



Государственное бюджетное образовательное учреждение высшего образования
Московской области

ТЕХНОЛОГИЧЕСКИЙ УНИВЕРСИТЕТ

Колледж космического машиностроения и технологии

КУРСОВОЙ ПРОЕКТ

По дисциплине «Прикладное программирование»

Тема: «Разработка приложения "Автотехцентр": главный модуль».

Выполнил студент

Бобнев Алексей Олегович

Группа П1-17

_____ (Подпись)

_____ (Дата сдачи работы)

Проверил преподаватель

Гусятинер Леонид Борисович

_____ (Подпись)

_____ (Оценка)

Королев 2020

Содержание

Введение	4
Глава 1. Теоретическая часть	5
1.1. Описание предметной области	5
1.2. Описание существующих разработок	6
Глава 2. Проектная часть	9
2.1. Диаграмма прецедентов	9
2.2. Выбор инструментов	11
2.3. Проектирование сценария	12
2.4. Диаграмма классов	13
2.5. Описание главного модуля	14
2.6. Описание спецификаций к модулям	19
2.7. Описание модулей	20
2.8. Описание тестовых наборов модулей	27
2.9. Описание применения средств отладки	29
2.10. Анализ оптимальности использования памяти и быстродействия	30
Глава 3. Эксплуатационная часть	32
Аннотация	32
3.1. Руководство оператора	33
3.2. Список функций	33
3.3. Требования к персоналу (пользователю)	33
3.4. Минимальный состав аппаратных средств	34
3.5. Сообщение оператору:	34
Заключение	35
Список литературы и интернет-источников	36
Приложение 1. Код главного меню	37
Приложение 2. Модуль «Часы»	40
Приложение 3. Окно «Финансы»	41

Приложение 4. Окно «Сотрудники»	43
Приложение 5. Окно «Тест-драйв»	45
Приложение 6. Окно «Автозапчасти».....	46
Приложение 7. Окно «Диагностика»	48
Приложение 8. Окно «Клиенты».....	51
Приложение 9. Окно «О программе».....	53
Приложение 10. Окно «Прайс-лист».....	54
Приложение 11. Окно «Разработчики».....	54

Введение

Данный курсовой проект посвящен созданию автомобильной программе. Это тема актуальна, так как, в России очень много автосервисов и у меня есть семейный бизнес по ремонту автомобилей, поэтому эта тема мне близка.

Цель курсового проекта заключалась в разработке: «Программного обеспечения для автотехцентра».

В 1 главе рассматривается теоретическая часть программы, для понимания специфики выбранной тематики.

Во 2 главе рассматривается проектная часть программы, в которой присутствует описание программы.

В 3 главе рассматривается эксплуатационная часть программы, в которой описывается, как в программе работать.

Глава 1. Теоретическая часть

1.1. Описание предметной области

Автомобильное дело - это очень интересная и прибыльная сфера бизнеса. Владелец может заниматься управлением самостоятельно и организовать работу так, чтобы предприятие приносило хорошую прибыль, но это сможет далеко не каждый. В таком случае лучше прибегнуть к услугам профессионала.

Администратор автотехцентра – это управляющий предприятием, который организует работу по предоставлению услуг клиентам по ремонту автомобилей.

Автослесарь автотехцентра – это работник предприятия, который выполняет ремонт автомобильной техники клиентов.

В основном станции технического обслуживания предпочитают пользоваться специальной программой. Она позволяет грамотно распланировать совершение всех операций в сервисе и обеспечивает удобное хранение информации о клиентах, что открывает новые перспективы для развития предприятия.

Интерфейс таких программ чаще всего простой и понятный для обычных пользователей.

Цели использования системы управления автотехцентра:

- Учет персонала.
- Учет выручки.
- Учет автозапчастей.
- Учет клиентов.

Основные требования:

- Консультация клиентов по поводу ремонта автомобиля.
- Оплата по готовности заказа.

Современная станция технического обслуживания представляет собой интересный комплекс, в управлении которым особо важна точность, оперативность и удобство, ведь если допустить ошибку в такой сфере деятельности может обернуться большой проблемой. Если не обеспечить максимальную эффективность учета этих процессов, то работа всего предприятия будет неэффективной.

1.2. Описание существующих разработок

Delphi DS150E – это программа для диагностики автомобиля, хороший помощник автомобильному предприятию.

- Программа универсальна, то есть осуществляет диагностику всех транспортных средств.
- Настраивается в зависимости от размера и объемов активности ведения бизнеса.
- Соответствует самым требовательным запросам.
- Гарантирует достоверную информацию.
- Имеет богатый функционал.

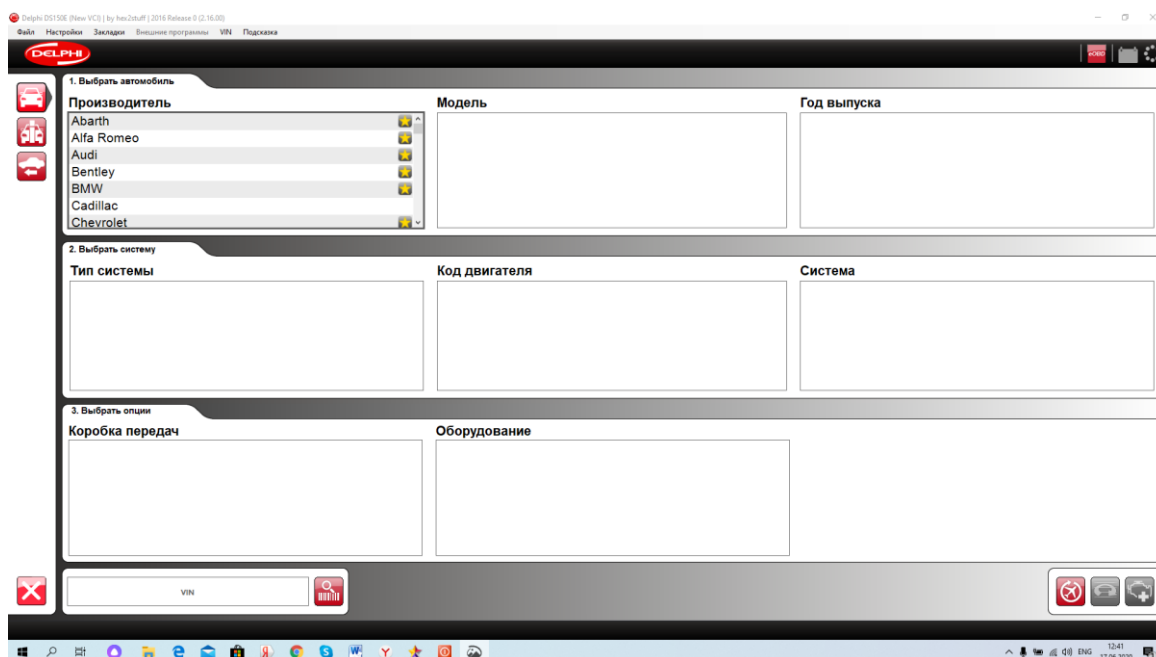


Рисунок 1. Интерфейс программы «Delphi DS150E».

Возможности автосканера:

Delphi DS150E считается наилучшим из бюджетных вариантов дешифровщиков для автомобилей.

Перечень его функциональных возможностей включает:

- Считывания, сброса неисправностей;
- Проверки в режиме реального времени;
- Адаптации значений приборов;
- Тестирование.

Чтение и сброс ошибок:

Основная функция программы, к названию которой часто добавляется качество «3 в 1» - считывание, расшифровка и опция сбрасывание ошибок. Речь идет о возможности сопряжения программы с легковыми, небольшими грузовыми авто, а также о возможности контроля качества по европейским стандартам.

Сканирование конкретных узлов или общего состояния автомобиля, а также выведение кодов найденных проблем. Дальнейшая расшифровка полученных значений и предоставление информации о том, что найденная нестыковка является ли неисправностью или просто сбоем в сообщении между элементами. Отдельно можно получить достоверные данные о пробеге автомобиля.

Проверка в режиме реального времени:

Позволяет выявить проблемы непосредственно при работе всех систем на данный момент.

Адаптация:

Проведение ремонта или профилактического обслуживания (чистки) узлов может привести к сбою нулевых настроек. Восстановить их состояние, адаптировать, способен Delphi DS150e.

Тестирование:

Сканирование состояния электроники способно выявить большую часть проблем.

Ссылка для скачивания программы:

- <https://softdroids.com/492-ds150e.html>

Глава 2. Проектная часть

2.1. Диаграмма прецедентов

В этом разделе представлено четыре диаграммы прецедентов. На первой диаграмме (рисунок № 2) показаны все возможности генерального директора. На второй диаграмме (рисунок № 3) показаны все возможные действия администратора. На третьей диаграмме (рисунок № 4) показаны все возможные действия автослесаря. И на четвертой диаграмме (рисунок №5) показана связь между клиентом и программой.



Рисунок 2. Генеральный директор.

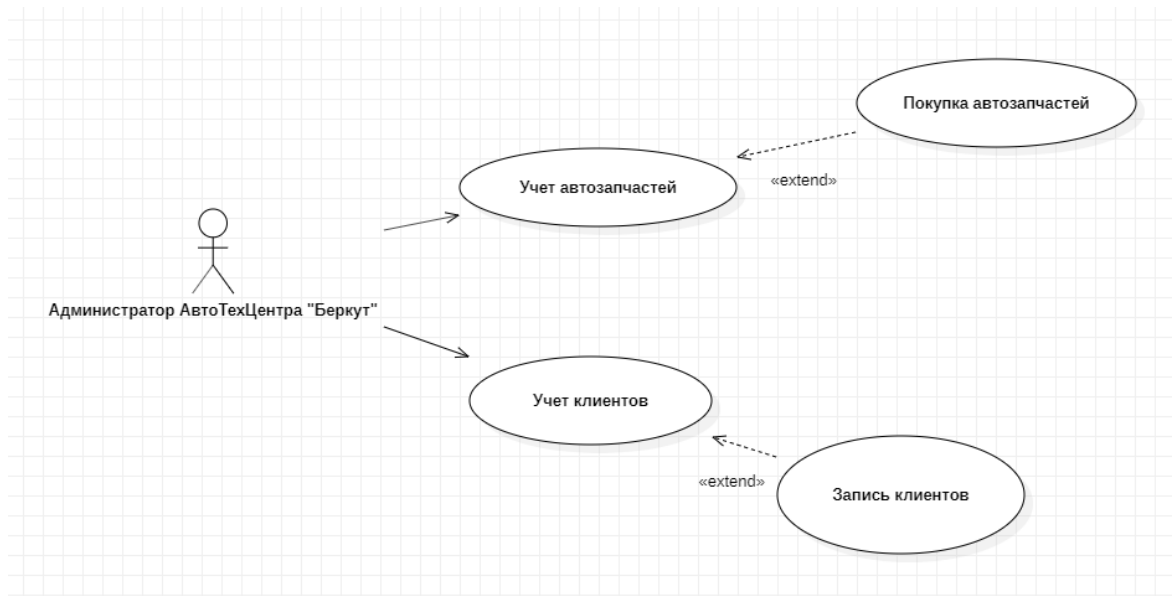


Рисунок 3. Администратор.

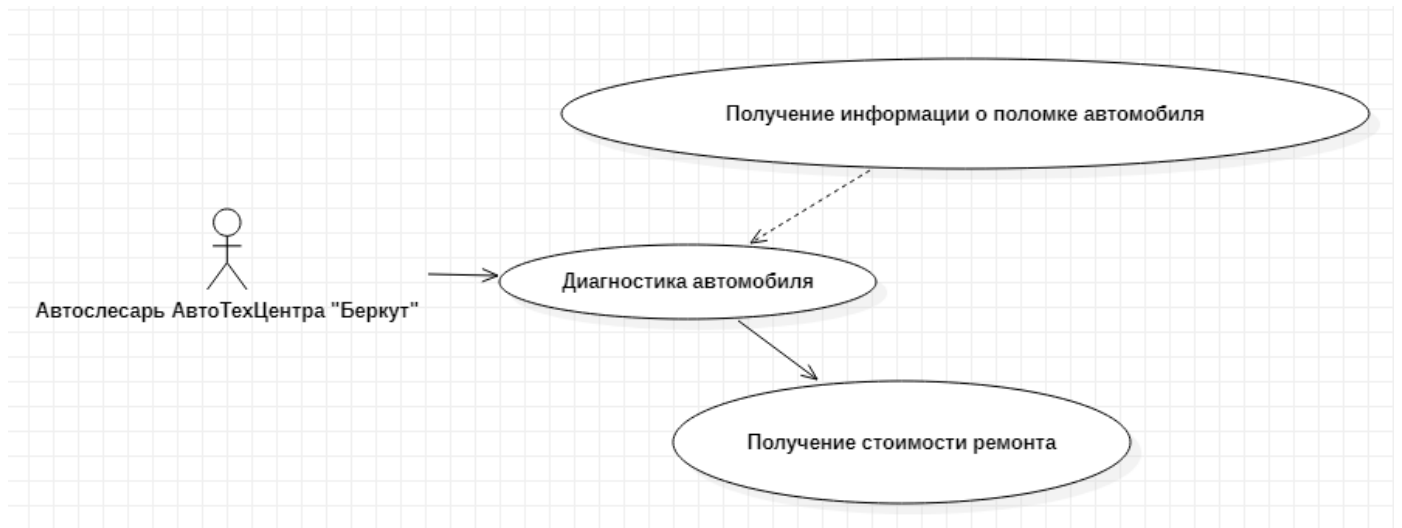


Рисунок 4. Автослесарь.

