Федеральное государственное образовательное учреждение

высшего образования

«**КАЛИНИНГРАДСКИЙ ГОСУДАРСТВЕННЫЙ ТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ**»

Кафедра систем управления и вычислительной техники

Курсовая работа

допущена к защите

руководитель

Курсовая работа защищена

с оценкой …………………

руководитель

«Кафедра»

Курсовая работа по дисциплине

“Разработка и стандартизация программных средств

и информационных технологий”

Пояснительная записка

65.18-ВТ.2

Нормоконтроллер

Доцент

Высоцкий Л.Г.

Проект выполнил

Студент гр. 18-ВТ-2

Подковыров Д.Р.

2021г.

Оглавление

[1. Задание 3](#_Toc69245209)

[2. Требования к проектируемой системе 4](#_Toc69245210)

[3. Дерево функций проекта 5](#_Toc69245211)

[4. Дерево модулей программной системы 6](#_Toc69245212)

[5. Распределение функций системы по модулям 7](#_Toc69245213)

[6. Структуры используемых данных 8](#_Toc69245214)

[7. Граф и видеоформы пользовательского интерфейса 10](#_Toc69245215)

[8. Блок-схемы процедур, реализующих специфические алгоритмы обработки информации 18](#_Toc69245216)

[9. Листинг 22](#_Toc69245217)

[10. Руководство системного программиста 58](#_Toc69245218)

[11. Руководство пользователя 59](#_Toc69245219)

[12. Текст справочной системы 60](#_Toc69245220)

[13. Литература 62](#_Toc69245221)

# Задание

Информационная система "Кафедра": план кафедры с указаниеме компьютерных классов. Указание номера пары показывает, в каких классах идут занятия и в какой группе.

Два вида пользователей: администратор с возможностью редактирования и пользователь.

# Требования к проектируемой системе

«Кафедра»

Функции(сервисы) системы:

1. функция «Проверка ввода пароля»,
2. функция «Выбор режима входа в систему»,
3. функция «Изменение пароля через каждые 10 дней»,
4. функция «Справочная система»,
5. функция «Просмотр расписания для 1 недели»,
6. функция «Просмотр расписания для 2 недели»,
7. функция «Добавление нового кабинета и группы»,
8. функция «Изменение кабинета и группы»,
9. функция «Удаление кабинета и группы»
10. функция «Переноса пары»
11. функция «Просмотр расписания для кабинета»
12. функция «Просмотр расписания для группы»

Ограничения на сервисы:

1. Два режима работы в программной системе. У режима пользователь доступны функции d, e, f, k, l. У режима администратора присутствуют те же функции, что и в режиме пользователь, а также имеется доступ к функциям g, h, i, j.
2. Срок действия пароля – 10 дней
3. Максимальное количество записей – 15

Руководитель КР Студент гр.

доц. Высоцкий Л.Г.

Дата

# Дерево функций проекта

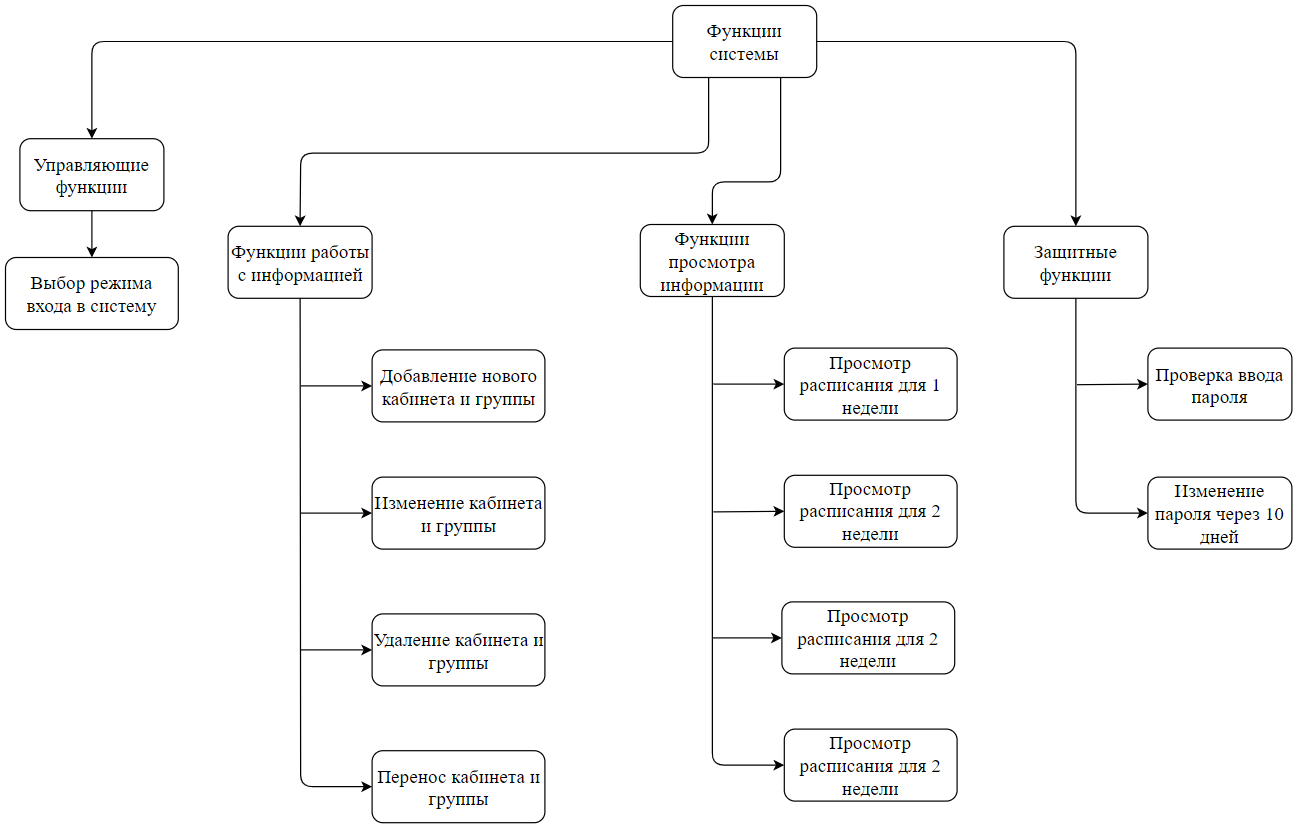


Рис.1.

# Дерево модулей программной системы

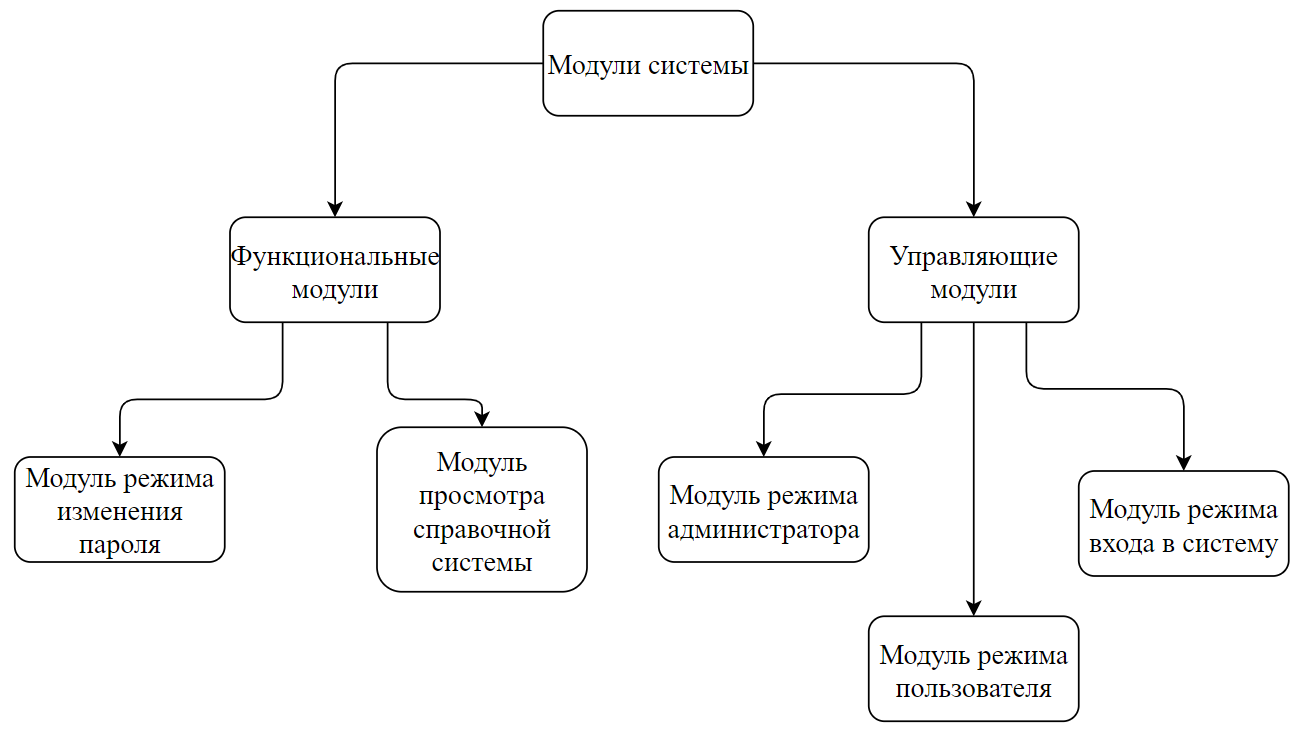


Рис.2.

# Распределение функций системы по модулям

Функции модуля режима входа в систему: функция «Проверка ввода пароля», функция «Выбор режима входа в систему»

Функции модуля изменения пароля: функция «Изменение пароля через каждые 10 дней»

Функции модуля просмотра справочной системы: функция «Справочная система»

Функция модуля режима пользователя: функция «Просмотр всего расписания», функция «Просмотр расписания для кабинета», функция «Просмотр расписания для группы»

Функции модуля режима администратора: функция: «Добавление нового кабинета и группы», функция «Изменение кабинета и группы», функция «Удаление кабинета и группы», функция «Переноса пары».

# Структуры используемых данных

БД «Курсовая.db»

Табл.1.

|  |  |  |  |
| --- | --- | --- | --- |
| Таблица «Расписание для 1 недели» | Название полей | Тип полей | Ограничения |
| Номер\_пары | Номер\_пары | TEXT |  |
| Пара №1 | Пара №1 | TEXT |  |
| Пара №2 | Пара №2 | TEXT |  |
| Пара №3 | Пара №3 | TEXT |  |
| Пара №4 | Пара №4 | TEXT |  |
| Пара №5 | Пара №5 | TEXT |  |
| Пара №6 | Пара №6 | TEXT |  |

Табл.2.

|  |  |  |  |
| --- | --- | --- | --- |
| Таблица «Расписание для 2 недели» | Название полей | Тип полей | Ограничения |
| Номер\_пары | Номер\_пары | TEXT |  |
| Пара №1 | Пара №1 | TEXT |  |
| Пара №2 | Пара №2 | TEXT |  |
| Пара №3 | Пара №3 | TEXT |  |
| Пара №4 | Пара №4 | TEXT |  |
| Пара №5 | Пара №5 | TEXT |  |
| Пара №6 | Пара №6 | TEXT |  |

Текстовый файл “secret.txt”

Первая строка текстового файла – значение текущего пароля.

