Министерство науки и высшего образования Российской Федерации

Федеральное государственное бюджетное образовательное учреждение

высшего образования

«Вятский государственный университет»

Колледж ВятГУ

**ОТЧЕТ**

**ПО лабораторной работе №9**

**«Работа с типизированными файлами»**

**ПО ДИСЦИПЛИНЕ**

**«УП 05.02 Разработка кода информационных систем»**

Выполнил: студент учебной группы

ИСПк-101-51-00

Дуркин Арсений Валерьевич

Преподаватель:

Сергеева Елизавета Григорьевна

Киров

2023

**Цель работы:** получить представление об организации хранения и использования информации посредством типизрованных файлов, закрепить навыки создания пользовательского интерфейса.

**Скриншот задания**

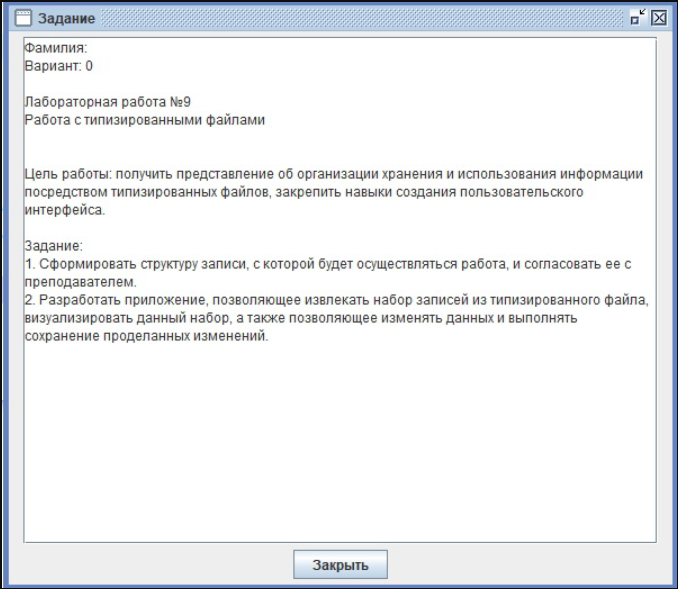
****

Рисунок 1 – Скриншот задания

**Формулировка заданий**

1. Сформировать структуру записи, с которой будет осуществляться работа, и согласовать ее с преподавателем.
2. Разработать приложение, позволяющее извлекать набор записей их типизированного файла, визуализировать данный набор, а также позволяющее изменять данных и выполнять сохранение проделанных изменений.

**Описание кода**

Программа реализована на программном коде Python, с помощью библиотеки PyQt5 и специальной библиотеки для связывания исходного кода с файлом Excel (pandas). С помощью этой библиотеки, можно удалять, изменять и добавлять новые записи из программы в файл Excel.

**Код программы**

import sys

import pandas as pd

from PyQt5.QtWidgets import QApplication, QWidget, QVBoxLayout, QHBoxLayout, QLabel, QLineEdit, QPushButton, QTableWidget, QTableWidgetItem, QFileDialog, QMessageBox

class MyWidget(QWidget):

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

super().\_\_init\_\_()

self.initUI()

def initUI(self):

# Создаем вертикальный макет

vbox = QVBoxLayout()

# Создаем горизонтальный макет для ввода имени файла

hbox\_file = QHBoxLayout()

label\_file = QLabel('Имя файла:')

self.line\_edit\_file = QLineEdit()

hbox\_file.addWidget(label\_file)

hbox\_file.addWidget(self.line\_edit\_file)

# Создаем горизонтальный макет для ввода номера строки

hbox\_row = QHBoxLayout()

label\_row = QLabel('Номер строки:')

self.line\_edit\_row = QLineEdit()

hbox\_row.addWidget(label\_row)

hbox\_row.addWidget(self.line\_edit\_row)

# Создаем кнопку для выбора файла

button\_choose\_file = QPushButton('Выбрать файл')

button\_choose\_file.clicked.connect(self.choose\_file)

# Создаем кнопку для извлечения данных из файла

button\_load = QPushButton('Загрузить данные')

button\_load.clicked.connect(self.load\_data)

# Создаем таблицу для отображения данных

self.table = QTableWidget()

# Создаем кнопку для сохранения изменений в файле

button\_save = QPushButton('Сохранить изменения')

button\_save.clicked.connect(self.save\_data)

# Добавляем все элементы в вертикальный макет

vbox.addLayout(hbox\_file)

vbox.addWidget(button\_choose\_file)

vbox.addLayout(hbox\_row)

vbox.addWidget(button\_load)

vbox.addWidget(self.table)

vbox.addWidget(button\_save)

self.setLayout(vbox)

self.setGeometry(100, 100, 600, 400)

self.setWindowTitle('Приложение для работы с файлом Excel')

def choose\_file(self):

# Открываем диалог выбора файла

filename, \_ = QFileDialog.getOpenFileName(self, 'Открыть файл', '.', 'Файлы Excel (\*.xlsx \*.xls)')

# Устанавливаем имя файла в поле ввода

if filename:

self.line\_edit\_file.setText(filename)

def load\_data(self):

