Практическое занятие №16

Тема: Разработка многооконного приложения для работы с однотабличной БД в IDE PyCharm Community.

Цель: закрепить усвоенные знания, понятия, алгоритмы, основные принципы составления программ, работы с БД в IDE PyCharm Community.

Постановка задачи: Приложение **СДАЧА В АРЕНДУ ТОРГОВЫХ ПЛОЩАДЕЙ** для некоторой организации. БД должна содержать таблицу **Торговая точка** со следующей структурой записи: этаж, площадь, наличие кондиционера и стоимость аренды в день. БД должна обеспечивать получение информации по стоимости аренды.

Текст программы:

```
import tkinter as tk
from tkinter import ttk
import sqlite3 as sq


class Main(tk.Frame):

    """Класс для главного окна"""

    def __init__(self, root):
        super().__init__(root)
        self.init_main()
        self.db = db
        self.view_records()

    def init_main(self):
        toolbar = tk.Frame(bg='#a0dea0', bd=4)
        toolbar.pack(side=tk.TOP, fill=tk.X)

        self.add_img = tk.PhotoImage(file="../PZ_16/11.gif")
        self.btn_open_dialog = tk.Button(toolbar, text='Добавить точку',
command=self.open_dialog, bg='#5da130', bd=0,
                                         compound=tk.TOP, image=self.add_img)
        self.btn_open_dialog.pack(side=tk.LEFT)

        self.update_img = tk.PhotoImage(file="../PZ_16/12.gif")
        btn_edit_dialog = tk.Button(toolbar, text="Редактировать",
command=self.open_update_dialog, bg='#5da130',
                                    bd=0, compound=tk.TOP,
image=self.update_img)
        btn_edit_dialog.pack(side=tk.LEFT)

        self.delete_img = tk.PhotoImage(file="../PZ_16/13.gif")
        btn_delete = tk.Button(toolbar, text="Удалить точку",
command=self.delete_records, bg='#5da130',
                               bd=0, compound=tk.TOP,
image=self.delete_img)
        btn_delete.pack(side=tk.LEFT)

        self.search_img = tk.PhotoImage(file="../PZ_16/14.gif")
        btn_search = tk.Button(toolbar, text="Поиск точки",
command=self.open_search_dialog, bg='#5da130',
                               bd=0, compound=tk.TOP, image=self.search_img)
        btn_search.pack(side=tk.LEFT)

        self.refresh_img = tk.PhotoImage(file="../PZ_16/15.gif")
```

```
        btn_refresh = tk.Button(toolbar, text="Обновить экран",
command=self.view_records, bg='#5da130',
                                bd=0, compound=tk.TOP, image=self.refresh_img)
        btn_refresh.pack(side=tk.LEFT)

        self.tree = ttk.Treeview(self, columns=('t_id', 'name_t', 'name_m',
'report'), height=20, show='headings')

        self.tree.column('t_id', width=100, anchor=tk.CENTER)
        self.tree.column('name_t', width=180, anchor=tk.CENTER)
        self.tree.column('name_m', width=160, anchor=tk.CENTER)
        self.tree.column('report', width=160, anchor=tk.CENTER)

        self.tree.heading('t_id', text='Этаж')
        self.tree.heading('name_t', text='Площадь')
        self.tree.heading('name_m', text='Наличие кондиционера')
        self.tree.heading('report', text='Стоимость аренды в день')

        self.tree.pack()

    def records(self, t_id, name_t, name_m, report):
        self.db.insert_data(t_id, name_t, name_m, report)
        self.view_records()

    def update_record(self, t_id, name_t, name_m, report):
        self.db.cur.execute("""UPDATE base SET t_id=?, name_t=?, name_m=?,
report=? WHERE t_id=?""",
                            (t_id, name_t, name_m, report,
self.tree.set(self.tree.selection()[0], '#1')))
        self.db.con.commit()
        self.view_records()

    def view_records(self):
        self.db.cur.execute("""SELECT * FROM base""")
        [self.tree.delete(i) for i in self.tree.get_children()]
        [self.tree.insert('', 'end', values=row) for row in
self.db.cur.fetchall()]

    def delete_records(self):
        for selection_item in self.tree.selection():
            self.db.cur.execute("""DELETE FROM base WHERE t_id=?""",
(self.tree.set(selection_item, '#1'),))
        self.db.con.commit()
        self.view_records()

    def search_name_t(self, user_id):
        user_id = ("%" + user_id + "%",)
        self.db.cur.execute("""SELECT * FROM base WHERE report LIKE ?""",
user_id)
        [self.tree.delete(i) for i in self.tree.get_children()]