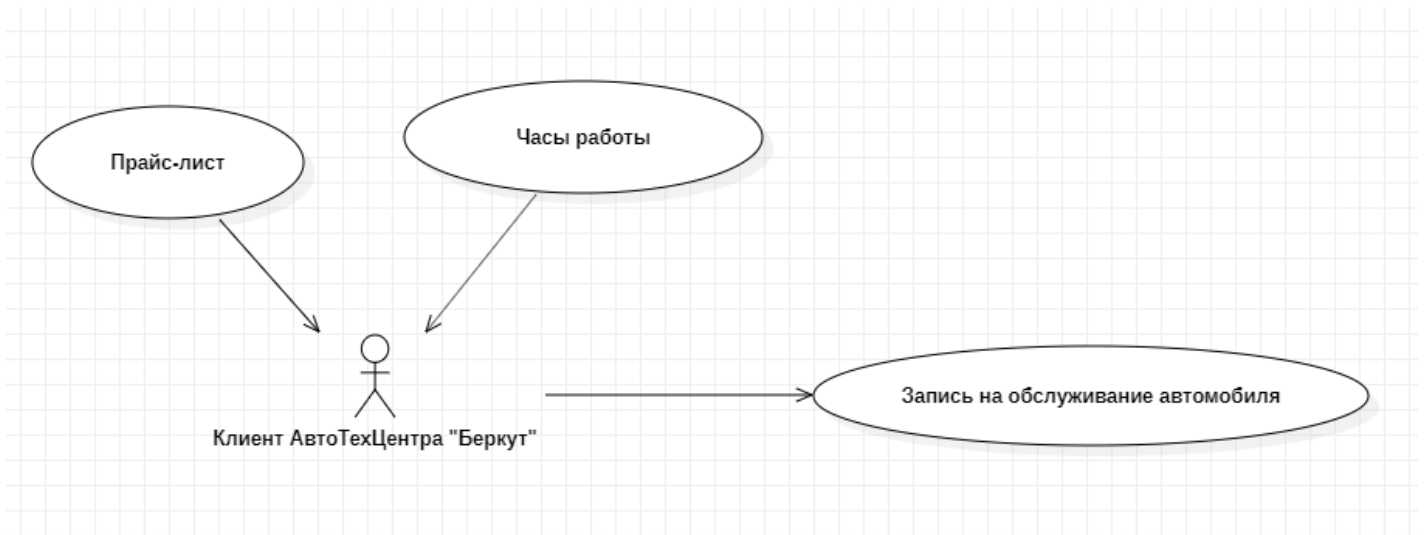


Рисунок 5. Клиент.

2.2. Выбор инструментов

Мой выбор инструментов зависел от нескольких факторов:

- Наличие документации на русском языке.
- Участие в корпоративном проекте.

Так же немало важные факторы:

- Какие библиотеки существуют, в данном случае мне нужна была библиотека, в которой можно было работать с интерфейсом - это Tkinter.
- Скорость разработки программы.

Все выше сказанные условия сыграли на выбор Python как среда разработки и Tkinter как библиотека.

Оба этих средства являются бесплатными, доступными и оба обеспечивают быструю скорость разработки.

Ссылка для скачивания программы Python:

- <https://www.python.org/downloads/release/python-383/>

2.3. Проектирование сценария

На рисунке №6 показан сценарий автомобильной программы.

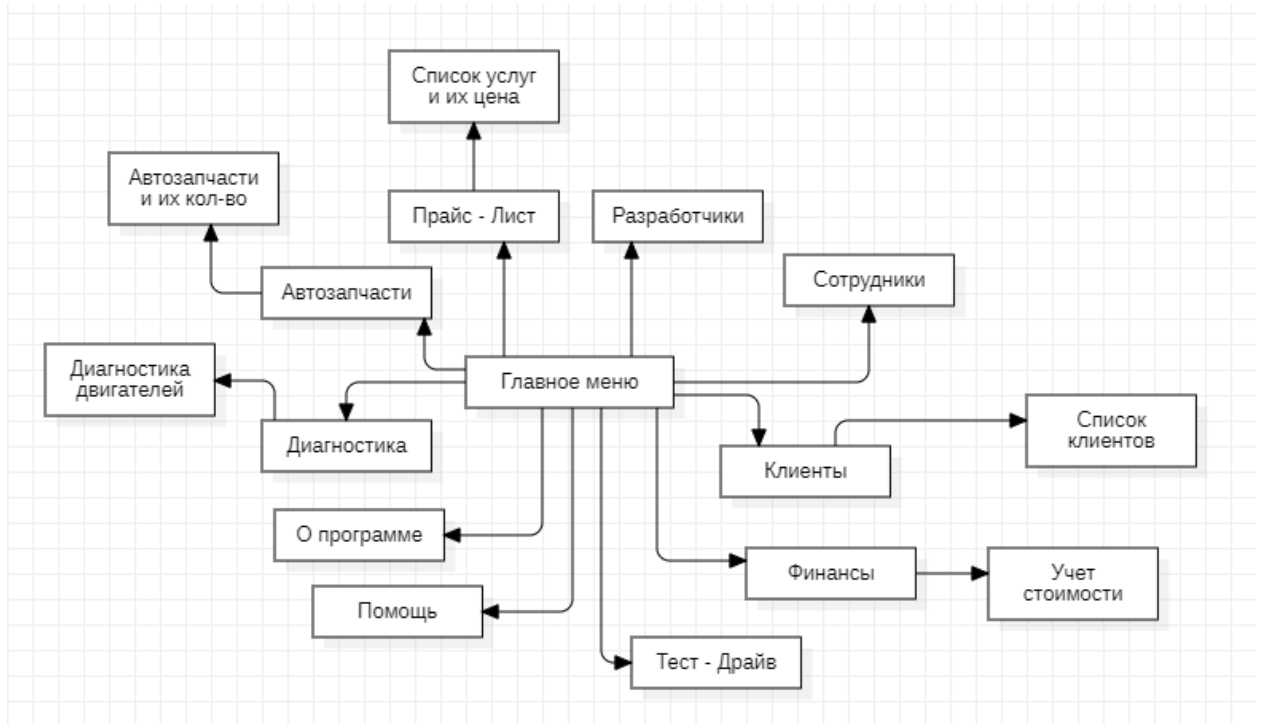


Рисунок 6. Сценарий программы.

Пользователь после запуска программы увидит: меню из 10 кнопок, которые ему доступны.

2.4. Диаграмма классов

В данном разделе представлены все классы, использующиеся в проекте (рисунок №7).

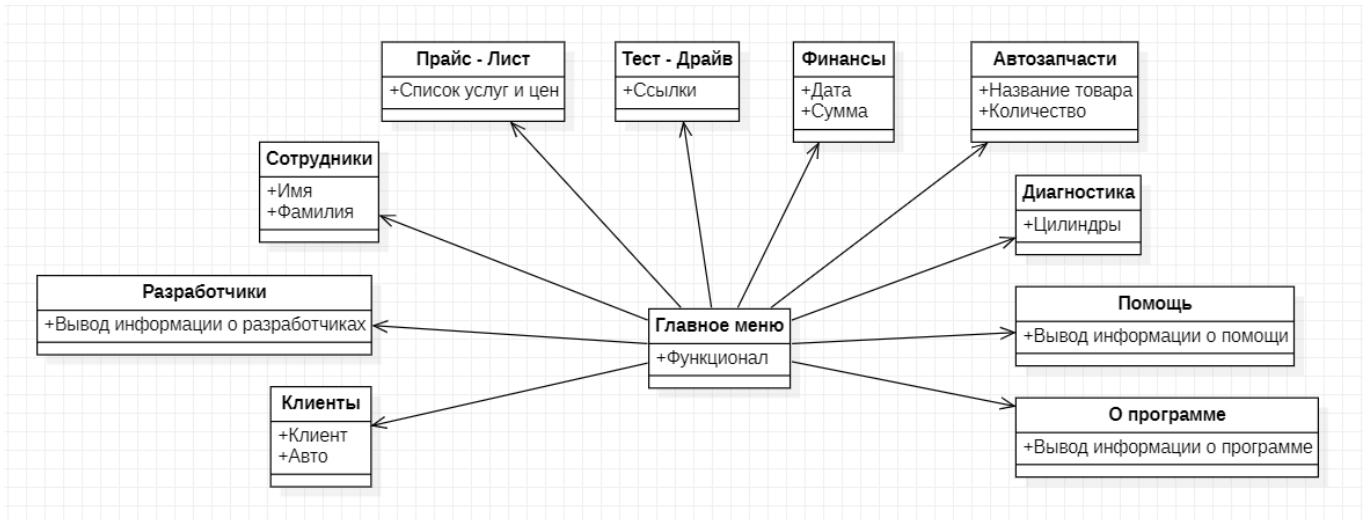


Рисунок 7. Диаграмма классов.

2.5. Описание главного модуля

Главный модуль - представляет из себя 10 кнопок с фоном на заднем плане программы.

Пользователь выбирает ему нужную кнопку и нажимает на нее, и в результате появляется окно, которое он выбрал.

Так же слева вверху мы видим три вкладки, которые тоже необходимы для работы пользователя.

На рисунке №8 представлен интерфейс программы.

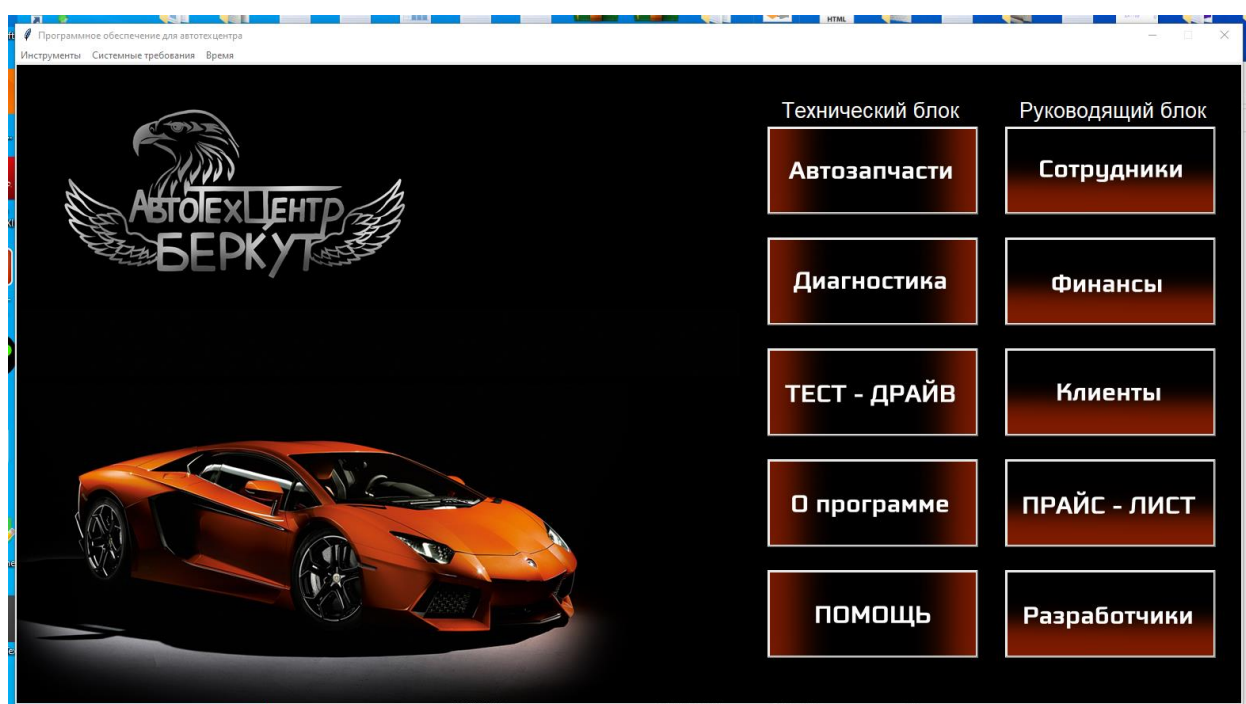


Рисунок 8. Интерфейс программы.