Вторая строка текстового файла – дата последнего изменения пароля

Пример:

fsdafafn

2011-11-2

# Граф и видеоформы пользовательского интерфейса

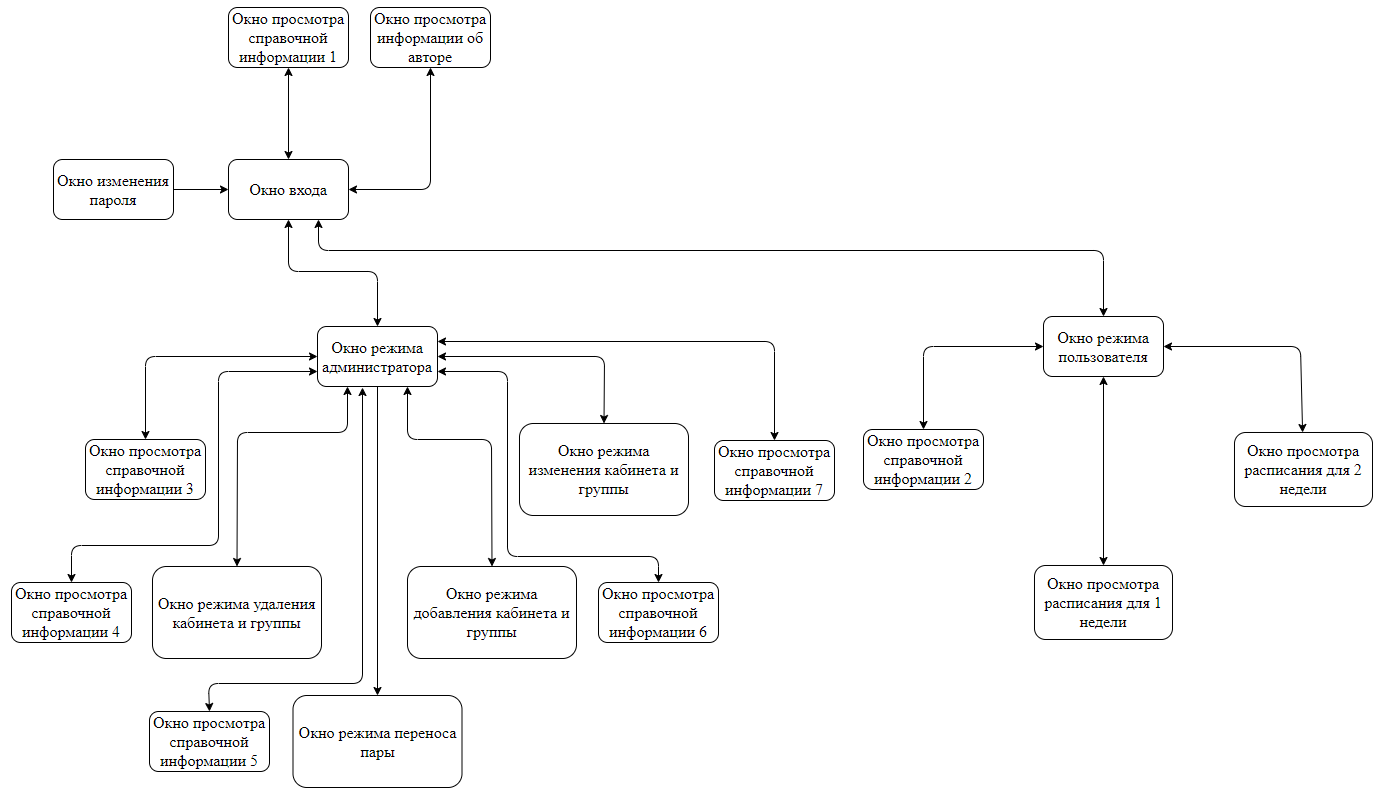


Рис. 3.

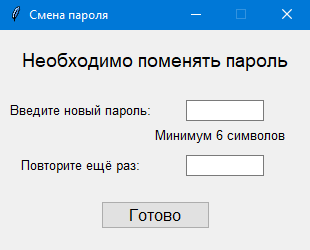


Рис. 4. Окно изменения пароля

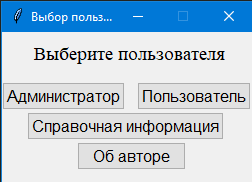


Рис.5. Окно входа

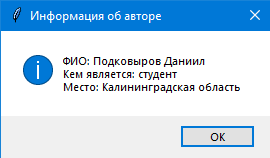


Рис.6. Окно просмотра информации об авторе

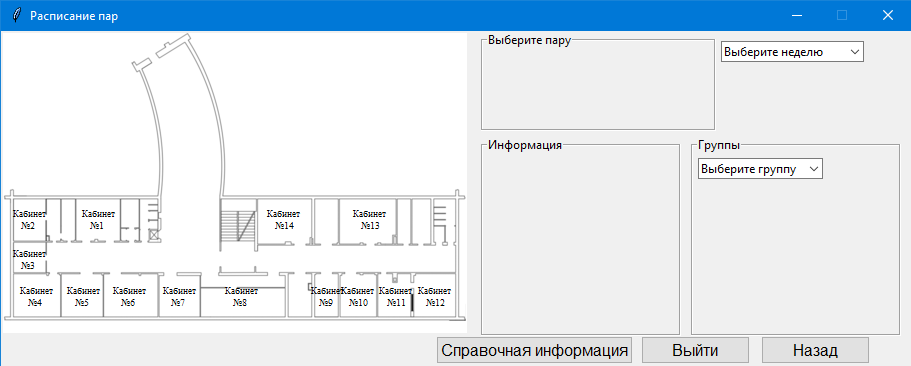


Рис.7. Окно режима пользователя

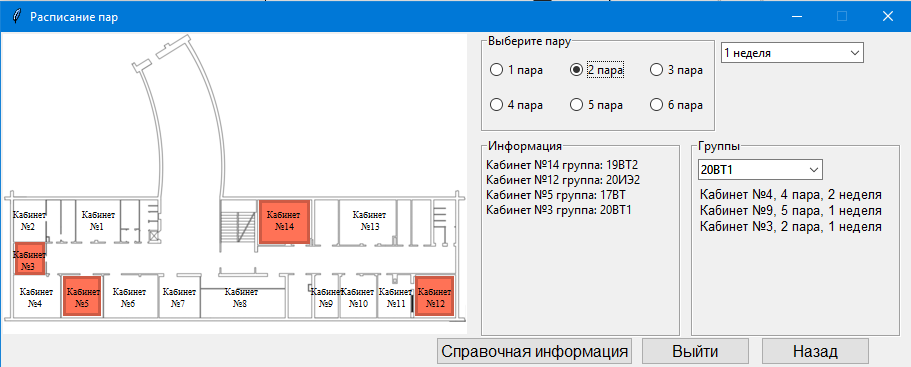


Рис.8. Окно просмотра расписания для 1 недели

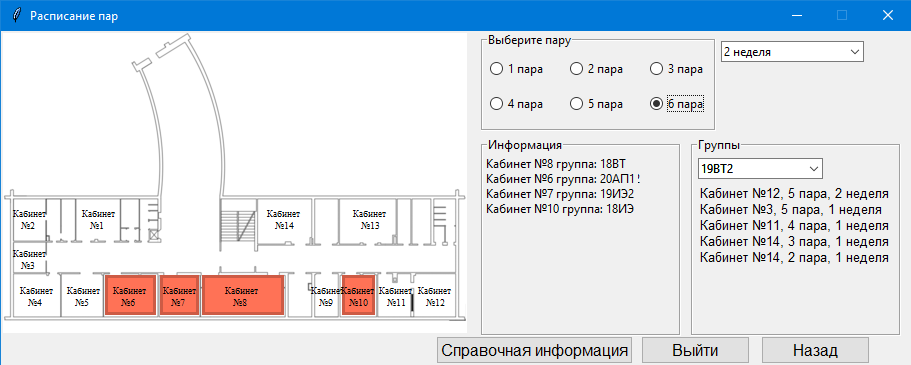


Рис.9. Окно просмотра расписания для 2 недели

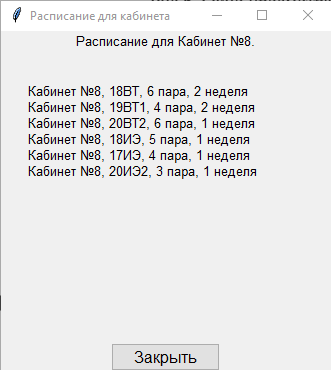


Рис.10. Окно расписания для конкретного кабинета

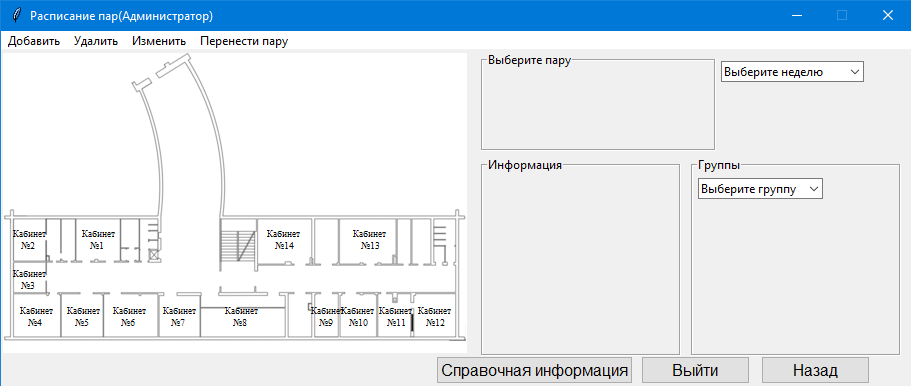


Рис.11. Окно режима администратора

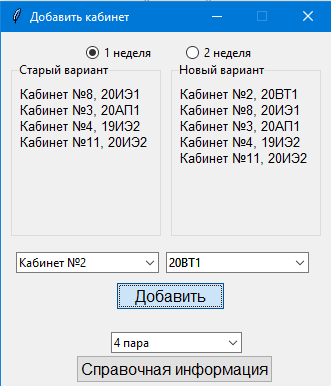


Рис.12. Окно режима добавления кабинета и группы

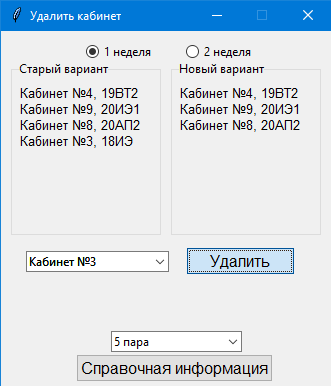


Рис.13. Окно режима удаления кабинета и группы

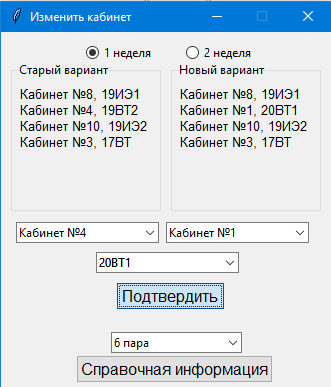


Рис.14. Окно режима изменения кабинета и группы

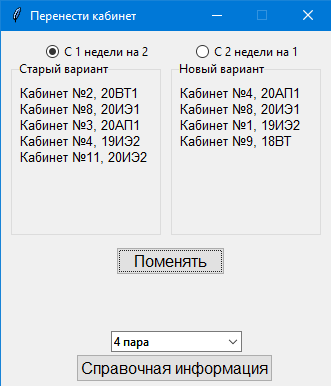


Рис.15. Окно режима переноса кабинета

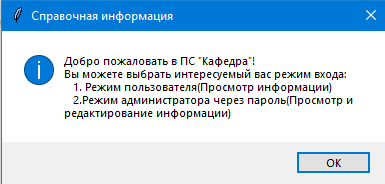


Рис.16. Окно просмотра справочной информации 1

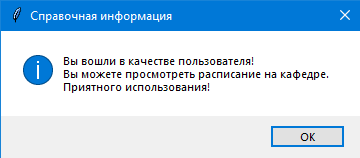


Рис.17. Окно просмотра справочной информации 2

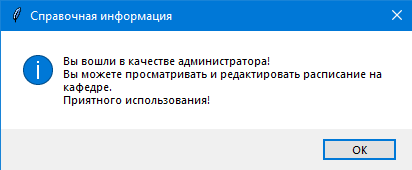


Рис.18. Окно просмотра справочной информации 3

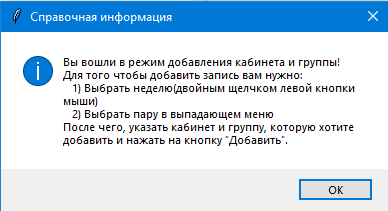


Рис.19. Окно просмотра справочной информации 4

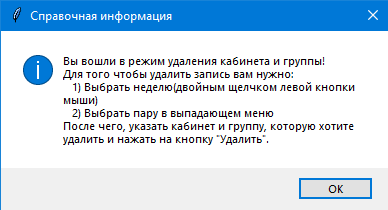


Рис.20. Окно просмотра справочной информации 3

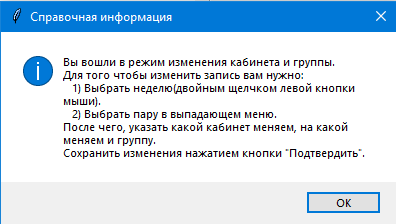


Рис.21. Окно просмотра справочной информации 3

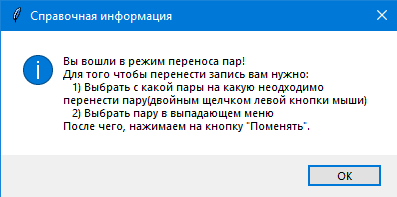


Рис.22. Окно просмотра справочной информации 3

# Блок-схемы процедур, реализующих специфические алгоритмы обработки информации

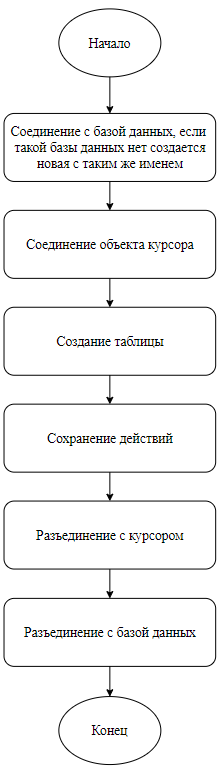
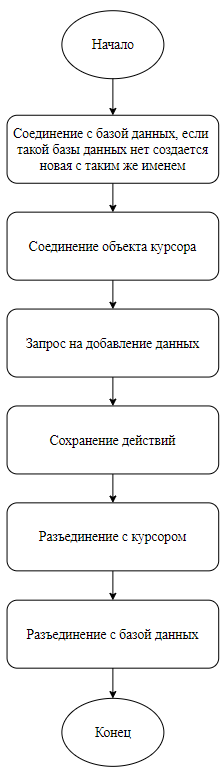
 

