

Студент группы ИС-22 Зоренко Константин  
Сергеевич.

### **Практическое занятие № 15**

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

**Количество часов:** 4.

**Цели практического занятия:** закрепить усвоенные знания, понятия, алгоритмы, основные принципы составления программ, приобрести навыки составления программ с использованием с использованием регулярных выражений вIDE PyCharm Community.

**Постановка задачи:** *Для автоматического контроля работ по ремонту бытовой техники БД должна содержать таблицу ремонт телевизоров, имеющую следующую структуру записи: Марка телевизора, Завод-производитель, Цена, Дата ремонта, Документ, Мастер, Сумма оплаты.*

Студент группы ИС-22 Зоренко Константин  
Сергеевич.

## PZ\_15

### Текст программы:

```
"""
/*телемастерская ->
/*telemaster
Для автоматического контроля работ по ремонту бытовой техники БД должна
содержать таблицу ремонт телевизоров, имеющую следующую структуру записи:
Марка телевизора, Завод-производитель, Цена, Дата ремонта, Документ, Мастер,
Сумма оплаты.
"""

#Сделал для того чтобы в терминале глазкам было смотреть приятнее
class color:
    DARKCYAN = '\033[36m'
    BOLD = '\033[1m'
    UNDERLINE = '\033[4m'
    END = '\033[0m'

import sqlite3
from mark import mark_list
with sqlite3.Connection("database.db") as cur:
    con = cur.cursor()
    con.execute("DROP TABLE IF EXISTS users")
    con.execute("""CREATE TABLE IF NOT EXISTS users(
prod_id INTEGER PRIMARY KEY AUTOINCREMENT,
mark TEXT NOT NULL DEFAULT 'Default',
manufacture TEXT NOT NULL DEFAULT 'Default',
price INTEGER,
data DATA,
document TEXT,
master TEXT,
cheque INTEGER
)""")

    """    con.execute("INSERT INTO users VALUES(1,'JVC', 'JVC', 20000, 2000-01-01,
'Лицензия'гарантийный талон'инструкция', 'Тимофей', 11000)")

    con.execute("INSERT INTO users VALUES(2,'JVC', 'JVC', 80000, 2000-01-01,
'Лицензия'гарантийный талон'инструкция', 'Тимофей', 50000)")

    con.execute("INSERT INTO users VALUES(3,'JVC', 'JVC', 50000, 2000-01-01,
'Лицензия'гарантийный талон'инструкция', 'Тимофей', 20000)")

    con.execute("SELECT * FROM users")
    result = con.fetchall()
    print(f"МЫ ПОЛУЧАЕМ ТРИ ИНСЕРТА, ДАЛЕЕ БУДЕТ РАБОТА НЕПОСРЕДСТВЕННО С
ИМПОРТИРОВАННОЙ ДАТА БАЗОЙ)\n{result}\n"+"-"*264)"""

with sqlite3.Connection('database.db') as cur:
    con = cur.cursor()
    con.executemany("INSERT INTO users VALUES(?, ?, ?, ?, ?, ?, ?, ?)", mark_list)
    con.execute("SELECT * FROM users")
    data = con.fetchall()

    print(color.DARKCYAN + "-" * 220)
    print(f"{color.DARKCYAN}{color.BOLD}| Вставка моего списка на 10 штук: \n|
{data}\n"+"-"*220+"\n"*2)
```

```

with sqlite3.Connection('database.db')as cur:
    con = cur.cursor()
    con.execute("SELECT master, data, mark FROM users ORDER BY data DESC")
    info_master = con.fetchall()
    print("-" * 220)
    print(f"{color.DARKCYAN}{color.BOLD}| МАСТЕРА И ИНФОРМАЦИЯ О ИХ ПРОДЕЛАННОЙ РАБОТЕ:\n| {info_master}\n"+"-"*220+"\n"*2)
    con.execute("SELECT master, data, mark FROM users ORDER BY data DESC")
    last_master = con.fetchmany(1)
    print("-" * 220)
    print(f"{color.DARKCYAN}{color.BOLD}| Последний работавший сотрудник(задействована дата)\n| {last_master}\n"+"-"*220+"\n"*2)