Листинг 1. Код главного меню.

```
#=====
#Главный модуль программы
#=====
from tkinter import *
from tkinter import messagebox
from tkinter import scrolledtext
from tkinter import ttk
from datetime import datetime
import ctypes
import Avto_zapchasti, Finance
import Klient, Sotrudniki, Price, Diagnostika
import Razrabotchiki, Chasi, O_prodramme, Test_drive
```

```

def Kalkulator():
    root = Tk()
    root.geometry("240x230")
    root.resizable(False, False)
    root.title("Калькулятор")
    root.configure(bg = 'black')
    def calc(key):
        global memory
        if key == "=":
            str1 = "-+0123456789.*/"
            if calc_entry.get()[0] not in str1:
                calc_entry.insert(END, "Первый символ не число")
                messagebox.showerror("Ошибка!", "Вы ввели не число!")
            try:
                result = eval(calc_entry.get())
                calc_entry.insert(END, "=" + str(result))
            except:
                calc_entry.insert(END, "Ошибка!")
                messagebox.showerror("Ошибка!", "Проверьте правильность
данных.")
        elif key == "C":
            calc_entry.delete(0, END)
        elif key == "-/+":
            if "=" in calc_entry.get():
                calc_entry.delete(0, END)
            try:
                if calc_entry.get()[0] == "-":
                    calc_entry.delete(0)
                else:
                    calc_entry.insert(0, "-")
            except IndexError:
                pass
        else:
            if "=" in calc_entry.get():
                calc_entry.delete(0, END)
            calc_entry.insert(END, key)
    btnn_list = ["7", "8", "9", "+", "-",
                  "4", "5", "6", "*", "/",
                  "1", "2", "3", "-/+", "=",
                  "0", ".", "C", "", ""]

    r = 1
    c = 0
    for i in btnn_list:
        rel = ""
        cmd = lambda x = i: calc(x)
        b = Button(root, bg = 'black', fg = 'white', font = 20, text = i,
width = 4, height = 2, command = cmd).grid(row = r, column = c)
        c += 1
        if c > 4:
            c = 0
            r += 1
    calc_entry = Entry(root, width = 33)
    calc_entry.grid(row = 0, column = 0, columnspan = 5)

```

```

    root.mainloop()

def sprav():
    okno1 = Tk()
    okno1.geometry("1600x860")
    okno1.title("Программное обеспечение для автосервиса")
    okno1.resizable(False, False)
    okno1.configure(bg = 'black')
    bbtn1 = Label(okno1)
    bbtn1.configure(text = 'В разработке', font = ('Batang 20', 70), bg =
'black', fg = 'white')
    bbtn1.place(x = 500, y = 400)
    okno1.mainloop()

def system():
    okno1 = Toplevel()
    okno1.geometry("1000x700")
    okno1.title("Системные требования")
    okno1.resizable(False, False)
    okno1.configure(bg = 'black')
    c = Canvas(okno1, width = 1800, height = 800, bg = 'white')
    c.pack()
    osnovnoi_fon = PhotoImage(file = 'Sis.png')
    osnovnoi_fon1 = c.create_image(500, 300, image = osnovnoi_fon)
    sis1 = Label(okno1)
    sis1.configure(text = 'Минимальные требования\n ОС: Windows 7/8/10\n
Процессор: 2.0 GHz\n Оперативная память: 4 GB ОЗУ\n Видеокарта: NVIDIA
GeForce GT 520\n Место на диске: 4 MB' , font = 'Batang 20', bg = 'black', fg
= 'white')
    sis1.place(x = 500, y = 40)
    okno1.mainloop()

def celi():
    okno1 = Toplevel()
    okno1.geometry("1600x860")
    okno1.title("Цели")
    okno1.resizable(False, False)
    okno1.configure(bg = 'black')
    c = Canvas(okno1, width = 1920, height = 1080, bg = 'white')
    c.pack()
    osnovnoi_fon = PhotoImage(file = 'Sis.png')
    osnovnoi_fon1 = c.create_image(700, 400, image = osnovnoi_fon)
    okno1.mainloop()

def Help1():
    okno1 = Toplevel()
    okno1.geometry("1000x600")
    okno1.title("Помощь")
    okno1.resizable(False, False)
    okno1.configure(bg = 'black')
    c = Canvas(okno1, width = 1800, height = 800, bg = 'white')
    c.pack()
    osnovnoi_fon = PhotoImage(file = 'Help1.png')
    osnovnoi_fon1 = c.create_image(500, 300, image = osnovnoi_fon)
    sis1 = Label(okno1)
    sis1.configure(text = 'Позвоните по телефону:\n 8 498 601 22 01\n
Сообщите в техническую поддержку:\n версию программы и индивидуальный номер

```



```

программы\n Версия 0.0.4\n WBK067JPA' , font = 'Batang 20', bg = 'black', fg
= 'white')
    sis1.place(x = 200, y = 40)
    okno1.mainloop()

def clock():
    import sekyndomer
okno1 = Tk()
okno1.geometry("1550x810")
okno1.title("Программное обеспечение для автотехцентра")
okno1.resizable(False, False)
c = Canvas(okno1, width = 1920, height = 1080, bg = 'white')
c.pack()
gg1 = Label(okno1)
gg1.configure(text = 'Технический блок', font = 'Batang 20', bg = 'black', fg
= 'white')
gg1.place(x = 965, y = 40)
gg2 = Label(okno1)
gg2.configure(text = 'Руководящий блок', font = 'Batang 20', bg = 'black', fg
= 'white')
gg2.place(x = 1265, y = 40)
main_menu = Menu()
osnovnoi_fon = PhotoImage(file='osnovnoi_fon.png')
z = PhotoImage(file = 'avtozapchasti.png')
f = PhotoImage(file = 'finansi.png')
t = PhotoImage(file = 'test_drive.png')
h = PhotoImage(file = 'help.png')
cr = PhotoImage(file = 'razrabotchiki.png')
kl = PhotoImage(file = 'klienti.png')
opr = PhotoImage(file = 'o_programme.png')
prl = PhotoImage(file = 'price_list.png')
s = PhotoImage(file = 'sotrydniki.png')
dd = PhotoImage(file = 'diagnostika.png')
osnovnoi_fon1 = c.create_image(830, 455, image = osnovnoi_fon)
Button(okno1, width = 260, height = 105, font = "Batang 20", text =
"Автозапчасти", command = Avto_zapchasti.Avto_zapchasti, image = z).place(x =
950, y = 80)
Button(okno1, width = 260, height = 105, font = "Batang 20", text =
"Финансы", command = Finance.Finance, image = f).place(x = 1250, y = 220)
Button(okno1, width = 260, height = 105, font = "Batang 20", text = "Тест -
драйв", command = Test_drive.Test_drive, image = t).place(x = 950, y = 360)
Button(okno1, width = 260, height = 105, font = "Batang 20", text = "О
программе", command = O_prodramme.O_prodramme, image = opr).place(x = 950, y
= 500)
Button(okno1, width = 260, height = 105, font = "Batang 20", text =
"Разработчики", command = Razrabotchiki.Razrabotchiki, image = cr).place(x =
1250, y = 640)
Button(okno1, width = 260, height = 105, font = "Batang 20", text =
"Сотрудники", command = Sotrudniki.Sotrudniki, image = s).place(x = 1250, y =
80)
Button(okno1, width = 260, height = 105, font = "Batang 20", text =
"Диагностика\ндвигателя", command = Diagnostika.Diagnostika, image =
dd).place(x = 950, y = 220)

```

```

Button(okno1, width = 260, height = 105, font = "Batang 20", text =
"Клиенты", command = Klient.Klient, image = kl).place(x = 1250, y = 360)
Button(okno1, width = 260, height = 105, font = "Batang 20", text = "Прайс -
лист", command = Price.Price, image = prl).place(x = 1250, y = 500)
Button(okno1, width = 260, height = 105, font = "Batang 20", text = "Помощь",
command = Help1, image = h).place(x = 950, y = 640)
file_menu = Menu(tearoff = 0)
file_menu1 = Menu(tearoff = 0)
file_menu2 = Menu(tearoff = 0)
file_menu3 = Menu(tearoff = 0)
file_menu4 = Menu(tearoff = 0)
file_menu.add_command(label = "Калькулятор", command = Kalkulator)
file_menu1.add_command(label = "Требования", command = system)
file_menu3.add_command(label = "Часы", command = Chasi.clock)
main_menu.add_cascade(label="Инструменты", menu = file_menu)
main_menu.add_cascade(label="Системные требования", menu = file_menu1)
main_menu.add_cascade(label="Время", menu = file_menu3)
okno1.config(menu = main_menu)
okno1.mainloop()

```

2.6. Описание спецификаций к модулям

В данном разделе описаны спецификации к модулям. Спецификации к модулям заключается в следующем: использование методов библиотеки Tkinter.

Методы:

- Geometry - размер окна.
- Title - название окна.
- Resizable – метод для изменения окна.
- Bg – фон окна.
- Place – место объекта на экране.
- Command – переход на команду.
- Label – метка.
- PhotoImage – импорт фотографии.
- Configure – настройки окна.

Данные методы позволили работать с интерфейсом автомобильной программы.

2.7. Описание модулей

Рассмотрим один из модулей программы - это модуль «Финансы».

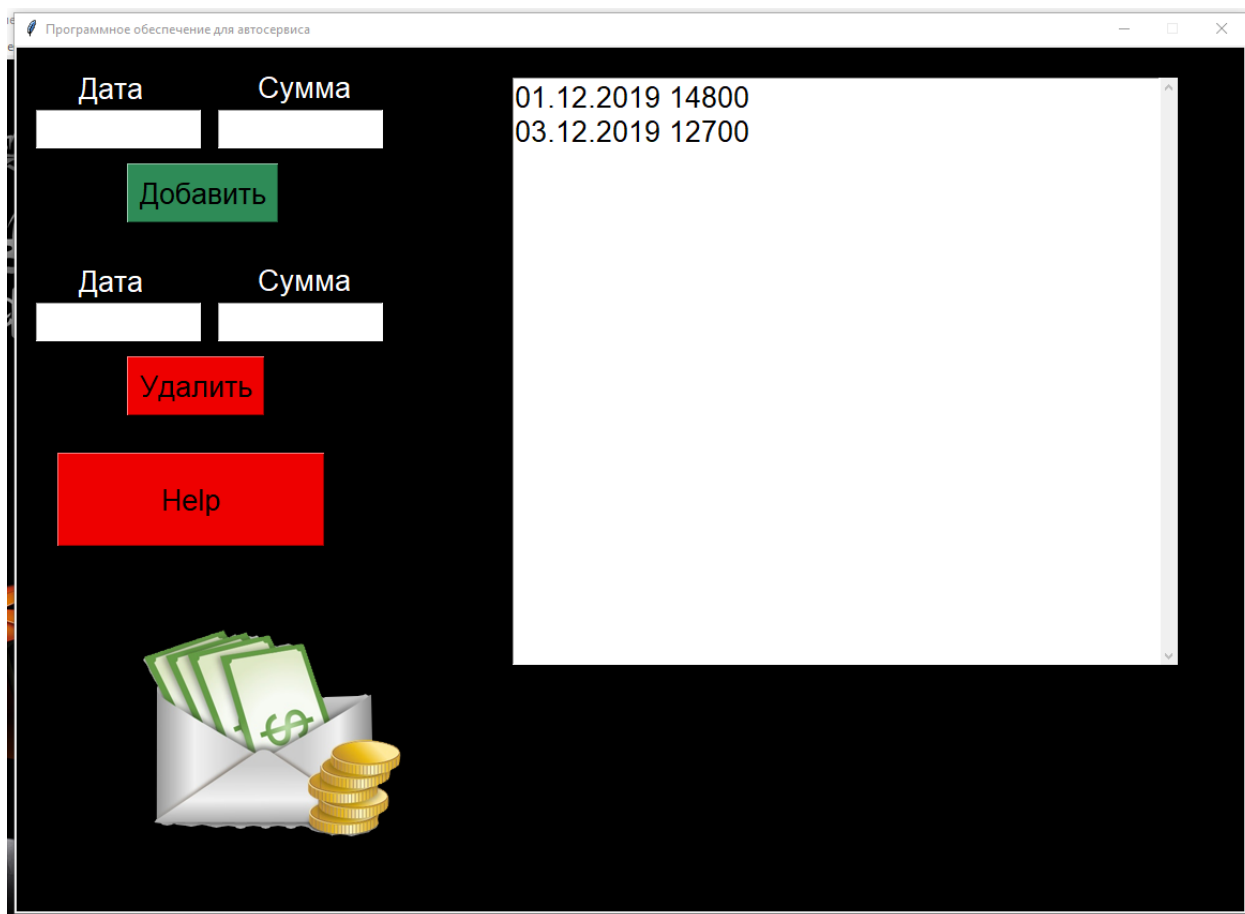


Рисунок 11. Окно «Финансы».

В данном окне мы можем добавить или удалить запись, которая информирует нас о финансовых делах предприятия.

Листинг 3. Окно «Финансы».

```
#=====
#С помощью этого модуля можно ввести учет финансовой деятельности
#=====
from tkinter import *
from tkinter import messagebox
from tkinter import scrolledtext
from tkinter import ttk
from datetime import datetime
import ctypes

def Finance():
    window = Toplevel()
    window.geometry("1150x810")
    window.title("Финансы")
```

```

window.resizable(False, False)
c = Canvas(window, width = 1920, height = 1080, bg = 'white')
c.pack()
osnovnoi_fon = PhotoImage(file='fin2.png')
osnovnoi_fon1 = c.create_image(700, 400, image = osnovnoi_fon)
user32 = ctypes.windll.user32
widthButtonPlace = (user32.GetSystemMetrics(0) // 15)
widthButton = widthButtonPlace // 7
f = open('bug.txt', 'r', encoding = 'utf-8')
text = Text()
text = scrolledtext.ScrolledText(window, height = '17', font = 'Batang
20', width = '40')
text.insert(INSERT, f.read())
text.place(x = 465, y = 30)
f.close()
def clicked():
    z = open('bug.txt', 'r', encoding = 'utf-8')
    y = z.read()
    x = y.split()
    t = []
    for i in range (len(x)):
        if i%2 == 0:
            u = x[i] + ' ' + x[i+1]
            t.append(u)
    b = txt1.get()
    b1 = txt2.get()
    f = open('bug.txt', 'a', encoding = 'utf-8')
    u11 = txt1.get() + ' ' + txt2.get() + '\n'
    f.write(u11)
    f.close()
    f = open('bug.txt', 'r', encoding = 'utf-8')
    text = Text()
    text = scrolledtext.ScrolledText(window, height = '17', font =
'Batang 20', width = '40')
    text.insert(INSERT, f.read())
    text.place(x = 465, y = 30)
    f.close()
def clicked1():
    z = open('bug.txt', 'r', encoding = 'utf-8')
    y = z.read()
    x = y.split()
    t = []
    for i in range (len(x)):
        if i%2 == 0:
            u = x[i] + ' ' + x[i+1]
            t.append(u)
    b = txt3.get()
    b1 = txt4.get()
    v = open('bug.txt', 'w', encoding = 'utf-8')
    for i in range(len(x)):
        j = t[i].split()
        if i%2 == 0 and j[0] != b:
            v.write(j[0])

```

```

        v.write(' ')
        v.write(j[1])
        v.write('\n')
    v.close()
    f = open('bug.txt', 'r', encoding = 'utf-8')
    text = Text()
    text = scrolledtext.ScrolledText(window, height = '17', font =
'Batang 20', width = '40')
    text.insert(INSERT, f.read())
    text.place(x = 465, y = 30)
    f.close()
    lbl1 = Label(window)
    lbl2 = Label(window)
    lbl1.configure(font = 'Batang 20', text = '    Дата', bg = 'black', fg =
'white')
    lbl1.place(x = 25, y = 21)
    lbl2.configure(font = 'Batang 20', text = '    Сумма', bg = 'black', fg =
'white')
    lbl2.place(x = 200, y = 20)
    txt1 = Entry(window, width = 10, font = 'Batang 20')
    txt1.place(x = 20, y = 60)
    txt2 = Entry(window, width = 10, font = 'Batang 20')
    txt2.place(x = 190, y = 60)
    btn = Button(window, text = "Добавить", font = 'Batang 20',
command=clicked, bg = 'seagreen')
    btn.place(x = 105, y = 110)
    lbl3 = Label(window)
    lbl4 = Label(window)
    lbl3.configure(font = 'Batang 20', text = '    Дата', bg = 'black', fg =
'white')
    lbl3.place(x = 25, y = 201)
    lbl4.configure(font = 'Batang 20', text = '    Сумма', bg = 'black', fg =
'white')
    lbl4.place(x = 200, y = 200)
    txt3 = Entry(window, width = 10, font = 'Batang 20')
    txt3.place(x = 20, y = 240)
    txt4 = Entry(window, width = 10, font = 'Batang 20')
    txt4.place(x = 190, y = 240)
    btn1 = Button(window, text = "Удалить", font = 'Batang 20', command =
clicked1, bg = 'red2')
    btn1.place(x = 105, y = 290)
    window.mainloop()

```

Рассмотрим модуль автомобильной программы – это окно «Сотрудники». На рисунке №12 представлен интерфейс модуля.

