# **ФГБОУ ВО «Московский политехнический университет»**

## **Лабораторная работа №7**

GUI, классы, модуль Tkinter

**Задание 1 Вариант №19**

По дисциплине:

Программирование

Выполнил

студент 1 курса

группы 201-321

Сметанина А.А.

Проверил:

\_\_\_\_\_\_\_\_\_\_\_\_ Никишина И.Н.

# **МОСКВА 2020**

**Цель работы**

Получить практический навык в написании программ, использующих графический

интерфейс пользователя. Познакомиться с такими понятиями, как класс, виджеты, массивы записей, а также с атрибутами класса и методами передачи данных между классами.

**Постановка задачи**

1. Описать запись с именем Price, содержащую следующие поля:

* название товара;
* название магазина, в котором продаётся товар;
* стоимость товара в рублях.

1. Написать программу, выполняющую следующие действия:

* ввод данных с клавиатуры в массив, состоящий из восьми элементов типа Price; записи должны быть размещены в алфавитном порядке по названиям товаров;
* вывод на экран информации о товаре, название которого введено с клавиатуры; если таких товаров нет, вывести на экран соответствующее сообщение;
* запись массива в файл под заданным с клавиатуры именем.

**Теоретическая часть**

Для решения задачи используется графический модуль PyQt5.

**Структура**

**Класс Price**

Описывает товар, хранит имя товара, названия магазина, в котором он продаётся, его стоимость.

**Класс Shop**

Описывает список товаров. Выполняет поиск товаров по названиям магазина, вывод в CSV файл, сохраняет при этом структуру списка, то есть товары отсортированы в алфавитном порядке по названиям.

**Класс MainWindow**

Обеспечивает работу главного окна. Представлен вывод товаров по

мере их ввода, кнопки «Добавить», «В файл» и «Найти», вызывающие соответствующие окна, и кнопка «Выйти», завершающая выполнение программы.

**Класс InputWindow**

Обеспечивает ввод товаров, контролирует корректность данных: не допускается ввод пустых полей, нецифровых символов в строке ввода цены. Пополняет список товаров. Окно ввода нельзя открыть, когда длина списка товаров равна восьми.

**Класс SearchWindow**

Выполняет поиск товаров по названию магазина. Принимает на вход имя товара, выводит «Товары из...» или «Товары из … не найдены».

**Класс SaveWindow**

Обеспечивает ввод имени файла и вывод в него списка товаров. Если

передано имя файла, расширение которого не ‘.csv’, то автоматически добавит это расширение. Если имя файла не введено, но кнопка «Сохранить» нажата, то список выводится в файл «file.csv».

**Класс ErrorWindow**

Обеспечивает вывод сообщений об ошибках в отдельном окне.

Сообщение выводится, например, если нажата кнопка «В файл» или «Найти» в главном окне, когда список товаров пуст.

**Описание программы**

Программа написана на алгоритмическом языке Python 3.6, реализована в среде macOS

Catalina и состоит из частей, отвечающих за ввод данных, вычисление и представление данных на экране монитора.

**Описание алгоритма**

1. Получить на вход товары в количестве восьми штук.
2. Отобразить их в главном окне, отсортировав список по названию магазина в алфавитном порядке.
3. Ожидать от пользователя нажатие какой-либо кнопки, то есть ожидать некоторое событие, которое будет передано соответствующему обработчику.

**Описание входных и выходных данных**

Программа принимает на вход числа и строки, выводит GUI, а также информацию в файлы.

**Листинг программы**

import csv

import os

import sys

import PyQt5.QtWidgets as W

from PyQt5 import uic, QtCore

class Price:

def \_\_init\_\_(self, a, b, c):

self.name = a

self.price = b

self.shop = c

def \_\_str\_\_(self):

return f"Товар {self.name} {self.price}р\nПродаётся в {self.shop}"

class Shop:

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

self.items = a

def to\_file(self, a):

headers = ['Товар', 'Цена', 'Магазин']

with open(a, 'w', encoding='utf-8', newline='') as f:

writer = csv.writer(f)

writer.writerow(headers)

for item in self.items:

writer.writerow([item.name, item.price, item.shop])

def append(self, a):

self.items.append(a)

self.items.sort(key=lambda x: x.shop)

def find(self, a):

res = []

a = a.strip()

for i in self.items:

if i.shop.strip() == a:

res.append(i)

return Shop(res)

def \_\_str\_\_(self):

if len(self.items) == 0:

return ''

return '\n\n'.join(str(item) for item in self.items)

shop = Shop([])

class MainWindow(W.QMainWindow):

def \_\_init\_\_(self, \*args):

