**Практическая работа №4. Создание класса базы данных. Реализация основных операций с данными (CRUD) в консольном приложении.**

Выполнил студент группы ИСП21.1А Жигач Дмитрий

Предметная область: сельское хозяйство

Задание.

1. Создать проект PyCharm.
2. Создать в нем 5 модулей .py
3. В первом модуле описать класс, соответствующий предметной области по варианту. Класс должен содержать:

- конструктор с подключением к базе данных, запросы на создание таблиц (CREATE TABLE);

- методы отключения базы данных, просмотра данных из таблиц, добавления данных, удаления данных, редактирования данных, поиска данных по критерию.

1. Остальные модули содержат методы и программный код, демонстрирующие работу с данными из базы данных.
2. Оформить отчет о выполненной работе:

- заголовок,

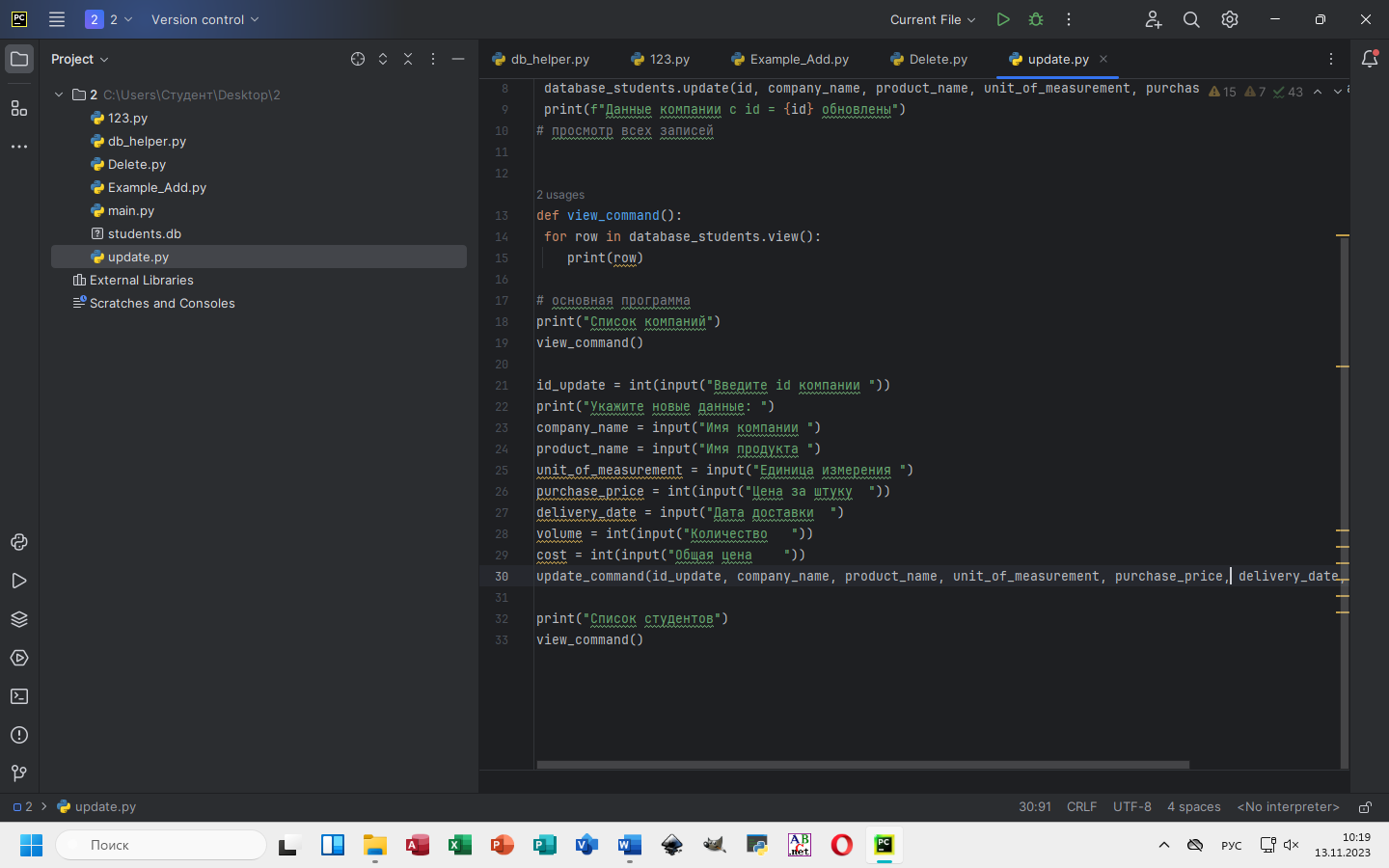
- кто выполнил,

- описание структуры проекта,

- программный код модулей,

- результаты выполнения модулей (скрины).