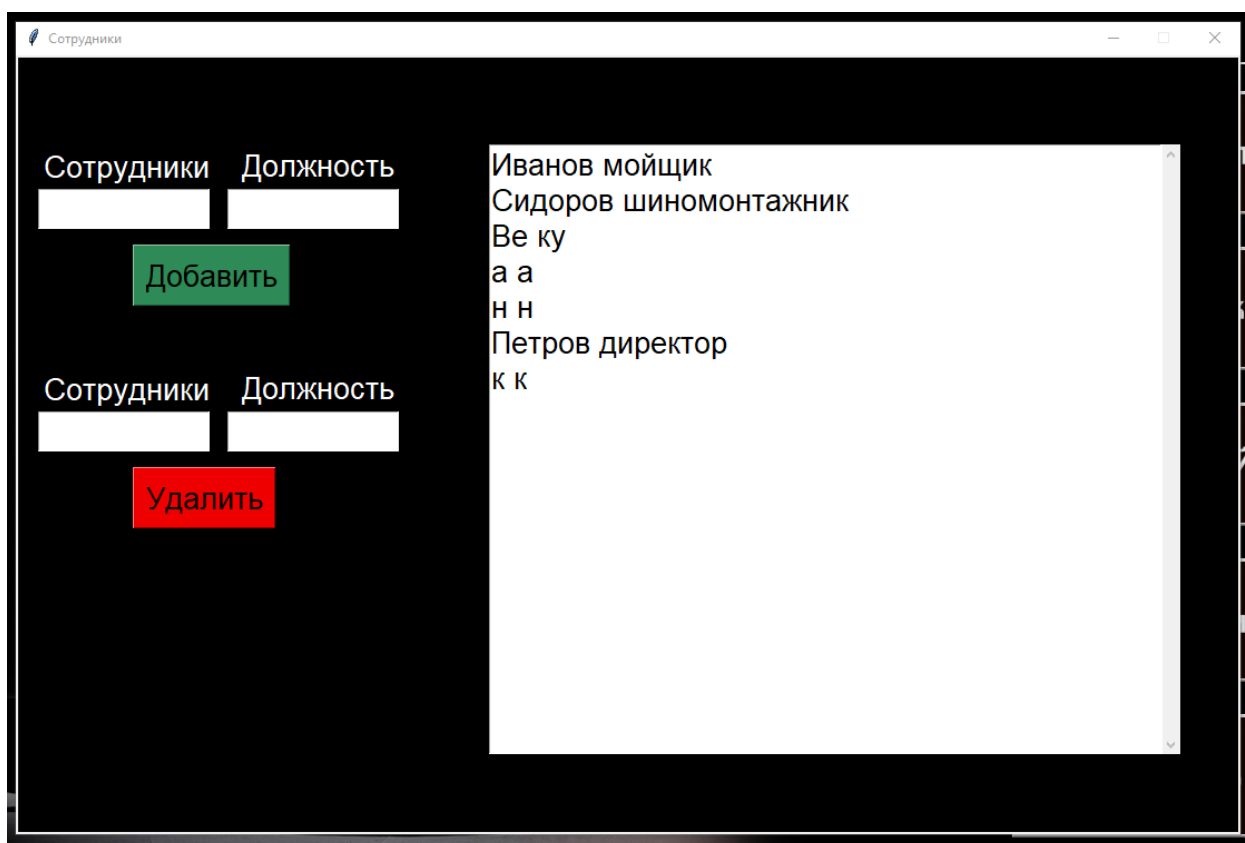


Рисунок 12. Окно «Сотрудники».

В данном окне мы можем добавить или удалить запись, которая информирует нас о сотрудниках предприятия.

Листинг 4. Окно «Сотрудники».

```
#=====
#С помощью этого модуля можно ввести учет сотрудников
#=====
from tkinter import *
import tkinter as tk
import random
from tkinter import scrolledtext
import ctypes

def Sotrudniki():
    window = Toplevel()
    window.resizable(False, False)
    window.title("Сотрудники")
    window.geometry('1100x700')
    user32 = ctypes.windll.user32
    widthButtonPlace = (user32.GetSystemMetrics(0) // 15)
    widthButton = widthButtonPlace // 7
    canvas = Canvas(window, width = user32.GetSystemMetrics(0), height =
user32.GetSystemMetrics(1))
    canvas.configure(bg = 'black')
    canvas.pack()
```

```

f = open('ss.txt', 'r', encoding = 'utf-8')
text = Text()
text = scrolledtext.ScrolledText(window, height = '17', font = 'Batang
20', width = '40')
text.insert(INSERT, f.read())
text.place(x = 425, y = 80)
f.close()
def clicked():
    z = open('ss.txt', 'r', encoding = 'utf-8')
    y = z.read()
    x = y.split()
    t = []
    for i in range (len(x)):
        if i%2 == 0:
            u = x[i] + ' ' + x[i+1]
            t.append(u)
    b = txt1.get()
    b1 = txt2.get()
    f = open('ss.txt', 'a', encoding = 'utf-8')
    u11 = txt1.get() + ' ' + txt2.get() + '\n'
    f.write(u11)
    f.close()
    f = open('ss.txt', 'r', encoding = 'utf-8')
    text = Text()
    text = scrolledtext.ScrolledText(window, height = '17', font =
'Batang 20', width = '40')
    text.insert(INSERT, f.read())
    text.place(x = 425, y = 80)
    f.close()
def clicked1():
    z = open('ss.txt', 'r', encoding = 'utf-8')
    y = z.read()
    x = y.split()
    t = []
    for i in range (len(x)):
        if i%2 == 0:
            u = x[i] + ' ' + x[i+1]
            t.append(u)
    b = txt3.get()
    b1 = txt4.get()
    v = open('ss.txt','w', encoding = 'utf-8')
    for i in range(len(x)):
        j = t[i].split()
        if i%2 == 0 and j[0] != b:
            v.write(j[0])
            v.write(' ')
            v.write(j[1])
            v.write('\n')
    v.close()
    f = open('ss.txt', 'r', encoding = 'utf-8')
    text = Text()
    text = scrolledtext.ScrolledText(window, height = '17', font =
'Batang 20', width = '40')

```



```

text.insert(INSERT, f.read())
text.place(x = 425, y = 80)
f.close()
lbl1 = Label(window)
lbl2 = Label(window)
lbl1.configure(font = 'Batang 20', text = 'Сотрудники', bg = 'black', fg
= 'white')
lbl1.place(x = 22, y = 81)
lbl2.configure(font = 'Batang 20', text = 'Должность', bg = 'black', fg =
'white')
lbl2.place(x = 200, y = 80)
txt1 = Entry(window, width = 10, font = 'Batang 20')
txt1.place(x = 20, y = 120)
txt2 = Entry(window, width = 10, font = 'Batang 20')
txt2.place(x = 190, y = 120)
btn = Button(window, text = "Добавить", font = 'Batang 20', command =
clicked, bg = 'seagreen')
btn.place(x = 105, y = 170)
lbl3 = Label(window)
lbl4 = Label(window)
lbl3.configure(font = 'Batang 20', text = 'Сотрудники', bg = 'black', fg
= 'white')
lbl3.place(x = 22, y = 281)
lbl4.configure(font = 'Batang 20', text = 'Должность', bg = 'black', fg =
'white')
lbl4.place(x = 200, y = 280)
txt3 = Entry(window, width = 10, font = 'Batang 20')
txt3.place(x = 20, y = 320)
txt4 = Entry(window, width = 10, font = 'Batang 20')
txt4.place(x = 190, y = 320)
btn1 = Button(window, text = "Удалить", font = 'Batang 20', command =
clicked1, bg = 'red2')
btn1.place(x = 105, y = 370)

```

Рассмотрим модуль автомобильной программы – это окно «Тест-драйв». На рисунке №13 представлен интерфейс модуля.

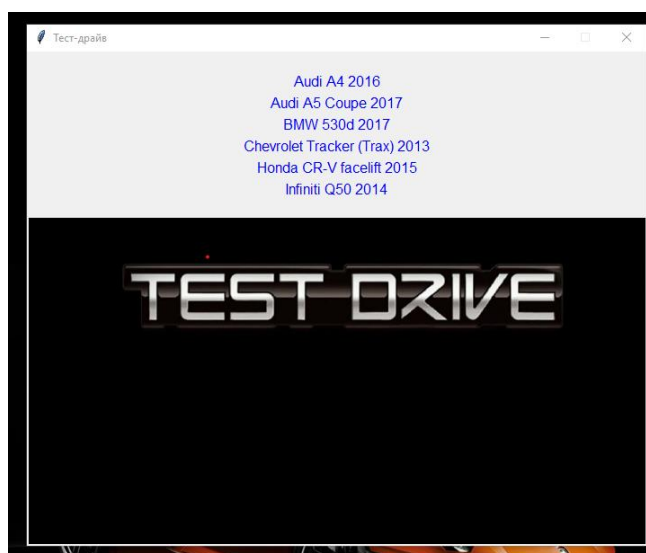


Рисунок 13. Окно «Тест-драйв».

В данном окне мы можем увидеть ссылки на тест-драйвы автомобилей. После нажатия на ссылку пользователь, перейдет по этой ссылке в интернет. После чего он получит из видео информацию об автомобиле.

Листинг 5. Окно «Тест-драйв».

```
#=====
#Этот модуль предоставляет ссылки на тест-драйвы автомобилей
#=====
from tkinter import *
import webbrowser

def Test_drive():
    def callback(url):
        webbrowser.open_new(url)
    window = Toplevel()
    window.geometry("690x550")
    window.resizable(False, False)
    window.title("Тест-драйв")
    link1 = Label(window)
    link1.pack()
    link2 = Label(window, text="Audi A4 2016", fg="blue", cursor="hand2",
font = '1')
    link2.pack()
    link2.bind("<Button-1>", lambda e:
callback("https://www.youtube.com/watch?v=cB-66nJOJ08&feature=emb_logo"))
    link3 = Label(window, text="Audi A5 Coupe 2017", fg="blue",
cursor="hand2", font = '1')
    link3.pack()
    link3.bind("<Button-1>", lambda e:
callback("https://www.youtube.com/watch?v=dnqR-aopD6Y&feature=emb_logo"))
    link4 = Label(window, text="BMW 530d 2017", fg="blue", cursor="hand2",
font = '1')
```

```

link4.pack()
link4.bind("<Button-1>", lambda e:
callback("https://www.youtube.com/watch?v=Nfnwhy0fPzw&feature=emb_logo"))
link5 = Label(window, text="Chevrolet Tracker (Trax) 2013", fg="blue",
cursor="hand2", font = '1')
link5.pack()
link5.bind("<Button-1>", lambda e:
callback("https://www.youtube.com/watch?v=vHAne0XgScQ&feature=emb_logo"))
link6 = Label(window, text="Honda CR-V facelift 2015", fg="blue",
cursor="hand2", font = '1')
link6.pack()
link6.bind("<Button-1>", lambda e:
callback("https://www.youtube.com/watch?v=XKPO33mrAgE&feature=emb_logo"))
link7 = Label(window, text="Infiniti Q50 2014\n", fg="blue",
cursor="hand2", font = '1')
link7.pack()
link7.bind("<Button-1>", lambda e:
callback("https://www.youtube.com/watch?v=o14FDZcvR8c&feature=emb_logo"))
c = Canvas(window, width = 1200, height = 500, bg = 'white')
c.pack()
osnovnoi_fon = PhotoImage(file = 'T_D.png')
osnovnoi_fon1 = c.create_image(340, 300, image = osnovnoi_fon)
window.mainloop()

```

2.8. Описание тестовых наборов модулей

В каждом модуле требуется вводить определенные значения. Там, где имеется выпадающий список следует выбирать значения только из него. Если, не соблюдать это правило, то программа не выполнит свою функцию. Пример выпадающего списка можно увидеть на рисунке № 12.