Рис.23. Создание таблиц Рис.24. Запрос на добавление данных

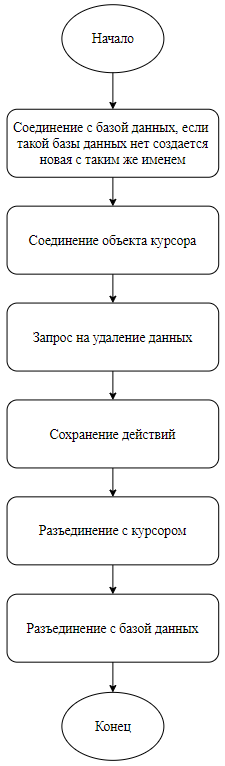


Рис.25. Запрос на удаление данны*х*

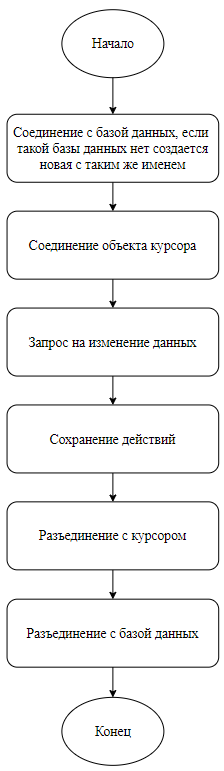


Рис.26. Запрос на изменение данных

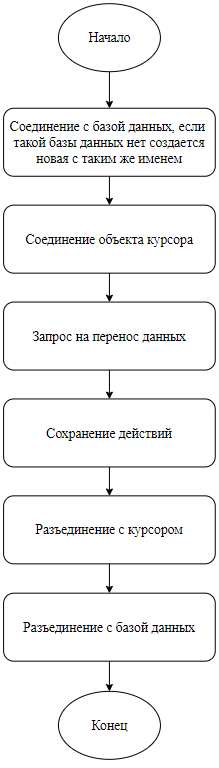


Рис.27. Запрос на изменение данных

# Листинг

import sqlite3

'''

Информационная система "Кафедра": план кафедры с указаниеме компьютерных классов.

Указание номера пары показывает, в каких классах идут занятия и в какой группе.

Два вида пользователей: администратор с возможностью редактирования и пользователь.

'''

from tkinter import\*

from tkinter import messagebox as mb

from tkinter import ttk

import random

import datetime

root = Tk()

root.title("Выбор пользователя")

root.resizable(False, False)

w = root.winfo\_screenwidth()

h = root.winfo\_screenheight()

w = w//2

h = h//2

w = w - 125

h = h - 50

root.geometry('250x150+{}+{}'.format(w, h))

file\_plan = 'plan.gif'

less1 = IntVar()

less2 = IntVar()

k = 0

check = 0

x = 0

y = 0

cab\_list = ['Кабинет №1', 'Кабинет №2', 'Кабинет №3', 'Кабинет №4',

'Кабинет №5', 'Кабинет №6', 'Кабинет №7', 'Кабинет №8',

'Кабинет №9', 'Кабинет №10', 'Кабинет №11', 'Кабинет №12',

'Кабинет №13', 'Кабинет №14']

group\_list = ['20ВТ1', '20ВТ2', '19ВТ1', '19ВТ2', '18ВТ', '17ВТ', '20ИЭ1',

'20ИЭ2', '19ИЭ1', '19ИЭ2', '18ИЭ', '17ИЭ', '20АП1', '20АП2']

para11 = ['Кабинет №12', 'Кабинет №2', 'Кабинет №6', 'Кабинет №1']

group11 = random.sample(group\_list, 4)

para12 = ['Кабинет №3', 'Кабинет №5', 'Кабинет №12', 'Кабинет №14']

group12 = random.sample(group\_list, 4)

para13 = ['Кабинет №6', 'Кабинет №8', 'Кабинет №7', 'Кабинет №14']

group13 = random.sample(group\_list, 4)

para14 = ['Кабинет №11', 'Кабинет №4', 'Кабинет №3', 'Кабинет №8']

group14 = random.sample(group\_list, 4)

para15 = ['Кабинет №3', 'Кабинет №8', 'Кабинет №9', 'Кабинет №4']

group15 = random.sample(group\_list, 4)

para16 = ['Кабинет №3', 'Кабинет №10', 'Кабинет №4', 'Кабинет №8']

group16 = random.sample(group\_list, 4)

para21 = ['Кабинет №3', 'Кабинет №12', 'Кабинет №4', 'Кабинет №14']

group21 = random.sample(group\_list, 4)

para22 = ['Кабинет №12', 'Кабинет №1', 'Кабинет №7', 'Кабинет №3']

group22 = random.sample(group\_list, 4)

para23 = ['Кабинет №1', 'Кабинет №9', 'Кабинет №5', 'Кабинет №10']

group23 = random.sample(group\_list, 4)

para24 = ['Кабинет №9', 'Кабинет №1', 'Кабинет №8', 'Кабинет №4']

group24 = random.sample(group\_list, 4)

para25 = ['Кабинет №10', 'Кабинет №12', 'Кабинет №9', 'Кабинет №1']

group25 = random.sample(group\_list, 4)

para26 = ['Кабинет №10', 'Кабинет №7', 'Кабинет №6', 'Кабинет №8']

group26 = random.sample(group\_list, 4)

para11copy = []

para12copy = []

para13copy = []

para14copy = []

para15copy = []

para16copy = []

para21copy = []

para22copy = []

para23copy = []

para24copy = []

para25copy = []

para26copy = []

some\_para = []

some\_para1 = []

some\_group2 = []

some\_group4 = []

some\_cab2 = []

cabinet = []

style = ttk.Style()

style.configure('TButton', font = "Times, 12")

style.configure('TLabel', font = "Times, 10")

style.map('TButton', background = [("active", "pink")])

#функция отслеживания мышки

def popup(event):

global x, y

x = event.x

y = event.y

menu.post(event.x\_root, event.y\_root)

print(x,y)

#функция маски ввода для entry1

def mask1(event):

global entry1

entry1.destroy()

entry1 = ttk.Entry(frame\_for\_admin, width = 25, font = "Times, 12", show = "\*")

entry1.place(x = 10, y = 2)

entry1.focus()

#функция маски ввода для entry2

def mask2(event):

global entry2

entry2.destroy()

entry2 = ttk.Entry(frame\_for\_admin, width = 25, font = "Times, 12", show = "\*")

entry2.place(x = 10, y = 30)

entry2.focus()

#функция проверки ввода пароля

def verify():

if entry1.get() == password:

if entry2.get() == password:

window\_admin()

else:

mb.showerror("Ошибка", 'Пароль не совпадает')

else:

mb.showerror("Ошибка", 'Пароль не совпадает')

def view(cabinet):

root5 = Toplevel()

root5.title("Расписание для кабинета")

w = root5.winfo\_screenwidth()

h = root5.winfo\_screenheight()

w = w//2

h = h//2

w = w - 770

h = h - 325

root5.geometry('330x340+{}+{}'.format(w, h))

label\_for\_cab = ttk.Label(root5, text = "Расписание для " + cabinet + ".")

label\_for\_cab.pack(side = TOP)

label = ttk.Label(root5)

label.place(x = 25, y = 50)

btttn\_close = ttk.Button(root5, text = "Закрыть", command = root5.destroy)

btttn\_close.pack(side = BOTTOM)

for i in range(len(para11)):

if cabinet == para11[i]:

index\_for\_para11 = para11.index(cabinet)

group\_for\_para11 = group11[index\_for\_para11]

label["text"] = para11[i] + ', ' + group11[index\_for\_para11] + ', 1 пара' ', 1 неделя' + "\n" + label["text"]

for i in range(len(para12)):

if cabinet == para12[i]:

index\_for\_para12 = para12.index(cabinet)

group\_for\_para12 = group12[index\_for\_para12]

label["text"] = para12[i] + ', ' + group12[index\_for\_para12] + ', 2 пара' + ', 1 неделя' + "\n" + label["text"]

for i in range(len(para13)):

if cabinet == para13[i]:

index\_for\_para13 = para13.index(cabinet)

group\_for\_para13 = group13[index\_for\_para13]

label["text"] = para13[i] + ', ' + group13[index\_for\_para13] + ', 3 пара' + ', 1 неделя' + "\n" + label["text"]

for i in range(len(para14)):

if cabinet == para14[i]:

index\_for\_para14 = para14.index(cabinet)

group\_for\_para14 = group14[index\_for\_para14]

label["text"] = para14[i] + ', ' + group14[index\_for\_para14] + ', 4 пара' + ', 1 неделя' + "\n" + label["text"]

for i in range(len(para15)):

if cabinet == para15[i]:

index\_for\_para15 = para15.index(cabinet)

group\_for\_para15 = group15[index\_for\_para15]

label["text"] = para15[i] + ', ' + group15[index\_for\_para15] + ', 5 пара' + ', 1 неделя' + "\n" + label["text"]

for i in range(len(para16)):

if cabinet == para16[i]:

index\_for\_para16 = para16.index(cabinet)

group\_for\_para16 = group16[index\_for\_para16]

label["text"] = para16[i] + ', ' + group16[index\_for\_para16] + ', 6 пара' + ', 1 неделя' + "\n" + label["text"]

for i in range(len(para21)):

if cabinet == para21[i]:

index\_for\_para21 = para21.index(cabinet)

group\_for\_para21 = group21[index\_for\_para21]

label["text"] = para21[i] + ', ' + group21[index\_for\_para21] + ', 1 пара' ', 2 неделя' + "\n" + label["text"]

for i in range(len(para22)):

if cabinet == para22[i]:

index\_for\_para22 = para22.index(cabinet)

group\_for\_para22 = group22[index\_for\_para22]

label["text"] = para22[i] + ', ' + group22[index\_for\_para22] + ', 2 пара' + ', 2 неделя' + "\n" + label["text"]

for i in range(len(para23)):

if cabinet == para23[i]:

index\_for\_para23 = para23.index(cabinet)

group\_for\_para23 = group23[index\_for\_para23]

label["text"] = para23[i] + ', ' + group23[index\_for\_para23] + ', 3 пара' + ', 2 неделя' + "\n" + label["text"]

for i in range(len(para24)):

if cabinet == para24[i]:

index\_for\_para24 = para24.index(cabinet)

group\_for\_para24 = group24[index\_for\_para24]

label["text"] = para24[i] + ', ' + group24[index\_for\_para24] + ', 4 пара' + ', 2 неделя' + "\n" + label["text"]

for i in range(len(para25)):

if cabinet == para25[i]:

index\_for\_para25 = para25.index(cabinet)

group\_for\_para25 = group25[index\_for\_para25]

label["text"] = para25[i] + ', ' + group25[index\_for\_para25] + ', 5 пара' + ', 2 неделя' + "\n" + label["text"]

for i in range(len(para26)):

if cabinet == para26[i]:

index\_for\_para26 = para26.index(cabinet)

group\_for\_para26 = group26[index\_for\_para26]

label["text"] = para26[i] + ', ' + group26[index\_for\_para26] + ', 6 пара' + ', 2 неделя' + "\n" + label["text"]

def trace():

global x, y

case\_cab = 0

if x >= 75 and x <= 120 and y <= 210 and y >= 165:

case\_cab = 'Кабинет №1'

view(case\_cab)

if x >= 15 and x <= 45 and y <= 210 and y >= 165:

case\_cab = 'Кабинет №2'

view(case\_cab)

if x >= 15 and x <= 45 and y <= 245 and y >= 210:

case\_cab = 'Кабинет №3'

view(case\_cab)

if x >= 15 and x <= 60 and y <= 285 and y >= 245:

case\_cab = 'Кабинет №4'

view(case\_cab)

if x >= 60 and x <= 100 and y <= 285 and y >= 245:

case\_cab = 'Кабинет №5'

view(case\_cab)

if x >= 103 and x <= 155 and y <= 285 and y >= 245:

case\_cab = 'Кабинет №6'

view(case\_cab)

if x >= 160 and x <= 200 and y <= 285 and y >= 245:

case\_cab = 'Кабинет №7'

view(case\_cab)

if x >= 205 and x <= 280 and y <= 285 and y >= 245:

case\_cab = 'Кабинет №8'

view(case\_cab)

if x >= 315 and x <= 335 and y <= 285 and y >= 245:

case\_cab = 'Кабинет №9'

view(case\_cab)

if x >= 340 and x <= 377 and y <= 285 and y >= 245:

case\_cab = 'Кабинет №10'

view(case\_cab)

if x >= 380 and x <= 407 and y <= 285 and y >= 245:

case\_cab = 'Кабинет №11'

view(case\_cab)

if x >= 415 and x <= 452 and y <= 285 and y >= 245:

case\_cab = 'Кабинет №12'

view(case\_cab)

if x >= 342 and x <= 394 and y <= 210 and y >= 165:

case\_cab = 'Кабинет №13'

view(case\_cab)

if x >= 257 and x <= 310 and y <= 210 and y >= 165:

case\_cab = 'Кабинет №14'

view(case\_cab)