        [self.tree.insert('', 'end', values=row) for row in
self.db.cur.fetchall()]


    def open_dialog(self):
        Child(root, app)

    def open_update_dialog(self):
        Update()

    def open_search_dialog(self):
        Search()

class Child(tk.Toplevel):
```

```
    """Класс для дочернего окна"""

    def __init__(self, root, app):
        super().__init__(root)
        self.init_child()
        self.view = app

    def init_child(self):
        self.title('Добавить точку')
        self.geometry('600x170+600+300')
        self.resizable(False, False)

        label_description = tk.Label(self, text='Этаж')
        label_description.place(x=50, y=25)
        self.entry_des = ttk.Entry(self)
        self.entry_des.place(x=220, y=25)

        label_name = tk.Label(self, text='Площадь')
        label_name.place(x=50, y=50)
        self.entry_name = ttk.Entry(self)
        self.entry_name.place(x=220, y=50)

        label_name1 = tk.Label(self, text='Наличие кондиционера')
        label_name1.place(x=50, y=75)
        self.entry1_name = ttk.Combobox(self, values=[u'Имеется',
u'Отсутствует'])
        self.entry1_name.current(0)
        self.entry1_name.place(x=220, y=75)

        label_old = tk.Label(self, text='Стоимость аренды в день')
        label_old.place(x=50, y=100)
        self.entry_z = ttk.Entry(self)
        self.entry_z.place(x=220, y=100)

        btn_cancel = ttk.Button(self, text='Закрыть', command=self.destroy)
        btn_cancel.place(x=300, y=140)

        self.btn_ok = ttk.Button(self, text='Добавить')
        self.btn_ok.place(x=220, y=140)
        self.btn_ok.bind('<Button-1>', lambda event:
self.view.records(self.entry_des.get(),

self.entry_name.get(),

self.entry1_name.get(),

self.entry_z.get()))

        self.grab_set()
        self.focus_set()

class Update(Child):
    def __init__(self):
        super().__init__(root, app)
        self.init_edit()
        self.view = app

    def init_edit(self):
        self.title("Редактировать запись")
        btn_edit = ttk.Button(self, text="Редактировать")
        btn_edit.place(x=205, y=140)
        btn_edit.bind('<Button-1>', lambda event:
self.view.update_record(self.entry_des.get(),
```

```
self.entry_name.get(),

self.entry1_name.get(),

self.entry_z.get()))
        self.btn_ok.destroy()

class Search(tk.Toplevel):
    def __init__(self):
        super().__init__()
        self.init_search()
        self.view = app

    def init_search(self):
        self.title("Поиск")
        self.geometry("400x100+400+300")
        self.resizable(False, False)

        label_search = tk.Label(self, text="Поиск по стоимости аренды")
        label_search.place(x=50, y=20)

        self.entry_search = ttk.Entry(self)
        self.entry_search.place(x=225, y=20, width=150)

        btn_cancel = ttk.Button(self, text="Закрыть", command=self.destroy)
        btn_cancel.place(x=185, y=50)

        btn_search = ttk.Button(self, text="Поиск")
        btn_search.place(x=105, y=50)
        btn_search.bind('<Button-1>', lambda event:
self.view.search_name_t(self.entry_search.get()))
        btn_search.bind('<Button-1>', lambda event: self.destroy(), add='+')

class DB:
    def __init__(self):

        with sq.connect('../PZ_16/ob.db') as self.con:
            self.cur = self.con.cursor()
            self.cur.execute("""CREATE TABLE IF NOT EXISTS base (
                t_id INTEGER PRIMARY KEY AUTOINCREMENT,
                name_t TEXT NOT NULL,
                name_m INTEGER NOT NULL DEFAULT 1,
                report TEXT NOT NULL
                )""")

    def insert_data(self, t_id, name_t, name_m, report):
        self.cur.execute("""INSERT INTO base(t_id, name_t, name_m, report)
VALUES (?, ?, ?, ?)""",
                         (t_id, name_t, name_m, report))
        self.con.commit()

if __name__ == "__main__":
    root = tk.Tk()
    db = DB()
    app = Main(root)
    app.pack()
    root.title("Сдача в аренду торговых площадей")
    root.geometry("600x450+300+200")
    root.resizable(False, False)
    root.mainloop()
```

Протокол работы программы:

Process finished with exit code 0

Вывод: закрепила усвоенные знания, понятия, алгоритмы, основные принципы составления программ, работы с БД в IDE PyCharm Community.