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

uic.loadUi('./windows/MainWindow.ui', self)

self.InputWindow = InputWindow(self, [])

self.ErrorWindow = ErrorWindow(self, [])

self.SaveWindow = SaveWindow(self, [])

self.SearchWindow = SearchWindow(self, [])

self.setWindowTitle("Товары")

self.SearchButton.clicked.connect(self.search)

self.DumpButton.clicked.connect(self.to\_file)

self.InputButton.clicked.connect(self.input)

self.ExitButton.clicked.connect(self.close)

self.timer = QtCore.QTimer(self)

self.timer.setInterval(500)

self.timer.start()

self.timer.timeout.connect(self.show)

def input(self):

if len(shop.items) == 8:

self.ErrorWindow.display("8 товаров уже введено")

return

self.InputWindow.show()

def show(self):

a = Shop(shop.items[:4])

b = Shop(shop.items[4:])

self.ItemsBrowser1.setText(str(a))

self.ItemsBrowser2.setText(str(b))

super().show()

def to\_file(self):

if len(shop.items) == 0:

self.ErrorWindow.display("Товары не введены")

return

self.SaveWindow.show()

def search(self):

if len(shop.items) == 0:

self.ErrorWindow.display("Товары не введены")

return

self.SearchWindow.show()

def error(self, a):

self.ErrorWindow.close()

self.ErrorWindow.display(a)

def close(self):

self.ErrorWindow.close()

self.InputWindow.close()

self.SaveWindow.close()

self.SearchWindow.close()

super().close()

class InputWindow(W.QWidget):

def \_\_init\_\_(self, \*args):

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

uic.loadUi('./windows/InputWindow.ui', self)

self.EnterButton.clicked.connect(self.input)

self.ClearButton.clicked.connect(self.clear)

self.setWindowTitle("Ввод товаров")

def clear(self):

self.NameInput.clear()

self.PriceInput.clear()

self.ShopInput.clear()

def input(self):

if not (len(self.NameInput.text().strip()) > 0 and

len(self.PriceInput.text().strip()) > 0 and

len(self.ShopInput.text().strip()) > 0):

self.clear()

return

p = self.PriceInput.text().strip()

try:

p = float(p)

except ValueError:

self.clear()

return

n = self.NameInput.text().strip()

s = self.ShopInput.text().strip()

shop.append(Price(n, p, s))

self.clear()

if len(shop.items) == 8:

self.close()

self.ItemsRemain.setText(f"Осталось ввести: {8 - len(shop.items)}")

def show(self):

self.ItemsRemain.setText(f"Осталось ввести: {8 - len(shop.items)}")

super().show()

class ErrorWindow(W.QWidget):

def \_\_init\_\_(self, \*args):

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

uic.loadUi('./windows/ErrorWindow.ui', self)

self.OkButton.clicked.connect(self.close)

self.setWindowTitle("Ошибка")

def display(self, message):

self.ErrorBrowser.setText(f"Ошибка!\n\n{message}")

self.show()

class SaveWindow(W.QWidget):

def \_\_init\_\_(self, \*args):

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

uic.loadUi('./windows/SaveWindow.ui', self)

self.ExitButton.clicked.connect(self.close)

self.SaveButton.clicked.connect(self.save)

self.setWindowTitle("Сохранить")

def path(self):

name = self.FilenameInput.text().strip()

if len(name) == 0:

raise ValueError("Имя файла не введено")

if not name.endswith('.csv'):

name += '.csv'

return os.getcwd() + '/' + name

def save(self):

try:

path = self.path()

except ValueError:

path = os.getcwd() + '/' + 'file.csv'

shop.to\_file(path)

self.clear()

self.close()

def clear(self):

self.FilenameInput.clear()

def close(self):

self.clear()

super().close()

class SearchWindow(W.QWidget):

def \_\_init\_\_(self, \*args):

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

uic.loadUi('./windows/SearchWindow.ui', self)

self.SearchButton.clicked.connect(self.res)

self.ExitButton.clicked.connect(self.close)

self.setWindowTitle("Поиск по магазинам")

def res(self):

if len(self.QueryInput.text().strip()) == 0:

self.ResultsBrowser.setText("Введите запрос")

return

query = self.QueryInput.text()

self.clear()

shops = shop.find(query)

if not shops.items:

self.ResultsBrowser.setText(f"Товары из '{query}' не найдены")

else:

text = f"Товары из '{query}': "

self.ResultsBrowser.setText(f"{text}\n{shops}")

def clear(self):

self.ResultsBrowser.clear()

self.QueryInput.clear()

def close(self):

self.clear()

super().close()

app = W.QApplication(sys.argv)

main = MainWindow()

main.show()

main.input()

exit(app.exec\_())

**Результат работы программы**



**Список используемой литературы**

1. Лутц М. Программирование на Python, том I, 4-е издание. – Пер. с англ. – СПб.: Символ-Плюс, 2011. – 992 с.
2. Доусон М. Программируем на Python. – СПб.: Питер, 2014. – 416 с.