#функция просмотра кабинетов

def cab(\*args):

global canv

lbl\_inf = Label(frame2, justify = LEFT)

lbl\_inf.place(x = 0, y = 0)

lbl\_inf['text'] = ' '

choice\_cab = []

canv.delete('cab1','cab2','cab3','cab4','cab5','cab6','cab7','cab8','cab9','cab10','cab11','cab12','cab13','cab14')

if combobox\_for\_week.get() == "1 неделя":

if less1.get() == 1:

choice\_cab = para11

choice\_group = group11

elif less1.get() == 2:

choice\_cab = para12

choice\_group = group12

elif less1.get() == 3:

choice\_cab = para13

choice\_group = group13

elif less1.get() == 4:

choice\_cab = para14

choice\_group = group14

elif less1.get() == 5:

choice\_cab = para15

choice\_group = group15

elif less1.get() == 6:

choice\_cab = para16

choice\_group = group16

if combobox\_for\_week.get() == "2 неделя":

if less2.get() == 1:

choice\_cab = para21

choice\_group = group21

elif less2.get() == 2:

choice\_cab = para22

choice\_group = group22

elif less2.get() == 3:

choice\_cab = para23

choice\_group = group23

elif less2.get() == 4:

choice\_cab = para24

choice\_group = group24

elif less2.get() == 5:

choice\_cab = para25

choice\_group = group25

elif less2.get() == 6:

choice\_cab = para26

choice\_group = group26

for i in range(len(choice\_cab)):

if choice\_cab[i] == 'Кабинет №1':

canv.delete('cab1text')

canv.create\_rectangle(77, 169, 116, 207, width = 3, fill = 'coral1', outline = 'coral3', tag = 'cab1')

canv.create\_text(97, 188, text = 'Кабинет\n №1', font = 'Times 7', tag = 'cab1text')

lbl\_inf['text'] = 'Кабинет №1 группа: ' + choice\_group[i] + "\n" + lbl\_inf['text']

if choice\_cab[i] == 'Кабинет №2':

canv.delete('cab2text')

canv.create\_rectangle(15, 169, 42, 207, width = 3, fill = 'coral1', outline = 'coral3', tag = 'cab2')

canv.create\_text(28, 188, text = 'Кабинет\n №2', font = 'Times 7', tag = 'cab2text')

lbl\_inf['text'] = 'Кабинет №2 группа: ' + choice\_group[i] + "\n" + lbl\_inf['text']

if choice\_cab[i] == 'Кабинет №3':

canv.delete('cab3text')

canv.create\_rectangle(15, 211, 42, 241, width = 3, fill = 'coral1', outline = 'coral3', tag = 'cab3')

canv.create\_text(28, 228, text = 'Кабинет\n №3', font = 'Times 7', tag = 'cab3text')

lbl\_inf['text'] = 'Кабинет №3 группа: ' + choice\_group[i] + "\n" + lbl\_inf['text']

if choice\_cab[i] == 'Кабинет №4':

canv.delete('cab4text')

canv.create\_rectangle(15, 245, 56, 282, width = 3, fill = 'coral1', outline = 'coral3', tag = 'cab4')

canv.create\_text(35, 265, text = 'Кабинет\n №4', font = 'Times 7', tag = 'cab4text')

lbl\_inf['text'] = 'Кабинет №4 группа: ' + choice\_group[i] + "\n" + lbl\_inf['text']

if choice\_cab[i] == 'Кабинет №5':

canv.delete('cab5text')

canv.create\_rectangle(63, 245, 98, 282, width = 3, fill = 'coral1', outline = 'coral3', tag = 'cab5')

canv.create\_text(82, 265, text = 'Кабинет\n №5', font = 'Times 7', tag = 'cab5text')

lbl\_inf['text'] = 'Кабинет №5 группа: ' + choice\_group[i] + "\n" + lbl\_inf['text']

if choice\_cab[i] == 'Кабинет №6':

canv.delete('cab6text')

canv.create\_rectangle(105, 245, 153, 282, width = 3, fill = 'coral1', outline = 'coral3', tag = 'cab6')

canv.create\_text(128, 265, text = 'Кабинет\n №6', font = 'Times 7', tag = 'cab6text')

lbl\_inf['text'] = 'Кабинет №6 группа: ' + choice\_group[i] + "\n" + lbl\_inf['text']

if choice\_cab[i] == 'Кабинет №7':

canv.delete('cab7text')

canv.create\_rectangle(160, 245, 196, 282, width = 3, fill = 'coral1', outline = 'coral3', tag = 'cab7')

canv.create\_text(178, 265, text = 'Кабинет\n №7', font = 'Times 7', tag = 'cab7text')

lbl\_inf['text'] = 'Кабинет №7 группа: ' + choice\_group[i] + "\n" + lbl\_inf['text']

if choice\_cab[i] == 'Кабинет №8':

canv.delete('cab8text')

canv.create\_rectangle(202, 245, 281, 282, width = 3, fill = 'coral1', outline = 'coral3', tag = 'cab8')

canv.create\_text(240, 265, text = 'Кабинет\n №8', font = 'Times 7', tag = 'cab8text')

lbl\_inf['text'] = 'Кабинет №8 группа: ' + choice\_group[i] + "\n" + lbl\_inf['text']

if choice\_cab[i] == 'Кабинет №9':

canv.delete('cab9text')

canv.create\_rectangle(316, 245, 334, 282, width = 3, fill = 'coral1', outline = 'coral3', tag = 'cab9')

canv.create\_text(326, 265, text = 'Кабинет\n №9', font = 'Times 7', tag = 'cab9text')

lbl\_inf['text'] = 'Кабинет №9 группа: ' + choice\_group[i] + "\n" + lbl\_inf['text']

if choice\_cab[i] == 'Кабинет №10':

canv.delete('cab10text')

canv.create\_rectangle(342, 245, 372, 282, width = 3, fill = 'coral1', outline = 'coral3', tag = 'cab10')

canv.create\_text(356, 265, text = 'Кабинет\n №10', font = 'Times 7', tag = 'cab10text')

lbl\_inf['text'] = 'Кабинет №10 группа: ' + choice\_group[i] + "\n" + lbl\_inf['text']

if choice\_cab[i] == 'Кабинет №11':

canv.delete('cab11text')

canv.create\_rectangle(379, 245, 406, 282, width = 3, fill = 'coral1', outline = 'coral3', tag = 'cab11')

canv.create\_text(394, 265, text = 'Кабинет\n №11', font = 'Times 7', tag = 'cab11text')

lbl\_inf['text'] = 'Кабинет №11 группа: ' + choice\_group[i] + "\n" + lbl\_inf['text']

if choice\_cab[i] == 'Кабинет №12':

canv.delete('cab12text')

canv.create\_rectangle(415, 245, 451, 282, width = 3, fill = 'coral1', outline = 'coral3', tag = 'cab12')

canv.create\_text(433, 265, text = 'Кабинет\n №12', font = 'Times 7', tag = 'cab12text')

lbl\_inf['text'] = 'Кабинет №12 группа: ' + choice\_group[i] + "\n" + lbl\_inf['text']

if choice\_cab[i] == 'Кабинет №13':

canv.delete('cab13text')

canv.create\_rectangle(342, 169, 390, 210, width = 3, fill = 'coral1', outline = 'coral3', tag = 'cab13')

canv.create\_text(368, 188, text = 'Кабинет\n №13', font = 'Times 7', tag = 'cab13text')

lbl\_inf['text'] = 'Кабинет №13 группа: ' + choice\_group[i] + "\n" + lbl\_inf['text']

if choice\_cab[i] == 'Кабинет №14':

canv.delete('cab14text')

canv.create\_rectangle(259, 169, 307, 210, width = 3, fill = 'coral1', outline = 'coral3', tag = 'cab14')

canv.create\_text(282, 188, text = 'Кабинет\n №14', font = 'Times 7', tag = 'cab14text')

lbl\_inf['text'] = 'Кабинет №14 группа: ' + choice\_group[i] + "\n" + lbl\_inf['text']

#функция для разделения расписания на 1 и 2 недели

def week(event):

if combobox\_for\_week.get() == "1 неделя":

rb1 = ttk.Radiobutton(frame1, text = '1 пара', value = 1, variable = less1)

rb1.place(x = 5, y = 10)

rb1.bind('<Double-Button-1>', cab)

rb2 = ttk.Radiobutton(frame1, text = '2 пара', value = 2, variable = less1)

rb2.place(x = 85, y = 10)

rb2.bind('<Double-Button-1>', cab)

rb3 = ttk.Radiobutton(frame1, text = '3 пара', value = 3, variable = less1)

rb3.place(x = 165, y = 10)

rb3.bind('<Double-Button-1>', cab)

rb4 = ttk.Radiobutton(frame1, text = '4 пара', value = 4, variable = less1)

rb4.place(x = 5, y = 45)

rb4.bind('<Double-Button-1>', cab)

rb5 = ttk.Radiobutton(frame1, text = '5 пара', value = 5, variable = less1)

rb5.place(x = 85, y = 45)

rb5.bind('<Double-Button-1>', cab)

rb6 = ttk.Radiobutton(frame1, text = '6 пара', value = 6, variable = less1)

rb6.place(x = 165, y = 45)

rb6.bind('<Double-Button-1>', cab)

elif combobox\_for\_week.get() == "2 неделя":

rb1 = ttk.Radiobutton(frame1, text = '1 пара', value = 1, variable = less2)

rb1.place(x = 5, y = 10)

rb1.bind('<Double-Button-1>', cab)

rb2 = ttk.Radiobutton(frame1, text = '2 пара', value = 2, variable = less2)

rb2.place(x = 85, y = 10)

rb2.bind('<Double-Button-1>', cab)

rb3 = ttk.Radiobutton(frame1, text = '3 пара', value = 3, variable = less2)

rb3.place(x = 165, y = 10)

rb3.bind('<Double-Button-1>', cab)

rb4 = ttk.Radiobutton(frame1, text = '4 пара', value = 4, variable = less2)

rb4.place(x = 5, y = 45)

rb4.bind('<Double-Button-1>', cab)

rb5 = ttk.Radiobutton(frame1, text = '5 пара', value = 5, variable = less2)

rb5.place(x = 85, y = 45)

rb5.bind('<Double-Button-1>', cab)

rb6 = ttk.Radiobutton(frame1, text = '6 пара', value = 6, variable = less2)

rb6.place(x = 165, y = 45)

rb6.bind('<Double-Button-1>', cab)

#функция изменения кабинета и группы

def switch(event):

global label\_inf\_pick\_less2

oldCab = cb1.get()

newCab = cb2.get()

newGroup = cb3.get()

if newGroup == "Группа, на которую меняем":

indexcab = some\_para.index(oldCab)

if newCab in some\_para:

mb.showerror("Ошибка!", "Попробуйте еще раз")

elif (newCab not in some\_para and newGroup not in some\_group2):

indexcab = some\_para.index(oldCab)

some\_para.pop(indexcab)

some\_para.insert(indexcab, newCab)

mb.showinfo("Успех!", 'Запись успешно изменена!')