1. Создал проект в пайчарме и создал в нём 5 модулей



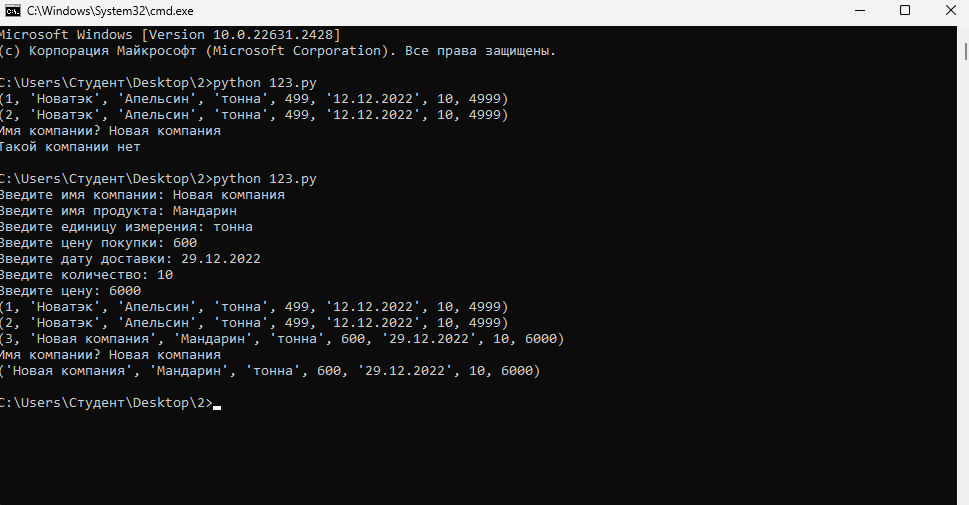
2. Перый модуль db\_helper.py

import sqlite3  
# создание класса БД  
  
class Student:  
 # конструктор класса  
 def \_\_init\_\_(self):  
 self.con = sqlite3.connect("students.db")  
 self.cur = self.con.cursor()  
 self.cur.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.con.commit()  
 # сохранить изменения  
 self.con.commit()  
  
  
 def \_\_del\_\_(self):  
 # отключение от БД  
 self.con.close()  
  
  
 def view(self):  
 # просмотр всех записей в таблице БД  
 self.cur.execute("SELECT \* FROM products")  
 # список всех записей из таблицы  
 rows = self.cur.fetchall()  
 return rows  
  
  
 def insert(self, company\_name, product\_name, unit\_of\_measurement, purchase\_price, delivery\_date, volume, cost):  
 # добавить запись  
 self.cur.execute("INSERT INTO products "  
 "VALUES (NULL, ?, ?, ?, ?, ?, ?, ?)", (company\_name, product\_name, unit\_of\_measurement, purchase\_price, delivery\_date, volume, cost,))  
 self.con.commit()  
  
  
 def update(self, id, company\_name, product\_name, unit\_of\_measurement, purchase\_price, delivery\_date, volume, cost):  
 # редактирование записи  
 self.cur.execute("UPDATE products SET "  
 "company\_name=?, product\_name=?, unit\_of\_measurement=?, purchase\_price=?, delivery\_date=?, volume=?, cost=?", (company\_name, product\_name, unit\_of\_measurement, purchase\_price, delivery\_date, volume, cost,))  
 self.con.commit()  
  
 def delete(self, id):  
 self.cur.execute("DELETE FROM products "  
 "WHERE ID=?", (id,))  
 self.con.commit()  
  
  
 def search(self, company\_name):  
 self.cur.execute("SELECT company\_name, product\_name, unit\_of\_measurement, purchase\_price, delivery\_date, volume, cost FROM products "  
 "WHERE company\_name=?", (company\_name,))  
 rows = self.cur.fetchall()  
 return rows