# Извлекаем имя файла и номер строки из полей ввода

filename = self.line\_edit\_file.text()

try:

row\_num = int(self.line\_edit\_row.text())

except ValueError:

row\_num = 0

# Загружаем данные из файла в формате Excel с помощью pandas

try:

df = pd.read\_excel(filename, skiprows=row\_num - 1)

except Exception as e:

self.show\_error\_dialog(str(e))

return

# Очищаем таблицу

self.table.clear()

# Устанавливаем количество строк и столбцов в таблице

self.table.setRowCount(df.shape[0])

self.table.setColumnCount(df.shape[1])

# Заполняем таблицу данными из DataFrame

for i in range(df.shape[0]):

for j in range(df.shape[1]):

item = QTableWidgetItem(str(df.iloc[i, j]))

self.table.setItem(i, j, item)

def save\_data(self):

# Извлекаем имя файла из поля ввода

filename = self.line\_edit\_file.text()

# Получаем данные из таблицы

data = []

for i in range(self.table.rowCount()):

row = []

for j in range(self.table.columnCount()):

item = self.table.item(i, j)

if item is not None:

row.append(item.text())

else:

row.append('')

data.append(row)

# Создаем DataFrame из данных и сохраняем в файл Excel

df = pd.DataFrame(data)

try:

df.to\_excel(filename, index=False)

except Exception as e:

self.show\_error\_dialog(str(e))

return

# Показываем сообщение об успешном сохранении

QMessageBox.information(self, 'Сохранение', 'Изменения сохранены в файле')

def show\_error\_dialog(self, message):

# Показываем диалог с ошибкой

QMessageBox.critical(self, 'Ошибка', message)

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

app = QApplication(sys.argv)

widget = MyWidget()

# Показываем виджет

widget.show()

# Запускаем главный цикл приложения

sys.exit(app.exec\_())

**Результат выполнения программы**

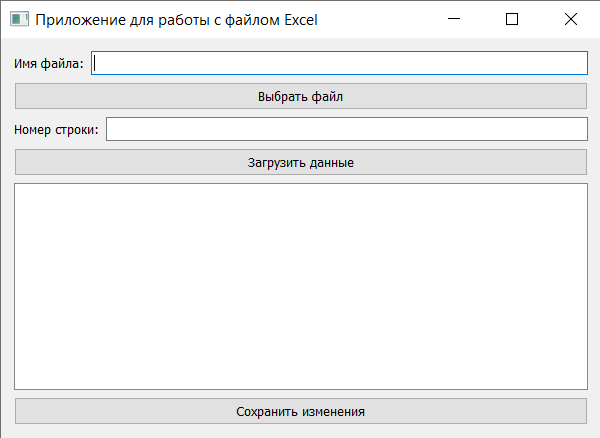
****

Рисунок 2 – Окно для управления программой (Начальное окно)

