

МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ  
РОССИЙСКОЙ ФЕДЕРАЦИИ  
ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ БЮДЖЕТНОЕ  
ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ  
«ОРЛОВСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ  
ИМЕНИ И.С. ТУРГЕНЕВА»

Кафедра программной инженерии

**ОТЧЕТ**

по лабораторной работе № 5  
на тему: «Работа с файлами»  
по дисциплине: «Программирование на языке Python»  
Вариант № 7

Выполнила: Карпикова С.П.

Шифр: 170580

Институт приборостроения, автоматизации и информационных технологий

Направление: 09.03.04 «Программная инженерия»

Группа: 71-ПГ

Проверили: Захарова О.В., Раков В.И.

Отметка о зачете:

Дата: «\_\_\_\_» \_\_\_\_\_ 2019 г.

Орел, 2019 г.

## Задание:

В программу лабораторной работы № 4 добавить главное меню с пунктами «Файл», «Справка». Пункт главного меню «Файл» должен включать подпункты «Создать» (очистка формы для ввода новых данных), «Открыть» (считывание из файла информации об объектах и отображение её на форме), «Сохранить» (обновление текущего файла; если файл не создан, то действия пункта «Сохранить как...»), «Сохранить как...» (сохранение информации об объектах в файл по указанному пути), «Выход» (дружественный выход из программы). Пункт главного меню «Справка» должен показывать информацию о разработчике программы.

## Решение:

```
from tkinter import *
from tkinter.ttk import *
from tkinter import messagebox as mb
from tkinter import filedialog

class Medicament(object):
    vendor_code = None
    name = None
    cost = None
    has_prescription = False
    description = None

    def __init__(self, **kwargs):
        self.__dict__.update(kwargs)

    def insert(self, table):
        table.insert("", 'end', text=self.vendor_code,
            values=(
                self.name,
                self.cost,
                'да' if self.has_prescription else 'нет',
                self.description
            )
        )

    def without_prescription(self, table):
        if not self.has_prescription:
            self.insert(table)

    def save(self):
        return '{0} {1} {2} {3} {4}'.format(self.vendor_code, self.name, self.cost,
            'да' if self.has_prescription else 'нет', self.description)

class MainFrame(Frame):
    def __init__(self, parent):
```

```

Frame.__init__(self, parent)
self.parent = parent
self.medicaments = []
self.file_name = ""

self.initUI()
self.bindEvents()

def initUI(self):
    self.parent.title("Лекарства")
    self.pack(fill=BOTH, expand=True)

    self.frame_add_1 = Frame(self)
    self.frame_add_1.pack(fill=X)

    lbl_vendor_code = Label(self.frame_add_1, text="Артикул", width=8)
    lbl_vendor_code.pack(side=LEFT, padx=5, pady=5)

    self.entry_vendor_code = Entry(self.frame_add_1, width=15)
    self.entry_vendor_code.pack(side=LEFT, padx=5)

    lbl_name = Label(self.frame_add_1, text="Название", width=10)
    lbl_name.pack(side=LEFT, anchor=N, padx=5, pady=5)

    self.entry_name = Entry(self.frame_add_1, width=25)
    self.entry_name.pack(side=LEFT, padx=5)

    lbl_cost = Label(self.frame_add_1, text="Стоимость", width=10)
    lbl_cost.pack(side=LEFT, anchor=N, padx=5, pady=5)

    self.entry_cost = Entry(self.frame_add_1, width=15)
    self.entry_cost.pack(side=LEFT, padx=5)

    self.cvar1 = BooleanVar()
    self.cvar1.set(0)
    self.check_prescription = Checkbutton(self.frame_add_1, text="Рецепт", variable=self.cvar1, onvalue=1,
                                           offvalue=0)
    self.check_prescription.pack(anchor=W)

    self.frame_add_2 = Frame(self)
    self.frame_add_2.pack(fill=X)

    lbl_desc = Label(self.frame_add_2, text="Описание", width=10)
    lbl_desc.pack(side=LEFT, anchor=N, padx=5, pady=5)

    self.txt_desc = Text(self.frame_add_2, width=25, height=3)
    self.txt_desc.pack(fill=BOTH, pady=5, padx=5)

    self.btn_add = Button(self.frame_add_2, text="Добавить", width=15)
    self.btn_add.pack()

    self.frame_remove = Frame(self)
    self.frame_remove.pack(fill=X)

    lbl_remove = Label(self.frame_remove, text="Артикул", width=8)
    lbl_remove.pack(side=LEFT, padx=5, pady=5)

    self.entry_remove = Entry(self.frame_remove, width=15)
    self.entry_remove.pack(side=LEFT, padx=5)

    self.btn_remove = Button(self.frame_remove, text="Удалить", width=15)
    self.btn_remove.pack(side=LEFT, padx=5, pady=5)

