**Практическая работа №5. Создание оконного приложения на основе класса базы данных.**

Выполнил студент группы исп 21.1А Жигач Дмитрий

Задание.

1

Создать проект в среде PyCharm «Приложение для ведения бюджета» по примеру

по ссылке:

https://thecode.media/sqlite-project/

2

На основе примера создать оконное приложение по своей предметной области.

Использовать класс, описанный в ПР№4 (из консольного приложения).

1. Модуль database.py

import sqlite3

class ProductDB:

    def \_\_init\_\_(self):

        self.connection = sqlite3.connect("products.db")

        self.cursor = self.connection.cursor()

        self.cursor.execute(

            """CREATE TABLE IF NOT EXISTS products

            (ID INTEGER PRIMARY KEY,

            company\_name TEXT,

            product\_name TEXT,

            unit\_of\_measurement TEXT,

            purchase\_price INTEGER,

            delivery\_date DATE,

            volume INTEGER,

            cost INTEGER)"""

        )

        self.connection.commit()

    def \_\_del\_\_(self):

        self.connection.close()

    def view(self):

        self.cursor.execute("SELECT \* FROM products")

        rows = self.cursor.fetchall()

        return rows

    def insert(self, company\_name, product\_name, unit\_of\_measurement, purchase\_price, delivery\_date, volume, cost):

        self.cursor.execute("INSERT INTO products VALUES (NULL, ?, ?, ?, ?, ?, ?, ?)",

                            (company\_name, product\_name, unit\_of\_measurement, purchase\_price, delivery\_date, volume, cost))

        self.connection.commit()

    def update(self, id, company\_name, product\_name, unit\_of\_measurement, purchase\_price, delivery\_date, volume, cost):

        self.cursor.execute("UPDATE products SET company\_name = ?, product\_name = ?, unit\_of\_measurement = ?, purchase\_price = ?, delivery\_date = ?, volume = ?, cost = ? WHERE ID = ?",

                            (company\_name, product\_name, unit\_of\_measurement, purchase\_price, delivery\_date, volume, cost, id))

        self.connection.commit()

    def delete(self, id):

        self.cursor.execute("DELETE FROM products WHERE ID = ?", (id,))

        self.connection.commit()

    def search(self, company\_name):

        self.cursor.execute("SELECT product\_name, volume, cost FROM products WHERE company\_name=?", (company\_name,))

        rows = self.cursor.fetchall()

        return rows

1. Модуль main.py

from tkinter import \*

from tkinter import ttk, messagebox

from database import ProductDB

global selected\_tuple

def get\_selected\_row(event):

    global selected\_tuple

    index = list1.curselection()[0]

    selected\_tuple = list1.get(index)

    company\_name\_entry.delete(0, END)

    company\_name\_entry.insert(END, selected\_tuple[1])

    product\_name\_entry.delete(0, END)

    product\_name\_entry.insert(END, selected\_tuple[2])

    unit\_of\_measurement\_entry.delete(0, END)

    unit\_of\_measurement\_entry.insert(END, selected\_tuple[3])

    purchase\_price\_entry.delete(0, END)

    purchase\_price\_entry.insert(END, selected\_tuple[4])

    delivery\_date\_entry.delete(0, END)

    delivery\_date\_entry.insert(END, selected\_tuple[5])

    volume\_entry.delete(0, END)

    volume\_entry.insert(END, selected\_tuple[6])

    cost\_entry.delete(0, END)

    cost\_entry.insert(END, selected\_tuple[7])

def view\_command():

    list1.delete(0, END)

    for row in database\_products.view():

        list1.insert(END, row)

def search\_command():

    list1.delete(0, END)

    for row in database\_products.search(company\_name\_text.get()):

        list1.insert(END, row)

def add\_command():

    database\_products.insert(

        company\_name\_text.get(),

        product\_name\_text.get(),

        unit\_of\_measurement\_text.get(),

        purchase\_price\_text.get(),

        delivery\_date\_text.get(),

        volume\_text.get(),

        cost\_text.get(),

    )

    view\_command()

def delete\_command():

    database\_products.delete(selected\_tuple[0])

    view\_command()

def update\_command():

    database\_products.update(

        selected\_tuple[0],

        company\_name\_text.get(),

        product\_name\_text.get(),

        unit\_of\_measurement\_text.get(),

        purchase\_price\_text.get(),

        delivery\_date\_text.get(),

        volume\_text.get(),

        cost\_text.get(),

    )

    view\_command()

def on\_closing():

    if messagebox.askokcancel("", "Закрыть программу?"):

        window.destroy()

window = Tk()

window.title("Product DB")

window.protocol("WM\_DELETE\_WINDOW", on\_closing)

fields = {

    "company\_name": "Company Name",

    "product\_name": "Product Name",

    "unit\_of\_measurement": "Unit of Measurement",

    "purchase\_price": "Purchase Price",

    "delivery\_date": "Delivery Date",

    "volume": "Volume",

    "cost": "Cost",

}

for index, (field\_name, field\_label) in enumerate(fields.items()):

    frame = ttk.Frame(borderwidth=1, relief=SOLID, padding=5)

    label = Label(frame, text=field\_label)

    label.pack()

    field\_text = StringVar()

    field\_entry = ttk.Entry(frame, textvariable=field\_text)

    field\_entry.pack()

    frame.grid(row=0, column=index)

    globals()[f"{field\_name}\_text"] = field\_text

    globals()[f"{field\_name}\_entry"] = field\_entry

frame = ttk.Frame(borderwidth=1, relief=SOLID, padding=5)

list1 = Listbox(frame, height=25, width=65)

list1.pack(side=LEFT, fill=BOTH, expand=1)

sb1 = Scrollbar(frame)

sb1.pack(side=RIGHT, fill=Y)

list1.configure(yscrollcommand=sb1.set)

sb1.configure(command=list1.yview)

frame.grid(row=1, column=0, columnspan=2)

list1.bind('<<ListboxSelect>>', get\_selected\_row)

frame = ttk.Frame(borderwidth=1, relief=SOLID, padding=5)

b1 = Button(frame, text="Посмотреть все", width=12, command=view\_command)

b1.pack()

b2 = Button(frame, text="Поиск", width=12, command=search\_command)

b2.pack()

b3 = Button(frame, text="Добавить", width=12, command=add\_command)

b3.pack()

b4 = Button(frame, text="Обновить", width=12, command=update\_command)

b4.pack()

b5 = Button(frame, text="Удалить", width=12, command=delete\_command)

b5.pack()

b6 = Button(frame, text="Закрыть", width=12, command=on\_closing)

b6.pack()

frame.grid(row=1, column=2)

database\_products = ProductDB()

window.mainloop()

1. Результат