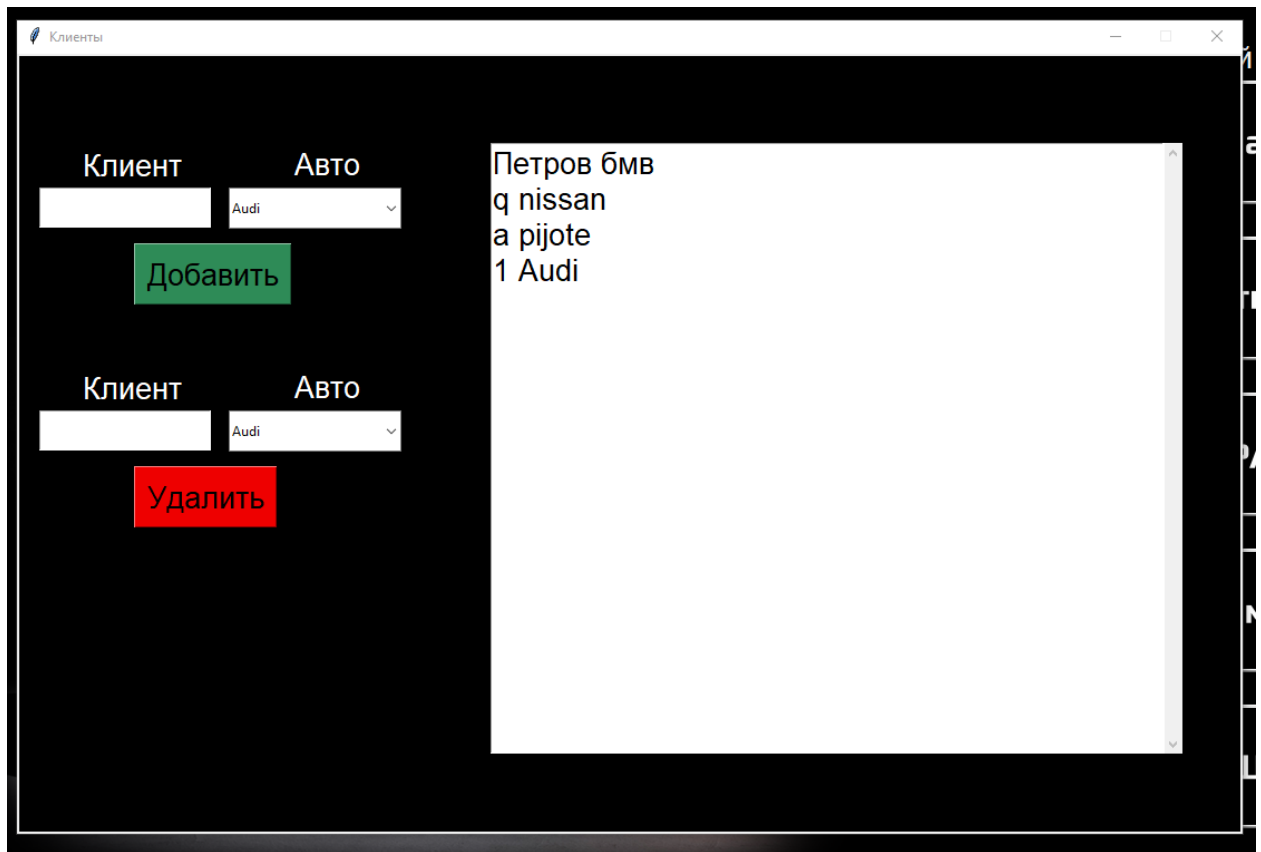


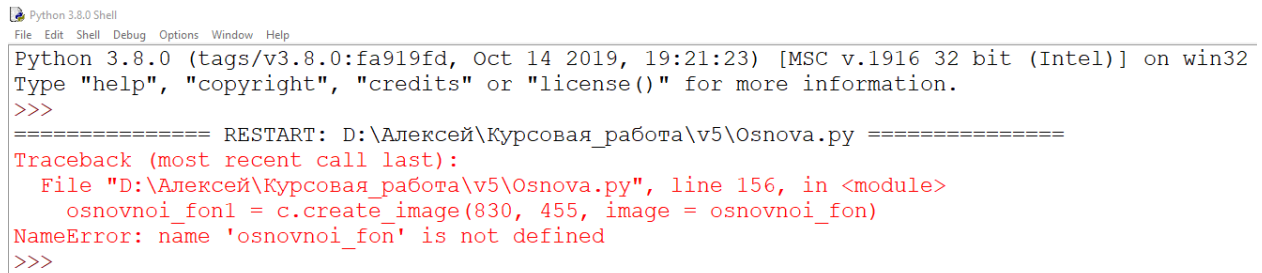
Рисунок 14. Выпадающий список.

Приведу пример грамотного добавления информации. В поле клиент пишем имя клиента, а в поле авто выбираем автомобиль из выпадающего списка.

2.9. Описание применения средств отладки

При проведении теста №1 (белый ящик) было выявлено:

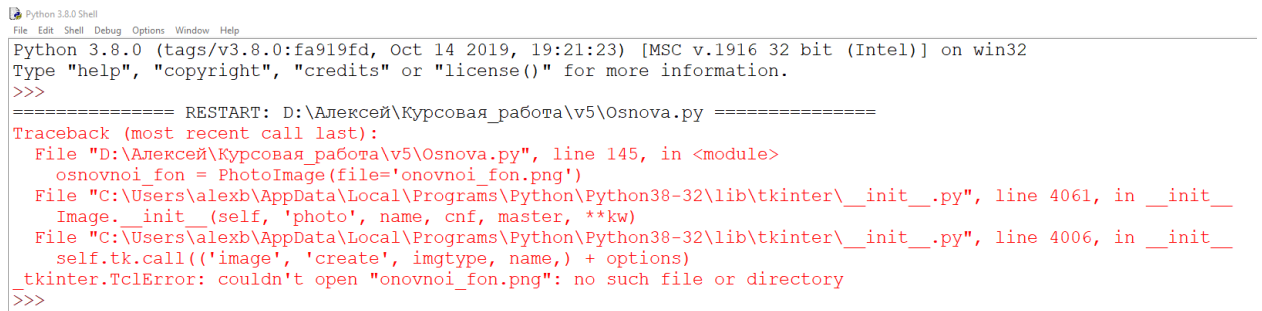
До отладки программы была ошибка с неизвестным именем (рисунок №15). Эта ошибка данных, то есть не объявлена переменная. Ее можно исправить, объявив ее в данной программе.



```
Python 3.8.0 Shell
File Edit Shell Debug Options Window Help
Python 3.8.0 (tags/v3.8.0:fa919fd, Oct 14 2019, 19:21:23) [MSC v.1916 32 bit (Intel)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>>
===== RESTART: D:\Алексей\Курсовая_работа\v5\Osnova.py =====
Traceback (most recent call last):
  File "D:\Алексей\Курсовая_работа\v5\Osnova.py", line 156, in <module>
    osnovnoi_fonl = c.create_image(830, 455, image = osnovnoi_fon)
NameError: name 'osnovnoi_fon' is not defined
>>>
```

Рисунок 15. Ошибка №1.

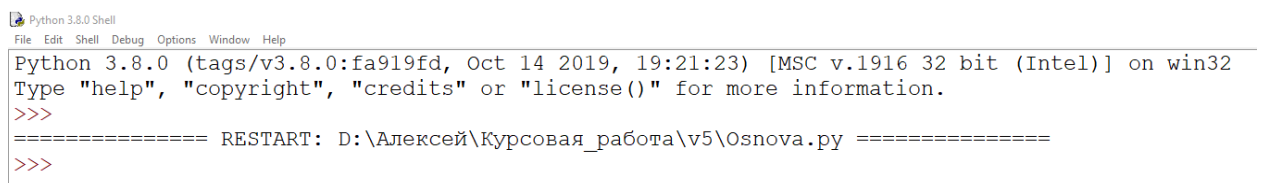
Еще одна ошибка была – это неверный путь к фотографии (рисунок №16). Вид у этой программной ошибки: синтаксический. Ее можно исправить, объявив правильный путь к фотографии в формате png.



```
Python 3.8.0 Shell
File Edit Shell Debug Options Window Help
Python 3.8.0 (tags/v3.8.0:fa919fd, Oct 14 2019, 19:21:23) [MSC v.1916 32 bit (Intel)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>>
===== RESTART: D:\Алексей\Курсовая_работа\v5\Osnova.py =====
Traceback (most recent call last):
  File "D:\Алексей\Курсовая_работа\v5\Osnova.py", line 145, in <module>
    osnovnoi_fon = PhotoImage(file='onovnoi_fon.png')
  File "C:\Users\alexb\AppData\Local\Programs\Python\Python38-32\lib\tkinter\_init_.py", line 4061, in __init__
    Image.__init__(self, 'photo', name, cnf, master, **kw)
  File "C:\Users\alexb\AppData\Local\Programs\Python\Python38-32\lib\tkinter\_init_.py", line 4006, in __init__
    self.tk.call(('image', 'create', imgtype, name,) + options)
tclinter.TclError: couldn't open "onovnoi_fon.png": no such file or directory
>>>
```

Рисунок 16. Ошибка №2.

После исправления всех ошибок, которые были, программа запустилась корректно.



```
Python 3.8.0 Shell
File Edit Shell Debug Options Window Help
Python 3.8.0 (tags/v3.8.0:fa919fd, Oct 14 2019, 19:21:23) [MSC v.1916 32 bit (Intel)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>>
===== RESTART: D:\Алексей\Курсовая_работа\v5\Osnova.py =====
>>>
```

Рисунок 17. Поле отладки программы после исправления всех ошибок.

2.10. Анализ оптимальности использования памяти и быстродействия

Данная разработка не занимает при работе большой объем памяти. В основном объем памяти занимают фотографии, но есть некоторые модули программы, которые в дальнейшем необходимо будет их оптимизировать для быстрой работы программы. Вид данной оптимизации: машинно-зависимый. То есть ориентирована на конкретный язык программирования, в данном случае – это Python.

Приведу два примера оптимизации, которые были использованы в этой программе:

1. Использование библиотеки «Time». Эта библиотека позволяет работать с готовыми конструкциями, например как strftime – это формат часов.

```
#####
#Этот модуль показывает время
#####

from tkinter import *
from tkinter.ttk import *
from time import strftime

def clock():
    root = Tk()
    root.resizable(False, False)
    root.title('Часы')
    def time():
        string = strftime('%H:%M:%S %p')
        lbl.config(text = string)
        lbl.after(1000, time)
    lbl = Label(root, font = ('Ubuntu', 40, 'bold'), background = 'black', foreground = 'white')
    lbl.pack(anchor = 'center')
    time()
    mainloop()
```

Рисунок 18. Оптимизация №1.

2. Использование длины строки в цикле экономит место из-за того, что в цикле есть граница.

```
for i in range (len(x)) :  
    if i%2 == 0 :  
        u = x[i] + ' ' + x[i+1]
```

Рисунок 19. Оптимизация №2.

Глава 3. Эксплуатационная часть

Аннотация

В данной главе приведено руководство оператора по применению и эксплуатации программного обеспечения для автотехцентра.

В разделе «Функциональное назначение программы» указаны сведения о назначении программы и информация, достаточная для понимания функций программы и ее эксплуатации.

В разделе «Список функций» указан список функций, который доступен пользователю в данной программе.

В разделе «Требования к персоналу (пользователю)» указаны требования к пользователю программы.

В разделе «Минимальный состав аппаратных средств» указаны требования к аппаратному средству.

3.1. Руководство оператора

Функциональное назначение программы:

Программное обеспечение для автотехцентра используется для мониторинга и управления организации, входящим в автомобильную сферу.

Эксплуатационное назначение программы:

Специальное программное обеспечение должно эксплуатироваться на объектах любого масштаба для организации комплексной системы обеспечения работы объекта.

3.2. Список функций

- Учет автозапчастей на складе.
- Учет финансов предприятия.
- Учет сотрудников предприятия.
- Учет клиентов предприятия.
- Вывод информации на экран о разработчиках программы.
- Вывод информации на экран о программе.
- Показ времени в данный момент.
- Калькулятор.
- Вывод информации на экран о системном требовании.
- Вывод на экран ссылок тест-драйвов автомобилей.
- Помощь пользователю.
- Учет неисправностей автомобиля клиента.
- Вывод на экран стоимость услуг.

3.3. Требования к персоналу (пользователю)

Конечный пользователь программы (оператор) должен обладать практическими навыками работы с графическим пользовательским интерфейсом программы.

3.4. Минимальный состав аппаратных средств

- ОС: Windows 7/8/10
- Процессор: 2.0 GHz
- Оперативная память: 4 GB ОЗУ
- Видеокарта: NVIDIA GeForce GT 520
- Место на диске: 4 MB

3.5. Сообщение оператору:

Если у пользователя возникла трудность с программой, то он может обратиться в техническую поддержку.

