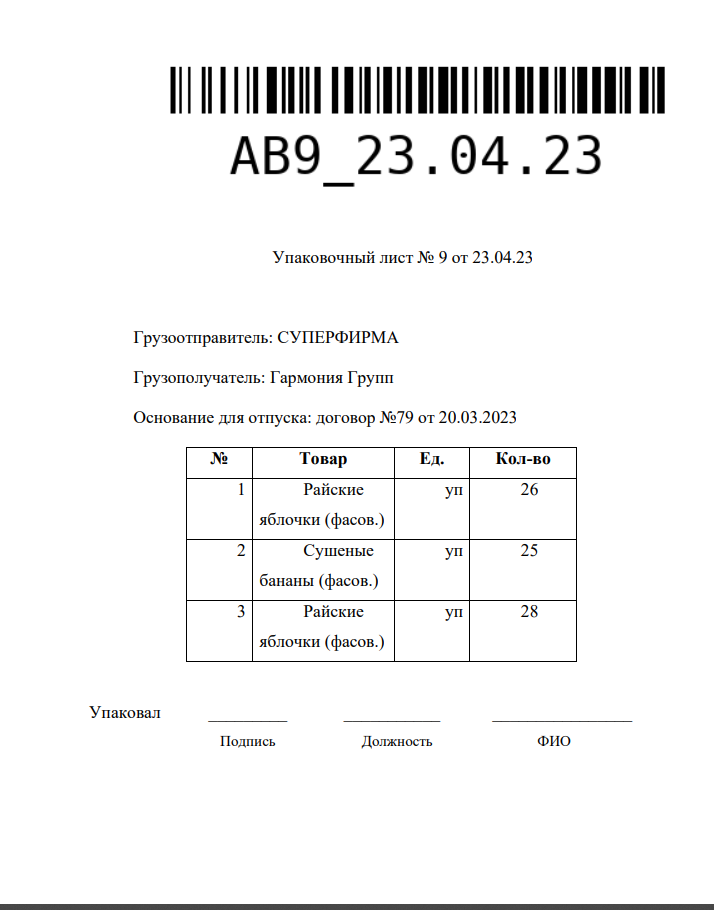


Рисунок 6 – Содержимое файла накладная\_09\_Гармония Групп.pdf

**Задание 3. Реализовать следующий сценарий работы:**

**➔ клиент приходит на склад с накладной, которая содержит штрих-код;**

**➔ работник склада сканирует штрих-код через веб-камеру;**

**➔ программа считывает штрих-код, определяет название файла с упаковочным листом;**

**➔ программно открывается файл с упаковочным листом (только один раз);**

**➔ программа ожидает следующего сканирования.**

На рисунке 7 представлен этап сканирования штрих-кода

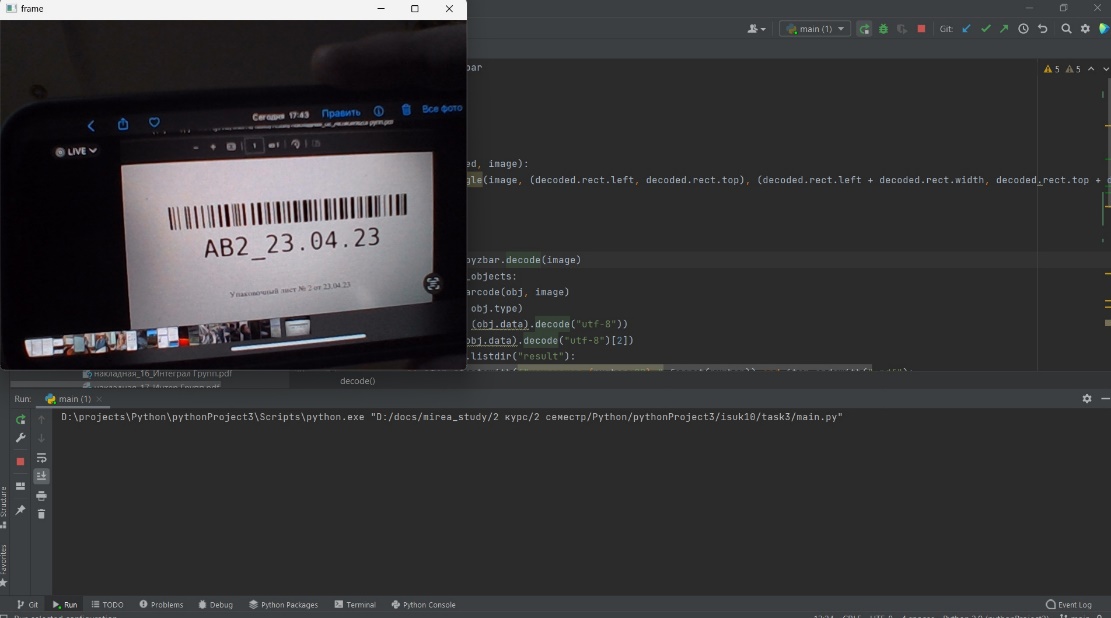


Рисунок 7 – Сканирование штрих-кода

На листинге 7 представлен код функции по декодированию штрих-кода и запуску файла pdf

Листинг 7 – Код функции decode

def decode(image):

decoded\_objects = pyzbar.decode(image)

for obj in decoded\_objects:

image = draw\_barcode(obj, image)

print("Type:", obj.type)

print("Data:", (obj.data).decode("utf-8"))

number = int((obj.data).decode("utf-8")[2])

for item in os.listdir("result"):

if item.startswith(f"накладная\_{number:02}\_".format(number)) and item.endswith(".pdf"):

path = f"result/{item}"

subprocess.Popen([path], shell=True)

break

print()

break

return image