```

```

self.frame_table = Frame(self)
self.frame_table.pack(fill=X)

self.table = Treeview(self.frame_table)

self.table["columns"] = ("1", "2", "3", "4")

self.table.column("#0", width=80, minwidth=50)
self.table.column("1", width=100, minwidth=70)
self.table.column("2", width=80, minwidth=50)
self.table.column("3", width=50, minwidth=30)
self.table.column("4", width=120, minwidth=100)

self.table.heading("#0", text="Артикул", anchor=W)
self.table.heading("1", text="Название", anchor=W)
self.table.heading("2", text="Стоимость", anchor=W)
self.table.heading("3", text="Рецепт", anchor=W)
self.table.heading("4", text="Описание", anchor=W)

self.table.pack(side=TOP, fill=X)

self.main_frame = Frame(self)
self.main_frame.pack(fill=X)

lb11 = Label(self.main_frame, text="Меню", width=6)
lb11.pack(side=LEFT, padx=5, pady=5)

self.v = StringVar()
self.v.trace('w', self.on_combo_change)

self.combo = Combobox(self.main_frame, state='readonly', width=25, textvar=self.v)
self.combo['values'] = (
    "Добавить лекарство", "Удалить лекарство", "Вывод списка лекарств", "Лекарства без рецепта")
self.combo.current(0)
self.combo.pack(side=LEFT, padx=5, pady=5)

self.mainmenu = Menu(self.parent)

self.parent.config(menu=self.mainmenu)

self.filemenu = Menu(self.mainmenu, tearoff=0)
self.filemenu.add_command(label="Создать", command=self.create)
self.filemenu.add_command(label="Открыть...", command=self.download_file)
self.filemenu.add_command(label="Сохранить...", state=DISABLED, command=self.save_file)
self.filemenu.add_command(label="Сохранить как...", state=DISABLED, command=self.save_as_file)
self.filemenu.add_command(label="Выход", command=self.ask_exit)

helpmenu = Menu(self.mainmenu, tearoff=0)
helpmenu.add_command(label="О программе", command=self.about)

self.mainmenu.add_cascade(label="Файл", menu=self.filemenu)
self.mainmenu.add_cascade(label="Справка", menu=helpmenu)

def bindEvents(self):
    self.btn_add.bind('<Button-1>', self.add_new_medicament)
    self.btn_remove.bind('<Button-1>', self.delete_medicament)

def hide(self):
    self.frame_add_1.pack_forget()
    self.frame_add_2.pack_forget()
    self.frame_remove.pack_forget()