for x in range(len(some\_para)):

label\_inf\_pick\_less2["text"] = some\_para[x] + ", " + some\_group2[x] + "\n" + label\_inf\_pick\_less2["text"]

cab()

else:

if newCab in some\_para:

mb.showerror("Ошибка!", "Попробуйте еще раз")

elif (newCab not in some\_para and newGroup not in some\_group2):

indexcab = some\_para.index(oldCab)

some\_group2.pop(indexcab)

some\_para.pop(indexcab)

some\_group2.insert(indexcab, newGroup)

some\_para.insert(indexcab, newCab)

mb.showinfo("Успех!", 'Запись успешно изменена!')

for x in range(len(some\_para)):

label\_inf\_pick\_less2["text"] = some\_para[x] + ", " + some\_group2[x] + "\n" + label\_inf\_pick\_less2["text"]

else:

mb.showerror("Ошибка!", "Попробуйте еще раз")

cab()

#функция построения окна изменения

def convert(some\_less, some\_group):

global label\_inf\_pick\_less1, root1, cb1, cb2, some\_para, label\_inf\_pick\_less2, some\_group2, cb3

label\_inf\_pick\_less1["text"] = " "

label\_inf\_pick\_less2["text"] = " "

for x in range(len(some\_less)):

label\_inf\_pick\_less1["text"] = some\_less[x] + ", " + some\_group[x] + "\n" + label\_inf\_pick\_less1["text"]

cb1 = ttk.Combobox(root1, value = some\_less)

cb1.place(x = 330/2 - 150, y = 190)

cb1.set("Кабинет, который нужно заменить")

cb2 = ttk.Combobox(root1, value = cab\_list)

cb2.place(x = 330/2, y = 190)

cb2.set("Кабинет, на который меняем")

cb3 = ttk.Combobox(root1, value = group\_list)

cb3.place(x = 330/2 - 70, y = 220)

cb3.set("Группа, на которую меняем")

bttn\_accept1 = ttk.Button(root1, text = "Подтвердить")

bttn\_accept1.place(x = 330/2 - 50, y = 250)

bttn\_accept1.bind("<Button-1>", switch)

some\_para = some\_less

some\_group2 = some\_group

#функция для распределения выбранной пары с текущей

def output\_less(event):

global label\_inf\_pick\_less1

if rbweek.get() == 1:

if combobox\_for\_pick\_less.get() == "1 пара":

para11copy = para11

convert(para11copy, group11)

elif combobox\_for\_pick\_less.get() == "2 пара":

para12copy = para12

convert(para12copy, group12)

elif combobox\_for\_pick\_less.get() == "3 пара":

para13copy = para13

convert(para13copy, group13)

elif combobox\_for\_pick\_less.get() == "4 пара":

para14copy = para14

convert(para14copy, group14)

elif combobox\_for\_pick\_less.get() == "5 пара":

para15copy = para15

convert(para15copy, group15)

elif combobox\_for\_pick\_less.get() == "6 пара":

para16copy = para16

convert(para16copy, group16)

if rbweek.get() == 2:

if combobox\_for\_pick\_less.get() == "1 пара":

para21copy = para21

convert(para21copy, group21)

elif combobox\_for\_pick\_less.get() == "2 пара":

para22copy = para22

convert(para22copy, group22)

elif combobox\_for\_pick\_less.get() == "3 пара":

para23copy = para23

convert(para23copy, group23)

elif combobox\_for\_pick\_less.get() == "4 пара":

para24copy = para24

convert(para24copy, group24)

elif combobox\_for\_pick\_less.get() == "5 пара":

para25copy = para25

convert(para25copy, group25)

elif combobox\_for\_pick\_less.get() == "6 пара":

para26copy = para26

convert(para26copy, group26)

def root1\_inf(event):

mb.showinfo("Справочная информация", "Вы вошли в режим изменения кабинета и группы.\nДля того чтобы изменить запись вам нужно:\n 1) Выбрать неделю(двойным щелчком левой кнопки мыши).\n 2) Выбрать пару в выпадающем меню.\nПосле чего, указать какой кабинет меняем, на какой меняем и группу.\nСохранить изменения нажатием кнопки \"Подтвердить\".")

#функция построения окна изменения

def change\_cab():

global rbweek, combobox\_for\_pick\_less, label\_inf\_pick\_less1, root1, label\_inf\_pick\_less2

rbweek = IntVar()

root1 = Toplevel()

root1.title("Изменить кабинет")

root1.resizable(False, False)

oldframe = ttk.LabelFrame(root1, text = "Старый вариант", width = 150, height = 150)

newframe = ttk.LabelFrame(root1, text = "Новый вариант", width = 150, height = 150)

oldframe. place(x = 10, y = 30)

newframe.place(x = 170, y = 30)

w = root1.winfo\_screenwidth()

h = root1.winfo\_screenheight()

w = w//2

h = h//2

w = w - 465

h = h - 52

root1.geometry('330x355+{}+{}'.format(w, h))

rb\_for\_week1 = ttk.Radiobutton(root1, text = "1 неделя", value = 1, variable = rbweek)

rb\_for\_week1.place(x = 265/2 - 50, y = 10)

rb\_for\_week2 = ttk.Radiobutton(root1, text = "2 неделя", value = 2, variable = rbweek)

rb\_for\_week2.place(x = 265/2 + 50, y = 10)

label\_inf\_pick\_less1 = ttk.Label(oldframe)

label\_inf\_pick\_less1.place(x = 5, y = 5)

label\_inf\_pick\_less2 = ttk.Label(newframe)

label\_inf\_pick\_less2.place(x = 5, y = 5)

combobox\_for\_pick\_less = ttk.Combobox(root1, values = ["1 пара", "2 пара", "3 пара", "4 пара", "5 пара", "6 пара"], width = 18)

combobox\_for\_pick\_less.place(x = 330/2 - 55, y = 300)

combobox\_for\_pick\_less.set("Выберите пару")

combobox\_for\_pick\_less.bind("<<ComboboxSelected>>", output\_less)

btttn\_inf = ttk.Button(root1, text = "Справочная информация")

btttn\_inf.place(x = 75, y = 323)

btttn\_inf.bind("<Button-1>", root1\_inf)

#функция для добавления кабинета и группы

def add(event):

global group11, label\_inf\_pick\_less1, label\_inf\_pick\_less2

newCab = cb2.get()

newGroup = cb1.get()

label\_inf\_pick\_less2["text"] = " "

if newGroup == "Выберите группу":

mb.showerror("Ошибка!", "Выберите группу!")

return newGroup

if newCab in some\_para1 or newGroup in some\_group1:

mb.showerror("Ошибка!", "Попробуйте еще раз!")

elif newCab not in some\_para1 and newGroup not in some\_group1:

some\_para1.append(newCab)

some\_group1.append(newGroup)

for x in range(len(some\_para1)):

label\_inf\_pick\_less2["text"] = some\_para1[x] + ", " + some\_group1[x] + "\n" + label\_inf\_pick\_less2["text"]

mb.showinfo("Успех!", 'Запись успешно добавлена!')

#elif newGroup in

cab()

#функция построения окна добавления

def choice\_cab\_add1(some\_cab, some\_group):

global some\_para1, cb2, cb1, label\_inf\_pick\_less1, label\_inf\_pick\_less2, some\_group1

label\_inf\_pick\_less1["text"] = " "

label\_inf\_pick\_less2["text"] = " "

print(some\_group)

for x in range(len(some\_cab)):

label\_inf\_pick\_less1["text"] = some\_cab[x] + ", " + some\_group[x] + "\n" + label\_inf\_pick\_less1["text"]

cb2 = ttk.Combobox(root2, value = cab\_list)

cb2.place(x = 330/2 - 150, y = 220)

cb2.set("Выберите кабинет")

cb1 = ttk.Combobox(root2, value = group\_list)

cb1.place(x = 330/2, y = 220)

cb1.set("Выберите группу")

bttn\_accept2 = ttk.Button(root2, text = "Добавить")

bttn\_accept2.place(x = 330/2 - 50, y = 250)

bttn\_accept2.bind("<Button-1>", add)

some\_para1 = some\_cab

some\_group1 = some\_group

#функция для распределения выбранной пары с текущей

def choice\_cab\_add(event):

if rbweek1.get() == 1:

if cb\_para.get() == "1 пара":

para11copy = para11

choice\_cab\_add1(para11copy, group11)

elif cb\_para.get() == "2 пара":

para12copy = para12

choice\_cab\_add1(para12copy, group12)

elif cb\_para.get() == "3 пара":

para13copy = para13

choice\_cab\_add1(para13copy, group13)

elif cb\_para.get() == "4 пара":

para14copy = para14

choice\_cab\_add1(para14copy, group14)

elif cb\_para.get() == "5 пара":

para15copy = para15

choice\_cab\_add1(para15copy, group15)

elif cb\_para.get() == "6 пара":

para16copy = para16

choice\_cab\_add1(para16copy, group16)

elif rbweek1.get() == 2:

if cb\_para.get() == "1 пара":

para21copy = para21

choice\_cab\_add1(para21copy, group21)

elif cb\_para.get() == "2 пара":

para22copy = para22

choice\_cab\_add1(para22copy, group22)

elif cb\_para.get() == "3 пара":

para23copy = para23

choice\_cab\_add1(para23copy, group23)

elif cb\_para.get() == "4 пара":

para24copy = para24

choice\_cab\_add1(para24copy, group24)

elif cb\_para.get() == "5 пара":

para25copy = para25

choice\_cab\_add1(para25copy, group25)

elif cb\_para.get() == "6 пара":

para26copy = para26

choice\_cab\_add1(para26copy, group26)

def root2\_inf(event):

mb.showinfo("Справочная информация", "Вы вошли в режим добавления кабинета и группы!\nДля того чтобы добавить запись вам нужно:\n 1) Выбрать неделю(двойным щелчком левой кнопки мыши)\n 2) Выбрать пару в выпадающем меню\nПосле чего, указать кабинет и группу, которую хотите добавить и нажать на кнопку \"Добавить\".")

#функция построения окна добавления

def add\_cab():

global cb\_para, rbweek1, label\_inf\_pick\_less1, label\_inf\_pick\_less2, root2

rbweek1 = IntVar()

root2 = Toplevel()

root2.title("Добавить кабинет")

root2.resizable(False, False)

oldframe = ttk.LabelFrame(root2, text = "Старый вариант", width = 150, height = 175)

newframe = ttk.LabelFrame(root2, text = "Новый вариант", width = 150, height = 175)

oldframe. place(x = 10, y = 30)

newframe.place(x = 170, y = 30)

w = root2.winfo\_screenwidth()

h = root2.winfo\_screenheight()

w = w//2

h = h//2

w = w - 465

h = h - 52

root2.geometry('330x355+{}+{}'.format(w, h))

label\_inf\_pick\_less1 = ttk.Label(oldframe)

label\_inf\_pick\_less1.place(x = 5, y = 5)

label\_inf\_pick\_less2 = ttk.Label(newframe)

label\_inf\_pick\_less2.place(x = 5, y = 5)

rb\_for\_week1 = ttk.Radiobutton(root2, text = "1 неделя", value = 1, variable = rbweek1)

rb\_for\_week1.place(x = 265/2 - 50, y = 10)

rb\_for\_week2 = ttk.Radiobutton(root2, text = "2 неделя", value = 2, variable = rbweek1)

rb\_for\_week2.place(x = 265/2 + 50, y = 10)

cb\_para = ttk.Combobox(root2, values = ["1 пара", "2 пара", "3 пара", "4 пара", "5 пара", "6 пара"], width = 18)

cb\_para.set("Выберите пару")

cb\_para.place(x = 330/2 - 55, y = 300)

cb\_para.bind("<<ComboboxSelected>>", choice\_cab\_add)

btttn\_inf = ttk.Button(root2, text = "Справочная информация")

btttn\_inf.place(x = 75, y = 323)

btttn\_inf.bind("<Button-1>", root2\_inf)

#функция для удаления кабинета и группы

def delete(event):

global label\_inf\_pick\_less2

delCab = cb2.get()

label\_inf\_pick\_less2["text"] = " "

if delCab in some\_cab2:

indexdelcab = some\_cab2.index(delCab)

some\_cab2.pop(indexdelcab)

some\_group4.pop(indexdelcab)

mb.showinfo("Успех!", 'Кабинет успешно удален!')

cab()

for i in range(len(some\_cab2)):

label\_inf\_pick\_less2["text"] = some\_cab2[i] + ", " + some\_group4[i] + "\n" + label\_inf\_pick\_less2["text"]