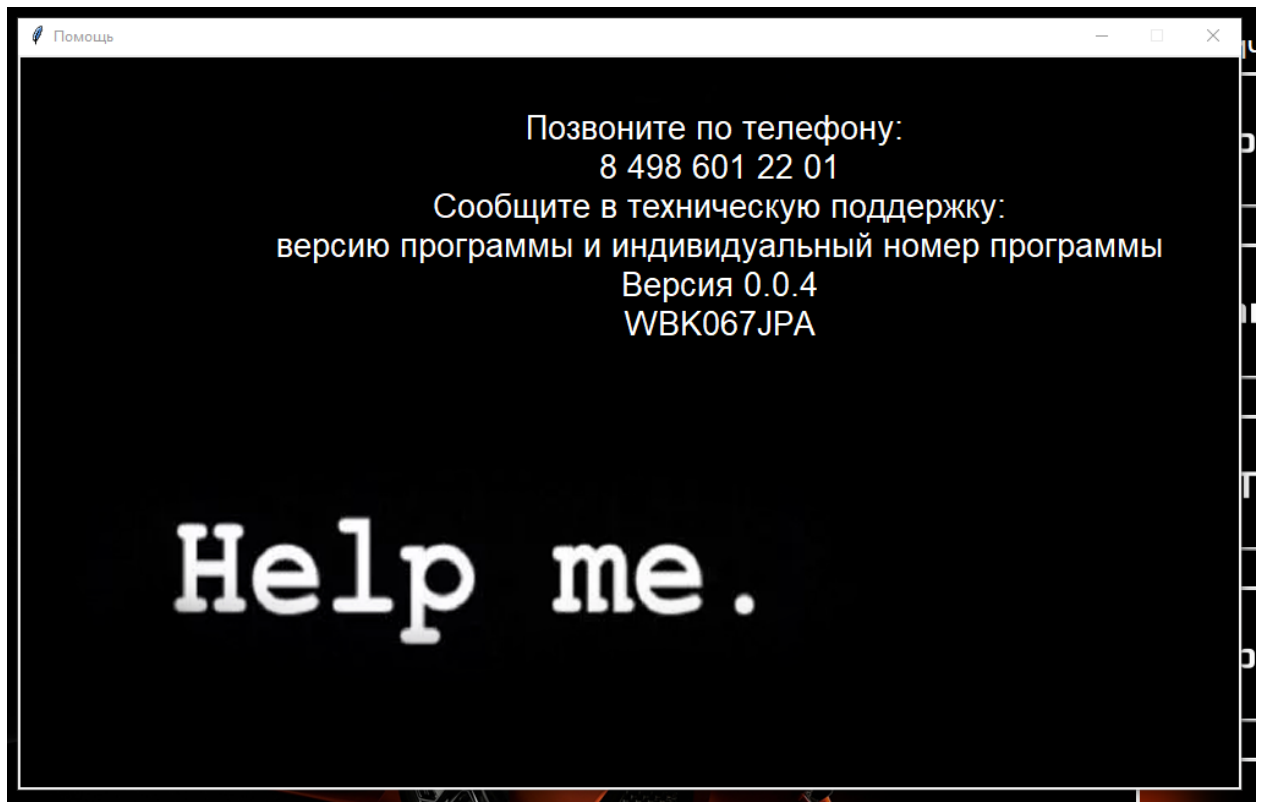


Рисунок 20. Окно «Помощь».

Заключение

В данном проекте было сделано: интерфейс и функционал автомобильной программы. В качестве развития я усилил навыки работы на языке программирования Python и научился работать с изображением.

Разработка данной программы принесла немало опыта в использовании библиотеки Tinter.

Список литературы и интернет-источников

- Бизнес-план автосервиса:
https://www.equipnet.ru/org-biz/avtobiznes/avtobiznes_457.html
- Сайт автосервиса «Bosch»:
<https://www.boschcarservice.com/ru/ru>
- Сайт автосервиса «Fit Service»:
<https://korolev.fitauto.ru/>

Приложение 1. Код главного меню.

```
#=====
#Главный модуль программы
#=====
from tkinter import *
from tkinter import messagebox
from tkinter import scrolledtext
from tkinter import ttk
from datetime import datetime
import ctypes
import Avto_zapchasti, Finance
import Klient, Sotrudniki, Price, Diagnostika
import Razrabotchiki, Chasi, O_prodramme, Test_drive
def Kalkulator():
    root = Tk()
    root.geometry("240x230")
    root.resizable(False, False)
    root.title("Калькулятор")
    root.configure(bg = 'black')
    def calc(key):
        global memory
        if key == "=":
            str1 = "+0123456789.*/"
            if calc_entry.get()[0] not in str1:
                calc_entry.insert(END, "Первый символ не число")
                messagebox.showerror("Ошибка!", "Вы ввели не число!")
            try:
                result = eval(calc_entry.get())
                calc_entry.insert(END, "=" + str(result))
            except:
                calc_entry.insert(END, "Ошибка!")
                messagebox.showerror("Ошибка!", "Проверьте правильность
данных.")
        elif key == "C":
            calc_entry.delete(0, END)
        elif key == "-/+":
            if "=" in calc_entry.get():
                calc_entry.delete(0, END)
            try:
                if calc_entry.get()[0] == "-":
                    calc_entry.delete(0)
                else:
                    calc_entry.insert(0, "-")
            except IndexError:
                pass
        else:
            if "=" in calc_entry.get():
                calc_entry.delete(0, END)
            calc_entry.insert(END, key)
    btnn_list = ["7", "8", "9", "+", "-",
                  "4", "5", "6", "*", "/",
                  "1", "2", "3", "-/+", "=",
```

```

        "0", ".", "C", "", ""]

r = 1
c = 0
for i in btnn_list:
    rel = ""
    cmd = lambda x = i: calc(x)
    b = Button(root, bg = 'black', fg = 'white', font = 20, text = i,
width = 4, height = 2, command = cmd).grid(row = r, column = c)
    c += 1
    if c > 4:
        c = 0
        r += 1
    calc_entry = Entry(root, width = 33)
    calc_entry.grid(row = 0, column = 0, columnspan = 5)
    root.mainloop()

def sprav():
    okno1 = Tk()
    okno1.geometry("1600x860")
    okno1.title("Программное обеспечение для автосервиса")
    okno1.resizable(False, False)
    okno1.configure(bg = 'black')
    btnn1 = Label(okno1)
    btnn1.configure(text = 'В разработке', font = ('Batang 20', 70), bg =
'black', fg = 'white')
    btnn1.place(x = 500, y = 400)
    okno1.mainloop()

def system():
    okno1 = Toplevel()
    okno1.geometry("1000x700")
    okno1.title("Системные требования")
    okno1.resizable(False, False)
    okno1.configure(bg = 'black')
    c = Canvas(okno1, width = 1800, height = 800, bg = 'white')
    c.pack()
    osnovnoi_fon = PhotoImage(file = 'Sis.png')
    osnovnoi_fon1 = c.create_image(500, 300, image = osnovnoi_fon)
    sis1 = Label(okno1)
    sis1.configure(text = 'Минимальные требования\n ОС: Windows 7/8/10\n
Процессор: 2.0 GHz\n Оперативная память: 4 GB ОЗУ\n Видеокарта: NVIDIA
GeForce GT 520\n Место на диске: 4 MB' , font = 'Batang 20', bg = 'black', fg
= 'white')
    sis1.place(x = 500, y = 40)
    okno1.mainloop()

def celi():
    okno1 = Toplevel()
    okno1.geometry("1600x860")
    okno1.title("Цели")
    okno1.resizable(False, False)
    okno1.configure(bg = 'black')
    c = Canvas(okno1, width = 1920, height = 1080, bg = 'white')
    c.pack()
    osnovnoi_fon = PhotoImage(file = 'Sis.png')
    osnovnoi_fon1 = c.create_image(700, 400, image = osnovnoi_fon)

```

```

    okno1.mainloop()
def Help1():
    okno1 = Toplevel()
    okno1.geometry("1000x600")
    okno1.title("Помощь")
    okno1.resizable(False, False)
    okno1.configure(bg = 'black')
    c = Canvas(okno1, width = 1800, height = 800, bg = 'white')
    c.pack()
    osnovnoi_fon = PhotoImage(file = 'Help1.png')
    osnovnoi_fon1 = c.create_image(500, 300, image = osnovnoi_fon)
    sis1 = Label(okno1)
    sis1.configure(text = 'Позвоните по телефону:\n 8 498 601 22 01\n
Сообщите в техническую поддержку:\n версию программы и индивидуальный номер
программы\n Версия 0.0.4\n WBK067JPA' , font = 'Batang 20', bg = 'black', fg
= 'white')
    sis1.place(x = 200, y = 40)
    okno1.mainloop()
def clock():
    import sekymdomer
okno1 = Tk()
okno1.geometry("1550x810")
okno1.title("Программное обеспечение для автотехцентра")
okno1.resizable(False, False)
c = Canvas(okno1, width = 1920, height = 1080, bg = 'white')
c.pack()
gg1 = Label(okno1)
gg1.configure(text = 'Технический блок', font = 'Batang 20', bg = 'black', fg
= 'white')
gg1.place(x = 965, y = 40)
gg2 = Label(okno1)
gg2.configure(text = 'Руководящий блок', font = 'Batang 20', bg = 'black', fg
= 'white')
gg2.place(x = 1265, y = 40)
main_menu = Menu()
osnovnoi_fon = PhotoImage(file='osnovnoi_fon.png')
z = PhotoImage(file = 'avtozapchasti.png')
f = PhotoImage(file = 'finans.png')
t = PhotoImage(file = 'test_drive.png')
h = PhotoImage(file = 'help.png')
cr = PhotoImage(file = 'razrabotchiki.png')
kl = PhotoImage(file = 'klienci.png')
opr = PhotoImage(file = 'o_programme.png')
prl = PhotoImage(file = 'price_list.png')
s = PhotoImage(file = 'sotrydniki.png')
dd = PhotoImage(file = 'diagnostika.png')
osnovnoi_fon1 = c.create_image(830, 455, image = osnovnoi_fon)
Button(okno1, width = 260, height = 105, font = "Batang 20", text =
"Автозапчасти", command = Avto_zapchasti.Avto_zapchasti, image = z).place(x =
950, y = 80)
Button(okno1, width = 260, height = 105, font = "Batang 20", text =
"Финансы", command = Finance.Finance, image = f).place(x = 1250, y = 220)

```

```

Button(okno1, width = 260, height = 105, font = "Batang 20", text = "Тест -
драйв", command = Test_drive.Test_drive, image = t).place(x = 950, y = 360)
Button(okno1, width = 260, height = 105, font = "Batang 20", text = "О
программе", command = O_prodramme.O_prodramme, image = opr).place(x = 950, y
= 500)
Button(okno1, width = 260, height = 105, font = "Batang 20", text =
"Разработчики", command = Razrabotchiki.Razrabotchiki, image = cr).place(x =
1250, y = 640)
Button(okno1, width = 260, height = 105, font = "Batang 20", text =
"Сотрудники", command = Sotrudniki.Sotrudniki, image = s).place(x = 1250, y =
80)
Button(okno1, width = 260, height = 105, font = "Batang 20", text =
"Диагностика\ндвигателя", command = Diagnostika.Diagnostika, image =
dd).place(x = 950, y = 220)
Button(okno1, width = 260, height = 105, font = "Batang 20", text =
"Клиенты", command = Klient.Klient, image = kl).place(x = 1250, y = 360)
Button(okno1, width = 260, height = 105, font = "Batang 20", text = "Прайс -
лист", command = Price.Price, image = prl).place(x = 1250, y = 500)
Button(okno1, width = 260, height = 105, font = "Batang 20", text = "Помощь",
command = Help1, image = h).place(x = 950, y = 640)
file_menu = Menu(tearoff = 0)
file_menu1 = Menu(tearoff = 0)
file_menu2 = Menu(tearoff = 0)
file_menu3 = Menu(tearoff = 0)
file_menu4 = Menu(tearoff = 0)
file_menu.add_command(label = "Калькулятор", command = Kalkulator)
file_menu1.add_command(label = "Требования", command = system)
file_menu3.add_command(label = "Часы", command = Chasi.clock)
main_menu.add_cascade(label="Инструменты", menu = file_menu)
main_menu.add_cascade(label="Системные требования", menu = file_menu1)
main_menu.add_cascade(label="Время", menu = file_menu3)
okno1.config(menu = main_menu)
okno1.mainloop()

```

Приложение 2. Модуль «Часы».

```

#=====
#Этот модуль показывает время
#=====
from tkinter import *
from tkinter.ttk import *
from time import strftime
def clock():
    root = Tk()
    root.resizable(False, False)
    root.title('Часы')
    def time():
        string = strftime('%H:%M:%S %p')
        lbl.config(text = string)
        lbl.after(1000, time)
    lbl = Label(root, font = ('Ubuntu', 40, 'bold'), background = 'black',
foreground = 'white')
    lbl.pack(anchor = 'center')
    time()

```


Приложение 3. Окно «Финансы».

```
#=====
#С помощью этого модуля можно ввести учет финансовой деятельности
#=====
from tkinter import *
from tkinter import messagebox
from tkinter import scrolledtext
from tkinter import ttk
from datetime import datetime
import ctypes

def Finance():
    window = Toplevel()
    window.geometry("1150x810")
    window.title("Финансы")
    window.resizable(False, False)
    c = Canvas(window, width = 1920, height = 1080, bg = 'white')
    c.pack()
    osnovnoi_fon = PhotoImage(file='fin2.png')
    osnovnoi_fon1 = c.create_image(700, 400, image = osnovnoi_fon)
    user32 = ctypes.windll.user32
    widthButtonPlace = (user32.GetSystemMetrics(0) // 15)
    widthButton = widthButtonPlace // 7
    f = open('bug.txt', 'r', encoding = 'utf-8')
    text = Text()
    text = scrolledtext.ScrolledText(window, height = '17', font = 'Batang
20', width = '40')
    text.insert(INSERT, f.read())
    text.place(x = 465, y = 30)
    f.close()
    def clicked():
        z = open('bug.txt', 'r', encoding = 'utf-8')
        y = z.read()
        x = y.split()
        t = []
        for i in range (len(x)):
            if i%2 == 0:
                u = x[i] + ' ' + x[i+1]
                t.append(u)
        b = txt1.get()
        b1 = txt2.get()
        f = open('bug.txt', 'a', encoding = 'utf-8')
        u11 = txt1.get() + ' ' + txt2.get() + '\n'
        f.write(u11)
        f.close()
        f = open('bug.txt', 'r', encoding = 'utf-8')
        text = Text()
        text = scrolledtext.ScrolledText(window, height = '17', font =
'Batang 20', width = '40')
        text.insert(INSERT, f.read())
        text.place(x = 465, y = 30)
        f.close()
    def clicked1():
```

```

z = open('bug.txt', 'r', encoding = 'utf-8')
y = z.read()
x = y.split()
t = []
for i in range (len(x)):
    if i%2 == 0:
        u = x[i] + ' ' + x[i+1]
        t.append(u)
b = txt3.get()
bl = txt4.get()
v = open('bug.txt','w', encoding = 'utf-8')
for i in range(len(x)):
    j = t[i].split()
    if i%2 == 0 and j[0] != b:
        v.write(j[0])
        v.write(' ')
        v.write(j[1])
        v.write('\n')
v.close()
f = open('bug.txt', 'r', encoding = 'utf-8')
text = Text()
text = scrolledtext.ScrolledText(window, height = '17', font =
'Batang 20', width = '40')
text.insert(INSERT, f.read())
text.place(x = 465, y = 30)
f.close()
lbl1 = Label(window)
lbl2 = Label(window)
lbl1.configure(font = 'Batang 20', text = '    Дата', bg = 'black', fg =
'white')
lbl1.place(x = 25, y = 21)
lbl2.configure(font = 'Batang 20', text = '    Сумма', bg = 'black', fg =
'white')
lbl2.place(x = 200, y = 20)
txt1 = Entry(window, width = 10, font = 'Batang 20')
txt1.place(x = 20, y = 60)
txt2 = Entry(window, width = 10, font = 'Batang 20')
txt2.place(x = 190, y = 60)
btn = Button(window, text = "Добавить", font = 'Batang 20',
command=clicked, bg = 'seagreen')
btn.place(x = 105, y = 110)
lbl3 = Label(window)
lbl4 = Label(window)
lbl3.configure(font = 'Batang 20', text = '    Дата', bg = 'black', fg =
'white')
lbl3.place(x = 25, y = 201)
lbl4.configure(font = 'Batang 20', text = '    Сумма', bg = 'black', fg =
'white')
lbl4.place(x = 200, y = 200)
txt3 = Entry(window, width = 10, font = 'Batang 20')
txt3.place(x = 20, y = 240)
txt4 = Entry(window, width = 10, font = 'Batang 20')
txt4.place(x = 190, y = 240)