```

```

self.frame_table.pack_forget()

def on_combo_change(self, index, value, op):
    self.hide()
    current_state = self.combo.get()
    if current_state == "Добавить лекарство":
        self.frame_add_1.pack(fill=X)
        self.frame_add_2.pack(fill=X)

    elif current_state == "Удалить лекарство":
        self.frame_remove.pack(fill=X)

    elif current_state == "Вывод списка лекарств":
        self.frame_table.pack(fill=X)
        self.get_medicaments()
    else:
        self.frame_table.pack(fill=X)
        self.get_without_prescription()

def add_new_medicament(self, event):
    vendor_code = self.entry_vendor_code.get()
    name = self.entry_name.get()
    cost = self.entry_cost.get()
    has_prescription = bool(self.cvar1.get())
    description = self.txt_desc.get("1.0", END)

    if len(vendor_code) == 0 or len(name) == 0 or len(cost) == 0 or len(description) == 0:
        mb.showerror("Ошибка", "Все поля должны быть заполнены!")
        return

    if not cost.isdigit():
        mb.showerror("Ошибка", "Стоимость должна иметь численное значение!")
        return

    self.medicaments.append(Medicament(
        vendor_code=vendor_code,
        name=name,
        cost=cost,
        has_prescription=has_prescription,
        description=description)
    )
    self.clear_text()
    mb.askokcancel("Успешно", "Информация добавлена!")
    self.filemenu.entryconfig(3, state=NORMAL)
    if self.file_name:
        self.filemenu.entryconfig(2, state=NORMAL)

def get_medicaments(self):
    self.table.delete(*self.table.get_children())

    for item in self.medicaments:
        item.insert(self.table)

def get_without_prescription(self):
    self.table.delete(*self.table.get_children())

    for item in self.medicaments:
        item.without_prescription(self.table)

def clear_text(self):
    self.entry_vendor_code.delete(0, 'end')
    self.entry_name.delete(0, 'end')

```

```

self.entry_cost.delete(0, 'end')
self.cvar1.set(0)
self.txt_desc.delete("1.0", END)
self.entry_remove.delete(0, 'end')

def delete_medicament(self, event):
    vendor_code = self.entry_remove.get()

    if len(vendor_code) == 0:
        mb.showerror("Ошибка", "Заполните поле!")
        return

    self.medicaments = [item for item in self.medicaments if item.vendor_code != vendor_code]
    self.clear_text()
    mb.askokcancel("Успешно", "Удалено!")
    self.filemenu.entryconfig(2, state=NORMAL)

def create(self):
    self.ask_save()
    self.file_name = ""
    self.clear_text()
    self.medicaments.clear()
    self.table.delete(*self.table.get_children())
    self.combo.current(0)
    self.filemenu.entryconfig(2, state=DISABLED)
    self.filemenu.entryconfig(3, state=DISABLED)

def download_file(self):
    self.ask_save()

    self.table.delete(*self.table.get_children())
    self.medicaments.clear()

    self.file_name = filedialog.askopenfilename(filetypes=[("Text files", "*.txt")])

    with open(self.file_name, "r", encoding='utf-8') as f:
        f.seek(0)
        for line in f:
            info = line.split()
            self.medicaments.append(Medicament(
                vendor_code=info[0],
                name=info[1],
                cost=info[2],
                has_prescription=True if info[3] == 'да' else False,
                description=info[4])
            )

    self.filemenu.entryconfig(2, state=NORMAL)
    self.combo.current(2)

def save_file(self):
    with open(self.file_name, "w", encoding='utf-8') as f:
        f.seek(0)
        for item in self.medicaments:
            f.write(item.save())

    mb.askokcancel("Успешно", "Файл успешно сохранён!")
    self.filemenu.entryconfig(2, state=DISABLED)

def save_as_file(self):
    self.file_name = filedialog.asksaveasfilename(defaulttextextension=".txt")
    if self.file_name:

```

```

        self.save_file()

    def ask_save(self):
        answer = mb.askyesno("Сохранить", "Сохранить изменения?")
        if answer == True:
            if len(self.file_name) > 0:
                self.save_file()
            else:
                self.save_as_file()

    def ask_exit(self):
        answer = mb.askyesno("Выход", "Вы действительно хотите выйти?")
        if answer == True:
            self.quit()

    def about(self):
        mb.askokcancel("о программе", "Сделала Сонька Карпикова из 71 ПГ!")

def main():
    root = Tk()
    root.geometry("700x300")
    app = MainFrame(root)
    root.mainloop()

if __name__ == '__main__':
    main()

```