#функция построения окна удаления

def choice\_cab\_del1(some\_cab1, some\_group3):

global some\_cab2, cb2, label\_inf\_pick\_less1, label\_inf\_pick\_less2, some\_group4

label\_inf\_pick\_less1["text"] = " "

label\_inf\_pick\_less2["text"] = " "

print(some\_group3)

print(some\_cab1)

for x in range(len(some\_cab1)):

label\_inf\_pick\_less1["text"] = some\_cab1[x] + ", " + some\_group3[x] + "\n" + label\_inf\_pick\_less1["text"]

cb2 = ttk.Combobox(root3, value = some\_cab1)

cb2.place(x = 350/2 - 150, y = 220)

cb2.set("Выберите кабинет")

bttn\_accept2 = ttk.Button(root3, text = "Удалить")

bttn\_accept2.place(x = 350/2 + 10, y = 216)

bttn\_accept2.bind("<Button-1>", delete)

some\_group4 = some\_group3

some\_cab2 = some\_cab1

#функция для распределения выбранной пары с текущей

def choice\_cab\_del(event):

if rbweek2.get() == 1:

print("1 неделя")

if cb\_para.get() == "1 пара":

print("1 пара")

para11copy = para11

choice\_cab\_del1(para11copy, group16)

elif cb\_para.get() == "2 пара":

print("2 пара")

para12copy = para12

choice\_cab\_del1(para12copy, group12)

elif cb\_para.get() == "3 пара":

print("3 пара")

para13copy = para13

choice\_cab\_del1(para13copy, group13)

elif cb\_para.get() == "4 пара":

print("4 пара")

para14copy = para14

choice\_cab\_del1(para14copy, group14)

elif cb\_para.get() == "5 пара":

print("5 пара")

para15copy = para15

choice\_cab\_del1(para15copy, group15)

elif cb\_para.get() == "6 пара":

print("6 пара")

para16copy = para16

choice\_cab\_del1(para16copy, group16)

elif rbweek2.get() == 2:

print("2 неделя")

if cb\_para.get() == "1 пара":

print("1 пара")

para21copy = para21

choice\_cab\_del1(para21copy, group21)

elif cb\_para.get() == "2 пара":

print("2 пара")

para22copy = para22

choice\_cab\_del1(para22copy, group22)

elif cb\_para.get() == "3 пара":

print("3 пара")

para23copy = para23

choice\_cab\_del1(para23copy, group23)

elif cb\_para.get() == "4 пара":

print("4 пара")

para24copy = para24

choice\_cab\_del1(para24copy, group24)

elif cb\_para.get() == "5 пара":

print("5 пара")

para25copy = para25

choice\_cab\_del1(para25copy, group25)

elif cb\_para.get() == "6 пара":

print("6 пара")

para26copy = para26

choice\_cab\_del1(para26copy, group26)

def root5\_inf(event):

mb.showinfo("Справочная информация", "Вы вошли в режим удаления кабинета и группы!\nДля того чтобы удалить запись вам нужно:\n 1) Выбрать неделю(двойным щелчком левой кнопки мыши)\n 2) Выбрать пару в выпадающем меню\nПосле чего, указать кабинет и группу, которую хотите удалить и нажать на кнопку \"Удалить\".")

#функция построения окна удаления

def del\_cab():

global cb\_para, rbweek2, label\_inf\_pick\_less1, label\_inf\_pick\_less2, root3

rbweek2 = IntVar()

root3 = Toplevel()

root3.title("Удалить кабинет")

root3.resizable(False, False)

oldframe = ttk.LabelFrame(root3, text = "Старый вариант", width = 150, height = 175)

newframe = ttk.LabelFrame(root3, text = "Новый вариант", width = 150, height = 175)

oldframe. place(x = 10, y = 30)

newframe.place(x = 170, y = 30)

w = root3.winfo\_screenwidth()

h = root3.winfo\_screenheight()

w = w//2

h = h//2

w = w - 465

h = h - 52

root3.geometry('330x355+{}+{}'.format(w, h))

label\_inf\_pick\_less1 = ttk.Label(oldframe)

label\_inf\_pick\_less1.place(x = 5, y = 5)

label\_inf\_pick\_less2 = ttk.Label(newframe)

label\_inf\_pick\_less2.place(x = 5, y = 5)

rb\_for\_week1 = ttk.Radiobutton(root3, text = "1 неделя", value = 1, variable = rbweek2)

rb\_for\_week1.place(x = 265/2 - 50, y = 10)

rb\_for\_week2 = ttk.Radiobutton(root3, text = "2 неделя", value = 2, variable = rbweek2)

rb\_for\_week2.place(x = 265/2 + 50, y = 10)

cb\_para = ttk.Combobox(root3, values = ["1 пара", "2 пара", "3 пара", "4 пара", "5 пара", "6 пара"], width = 18)

cb\_para.set("Выберите пару")

cb\_para.place(x = 330/2 - 55, y = 300)

cb\_para.bind("<<ComboboxSelected>>", choice\_cab\_del)

btttn\_inf = ttk.Button(root3, text = "Справочная информация")

btttn\_inf.place(x = 75, y = 323)

btttn\_inf.bind("<Button-1>", root5\_inf)

#функция для переноса кабинета и группы

def switchtc(event):

global para11, group11, para12, group12, para13, group13, para14, group14, para15, group15, para16, group16, para21, group21, para22, group22, para23, group23, para24, group24, para25, group25, para26, group26

if rbweek4.get() == 1:

print("1 неделя")

if cb\_para.get() == "1 пара":

para11 = para21

group11 = group21

para21 = some\_cab4

group21 = some\_group6

print(para21, group21)

for i in range(len(para11)):

label\_inf\_pick\_less2["text"] = para11[i] + ", " + group11[i] + "\n" + label\_inf\_pick\_less2["text"]

cab()

elif cb\_para.get() == "2 пара":

para12 = para22

group12 = group22

para22 = some\_cab4

group22 = some\_group6

for i in range(len(para12)):

label\_inf\_pick\_less2["text"] = para12[i] + ", " + group12[i] + "\n" + label\_inf\_pick\_less2["text"]

cab()

elif cb\_para.get() == "3 пара":

para13 = para23

group13 = group23

para23 = some\_cab4

group23 = some\_group6

for i in range(len(para13)):

label\_inf\_pick\_less2["text"] = para13[i] + ", " + group13[i] + "\n" + label\_inf\_pick\_less2["text"]

cab()

elif cb\_para.get() == "4 пара":

para14 = para24

group14 = group24

para24 = some\_cab4

group24 = some\_group6

for i in range(len(para14)):

label\_inf\_pick\_less2["text"] = para14[i] + ", " + group14[i] + "\n" + label\_inf\_pick\_less2["text"]

cab()

elif cb\_para.get() == "5 пара":

para15 = para25

group15 = group25

para25 = some\_cab4

group25 = some\_group6

for i in range(len(para15)):

label\_inf\_pick\_less2["text"] = para15[i] + ", " + group15[i] + "\n" + label\_inf\_pick\_less2["text"]

cab()

elif cb\_para.get() == "6 пара":

para16 = para26

group16 = group26

para26 = some\_cab4

group26 = some\_group6

for i in range(len(para16)):

label\_inf\_pick\_less2["text"] = para16[i] + ", " + group16[i] + "\n" + label\_inf\_pick\_less2["text"]

cab()

elif rbweek4.get() == 2:

print("2 неделя")

if cb\_para.get() == "1 пара":

para21 = para11

group21 = group11

para11 = some\_cab4

group11 = some\_group6

for i in range(len(para21)):

label\_inf\_pick\_less2["text"] = para21[i] + ", " + group21[i] + "\n" + label\_inf\_pick\_less2["text"]

cab()

elif cb\_para.get() == "2 пара":

para22 = para12

group22 = group12

para12 = some\_cab4

group12 = some\_group6

for i in range(len(para22)):

label\_inf\_pick\_less2["text"] = para22[i] + ", " + group22[i] + "\n" + label\_inf\_pick\_less2["text"]

cab()

elif cb\_para.get() == "3 пара":

para23 = para13

group23 = group13

para13 = some\_cab4

group13 = some\_group6

for i in range(len(para23)):

label\_inf\_pick\_less2["text"] = para23[i] + ", " + group23[i] + "\n" + label\_inf\_pick\_less2["text"]

cab()

elif cb\_para.get() == "4 пара":

para24 = para14

group24 = group14

para14 = some\_cab4

group14 = some\_group6

for i in range(len(para24)):

label\_inf\_pick\_less2["text"] = para24[i] + ", " + group24[i] + "\n" + label\_inf\_pick\_less2["text"]

cab()

elif cb\_para.get() == "5 пара":

para25 = para15

group25 = group15

para15 = some\_cab4

group15 = some\_group6

for i in range(len(para25)):

label\_inf\_pick\_less2["text"] = para25[i] + ", " + group25[i] + "\n" + label\_inf\_pick\_less2["text"]

cab()

elif cb\_para.get() == "6 пара":

para26 = para16

group26 = group16

para16 = some\_cab4

group16 = some\_group6

for i in range(len(para26)):

label\_inf\_pick\_less2["text"] = para26[i] + ", " + group26[i] + "\n" + label\_inf\_pick\_less2["text"]

cab()

#функция построения окна переноса пары

def choice\_cab\_switch(some\_cab3, some\_group5):

global label\_inf\_pick\_less1, label\_inf\_pick\_less2, some\_group6, some\_cab4

label\_inf\_pick\_less1["text"] = " "

label\_inf\_pick\_less2["text"] = " "

for x in range(len(some\_cab3)):

label\_inf\_pick\_less1["text"] = some\_cab3[x] + ", " + some\_group5[x] + "\n" + label\_inf\_pick\_less1["text"]

bttn\_accept3 = ttk.Button(root4, text = "Поменять")

bttn\_accept3.place(x = 350/2 - 60, y = 216)

bttn\_accept3.bind("<Button-1>", switchtc)

some\_group6 = some\_group5

some\_cab4 = some\_cab3

#функция для распределения выбранной пары с текущей

def choice\_switch\_del(event):

if rbweek4.get() == 1:

print("1 неделя")

if cb\_para.get() == "1 пара":

print("1 пара")

para11copy = para11

choice\_cab\_switch(para11copy, group11)

elif cb\_para.get() == "2 пара":

print("2 пара")

para12copy = para12

choice\_cab\_switch(para12copy, group12)

elif cb\_para.get() == "3 пара":

print("3 пара")

para13copy = para13

choice\_cab\_switch(para13copy, group13)

elif cb\_para.get() == "4 пара":

print("4 пара")

para14copy = para14

choice\_cab\_switch(para14copy, group14)

elif cb\_para.get() == "5 пара":

print("5 пара")

para15copy = para15

choice\_cab\_switch(para15copy, group15)

elif cb\_para.get() == "6 пара":

print("6 пара")

para16copy = para16

choice\_cab\_switch(para16copy, group16)

elif rbweek4.get() == 2:

print("2 неделя")

if cb\_para.get() == "1 пара":

print("1 пара")

para21copy = para21

choice\_cab\_switch(para21copy, group21)

elif cb\_para.get() == "2 пара":

print("2 пара")

para22copy = para22

choice\_cab\_switch(para22copy, group22)

elif cb\_para.get() == "3 пара":

print("3 пара")

para23copy = para23

choice\_cab\_switch(para23copy, group23)

elif cb\_para.get() == "4 пара":

print("4 пара")

para24copy = para24

choice\_cab\_switch(para24copy, group24)

elif cb\_para.get() == "5 пара":

print("5 пара")

para25copy = para25

choice\_cab\_switch(para25copy, group25)

elif cb\_para.get() == "6 пара":

print("6 пара")

para26copy = para26

choice\_cab\_switch(para26copy, group26)

def root4\_inf(event):

mb.showinfo("Справочная информация", "Вы вошли в режим переноса пар!\nДля того чтобы перенести запись вам нужно:\n 1) Выбрать с какой пары на какую неодходимо перенести пару(двойным щелчком левой кнопки мыши)\n 2) Выбрать пару в выпадающем меню\nПосле чего, нажимаем на кнопку \"Поменять\".")