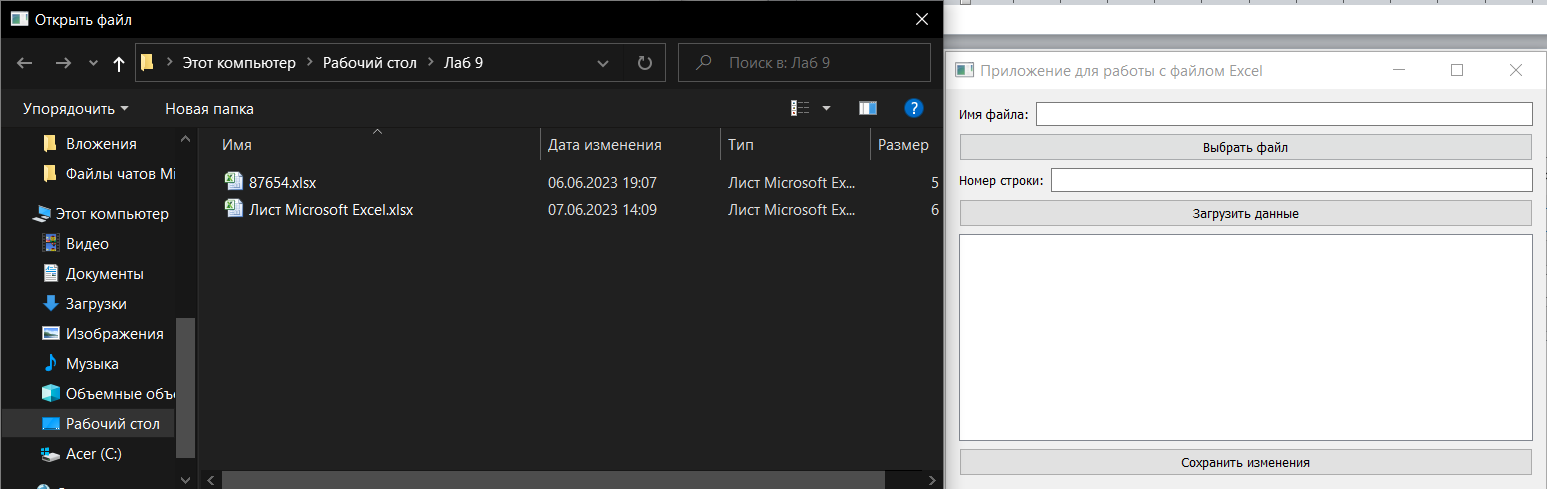


Рисунок 3 – Выбор файла Excel для взаимодействия

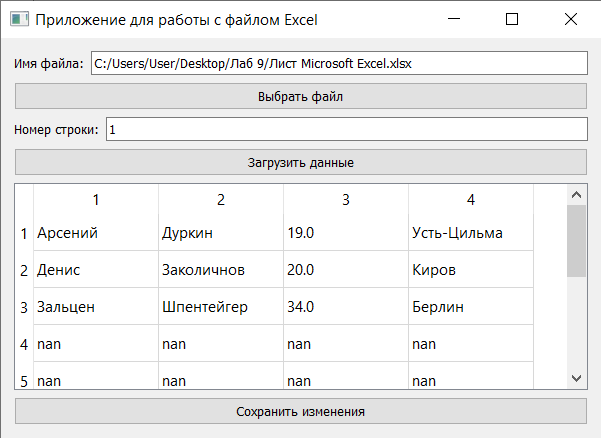


Рисунок 4 – Вывод записей с файла Excel.

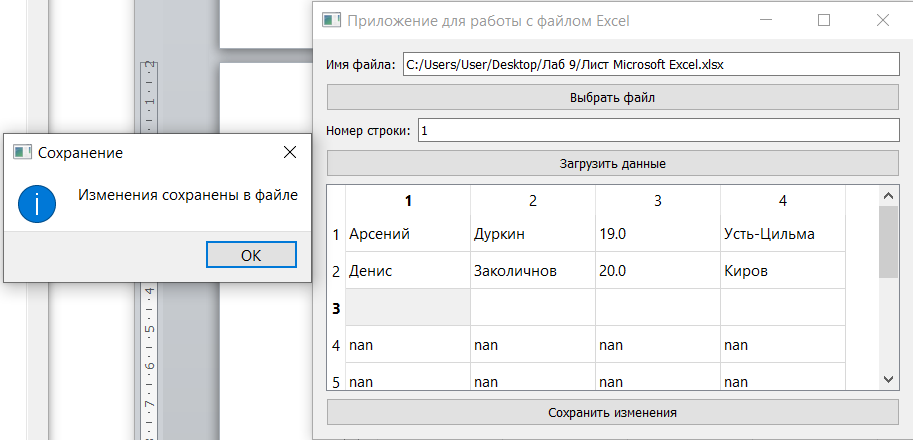


Рисунок 5 – Удаление записи из файла

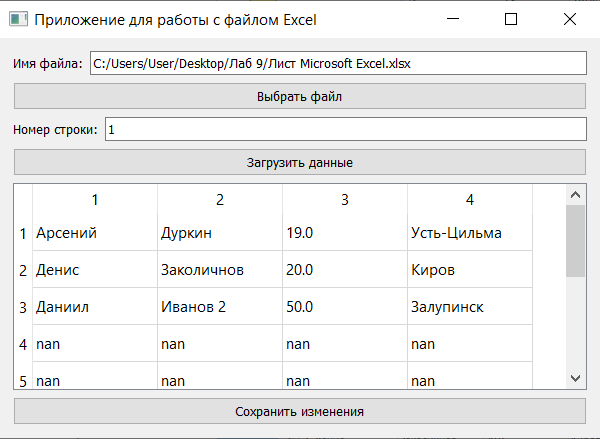


Рисунок 6 – Добавление информации в файл

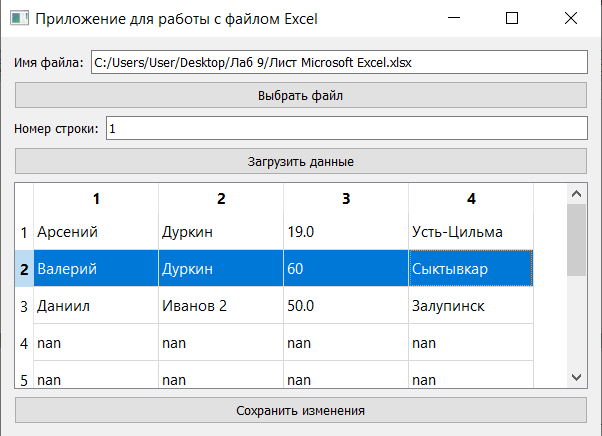


Рисунок 7 – Изменение информации в файле.

**Вывод**

В результате выполнения домашней контрольной работы №9, была создана программа, для добавления, удаления и изменения информации в файле Excel. Идет взаимодействие программы с Excel, с помощью библиотеки pandas. Также, получил представление об организации хранения и использования информации посредством типизированных файлов, закрепил навыки создания пользовательского интерфейса. Также, опвторил навыки создания программ с помощью библиотеки PyQt5, изучил новую библиотеку pandas и ее функции. Работа была выполнена в полном объеме.