```

```

with sqlite3.Connection('database.db')as cur:
    con = cur.cursor()
    con.execute("SELECT * FROM users WHERE price > 40000 AND master = 'Игорь'")
    master_check_norma = con.fetchall()
    print("-" * 220)
    print(f"{color.DARKCYAN}{color.BOLD}| ТУТ МЫ СМОТРИМ КАКИЕ ИГОРЁК БРАЛ ЗАКАЗЫ, КОТОРЫЕ БЫ ХОРОШО СКЛАДЫВАЛИСЬ НА ЕГО ЗАРПЛАТЕ\n| {master_check_norma}\n"+"-"*220+"\n"*2)
    with sqlite3.connect('database.db')as cur:
        con = cur.cursor()
        con.execute("SELECT * FROM users WHERE price < 40000 AND master = 'Игорь'")
        master_check_min = con.fetchall()
        print("-"*220)
        print(f"{color.BOLD}| ТУТ МЫ СМОТРИМ КАКИЕ ИГОРЁК БРАЛ ЗАКАЗЫ, ОПЛАТА КОТОРЫХ БЫЛА НИЗКОЙ\n| {master_check_min}\n"+"-"*220+"\n"*2)

```

```

with sqlite3.Connection('database.db')as cur:
    con = cur.cursor()
    con.execute("SELECT * FROM users WHERE master = 'Игорь' AND price > 30000")
    old_name = con.fetchone()
    con.execute("UPDATE users SET master = 'Игорявый' WHERE master = 'Игорь' AND price > 30000")
    print("-"*220)
    print(f"| Имя которое было у сотрудника изначально ->\n| {old_name}\n|")
    con.execute("SELECT * FROM users WHERE master = 'Игорявый' AND price > 30000")
    new_name = con.fetchone()
    print(f"| Изменения имени ->\n| {new_name}\n"+"-"*220+"\n"*2)

```

```

with sqlite3.connect('database.db')as cur:
    con = cur.cursor()
    con.execute("SELECT mark, manufacture FROM users WHERE mark = 'Logitek'")
    UNO_CARD = con.fetchone()
    print("-" * 220)
    print(f"| Старая марка, которая была изначальной -> (1 значение - Марка, 2 значение - Производитель) \n| {UNO_CARD}\n|")
    con.execute("UPDATE users SET mark = 'LG' AND manufacture = 'LG' WHERE mark = 'Logitek'")
    con.execute("SELECT mark, manufacture FROM users WHERE mark = 'LG'")
    UNO_REVERSE = con.fetchone()
    print(f"| Изменения в марке, а так же в производителе -> (1 значение - Марка, 2 значение - Производитель) \n| {UNO_REVERSE}\n"+"-"*264+"\n"*2)

```

```

with sqlite3.connect('database.db') as cur:
    con = cur.cursor()
    con.execute("SELECT document FROM users")
    document_old = con.fetchone()
    print("-" * 220)
    print(f"| Изначальное оформление документов -> \n| {document_old}\n|")
    con.execute("UPDATE users SET document = 'Паспорт на холодильник'")
    con.execute("SELECT document FROM users")
    document_new = con.fetchone()
    print(f"| Изменённое оформление вывода документов -> \n|
{document_new}\n" + "-" * 220 + "\n" * 2)

with sqlite3.connect('database.db') as cur:
    con = cur.cursor()
    con.execute("DELETE FROM users WHERE master = 'Кирилл' AND cheque > 20000")
    con.execute("SELECT * FROM users WHERE master = 'Кирилл'")
    kirilla = con.fetchall()
    print("-" * 220)
    print(f"| {color.UNDERLINE}Тут Кириллы, после удаления кириллов, у которых чек
оплаты около 20000, но не больше! {color.END} {color.DARKCYAN}\n|
{kirilla}\n" + "-" * 220 + "\n" * 2)

with sqlite3.connect('database.db') as cur:
    con = cur.cursor()
    con.execute("DELETE FROM users WHERE master = 'Игорявый' OR (master = 'Игорь') AND
cheque > 30000")
    con.execute("SELECT * FROM users WHERE master = 'Игорь'")
    Igor = con.fetchall()
    print("-" * 220)
    print(f"| {color.UNDERLINE}Тут Игорь, после удаления Игорей с зарплатой больше
30000 {color.END} {color.DARKCYAN}\n| {Igor}\n" + "-" * 220 + "\n" * 2)

```

### Протокол работы программы:

/home/student/Документы/Zorenko\_Py/venv/bin/python  
/home/student/Документы/Zorenko\_Py/pz\_15/database.py

---

| Вставка моего списка на 10 штук:

| [(1, 'Samsung', 'Samsung', 70000, 20000101, 'Лицензиягарантийный талонинструкция',  
'Константин', 35000), (2, 'Philips', 'Philips', 60000, 20000201, 'Лицензиягарантийный  
талонинструкция', 'Кирилл', 30000), (3, 'Sony', 'Sony', 50000, 20000501, 'Лицензия  
гарантийный талон инструкция', 'Игорь', 25000), (4, 'Pioneer', 'Pioneer', 40000, 20000201,  
'Лицензиягарантийный талонинструкция', 'Кирилл', 20000), (5, 'Toshiba', 'Toshiba', 30000,  
20000101, 'Лицензиягарантийный талонинструкция', 'Игорь', 15000), (6, 'LG', 'LG', 20000,  
20000501, 'Лицензиягарантийный талонинструкция', 'Игорь', 10000), (7, 'Logitek', 'Logitek',  
100000, 20000301, 'Лицензиягарантийный талонинструкция', 'Кирилл', 50000), (8, 'LGOG',  
'LGOG', 120000, 20000101, 'Лицензиягарантийный талонинструкция', 'Игорь', 60000), (9,  
'SAT', 'SAT', 160000, 20000901, 'Лицензиягарантийный талонинструкция', 'Константин',  
80000), (10, 'Hurma', 'Hurma', 130000, 20001201, 'Лицензиягарантийный талонинструкция',  
'Игорь', 65000)]

---

-----  
-----  
| МАСТЕРА И ИНФОРМАЦИЯ О ИХ ПРОДЕЛАННОЙ РАБОТЕ:

| [('Игорь', 20001201, 'Hurma'), ('Константин', 20000901, 'CAT'), ('Игорь', 20000501, 'Sony'),  
('Игорь', 20000501, 'LG'), ('Кирилл', 20000301, 'Logitek'), ('Кирилл', 20000201, 'Philips'),  
('Кирилл', 20000201, 'Pioneer'), ('Константин', 20000101, 'Samsung'), ('Игорь', 20000101,  
'Toshiba'), ('Игорь ', 20000101, 'LGOG')]  
-----  
-----

-----  
-----  
| Последний работавший сотрудник(задействована дата)

[('Игорь', 20001201, 'Hurma')]
-----

-----  
-----  
| ТУТ МЫ СМОТРИМ КАКИЕ ИГОРЁК БРАЛ ЗАКАЗЫ, КОТОРЫЕ БЫ ХОРОШО  
СКЛАДЫВАЛИСЬ НА ЕГО ЗАРПЛАТЕ

| [(3, 'Sony', 'Sony', 50000, 20000501, 'Лицензия гарантийный талон инструкция', 'Игорь',  
25000), (10, 'Hurma', 'Hurma', 130000, 20001201, 'Лицензия гарантийный талон инструкция',  
'Игорь', 65000)]  
-----  
-----

-----  
-----  
| ТУТ МЫ СМОТРИМ КАКИЕ ИГОРЁК БРАЛ ЗАКАЗЫ, ОПЛАТА КОТОРЫХ БЫЛА  
НИЗКОЙ

| [(5, 'Toshiba', 'Toshiba', 30000, 20000101, 'Лицензия гарантийный талон инструкция', 'Игорь',  
15000), (6, 'LG', 'LG', 20000, 20000501, 'Лицензия гарантийный талон инструкция', 'Игорь',  
10000)]  
-----  
-----

-----  
-----  
| Имя которое было у сотрудника изначально ->

| (3, 'Sony', 'Sony', 50000, 20000501, 'Лицензия гарантийный талон инструкция', 'Игорь', 25000)  
|

| Изменения имени ->

| (3, 'Sony', 'Sony', 50000, 20000501, 'Лицензия гарантийный талон инструкция', 'Игорявый',  
25000)  
-----  
-----

-----  
-----  
| Старая марка, которая была изначальной -> (1 значение - Марка, 2 значение -  
Производитель)  
| ('Logitek', 'Logitek')  
|  
| Изменения в марке, а так же в производителе -> (1 значение - Марка, 2 значение -  
Производитель)  
('LG', 'LG')
-----  
-----

-----  
-----  
| Изначальное оформление документов ->  
| ('Лицензия', 'гарантийный талон', 'инструкция',)  
|  
| Изменённое оформление вывода документов ->  
('Паспорт на холодильник',)
-----  
-----

-----  
-----  
| Тут Кириллы, после удаления кириллов, у которых чек оплаты около 20000, но не больше!  
[(4, 'Pioneer', 'Pioneer', 40000, 20000201, 'Паспорт на холодильник', 'Кирилл', 20000)]
-----  
-----

-----  
-----  
| Тут Игорь, после удаления Игорей с зарплатой больше 30000  
| [(5, 'Toshiba', 'Toshiba', 30000, 20000101, 'Паспорт на холодильник', 'Игорь', 15000), (6, 'LG',  
'LG', 20000, 20000501, 'Паспорт на холодильник', 'Игорь', 10000)]  
-----  
-----  
-----

Process finished with exit code 0

## **ВЫВОД:**

**закрепил усвоенные знания, понятия, алгоритмы, основные принципы составления программ, приобрести навыки составления программ с использованием с**

**использованием регулярных выражений в IDE PyCharm Community.**