#функция построения окна переноса пары

def switch\_cab():

global cb\_para, rbweek4, label\_inf\_pick\_less1, label\_inf\_pick\_less2, root4

rbweek4 = IntVar()

root4 = Toplevel()

root4.title("Перенести кабинет")

root4.resizable(False, False)

oldframe = ttk.LabelFrame(root4, text = "Старый вариант", width = 150, height = 175)

newframe = ttk.LabelFrame(root4, text = "Новый вариант", width = 150, height = 175)

oldframe. place(x = 10, y = 30)

newframe.place(x = 170, y = 30)

w = root4.winfo\_screenwidth()

h = root4.winfo\_screenheight()

w = w//2

h = h//2

w = w - 465

h = h - 52

root4.geometry('330x355+{}+{}'.format(w, h))

label\_inf\_pick\_less1 = ttk.Label(oldframe)

label\_inf\_pick\_less1.place(x = 5, y = 5)

label\_inf\_pick\_less2 = ttk.Label(newframe)

label\_inf\_pick\_less2.place(x = 5, y = 5)

rb\_for\_week1 = ttk.Radiobutton(root4, text = "С 1 недели на 2", value = 1, variable = rbweek4)

rb\_for\_week1.place(x = 265/2 - 90, y = 10)

rb\_for\_week2 = ttk.Radiobutton(root4, text = "С 2 недели на 1", value = 2, variable = rbweek4)

rb\_for\_week2.place(x = 265/2 + 60, y = 10)

cb\_para = ttk.Combobox(root4, values = ["1 пара", "2 пара", "3 пара", "4 пара", "5 пара", "6 пара"], width = 18)

cb\_para.set("Выберите пару")

cb\_para.place(x = 330/2 - 55, y = 300)

cb\_para.bind("<<ComboboxSelected>>", choice\_switch\_del)

btttn\_inf = ttk.Button(root4, text = "Справочная информация")

btttn\_inf.place(x = 75, y = 323)

btttn\_inf.bind("<Button-1>", root4\_inf)

def group(event):

global label\_for\_group

label\_for\_group["text"] = ""

print(combobox\_for\_group.get())

for i in range(len(group11)):

if combobox\_for\_group.get() == group11[i]:

index\_for\_group11 = group11.index(combobox\_for\_group.get())

para\_for\_group11 = para11[index\_for\_group11]

print(para\_for\_group11)

label\_for\_group["text"] = para\_for\_group11 + ", 1 пара" + ", 1 неделя" + "\n" + label\_for\_group["text"]

for i in range(len(group12)):

if combobox\_for\_group.get() == group12[i]:

index\_for\_group12 = group12.index(combobox\_for\_group.get())

para\_for\_group12 = para12[index\_for\_group12]

print(para\_for\_group12)

label\_for\_group["text"] = para\_for\_group12 + ", 2 пара" + ", 1 неделя" + "\n" + label\_for\_group["text"]

for i in range(len(group13)):

if combobox\_for\_group.get() == group13[i]:

index\_for\_group13 = group13.index(combobox\_for\_group.get())

para\_for\_group13 = para13[index\_for\_group13]

print(para\_for\_group13)

label\_for\_group["text"] = para\_for\_group13 + ", 3 пара" + ", 1 неделя" + "\n" + label\_for\_group["text"]

for i in range(len(group14)):

if combobox\_for\_group.get() == group14[i]:

index\_for\_group14 = group14.index(combobox\_for\_group.get())

para\_for\_group14 = para14[index\_for\_group14]

print(para\_for\_group14)

label\_for\_group["text"] = para\_for\_group14 + ", 4 пара" + ", 1 неделя" + "\n" + label\_for\_group["text"]

for i in range(len(group15)):

if combobox\_for\_group.get() == group15[i]:

index\_for\_group15 = group15.index(combobox\_for\_group.get())

para\_for\_group15 = para15[index\_for\_group15]

print(para\_for\_group15)

label\_for\_group["text"] = para\_for\_group15 + ", 5 пара" + ", 1 неделя" + "\n" + label\_for\_group["text"]

for i in range(len(group16)):

if combobox\_for\_group.get() == group16[i]:

index\_for\_group16 = group16.index(combobox\_for\_group.get())

para\_for\_group16 = para16[index\_for\_group16]

print(para\_for\_group16)

label\_for\_group["text"] = para\_for\_group16 + ", 6 пара" + ", 1 неделя" + "\n" + label\_for\_group["text"]

for i in range(len(group21)):

if combobox\_for\_group.get() == group21[i]:

index\_for\_group21 = group21.index(combobox\_for\_group.get())

para\_for\_group21 = para21[index\_for\_group21]

print(para\_for\_group21)

label\_for\_group["text"] = para\_for\_group21 + ", 1 пара" + ", 2 неделя" + "\n" + label\_for\_group["text"]

for i in range(len(group22)):

if combobox\_for\_group.get() == group22[i]:

index\_for\_group22 = group22.index(combobox\_for\_group.get())

para\_for\_group22 = para22[index\_for\_group22]

print(para\_for\_group22)

label\_for\_group["text"] = para\_for\_group22 + ", 2 пара" + ", 2 неделя" + "\n" + label\_for\_group["text"]

for i in range(len(group23)):

if combobox\_for\_group.get() == group23[i]:

index\_for\_group23 = group23.index(combobox\_for\_group.get())

para\_for\_group23 = para23[index\_for\_group23]

print(para\_for\_group23)

label\_for\_group["text"] = para\_for\_group23 + ", 3 пара" + ", 2 неделя" + "\n" + label\_for\_group["text"]

for i in range(len(group24)):

if combobox\_for\_group.get() == group24[i]:

index\_for\_group24 = group24.index(combobox\_for\_group.get())

para\_for\_group24 = para24[index\_for\_group24]

print(para\_for\_group24)

label\_for\_group["text"] = para\_for\_group24 + ", 4 пара" + ", 2 неделя" + "\n" + label\_for\_group["text"]

for i in range(len(group25)):

if combobox\_for\_group.get() == group25[i]:

index\_for\_group25 = group25.index(combobox\_for\_group.get())

para\_for\_group25 = para25[index\_for\_group25]

print(para\_for\_group25)

label\_for\_group["text"] = para\_for\_group25 + ", 5 пара" + ", 2 неделя" + "\n" + label\_for\_group["text"]

for i in range(len(group26)):

if combobox\_for\_group.get() == group26[i]:

index\_for\_group26 = group26.index(combobox\_for\_group.get())

para\_for\_group26 = para26[index\_for\_group26]

print(para\_for\_group26)

label\_for\_group["text"] = para\_for\_group26 + ", 6 пара" + ", 2 неделя" + "\n" + label\_for\_group["text"]

#функция для справочной информации

def root\_inf3():

mb.showinfo("Справочная информация", "Вы вошли в качестве администратора!\nВы можете просматривать и редактировать расписание на кафедре.\nПриятного использования!")

#функция окна администратора

def window\_admin():

global plan, combobox\_for\_week, frame1, canv, frame2, mainmenu, check, label\_for\_group, combobox\_for\_group, menu

frame = Frame(root, height = 308, width = 472)

frame.place(x = 0, y = 0)

frame1 = LabelFrame(root, height = 100, width = 235, text = 'Выберите пару')

frame1.place(x = 480, y = 0)

frame2 = LabelFrame(root, height = 200, width = 200, text = 'Информация')

frame2.place(x = 480, y = 105)

frame3 = LabelFrame(root, height = 200, width = 210, text = 'Группы')

frame3.place(x = 690, y = 105)

label\_for\_group = ttk.Label(frame3)

label\_for\_group.place(x = 5, y = 30)

check = 2

root.title('Расписание пар(Администратор)')

w = root.winfo\_screenwidth()

h = root.winfo\_screenheight()

w = w//2

h = h//2

w = w - 435

h = h - 325

root.geometry('910x355+{}+{}'.format(w, h))

canv = Canvas(frame, width = 464, height = 300, bg = "white")

canv.place(x = 0, y = 0)

plan = PhotoImage(file = file\_plan)

image = canv.create\_image(233, 145, image = plan)

mainmenu = Menu(root)

root.config(menu = mainmenu)

mainmenu.add\_command(label = "Добавить", command = add\_cab)

mainmenu.add\_command(label = "Удалить", command = del\_cab)

mainmenu.add\_command(label = "Изменить", command = change\_cab)

mainmenu.add\_command(label = "Перенести пару", command = switch\_cab)

menu = Menu(tearoff = 0)

menu.add\_command(label = "Показать всё расписание", command = trace)

combobox\_for\_week = ttk.Combobox(root, values = ["1 неделя", "2 неделя"])

combobox\_for\_week.place(x = 720, y = 10)

combobox\_for\_week.set("Выберите неделю")

combobox\_for\_week.bind("<<ComboboxSelected>>", week)

combobox\_for\_group = ttk.Combobox(frame3, values = ['20ВТ1', '20ВТ2', '19ВТ1', '19ВТ2', '18ВТ', '17ВТ', '20ИЭ1',

'20ИЭ2', '19ИЭ1', '19ИЭ2', '18ИЭ', '17ИЭ', '20АП1', '20АП2'],

width = 17)

combobox\_for\_group.place(x = 5, y = 5)

combobox\_for\_group.set("Выберите группу")

combobox\_for\_group.bind("<<ComboboxSelected>>", group)

canv.create\_text(97, 188, text = 'Кабинет\n №1', font = 'Times 7', tag = 'cab1text')

canv.create\_text(28, 188, text = 'Кабинет\n №2', font = 'Times 7', tag = 'cab2text')

canv.create\_text(28, 228, text = 'Кабинет\n №3', font = 'Times 7', tag = 'cab3text')

canv.create\_text(35, 265, text = 'Кабинет\n №4', font = 'Times 7', tag = 'cab4text')

canv.create\_text(82, 265, text = 'Кабинет\n №5', font = 'Times 7', tag = 'cab5text')

canv.create\_text(128, 265, text = 'Кабинет\n №6', font = 'Times 7', tag = 'cab6text')

canv.create\_text(178, 265, text = 'Кабинет\n №7', font = 'Times 7', tag = 'cab7text')

canv.create\_text(240, 265, text = 'Кабинет\n №8', font = 'Times 7', tag = 'cab8text')

canv.create\_text(326, 265, text = 'Кабинет\n №9', font = 'Times 7', tag = 'cab9text')

canv.create\_text(356, 265, text = 'Кабинет\n №10', font = 'Times 7', tag = 'cab10text')

canv.create\_text(394, 265, text = 'Кабинет\n №11', font = 'Times 7', tag = 'cab11text')

canv.create\_text(433, 265, text = 'Кабинет\n №12', font = 'Times 7', tag = 'cab12text')

canv.create\_text(368, 188, text = 'Кабинет\n №13', font = 'Times 7', tag = 'cab13text')

canv.create\_text(282, 188, text = 'Кабинет\n №14', font = 'Times 7', tag = 'cab14text')

bttn\_back1 = ttk.Button(root, text = "Назад", command = main).place(x = 760, y = 305)

bttn\_close1 = ttk.Button(root, text = "Выйти", command = root.destroy).place(x = 640, y = 305)

bttn\_inf2 = ttk.Button(root, text = "Справочная информация", command = root\_inf3).place(x = 435, y = 305)

canv.bind("<Button-3>", popup)

#функция окна ввода пароля

def admin\_verify():

global entry1, entry2, frame\_for\_admin

root.title("Введите логин и пароль")

frame\_for\_admin = Frame(root, height = 100, width = 250).place(x = 0, y = 0)

w = root.winfo\_screenwidth()

h = root.winfo\_screenheight()

w = w//2

h = h//2

w = w - 125

h = h - 50

root.geometry('250x100+{}+{}'.format(w, h))

entry1 = ttk.Entry(frame\_for\_admin, width = 25, font = "Times, 12")

entry1.place(x = 10, y = 2)

entry1.insert(0, "Введите пароль")

entry1.bind("<Button-1>", mask1)

entry2 = ttk.Entry(frame\_for\_admin, width = 25, font = "Times, 12")

entry2.place(x = 10, y = 30)

entry2.insert(0, "Введите пароль еще раз")

entry2.bind("<Button-1>", mask2)

bttn\_accept = ttk.Button(frame\_for\_admin, text = "Ок", command = verify).place(x = 5, y = 65)

bttn\_back = ttk.Button(frame\_for\_admin, text = "Назад", command = main).place(x = 135, y = 65)

#функция для справочной информации

def root\_inf2():

mb.showinfo("Справочная информация", "Вы вошли в качестве пользователя!\nВы можете просмотреть:\n 1)Полное расписание на кафедре.\n 2)Расписание для конкретного кабинета(нажатием правой клавиши мыши по кабинету).\n 3)Расписание для конкретной группы(в окне \"Группы\").\nПриятного использования!")