```

```

    btn1 = Button(window, text = "Удалить", font = 'Batang 20', command =
clicked1, bg = 'red2')
    btn1.place(x = 105, y = 290)
    window.mainloop()

```

Приложение 4. Окно «Сотрудники».

```

#=====
#С помощью этого модуля можно ввести учет сотрудников
#=====
from tkinter import *
import tkinter as tk
import random
from tkinter import scrolledtext
import ctypes

def Sotrudniki():
    window = Toplevel()
    window.resizable(False, False)
    window.title("Сотрудники")
    window.geometry('1100x700')
    user32 = ctypes.windll.user32
    widthButtonPlace = (user32.GetSystemMetrics(0) // 15)
    widthButton = widthButtonPlace // 7
    canvas = Canvas(window, width = user32.GetSystemMetrics(0), height =
user32.GetSystemMetrics(1))
    canvas.configure(bg = 'black')
    canvas.pack()
    f = open('ss.txt', 'r', encoding = 'utf-8')
    text = Text()
    text = scrolledtext.ScrolledText(window, height = '17', font = 'Batang
20', width = '40')
    text.insert(INSERT, f.read())
    text.place(x = 425, y = 80)
    f.close()
    def clicked():
        z = open('ss.txt', 'r', encoding = 'utf-8')
        y = z.read()
        x = y.split()
        t = []
        for i in range (len(x)):
            if i%2 == 0:
                u = x[i] + ' ' + x[i+1]
                t.append(u)
        b = txt1.get()
        b1 = txt2.get()
        f = open('ss.txt', 'a', encoding = 'utf-8')
        u11 = txt1.get() + ' ' + txt2.get() + '\n'
        f.write(u11)
        f.close()
        f = open('ss.txt', 'r', encoding = 'utf-8')
        text = Text()
        text = scrolledtext.ScrolledText(window, height = '17', font =
'Batang 20', width = '40')
        text.insert(INSERT, f.read())

```

```

text.place(x = 425, y = 80)
f.close()
def clicked1():
    z = open('ss.txt', 'r', encoding = 'utf-8')
    y = z.read()
    x = y.split()
    t = []
    for i in range (len(x)):
        if i%2 == 0:
            u = x[i] + ' ' + x[i+1]
            t.append(u)
    b = txt3.get()
    b1 = txt4.get()
    v = open('ss.txt','w', encoding = 'utf-8')
    for i in range(len(x)):
        j = t[i].split()
        if i%2 == 0 and j[0] != b:
            v.write(j[0])
            v.write(' ')
            v.write(j[1])
            v.write('\n')
    v.close()
    f = open('ss.txt', 'r', encoding = 'utf-8')
    text = Text()
    text = scrolledtext.ScrolledText(window, height = '17', font =
'Batang 20', width = '40')
    text.insert(INSERT, f.read())
    text.place(x = 425, y = 80)
    f.close()
    lbl1 = Label(window)
    lbl2 = Label(window)
    lbl1.configure(font = 'Batang 20', text = 'Сотрудники', bg = 'black', fg
= 'white')
    lbl1.place(x = 22, y = 81)
    lbl2.configure(font = 'Batang 20', text = 'Должность', bg = 'black', fg =
'white')
    lbl2.place(x = 200, y = 80)
    txt1 = Entry(window, width = 10, font = 'Batang 20')
    txt1.place(x = 20, y = 120)
    txt2 = Entry(window, width = 10, font = 'Batang 20')
    txt2.place(x = 190, y = 120)
    btn = Button(window, text = "Добавить", font = 'Batang 20', command =
clicked, bg = 'seagreen')
    btn.place(x = 105, y = 170)
    lbl3 = Label(window)
    lbl4 = Label(window)
    lbl3.configure(font = 'Batang 20', text = 'Сотрудники', bg = 'black', fg
= 'white')
    lbl3.place(x = 22, y = 281)
    lbl4.configure(font = 'Batang 20', text = 'Должность', bg = 'black', fg =
'white')
    lbl4.place(x = 200, y = 280)
    txt3 = Entry(window, width = 10, font = 'Batang 20')

```

```

txt3.place(x = 20, y = 320)
txt4 = Entry(window, width = 10, font = 'Batang 20')
txt4.place(x = 190, y = 320)
btn1 = Button(window, text = "Удалить", font = 'Batang 20', command =
clicked1, bg = 'red2')
btn1.place(x = 105, y = 370)

```

Приложение 5. Окно «Тест-драйв».

```

#=====
#Этот модуль предоставляет ссылки на тест-драйвы автомобилей
#=====
from tkinter import *
import webbrowser

def Test_drive():
    def callback(url):
        webbrowser.open_new(url)
    window = Toplevel()
    window.geometry("690x550")
    window.resizable(False, False)
    window.title("Тест-драйв")
    link1 = Label(window)
    link1.pack()
    link2 = Label(window, text="Audi A4 2016", fg="blue", cursor="hand2",
font = '1')
    link2.pack()
    link2.bind("<Button-1>", lambda e:
callback("https://www.youtube.com/watch?v=cB-66nJOJ08&feature=emb_logo"))
    link3 = Label(window, text="Audi A5 Coupe 2017", fg="blue",
cursor="hand2", font = '1')
    link3.pack()
    link3.bind("<Button-1>", lambda e:
callback("https://www.youtube.com/watch?v=dnqR-aopD6Y&feature=emb_logo"))
    link4 = Label(window, text="BMW 530d 2017", fg="blue", cursor="hand2",
font = '1')
    link4.pack()
    link4.bind("<Button-1>", lambda e:
callback("https://www.youtube.com/watch?v=Nfnwhy0fPzw&feature=emb_logo"))
    link5 = Label(window, text="Chevrolet Tracker (Trax) 2013", fg="blue",
cursor="hand2", font = '1')
    link5.pack()
    link5.bind("<Button-1>", lambda e:
callback("https://www.youtube.com/watch?v=vHAne0XgScQ&feature=emb_logo"))
    link6 = Label(window, text="Honda CR-V facelift 2015", fg="blue",
cursor="hand2", font = '1')
    link6.pack()
    link6.bind("<Button-1>", lambda e:
callback("https://www.youtube.com/watch?v=XKPO33mrAgE&feature=emb_logo"))
    link7 = Label(window, text="Infiniti Q50 2014\n", fg="blue",
cursor="hand2", font = '1')
    link7.pack()
    link7.bind("<Button-1>", lambda e:
callback("https://www.youtube.com/watch?v=o14FDZcvR8c&feature=emb_logo"))
    c = Canvas(window, width = 1200, height = 500, bg = 'white')

```

```

c.pack()
osnovnoi_fon = PhotoImage(file = 'T_D.png')
osnovnoi_fon1 = c.create_image(340, 300, image = osnovnoi_fon)
window.mainloop()

```

Приложение 6. Окно «Автозапчасти».

```

#=====
#С помощью этого модуля можно ввести учет автозапчастей
#=====
from tkinter import *
import tkinter as tk
import random
from tkinter import scrolledtext
import ctypes

def Avto_zapchasti():
    window = Toplevel()
    window.resizable(False, False)
    window.title("Склад")
    window.geometry('1100x700')
    user32 = ctypes.windll.user32
    widthButtonPlace = (user32.GetSystemMetrics(0) // 15)
    widthButton = widthButtonPlace // 7
    canvas = Canvas(window, width = user32.GetSystemMetrics(0), height =
user32.GetSystemMetrics(1))
    canvas.configure(bg = 'black')
    canvas.pack()
    f = open('skl.txt', 'r', encoding = 'utf-8')
    text = Text()
    text = scrolledtext.ScrolledText(window, height = '17', font = 'Batang
20', width = '40')
    text.insert(INSERT, f.read())
    text.place(x = 425, y = 80)
    f.close()
    def clicked_dobavlenie():
        z = open('skl.txt', 'r', encoding = 'utf-8')
        y = z.read()
        x = y.split()
        t = []
        for i in range (len(x)):
            if i%2 == 0:
                u = x[i] + ' ' + x[i+1]
                t.append(u)
        b = txt1.get()
        b1 = int(txt2.get())
        v = open('skl.txt','w', encoding = 'utf-8')
        for i in range(len(x)):
            j = t[i].split()
            if i%2 == 0 and j[0] != b:
                v.write(j[0])
                v.write(' ')
                v.write(j[1])
                v.write('\n')
            if j[0] == b and i%2 == 0:

```

```

        v.write(j[0])
        l = int(j[1]) + b1
        v.write(' ')
        v.write(str(l))
        v.write('\n')
    v.close()
    f = open('skl.txt', 'r', encoding = 'utf-8')
    text = Text()
    text = scrolledtext.ScrolledText(window, height = '17', font =
'Batang 20', width = '40')
    text.insert(INSERT, f.read())
    text.place(x = 425, y = 80)
    f.close()
def clicked_delete():
    z = open('skl.txt', 'r', encoding = 'utf-8')
    y = z.read()
    x = y.split()
    t = []
    for i in range (len(x)):
        if i%2 == 0:
            u = x[i] + ' ' + x[i+1]
            t.append(u)
    b = txt3.get()
    b1 = int(txt4.get())
    v = open('skl.txt','w', encoding = 'utf-8')
    for i in range(len(x)):
        j = t[i].split()
        if i%2 == 0 and j[0] != b:
            v.write(j[0])
            v.write(' ')
            v.write(j[1])
            v.write('\n')
        if j[0] == b and i%2 == 0:
            v.write(j[0])
            l = int(j[1]) - b1
            v.write(' ')
            v.write(str(l))
            v.write('\n')
    v.close()
    f = open('skl.txt', 'r', encoding = 'utf-8')
    text = Text()
    text = scrolledtext.ScrolledText(window, height = '17', font =
'Batang 20', width = '40')
    text.insert(INSERT, f.read())
    text.place(x = 425, y = 80)
    f.close()
    lbl1 = Label(window)
    lbl2 = Label(window)
    lbl1.configure(font = 'Batang 20', text = 'Название\нтовара', bg =
'black', fg = 'white')
    lbl1.place(x = 40, y = 51)
    lbl2.configure(font = 'Batang 20', text = 'Кол-во', bg = 'black', fg =
'white')

```

```

lbl2.place(x = 225, y = 80)
txt1 = Entry(window, width = 10, font = 'Batang 20')
txt1.place(x = 20, y = 120)
txt2 = Entry(window, width = 10, font = 'Batang 20')
txt2.place(x = 190, y = 120)
btn = Button(window, text = "Добавить", font = 'Batang 20', command =
clicked_dobavlenie, bg = 'seagreen')
btn.place(x = 105, y = 170)
lbl3 = Label(window)
lbl4 = Label(window)
lbl3.configure(font = 'Batang 20', text = 'Название\птовара', bg =
'black', fg = 'white')
lbl3.place(x = 40, y = 251)
lbl4.configure(font = 'Batang 20', text = 'Кол-во', bg = 'black', fg =
'white')
lbl4.place(x = 225, y = 280)
txt3 = Entry(window, width = 10, font = 'Batang 20')
txt3.place(x = 20, y = 320)
txt4 = Entry(window, width = 10, font = 'Batang 20')
txt4.place(x = 190, y = 320)
btn1 = Button(window, text = "Удалить", font = 'Batang 20', command =
clicked_delete, bg = 'red2')
btn1.place(x = 105, y = 370)
def Help():
    window = Toplevel()
    window.resizable(False, False)
    window.title("Помощь")
    window.geometry('555x120')
    user32 = ctypes.windll.user32
    widthButtonPlace = (user32.GetSystemMetrics(0) // 15)
    widthButton = widthButtonPlace // 7
    canvas = Canvas(window, width = user32.GetSystemMetrics(0), height =
user32.GetSystemMetrics(1))
    canvas.configure(bg = 'black')
    canvas.pack()
    lbl1 = Label(window)
    lbl1.configure(font = 'Batang 30', text = 'Вернитесь в главное меню
и\п нажмите "Помощь"', bg = 'black', fg = 'white')
    lbl1.place(x = 10, y = 10)
    Button(window, width = 15, height = 2, font = "Batang 20", text = "Help",
command = Help, bg = 'Red2', fg = 'Black').place(x = 30, y = 500)

```

Приложение 7. Окно «Диагностика».

```

#=====
#С помощью этого модуля можно ввести учет неисправностей автомобилей
#=====
from tkinter import *
from tkinter import messagebox
from tkinter import scrolledtext
from tkinter import ttk
from datetime import datetime
import ctypes

