

Отчёт

Практическое занятие № 15

Тема: составление программ для работ с базами данных в IDE PyCharm Community.

Цель: закрепить усвоенные знания, понятия, алгоритмы, основные принципы составления программ, приобрести навыки составления программ для работ с базами данных в IDE PyCharm Community.

Постановка задачи 1.

Вариант 13. Приложение ТОВАРНЫЙ ЗАПАС для автоматизированного учёта товарных запасов на складе. БД должна содержать таблицу Товары со следующей структурой записи: Код товара, Торговая марка, Тип, Цена, Количество на складе, Минимальный запас

Тип алгоритма 1: линейный

Текст программы 1:

```
#Вариант 13. Приложение ТОВАРНЫЙ ЗАПАС для автоматизированного
#учёта товарных запасов на складе. БД должна содержать
#таблицу Товары со следующей структурой записи: Код
#товара, Торговая марка, Тип, Цена, Количество на складе,
#Минимальный запас
import sqlite3 as sq
from info_zapas import info_items
with sq.connect('zapas.db') as con:
    cur=con.cursor()
    cur.execute("DROP TABLE IF EXISTS items")
    cur.execute("CREATE TABLE IF NOT EXISTS items("
                "item_id INTEGER PRIMARY KEY AUTOINCREMENT,"
                "brand TEXT NOT NULL,"
                "item_type TEXT NOT NULL,"
                "price INTEGER NOT NULL,"
                "quantity INTEGER NOT NULL,"
                "min_zapas INTEGER NOT NULL)")
with sq.connect('zapas.db') as con:
    cur=con.cursor()
    cur.executemany("INSERT INTO items VALUES(?,?,?,?,?,?)", info_items)
with sq.connect('zapas.db') as con:
    cur = con.cursor()
    cur.execute("SELECT * FROM items WHERE price>1000 AND quantity<1000")
    result_1=cur.fetchall()
    print(result_1)
with sq.connect('zapas.db') as con:
    cur = con.cursor()
    cur.execute("SELECT * FROM items WHERE brand LIKE 'N%')")
    result_2 = cur.fetchall()
    print(result_2)
```

```

with sq.connect('zapas.db') as con:
    cur = con.cursor()
    cur.execute("SELECT brand,price FROM items WHERE price<quantity")
    result_3 = cur.fetchall()
    print(result_3)
with sq.connect('zapas.db') as con:
    cur = con.cursor()
    cur.execute("UPDATE items SET price=price+25 WHERE price<1000")
    cur.execute("SELECT * FROM items")
    result_4=cur.fetchall()
    print(f"\n{result_4}")
with sq.connect('zapas.db') as con:
    cur = con.cursor()
    cur.execute("UPDATE items SET brand='Квас для вас' WHERE item_id=1")
    cur.execute("SELECT * FROM items WHERE item_id=1")
    result_5=cur.fetchall()
    print(result_5)
with sq.connect('zapas.db') as con:
    cur = con.cursor()
    cur.execute("UPDATE items SET item_type='лягушка' WHERE price<1000 AND
(min_zapas<1000 OR quantity>600)")
    cur.execute("SELECT * FROM items WHERE item_type='лягушка'")
    result_6=cur.fetchall()
    print(result_6)
with sq.connect('zapas.db') as con:
    cur = con.cursor()
    cur.execute("DELETE FROM items WHERE item_id%2==0")
    cur.execute("SELECT * FROM items")
    result_7=cur.fetchall()
    print(f"\n{result_7}")
with sq.connect('zapas.db') as con:
    cur = con.cursor()
    cur.execute("DELETE FROM items WHERE item_type LIKE 'т%'")
    cur.execute("SELECT * FROM items")
    result_8=cur.fetchall()
    print(result_8)
with sq.connect('zapas.db') as con:
    cur = con.cursor()
    cur.execute("DELETE FROM items WHERE price>1000")
    cur.execute("SELECT * FROM items")
    result_9=cur.fetchall()
    print(result_9)

```

Протокол работы программы 1:

[(1, 'Apple', 'телефон', 90000, 500, 50), (3, 'Gibson', 'гитара', 300000, 80, 20), (5, 'LG', 'телевизор', 100000, 400, 90), (9, 'Nike', 'кроссовки', 10000, 400, 300)]

[(7, 'Nescafe', 'кофе', 300, 1900, 1000), (9, 'Nike', 'кроссовки', 10000, 400, 300)]

[('Тавр', 400), ('Influence', 500), ('Colgate', 80), ('Nescafe', 300), ('Простоквашино', 60), ('Аютинский', 30)]

[(1, 'Apple', 'телефон', 90000, 500, 50), (2, 'Тавр', 'колбаса', 425, 700, 100), (3, 'Gibson', 'гитара', 300000, 80, 20), (4, 'Influence', 'тушь', 525, 1000, 300), (5, 'LG',

'телевизор', 100000, 400, 90), (6, 'Colgate', 'зубная паста', 105, 2000, 900), (7, 'Nescafe', 'кофе', 325, 1900, 1000), (8, 'Простоквашино', 'молоко', 85, 1000, 950), (9, 'Nike', 'кроссовки', 10000, 400, 300), (10, 'Аютинский', 'хлеб', 55, 980, 200)]

[(1, 'Квас для вас', 'телефон', 90000, 500, 50)]

[(2, 'Тавр', 'лягушка', 425, 700, 100), (4, 'Influence', 'лягушка', 525, 1000, 300), (6, 'Colgate', 'лягушка', 105, 2000, 900), (7, 'Nescafe', 'лягушка', 325, 1900, 1000), (8, 'Простоквашино', 'лягушка', 85, 1000, 950), (10, 'Аютинский', 'лягушка', 55, 980, 200)]

[(1, 'Квас для вас', 'телефон', 90000, 500, 50), (3, 'Gibson', 'гитара', 300000, 80, 20), (5, 'LG', 'телевизор', 100000, 400, 90), (7, 'Nescafe', 'лягушка', 325, 1900, 1000), (9, 'Nike', 'кроссовки', 10000, 400, 300)]

[(3, 'Gibson', 'гитара', 300000, 80, 20), (7, 'Nescafe', 'лягушка', 325, 1900, 1000), (9, 'Nike', 'кроссовки', 10000, 400, 300)]

[(7, 'Nescafe', 'лягушка', 325, 1900, 1000)]

Process finished with exit code 0

Вывод: в процессе выполнения практического занятия закрепила усвоенные знания, понятия, алгоритмы, основные принципы составления программ, приобрела навыки составления программ для работы с базами данных в IDE PyCharm Community.

Выполнены разработка кода, отладка, тестирование, оптимизация программного кода.

Готовые программные коды выложены на GitHub.