#функция окна пользователя

def window\_user():

global plan, combobox\_for\_week, frame1, canv, frame2, check, menu, label\_for\_group, combobox\_for\_group

frame = Frame(root, height = 308, width = 472)

frame.place(x = 0, y = 0)

frame1 = LabelFrame(root, height = 100, width = 235, text = 'Выберите пару')

frame1.place(x = 480, y = 0)

frame2 = LabelFrame(root, height = 200, width = 200, text = 'Информация')

frame2.place(x = 480, y = 105)

frame3 = LabelFrame(root, height = 200, width = 210, text = 'Группы')

frame3.place(x = 690, y = 105)

label\_for\_group = ttk.Label(frame3)

label\_for\_group.place(x = 5, y = 30)

check = 1

root.title('Расписание пар')

w = root.winfo\_screenwidth()

h = root.winfo\_screenheight()

w = w//2

h = h//2

w = w - 435

h = h - 325

root.geometry('910x335+{}+{}'.format(w, h))

canv = Canvas(frame, width = 464, height = 300, bg = "white")

canv.place(x = 0, y = 0)

plan = PhotoImage(file = file\_plan)

image = canv.create\_image(233, 145, image = plan)

combobox\_for\_week = ttk.Combobox(root, values = ["1 неделя", "2 неделя"])

combobox\_for\_week.place(x = 720, y = 10)

combobox\_for\_week.set("Выберите неделю")

combobox\_for\_week.bind("<<ComboboxSelected>>", week)

menu = Menu(tearoff = 0)

menu.add\_command(label = "Показать всё расписание", command = trace)

combobox\_for\_group = ttk.Combobox(frame3, values = ['20ВТ1', '20ВТ2', '19ВТ1', '19ВТ2', '18ВТ', '17ВТ', '20ИЭ1',

'20ИЭ2', '19ИЭ1', '19ИЭ2', '18ИЭ', '17ИЭ', '20АП1', '20АП2'],

width = 17)

combobox\_for\_group.place(x = 5, y = 5)

combobox\_for\_group.set("Выберите группу")

combobox\_for\_group.bind("<<ComboboxSelected>>", group)

canv.create\_text(97, 188, text = 'Кабинет\n №1', font = 'Times 7', tag = 'cab1text')

canv.create\_text(28, 188, text = 'Кабинет\n №2', font = 'Times 7', tag = 'cab2text')

canv.create\_text(28, 228, text = 'Кабинет\n №3', font = 'Times 7', tag = 'cab3text')

canv.create\_text(35, 265, text = 'Кабинет\n №4', font = 'Times 7', tag = 'cab4text')

canv.create\_text(82, 265, text = 'Кабинет\n №5', font = 'Times 7', tag = 'cab5text')

canv.create\_text(128, 265, text = 'Кабинет\n №6', font = 'Times 7', tag = 'cab6text')

canv.create\_text(178, 265, text = 'Кабинет\n №7', font = 'Times 7', tag = 'cab7text')

canv.create\_text(240, 265, text = 'Кабинет\n №8', font = 'Times 7', tag = 'cab8text')

canv.create\_text(326, 265, text = 'Кабинет\n №9', font = 'Times 7', tag = 'cab9text')

canv.create\_text(356, 265, text = 'Кабинет\n №10', font = 'Times 7', tag = 'cab10text')

canv.create\_text(394, 265, text = 'Кабинет\n №11', font = 'Times 7', tag = 'cab11text')

canv.create\_text(433, 265, text = 'Кабинет\n №12', font = 'Times 7', tag = 'cab12text')

canv.create\_text(368, 188, text = 'Кабинет\n №13', font = 'Times 7', tag = 'cab13text')

canv.create\_text(282, 188, text = 'Кабинет\n №14', font = 'Times 7', tag = 'cab14text')

bttn\_back1 = ttk.Button(root, text = "Назад", command = main).place(x = 760, y = 305)

bttn\_close1 = ttk.Button(root, text = "Выйти", command = root.destroy).place(x = 640, y = 305)

bttn\_inf1 = ttk.Button(root, text = "Справочная информация", command = root\_inf2).place(x = 435, y = 305)

canv.bind("<Button-3>", popup)

#функция для информации об авторе

def root\_inf():

mb.showinfo("Информация об авторе", "ФИО: Подковыров Даниил\nКем является: студент\nМесто: Калининградская область\n")

#функция для справочной информации

def root\_inf1():

mb.showinfo("Справочная информация", "Добро пожаловать в ПС \"Кафедра\"!\nВы можете выбрать интересуемый вас режим входа:\n 1. Режим пользователя(Просмотр информации)\n 2.Режим администратора через пароль(Просмотр и редактирование информации)")

#функция режима входа в систему

def main():

global root, mainmenu, menu

if check == 2:

mainmenu.delete("Добавить")

mainmenu.delete("Изменить")

mainmenu.delete("Удалить")

mainmenu.delete("Перенести пару")

menu.delete("Показать всё расписание")

root.title("Выбор пользователя")

root.resizable(False, False)

w = root.winfo\_screenwidth()

h = root.winfo\_screenheight()

w = w//2

h = h//2

w = w - 125

h = h - 50

root.geometry('250x150+{}+{}'.format(w, h))

frame21 = Frame(root, width = 250, height = 150)

frame21.place(x = 0, y = 0)

label\_choice = ttk.Label(frame21, text = 'Выберите пользователя', font = 'Times 14').place(x = 30, y = 10)

bttn\_admin = ttk.Button(frame21, text = 'Администратор', command = admin\_verify).place(x = 0, y = 50)

bttn\_user = ttk.Button(frame21, text = 'Пользователь', command = window\_user).place(x = 135, y = 50)

bttn\_inf\_avtor = ttk.Button(frame21, text = "Об авторе", command = root\_inf).place(x = 75, y = 110)

bttn\_inf = ttk.Button(frame21, text = "Справочная информация", command = root\_inf1).place(x = 25, y = 80)

main()

#функция изменения пароля через каждын 10 дней

def root\_change\_password():

global root

root1 = Tk()

root1.geometry('310x220')

root1.title("Cмена пароля")

print(123)

style = ttk.Style()

style.configure('TButton', font = "Times, 12")

style.configure('TLabel', font = "Times, 10")

style.configure('TEntry', font = "Times, 10")

style.map('TButton', background = [("active", "pink")])

main\_label = ttk.Label(root1, text = "Необходимо поменять пароль")

main\_label.place(x = 155, y = 30, anchor = CENTER)

password\_label = ttk.Label(root1, text = "Введите новый пароль:", justify = LEFT)

password\_entry = ttk.Entry(root1, width = 12, show = "•")

password\_label\_help = ttk.Label(root1, text = "Минимум 6 символов")

password\_label\_help.place(x = 220, y = 105 , anchor =CENTER)

password\_label.place(x = 80, y = 80, anchor = CENTER)

password\_entry.place(x = 225, y = 80, anchor = CENTER)

password\_label1 = ttk.Label(root1, text = "Повторите ещё раз:", justify = LEFT)

password\_entry1 = ttk.Entry(root1, width = 12, show = "•")

password\_label1.place(x = 80, y = 135, anchor = CENTER)

password\_entry1.place(x = 225, y = 135, anchor = CENTER)

def change\_password():

global password

if (password\_entry.get() == password\_entry1.get()):

if (len(password\_entry.get()) >= 6 and password\_entry.get() != password):

password = password\_entry.get()

password\_day = datetime.date.today()

f = open('secret.txt', 'r')

lines = f.readlines()

lines[0] = password + '\n'

lines[1] = str(password\_day) + '\n'

f.close()

save\_changes = open('secret.txt', 'w')

save\_changes.writelines(lines)

save\_changes.close()

main()

elif(len(password\_entry.get()) < 6):

mb.showerror("Ошибка!", "Пароль должен состоять минимум из 6 символов")

elif(password\_entry.get() == password):

mb.showerror("Ошибка!", "Новый пароль должен отличаться от предыдущего")

else:

mb.showerror("Ошибка!", "Пароли не совпадают")

button = ttk.Button(root1, text = "Готово")

button.place(x = 155, y = 185, anchor = CENTER)

f = open('secret.txt', 'r')

lines = f.readlines()

password = lines[0].rstrip()

password\_day = datetime.datetime.strptime(lines[1].rstrip(), "%Y-%m-%d").date()

f.close()

today = datetime.date.today()

difference\_password\_day = 1 - int((today - password\_day).days)

if (difference\_password\_day < 1):

pass

else:

main()

root.mainloop()

# Руководство системного программиста

Для работы данной программы и возможности вносить свои корректировки необходимо скачать компилятор Python с официально сайта <https://www.python.org>. В свою очередь, я предлагаю для вашего удобства скачать также программу для работы с Python Sublime Text 3.

Для полного понимания данное программы необходимо понимание языка SQL. В данном руководстве приводится некоторые методы языка SQLITE 3, который основывается на языке запросов SQL. В свою очередь необходимо подключить библиотеку языка SQLITE 3 через метод “import sqlite3”

Кроме того, для работы программы необходимы следующие библиотеки:

1. from tkinter import \* - для работы интерфейса программы
2. from tkinter import ttk - для работы интерфейса программы
3. from tkinter import messagebox as mb - для работы интерфейса программы
4. import datetime - для выполнения вычислений, связанных с датам

Первая строка файла secret.txt содержит пароль, вторая строка содержит последнюю дату изменения пароля типа 2011-11-11.

Файл kursovaya.db является базой данных, которая создается программно.

# Руководство пользователя

Работа программы может начинаться по-разному – либо с режима входа, либо с режима изменения пароля.

Если число дней после последней смены пароля больше либо равно 10, то вам будет предложено изменить пароль или покинуть программу. При изменении пароля вам будет предложено ввести пароль для входа в систему.

Если число дней меньше 10, то вам будет предложено ввести пароль для входа в систему. Также на данной странице можно получить информацию об авторе программной системы и справочную записку о работе программы.

После успешного входа в качестве пользователя вы сможете просмотреть расписание на кафедре для 1 и 2 недели. Кнопка назад вернет вас на окно режима входа.

После успешного входа в качестве администратора у вас будут более обширные возможности по отношению к режиму пользователя. Вы сможете редактировать расписание, удалять кабинеты и группы, добавлять новые кабинеты и группы и изменять уже существующие ,а также иметь полный спектр возможностей, что представлен в режиме пользователя. Кнопка назад вернет вас на окно режима входа.

# Текст справочной системы

Справка на окне входа:

Добро пожаловать в ПС "Кафедра"!

Вы можете выбрать интересуемый вас режим входа:

1. Режим пользователя (Просмотр расписания)

2.Режим администратора через пароль (Просмотр и редактирование расписания)

Информация об авторе на окне входа:

ФИО: Подковыров Даниил Русланович

Кем является: студент

Место: Калининградская область

Справка на окне режима пользователя:

Вы вошли в качестве пользователя!

Вы можете посмотреть:

1. Полное расписание на кафедре
2. Расписание для конкретного кабинета(нажатием правой клавиши мыши по кабинету)
3. Расписание для конкретной группы(в окне «Группы»)

Приятного использования!

Справка на окне режима администратора:

Вы вошли в качестве администратора!

Вы можете просматривать и редактировать расписание на кафедре.

Приятного использования!

Справка на окне переноса пар:

Вы вошли в режим переноса пар!

Для того чтобы перенести запись вам нужно:

1. Выбрать с какой пары на какую необходимо перенести пару(двойным щелчком левой клавиши мыши)
2. Выбрать пару в выпадающем меню

После чего, нажимаем на кнопку «Поменять».

Справка на окне удаления пар:

Вы вошли в режим удаление кабинета и группы!

Для того чтобы удалить запись вам нужно:

1. Выбрать неделю(двойным щелчком левой кнопки мыши)
2. Выбрать пару в выпадающем меню

После чего, указать кабинет, который хотите удалить и нажать кнопку «Удалить»

Справка на окне добавления пар:

Вы вошли в режим добавления кабинета и группы!

Для того чтобы удалить запись вам нужно:

1. Выбрать неделю(двойным щелчком левой кнопки мыши)
2. Выбрать пару в выпадающем меню

После чего, указать кабинет и группу, которую хотите добавить и нажать кнопку «Добавить»

Справка на окне изменения пар:

Вы вошли в режим изменения кабинета и группы!

Для того чтобы изменить запись вам нужно:

1. Выбрать неделю(двойным щелчком левой кнопки мыши)
2. Выбрать пару в выпадающем меню

После чего, указать какой кабинет меняем, на какой меняем и группу. Сохранить изменения нажатием кнопки «Подтвердить»

# Литература

1. <https://tproger.ru/translations/sql-recap/>
2. <https://python-scripts.com/tkinter>
3. <https://pythonworld.ru/tipy-dannyx-v-python/fajly-rabota-s-fajlami.html>
4. <https://python-scripts.com/datetime-time-python>
5. <https://it61.tk/python/Python-6.pdf>
6. <https://younglinux.info/sqlite/python>