2. Второй модуль 123.py

from db\_helper import Student  
# создать объект базы данных  
database\_students = Student()  
# логика  
# добавление записи  
  
  
def add\_command(company\_name, product\_name, unit\_of\_measurement, purchase\_price, delivery\_date, volume, cost):  
 database\_students.insert(company\_name, product\_name, unit\_of\_measurement, purchase\_price, delivery\_date, volume, cost)  
# просмотр всех записей  
  
  
def view\_command():  
 for row in database\_students.view():  
 print(row)  
# поиск по фамилии  
  
  
def search\_command(company\_name):  
 if len(database\_students.search(company\_name)) > 0:  
 for row in database\_students.search(company\_name):  
 print(row)  
 else:  
 print("Такой компании нет")  
 # основная программа в консоли  
 # добавление записи  
for i in range(0):  
 add\_command(input("Введите имя компании: "),  
 input("Введите имя продукта: "),  
 input("Введите единицу измерения: "),  
 int(input("Введите цену покупки: ")),  
 (input("Введите дату доставки: ")),  
 int(input("Введите количество: ")),  
 int(input("Введите цену: ")))  
# просмотр всех записей  
view\_command()  
search\_command(input("Имя компании? "))

Результат выполнения:



3. Модуль Example\_add.py

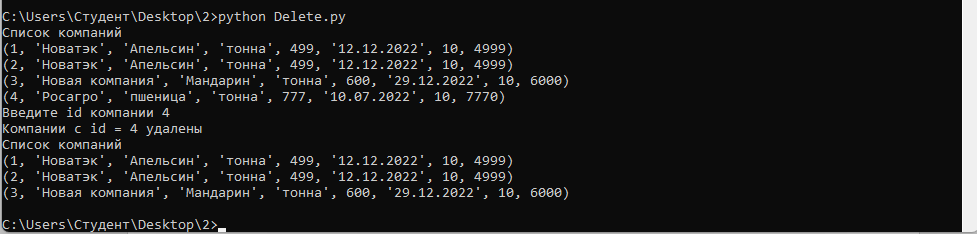
from db\_helper import Student  
# создать объект базы данных  
database\_students = Student()  
# логика  
# добавление записи  
  
  
def add\_command(company\_name, product\_name, unit\_of\_measurement, purchase\_price, delivery\_date, volume, cost):  
 database\_students.insert(company\_name, product\_name, unit\_of\_measurement, purchase\_price, delivery\_date, volume, cost)  
# просмотр всех записей  
  
  
def view\_command():  
 for i in range(1):  
 add\_command(input("Введите имя компании: "),  
 input("Введите имя продукта: "),  
 input("Введите единицу измерения: "),  
 int(input("Введите цену покупки: ")),  
 (input("Введите дату доставки: ")),  
 int(input("Введите количество: ")),  
 int(input("Введите цену: ")))  
 # просмотр всех записей  
view\_command()

Результат выполнения:



4. Модуль Delete.py

from db\_helper import Student  
# создать объект базы данных  
database\_students = Student()  
# удаление по id студента  
  
  
def delete\_command(id):  
 database\_students.delete(id)  
 print(f"Компании с id = {id} удалены")  
# просмотр всех записей  
  
  
def view\_command():  
 for row in database\_students.view():  
 print(row)  
# основная программа  
print("Список компаний")  
view\_command()  
id\_delete = int(input("Введите id компании "))  
delete\_command(id\_delete)  
print("Список компаний")  
view\_command()

Результат выполнения:  
  


5. Модуль update.py

from db\_helper import Student  
# создать объект базы данных  
database\_students = Student()  
# удаление по id студента  
  
  
def update\_command(id, company\_name, product\_name, unit\_of\_measurement, purchase\_price, delivery\_date, volume, cost):  
 database\_students.update(id, company\_name, product\_name, unit\_of\_measurement, purchase\_price, delivery\_date, volume, cost)  
 print(f"Данные компании с id = {id} обновлены")  
# просмотр всех записей  
  
  
def view\_command():  
 for row in database\_students.view():  
 print(row)  
  
# основная программа  
print("Список компаний")  
view\_command()  
  
id\_update = int(input("Введите id компании "))  
print("Укажите новые данные: ")  
company\_name = input("Имя компании ")  
product\_name = input("Имя продукта ")  
unit\_of\_measurement = input("Единица измерения ")  
purchase\_price = int(input("Цена за штуку "))  
delivery\_date = input("Дата доставки ")  
volume = int(input("Количество "))  
cost = int(input("Общая цена "))  
update\_command(id\_update, company\_name, product\_name, unit\_of\_measurement, purchase\_price, delivery\_date, volume, cost)  
  
print("Список студентов")  
view\_command()

Результат выполнения:

