

Практическое занятие №15

Тема: Составление программ для работы с базами данных в IDE PyCharm Community

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

Постановка задачи №1

Приложение **АБИТУРИЕНТ** для автоматизации работы приемной комиссии, которая обеспечивает обработку анкетных данных абитуриентов. Таблица *Анкета* содержит следующие данные об абитуриентах: *Регистрационный номер*, *Фамилия*, *Имя*, *Отчество*, *Дата Рождения*, *Награды* (наличие кр. Диплома или медали (ад/нет), *Адрес*, выбранная *Специальность*.

Текст программы:

```
import sqlite3

def create_db():
    with sqlite3.connect("abityrient1") as conn:
        cursor = conn.cursor()
        cursor.execute("""
            CREATE TABLE if not exists anketa(
                reg_number VARCHAR(5),
                surname VARCHAR(30),
                name VARCHAR(30),
                middle_name VARCHAR(30),
                birthday DATETIME,
                awards BOOLEAN DEFAULT(FALSE),
                address VARCHAR(255),
                profession VARCHAR(30)
            );
        """)
        conn.commit()
    print('ready!')

# def insert():
#     with sqlite3.connect("abityrient1") as conn:
#         cursor = conn.cursor()
#
#         commands = """ INSERT INTO anketa VALUES ('00001', 'Иващенко', 'Иван', 'Сергеевич',
#             '01.04.1998', FALSE, 'аллая 12', 'Мед брат');
#             INSERT INTO anketa VALUES ('00002', 'Орешников', 'Игорь', 'Радионович',
#             '05.07.1997', TRUE, 'задорная 18', 'Хирург');
#             INSERT INTO anketa VALUES ('00003', 'Чепрасов', 'Артем', 'Андреевич',
#             '21.08.2006', TRUE, 'красивая 25', 'Программист(плохой программист)');
#             INSERT INTO anketa VALUES ('00004', 'Прудько', 'Владимир', 'Борисович',
#             '08.11.2006', TRUE, 'слепая 12', 'Программист(плохой программист)');
```

```

# INSERT INTO anketa VALUES ('00005', 'Подугольников', 'Иван', 'Артемович',
# '12.05.2006', TRUE, 'глухая 14', 'Программист(плохой программист)');
# INSERT INTO anketa VALUES ('00005', 'Цыбина', 'Евгения', 'Эдуардовна',
# '04.10.2007', TRUE, 'толстова 166', 'Программист(хороший программист)');
# INSERT INTO anketa VALUES ('00006', 'Архипов', 'Валерий', 'Михайлович',
# '03.02.2006', TRUE, 'житная 20', 'Программист(нормальный программист)');
# INSERT INTO anketa VALUES ('00007', 'Рахмат', 'Исмаил', 'Мухамедов',
# '24.3.1980', TRUE, 'веселая 27', 'Водитель автобуса');
# INSERT INTO anketa VALUES ('00008', 'Тулипина', 'Анна', 'Александровна',
# '19.09.1976', TRUE, 'пушкинская 28', 'Учитель биологии');
# INSERT INTO anketa VALUES ('00009', 'Некрасов', 'Радион', 'Борисович',
# '12.07.1990', TRUE, 'советская 97', 'Инженер');
# INSERT INTO anketa VALUES ('00010', 'Коломыков', 'Петр', 'Филипович',
# '04.04.1965', TRUE, 'пушкинская 45', 'Филосов')""
#
# commands = commands.split(";")
# for i in commands:
#     cursor.execute(i)
#     conn.commit()

```

```

def insert(reg_number, surname, name, middle_name, birthday, awards, address, profession):
    with sqlite3.connect("abityrient1") as conn:
        cursor = conn.cursor()

```

```

        commands = """ INSERT INTO anketa VALUES (
            """ + reg_number + """,
            """ + surname + """,
            """ + name + """,
            """ + middle_name + """,
            """ + birthday + """,
            """ + awards + """,
            """ + address + """,
            """ + profession + """
        )"""

```

```

        cursor.execute(commands)
        conn.commit()

```

```

def show_all():
    with sqlite3.connect("abityrient1") as conn:
        cursor = conn.cursor()
        cursor.execute("select * FROM anketa")
        print(cursor.fetchall())

```

```

def show_data_reg_number(reg_number):
    with sqlite3.connect("abityrient1") as conn:
        cursor = conn.cursor()
        commands = """ SELECT * FROM anketa WHERE reg_number = """ + reg_number + """"
        cursor.execute(commands)

```

```

print(cursor.fetchall())

def show_data_surname(surname):
    with sqlite3.connect("abityrient1") as conn:
        cursor = conn.cursor()
        commands = """ SELECT * FROM anketa WHERE surname = """ + surname + """
        cursor.execute(commands)
        print(cursor.fetchall())

def show_data_name(name):
    with sqlite3.connect("abityrient1") as conn:
        cursor = conn.cursor()
        commands = """ SELECT * FROM anketa WHERE name = """ + name + """
        cursor.execute(commands)
        print(cursor.fetchall())

def show_data_middle_name(middle_name):
    with sqlite3.connect("abityrient1") as conn:
        cursor = conn.cursor()
        commands = """ SELECT * FROM anketa WHERE middle_name = """ + middle_name + """
        cursor.execute(commands)
        print(cursor.fetchall())

def show_data_birthday(birthday):
    with sqlite3.connect("abityrient1") as conn:
        cursor = conn.cursor()
        commands = """ SELECT * FROM anketa WHERE birthday = """ + birthday + """
        cursor.execute(commands)
        print(cursor.fetchall())

def show_data_address(address):
    with sqlite3.connect("abityrient1") as conn:
        cursor = conn.cursor()
        commands = """ SELECT * FROM anketa WHERE address = """ + address + """
        cursor.execute(commands)
        print(cursor.fetchall())

def show_data_profession(profession):
    with sqlite3.connect("abityrient1") as conn:
        cursor = conn.cursor()
        commands = """ SELECT * FROM anketa WHERE profession = """ + profession + """
        cursor.execute(commands)
        print(cursor.fetchall())

def delete_all():

```

```

with sqlite3.connect("abityrient1") as conn:
    cursor = conn.cursor()
    commands = """DELETE from anketa """
    cursor.execute(commands)
    conn.commit()

def delete_data_reg_number(reg_number):
    with sqlite3.connect("abityrient1") as conn:
        cursor = conn.cursor()
        commands = """DELETE from anketa WHERE reg_number = """ + reg_number + """
        print(commands)
        cursor.execute(commands)
        conn.commit()

def update_surname(surname, num):
    with sqlite3.connect("abityrient1") as conn:
        cursor = conn.cursor()
        commands = """update anketa set surname = """ + surname \
            + """ WHERE reg_number = """ + num + """
        print(commands)
        cursor.execute(commands)
        conn.commit()

def update_name(name, num):
    with sqlite3.connect("abityrient1") as conn:
        cursor = conn.cursor()
        commands = """update anketa set name = """ + name \
            + """ WHERE reg_number = """ + num + """
        print(commands)
        cursor.execute(commands)
        conn.commit()

def update_middle_name(middle_name, num):
    with sqlite3.connect("abityrient1") as conn:
        cursor = conn.cursor()
        commands = """update anketa set middle_name = """ + middle_name \
            + """ WHERE reg_number = """ + num + """
        print(commands)
        cursor.execute(commands)
        conn.commit()

def update_awars(awars, num):
    with sqlite3.connect("abityrient1") as conn:
        cursor = conn.cursor()
        commands = """update anketa set awars = """ + awars \
            + """ WHERE reg_number = """ + num + """

```

```

print(commands)
cursor.execute(commands)
conn.commit()

def update_address(address, num):
    with sqlite3.connect("abityrient1") as conn:
        cursor = conn.cursor()
        commands = """update anketa set addresss = """ + address \
            + """ WHERE reg_number = """ + num + """"""
        print(commands)
        cursor.execute(commands)
        conn.commit()

def update_profession(profession, num):
    with sqlite3.connect("abityrient1") as conn:
        cursor = conn.cursor()
        commands = """update anketa set profession = """ + profession \
            + """ WHERE reg_number = """ + num + """"""
        print(commands)
        cursor.execute(commands)
        conn.commit()

create_db()
while True:
    cikle = input("Введите 0 для выхода из программы, \n"
        "1 - добавить абитуриента \n"
        "2 - если хотите сделать поиск \n"
        "3 - для изменения значений \n"
        "4 - удаляет всю базу данных \n"
        "5 - удаление абитуриента по номеру \n"
        "6 - показывает всю базу данных")
    if cikle == "0":
        break

    elif cikle == '1':
        reg_number = input('reg_number=')
        surname = input('surname=')
        name = input('name=')
        middle_name = input('middle_name=')
        birthday = input('birthday=')
        awards = input('awards=')
        address = input('address=')
        profession = input('profession=')
        insert(reg_number, surname, name, middle_name, birthday, awards, address, profession)
        show_data_reg_number(reg_number)

    elif cikle == '2':
        poick = input("0 - reg_number \n"
            "1 - surname \n"
            "2 - name \n")

```

```

        "3 - middle_name \n"
        "4 - birthday \n"
        "5 - address \n"
        "6 - profession \n")
if poick == '0':
    reg_number = input('reg_number=')
    show_data_reg_number(reg_number)
elif poick == '1':
    surname = input('surname=')
    show_data_surname(surname)
elif poick == '2':
    name = input('name=')
    show_data_name(name)
elif poick == '3':
    middle_name = input('middle_name=')
    show_data_middle_name(middle_name)
elif poick == '4':
    birthday = input('birthday=')
    show_data_birthday(birthday)
elif poick == '5':
    address = input('address=')
    show_data_address(address)
elif poick == '6':
    profession = input('profession=')
    show_data_profession(profession)

elif cikle == "3":
    update = input("1 - surname \n"
        "2 - name \n"
        "3 - middle_name \n"
        "4 - address \n"
        "5 - profession \n")

    if update == '1':
        surname = input('new_surname=')
        num = input("Введите reg_number")
        update_surname(surname, num)
    elif update == '2':
        name = input('new_name=')
        num = input("Введите reg_number")
        update_name(name, num)
    elif update == '3':
        middle_name = input('middle_name=')
        num = input("Введите reg_number")
        update_middle_name(middle_name, num)
    elif update == '4':
        address = input('address=')
        num = input("Введите reg_number")
        update_address(address, num)
    elif update == '5':
        profession = input('profession=')
        num = input("Введите reg_number")
        update_profession(profession, num)

```

```
elif cikle == "4":
    delete_all()

elif cikle == "5":
    reg_number = input("Введите reg_number")
    delete_data_reg_number(reg_number)

elif cikle == "6":
    show_all()
```

Протокол работы программы:

Введите 0 для выхода из программы,

1 - добавить абитуриента

2 - если хотите сделать поиск

3 - для изменения значений

4 - удаляет всю базу данных

5 - удаление абитуриента по номеру

6 - показывает всю базу данных

: 1

reg_number=00001

surname=Архипов

name=Валерий

middle_name=Михайлович

birthday=03.02.2006

awards=False

address=Житная 20

profession=начинаючи программист

[('00001', 'Архипов', 'Валерий', 'Михайлович', '03.02.2006', 0, 'Житная 20', 'Начинающий программист'), ('00001', 'Архипов', 'Валерий', 'Михайлович', '03.02.2006', 0, 'Житная 20', 'начинаючи программист')]

Введите 0 для выхода из программы,

1 - добавить абитуриента

2 - если хотите сделать поиск

3 - для изменения значений

4 - удаляет всю базу данных

5 - удаление абитуриента по номеру

6 - показывает всю базу данных

: 0

Process finished with exit code 0

Вывод: в процессе выполнения практического занятия выработал навыки составления программ с использованием баз данных в IDE PyCharm Community..

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

Готовые программные коды выложены на GitHub.