```



```

def Diagnostika():
    window = Toplevel()
    window.geometry("1150x710")
    window.title("Диагностика")
    window.resizable(False, False)
    c = Canvas(window, width = 1920, height = 1080, bg = 'white')
    c.pack()
    osnovnoi_fon = PhotoImage(file='D_fon.png')
    osnovnoi_fon1 = c.create_image(600, 400, image = osnovnoi_fon)
    user32 = ctypes.windll.user32
    widthButtonPlace = (user32.GetSystemMetrics(0) // 15)
    widthButton = widthButtonPlace // 7
    f = open('d.txt', 'r', encoding = 'utf-8')
    text = Text()
    text = scrolledtext.ScrolledText(window, height = '17', font = 'Batang
20', width = '40')
    text.insert(INSERT, f.read())
    text.place(x = 465, y = 30)
    f.close()
    def clicked():
        z = open('d.txt', 'r', encoding = 'utf-8')
        y = z.read()
        x = y.split()
        t = []
        for i in range (len(x)):
            if i%2 == 0:
                u = x[i] + ' ' + x[i+1]
                t.append(u)
        b = txt1.get()
        b1 = txt2.get()

        f = open('d.txt', 'a', encoding = 'utf-8')
        u11 = txt1.get() + ' ' + txt2.get() + '\n'
        f.write(u11)
        f.close()
        f = open('d.txt', 'r', encoding = 'utf-8')
        text = Text()
        text = scrolledtext.ScrolledText(window, height = '17', font =
'Batang 20', width = '40')
        text.insert(INSERT, f.read())
        text.place(x = 465, y = 30)
        f.close()
    def clicked1():
        z = open('d.txt', 'r', encoding = 'utf-8')
        y = z.read()
        x = y.split()
        t = []
        for i in range (len(x)):
            if i%2 == 0:
                u = x[i] + ' ' + x[i+1]
                t.append(u)
        b = txt3.get()

```

```

b1 = txt4.get()
v = open('d.txt', 'w', encoding = 'utf-8')
for i in range(len(x)):
    j = t[i].split()
    if i%2 == 0 and j[0] != b:
        v.write(j[0])
        v.write(' ')
        v.write(j[1])
        v.write('\n')
v.close()
f = open('d.txt', 'r', encoding = 'utf-8')
text = Text()
text = scrolledtext.ScrolledText(window, height = '17', font =
'Batang 20', width = '40')
text.insert(INSERT, f.read())
text.place(x = 465, y = 30)
f.close()
lbl1 = Label(window)
lbl2 = Label(window)
lbl1.configure(font = 'Batang 20', text = 'Неисправность', bg = 'black',
fg = 'white')
lbl1.place(x = 22, y = 21)
lbl2.configure(font = 'Batang 20', text = 'Авто', bg = 'black', fg =
'white')
lbl2.place(x = 285, y = 20)
txt1 = Entry(window, width = 10, font = 'Batang 20')
txt1.place(x = 40, y = 60)
txt2 = Entry(window, width = 10, font = 'Batang 20')
txt2.place(x = 240, y = 60)
btn = Button(window, text = "Добавить", font = 'Batang 20',
command=clicked, bg = 'seagreen')
btn.place(x = 145, y = 110)
lbl3 = Label(window)
lbl4 = Label(window)
lbl3.configure(font = 'Batang 20', text = 'Неисправность', bg = 'black',
fg = 'white')
lbl3.place(x = 22, y = 201)
lbl4.configure(font = 'Batang 20', text = 'Авто', bg = 'black', fg =
'white')
lbl4.place(x = 285, y = 200)
txt3 = Entry(window, width = 10, font = 'Batang 20')
txt3.place(x = 40, y = 240)
txt4 = Entry(window, width = 10, font = 'Batang 20')
txt4.place(x = 240, y = 240)
btn1 = Button(window, text = "Удалить", font = 'Batang 20', command =
clicked1, bg = 'red2')
btn1.place(x = 152, y = 290)
def Help():
    window = Toplevel()
    window.resizable(False, False)
    window.title("Помощь")
    window.geometry('555x120')
    user32 = ctypes.windll.user32

```

```

widthButtonPlace = (user32.GetSystemMetrics(0) // 15)
widthButton = widthButtonPlace // 7
canvas = Canvas(window, width = user32.GetSystemMetrics(0), height =
user32.GetSystemMetrics(1))
canvas.configure(bg = 'black')
canvas.pack()
lbl1 = Label(window)
lbl1.configure(font = 'Batang 30', text = 'Вернитесь в главное меню
и\n нажмите "Помощь"', bg = 'black', fg = 'white')
lbl1.place(x = 10, y = 10)
Button(window, width = 15, height = 2, font = "Batang 20", text = "Help",
command = Help, bg = 'red2', fg = 'black').place(x = 90, y = 380)
window.mainloop()

```

Приложение 8. Окно «Клиенты».

```

#=====
#С помощью этого модуля можно ввести учет клиентов
#=====
from tkinter import *
from tkinter import ttk
import ctypes
import tkinter as tk
import random
def Klient():
    window = Tk()
    window.resizable(False, False)
    window.title("Клиенты")
    window.geometry('1100x700')
    user32 = ctypes.windll.user32
    widthButtonPlace = (user32.GetSystemMetrics(0) // 15)
    widthButton = widthButtonPlace // 7
    canvas = Canvas(window, width = user32.GetSystemMetrics(0), height =
user32.GetSystemMetrics(1))
    canvas.configure(bg = 'black')
    canvas.pack()
    f = open('k.txt', 'r', encoding = 'utf-8')
    text = Text()
    text = scrolledtext.ScrolledText(window, height = '17', font = 'Batang
20', width = '40')
    text.insert(INSERT, f.read())
    text.place(x = 425, y = 80)
    f.close()
    def clicked_dobavlenie():
        z = open('k.txt', 'r', encoding = 'utf-8')
        y = z.read()
        x = y.split()
        t = []
        for i in range (0, len(x), 2):
            u = x[i] + ' ' + x[i+1]
            t.append(u)
        b = txt1.get()

```

```

f = open('k.txt', 'a', encoding = 'utf-8')
u11 = txt1.get() + ' ' + com.get() + '\n'
f.write(u11)
f.close()
f = open('k.txt', 'r', encoding = 'utf-8')
text = Text()
text = scrolledtext.ScrolledText(window, height = '17', font =
'Batang 20', width = '40')
text.insert(INSERT, f.read())
text.place(x = 425, y = 80)
f.close()
def clicked_delete():
    z = open('k.txt', 'r', encoding = 'utf-8')
    y = z.read()
    x = y.split()
    t = []
    for i in range (len(x)):
        if i % 2 == 0:
            u = x[i] + ' ' + x[i+1]
            t.append(u)
    b = txt3.get()
    b1 = com2.get()
    v = open('k.txt','w', encoding = 'utf-8')
    for i in range(len(x)):
        j = t[i].split()
        if i%2 == 0 and j[0] != b:
            v.write(j[0])
            v.write(' ')
            v.write(j[1])
            v.write('\n')
    v.close()
    f = open('k.txt', 'r', encoding = 'utf-8')
    text = Text()
    text = scrolledtext.ScrolledText(window, height = '17', font =
'Batang 20', width = '40')
    text.insert(INSERT, f.read())
    text.place(x = 425, y = 80)
    f.close()
    lbl1 = Label(window)
    lbl2 = Label(window)
    lbl1.configure(font = 'Batang 20', text = ' Клиент', bg = 'black', fg =
'white')
    lbl1.place(x = 40, y = 81)
    lbl2.configure(font = 'Batang 20', text = ' Авто', bg = 'black', fg =
'white')
    lbl2.place(x = 230, y = 80)
    txt1 = Entry(window, width = 10, font = 'Batang 20')
    txt1.place(x = 20, y = 120)
    btn = Button(window, text = "Добавить", font = 'Batang 20', command =
clicked_dobavlenie, bg = 'seagreen')
    btn.place(x = 105, y = 170)
    lbl3 = Label(window)

```

```

lbl4 = Label(window)
lbl3.configure(font = 'Batang 20', text = '  Клиент', bg = 'black', fg =
'white')
lbl3.place(x = 40, y = 281)
lbl4.configure(font = 'Batang 20', text = '  Авто', bg = 'black', fg =
'white')
lbl4.place(x = 230, y = 280)
txt3 = Entry(window, width = 10, font = 'Batang 20')
txt3.place(x = 20, y = 320)
btn1 = Button(window, text = "Удалить", font = 'Batang 20', command =
clicked_delete, bg = 'red2')
btn1.place(x = 105, y = 370)
com = ttk.Combobox(window)
com.place(x = 190, y = 120, height = 37, width = 155)
com['values'] = ['Audi', 'Bmw', 'Chery', 'Chevrolet', 'Citroen',
'Daewoo', 'Ford', 'Great Wall', 'Honda',
'Hyundai', 'Infiniti', 'Jeep', 'Kia', 'Land Rover',
'Lexus', 'MINI', 'Mazda',
'Mercedes-Benz', 'Mitsubishi', 'Nissan', 'Opel',
'Peugeot', 'Porsche', 'Renault', 'Skoda']
com.current(0)
com2 = ttk.Combobox(window)
com2.place(x = 190, y = 320, height = 37, width = 155)
com2['values'] = ['Audi', 'Bmw', 'Chery', 'Chevrolet', 'Citroen',
'Daewoo', 'Ford', 'Great Wall', 'Honda',
'Hyundai', 'Infiniti', 'Jeep', 'Kia', 'Land Rover',
'Lexus', 'MINI', 'Mazda',
'Mercedes-Benz', 'Mitsubishi', 'Nissan', 'Opel',
'Peugeot', 'Porsche', 'Renault', 'Skoda']
com2.current(0)

```

Приложение 9. Окно «О программе».

```

#=====
#Этот модуль выводит на экран информацию о программе
#=====
from tkinter import *
import ctypes
def O_prodramme():
    window = Tk()
    window.resizable(False, False)
    window.title("О программе")
    user32 = ctypes.windll.user32
    widthButtonPlace = (user32.GetSystemMetrics(0) // 15)
    widthButton = widthButtonPlace // 7
    f = open('O_prog.txt', 'r', encoding='utf-8')
    canvas = Canvas(window, width = user32.GetSystemMetrics(0), height =
user32.GetSystemMetrics(1))
    canvas.configure(bg = 'Black')
    canvas.pack()
    window.geometry('800x600')
    txt = scrolledtext.ScrolledText(window, height = '15', font = 'Batang
20', width = '49')

```

```
txt.insert(INSERT, f.read())
txt.place(x = 20, y = 55)
window.mainloop()
```

Приложение 10. Окно «Прайс-лист».

```
#=====
#Этот модуль выводит на экран цены на предоставленные услуги
#=====
from tkinter import *
from tkinter import scrolledtext
import ctypes
def Price():
    window = Tk()
    window.resizable(False, False)
    window.title("Прайс-лист")
    user32 = ctypes.windll.user32
    widthButtonPlace = (user32.GetSystemMetrics(0) // 15)
    widthButton = widthButtonPlace // 7
    f = open('cenik.txt', 'r', encoding='utf-8')
    canvas = Canvas(window, width = user32.GetSystemMetrics(0), height =
user32.GetSystemMetrics(1))
    canvas.configure(bg = 'black')
    canvas.pack()
    window.geometry('800x600')
    txt = scrolledtext.ScrolledText(window, height = '15', font = 'Batang
20', width = '49')
    txt.insert(INSERT, f.read())
    txt.place(x = 20, y = 55)
    window.mainloop()
```

Приложение 11. Окно «Разработчики».

```
#=====
#Этот модуль показывает разработчиков данной программы
#=====
from tkinter import *
def Razrabotchiki():
    window = Toplevel()
    window.geometry("1450x470")
    window.resizable(False, False)
    window.title("Разработчики")
    window.configure(bg = 'black')
    R = Label(window)
    R.configure(text = 'Разработчики', font = 'Batang 40', bg = 'black', fg =
'white')
    R.place(x = 550, y = 40)
#=====
    Alex = PhotoImage(file = 'Alex.png')
    lbl1 = Label(window)
    lbl1.configure(image = Alex)
    lbl1.place(x = 25, y = 160)
    Fam1 = Label(window)
```

```

    Fam1.configure(text = 'Бобнев Алексей Олегович', font = 'Batang 20', bg =
'black', fg = 'white')
    Fam1.place(x = 285, y = 160)
    T1 = Label(window)
    T1.configure(text = '+7 (917) 517-75-27', font = 'Batang 20', bg =
'black', fg = 'white')
    T1.place(x = 285, y = 210)
    P1 = Label(window)
    P1.configure(text = 'alexbob17@yandex.ru', font = 'Batang 20', bg =
'black', fg = 'white')
    P1.place(x = 285, y = 260)
#=====
    Maria = PhotoImage(file = 'Maria.png')
    lbl2 = Label(window)
    lbl2.configure(image = Maria)
    lbl2.place(x = 725, y = 160)
    Fam2 = Label(window)
    Fam2.configure(text = 'Канавская Мария Владимировна', font = 'Batang 20',
bg = 'black', fg = 'white')
    Fam2.place(x = 990, y = 160)
    T2 = Label(window)
    T2.configure(text = '+7 (916) 586-96-76', font = 'Batang 20', bg =
'black', fg = 'white')
    T2.place(x = 990, y = 210)
    P2 = Label(window)
    P2.configure(text = 'maria.seledka666@gmail.com', font = 'Batang 20', bg
= 'black', fg = 'white')
    P2.place(x = 990, y = 260)
    P22 = Label(window)
    P22.configure(text = 'severdrakon@ya.ru', font = 'Batang 20', bg =
'black', fg = 'white')
    P22.place(x = 990, y = 310)
#=====
    window.mainloop()

```