

## Практическое занятие № 15

Наименование практического занятия: составление программ с использованием баз данных в IDE PyCharm Community.

Цели практического занятия: закрепить усвоенные знания, понятия навыки составления программ с использованием баз данных в IDE PyCharm Community.

Постановка задачи.

Приложение УЧЕТ ВНУТРИОФИСНЫХ РАСХОДОВ для некоторой организации. БД должна содержать таблицу Канцелярия со следующей структурой записи: ФИО работника, отдел, вид расхода, сумма.

**Тип алгоритма:** Линейный, Линейный

### Текст программы:

```
import sqlite3
import os

db_file = 'учет_внутриофисных_расходов.db'

if os.path.exists(db_file):
    os.remove(db_file)

def insert_expense(conn, fio, department, expense_type, amount):
    cursor = conn.cursor()
    cursor.execute("""
        INSERT INTO Kancelyariya (fio, zvanie, prichina, summa)
        VALUES (?, ?, ?, ?)
        """, (fio, department, expense_type, amount))
    conn.commit()

def update_expense(conn, expense_id, fio, department, expense_type, amount):
    cursor = conn.cursor()
    cursor.execute("""
        UPDATE Kancelyariya
        SET fio = ?,
            zvanie = ?,
            prichina = ?,
            summa = ?
        WHERE id = ?
        """, (fio, department, expense_type, amount, expense_id))
    conn.commit()

def delete_expense(conn, expense_id):
    cursor = conn.cursor()
    cursor.execute("DELETE FROM Kancelyariya WHERE id = ?", (expense_id,))
    conn.commit()

conn = sqlite3.connect('учет_внутриофисных_расходов.db')
cursor = conn.cursor()

cursor.execute('''
    CREATE TABLE IF NOT EXISTS Kancelyariya (
        id INTEGER PRIMARY KEY AUTOINCREMENT,
        fio TEXT NOT NULL,
        zvanie TEXT NOT NULL,
        prichina TEXT NOT NULL,
        summa REAL NOT NULL
    )
''')
```

```

loop = int(input("введите кол-во желаемых строк: "))
for i in range(loop):
    fio = input("Введите ФИО: ")
    zvanie = input("Введите звание: ")
    prichina = input("Введите причину расхода: ")
    summa = input("Введите сумму расхода: ")
    insert_expense(conn, fio, zvanie, prichina, summa)

conn.commit()
def search_expenses(conn, search_query):
    cursor = conn.cursor()
    cursor.execute("""
        SELECT * FROM Kancelyariya
        WHERE fio LIKE ? OR zvanie LIKE ? OR prichina LIKE ?
    """, ('%' + search_query + '%', '%' + search_query + '%', '%' +
search_query + '%'))
    rows = cursor.fetchall()

    print("Результаты поиска:")
    for row in rows:
        print("ID: {}, ФИО: {}, Звание: {}, Причина: {}, Сумма:
{}".format(row[0], row[1], row[2], row[3], row[4]))

def fetch_all_expenses(conn):
    cursor = conn.cursor()
    cursor.execute("SELECT * FROM Kancelyariya")
    rows = cursor.fetchall()

    for row in rows:
        print(row)

fetch_all_expenses(conn)

while True:
    action = input("Выберите действие (удалить - d, редактировать - e, поиск -
s, выход - q): ")
    if action == "d":
        delete_id = int(input("Введите ID записи для удаления: "))
        delete_expense(conn, delete_id)
    elif action == "e":
        edit_id = int(input("Введите ID записи для редактирования: "))
        new_fio = input("Введите новое ФИО: ")
        new_department = input("Введите новое звание: ")
        new_expense_type = input("Введите новую причину расхода: ")
        new_amount = float(input("Введите новую сумму расхода: "))
        update_expense(conn, edit_id, new_fio, new_department,
new_expense_type, new_amount)
    elif action == "s":
        search_query = input("Введите текст для поиска: ")
        search_expenses(conn, search_query)

    elif action == "q":
        break
    else:
        print("Неверное действие, попробуйте еще раз.")

print("Вывод")
fetch_all_expenses(conn)

fetch_all_expenses(conn)

conn.close()

```

## Протокол работы программы:

```
(3, 'Игорь Игнатьевич Земской', 'Фермер', 'Пшеница', 2000.0)
Выберите действие (удалить - d, редактировать - e, поиск - s, выход - q): e
Введите ID записи для редактирования: 2
Введите новое ФИО: Кирилл Алексеевич Претендент
Введите новое звание: Акционер
Введите новую причину расхода: акции
Введите новую сумму расхода: 10000
вывод
(1, 'Пётр Первый Великий', 'Царь', 'Оружие', 1000.0)
(2, 'Кирилл Алексеевич Претендент', 'Акционер', 'акции', 10000.0)
(3, 'Игорь Игнатьевич Земской', 'Фермер', 'Пшеница', 2000.0)
Выберите действие (удалить - d, редактировать - e, поиск - s, выход - q): d
Введите ID записи для удаления: 3
вывод
(1, 'Пётр Первый Великий', 'Царь', 'Оружие', 1000.0)
(2, 'Кирилл Алексеевич Претендент', 'Акционер', 'акции', 10000.0)
Выберите действие (удалить - d, редактировать - e, поиск - s, выход - q): s
Введите текст для поиска: Пётр
Результаты поиска:
ID: 1, ФИО: Пётр Первый Великий, Звание: Царь, Причина: Оружие, Сумма: 1000.0
вывод
(1, 'Пётр Первый Великий', 'Царь', 'Оружие', 1000.0)
(2, 'Кирилл Алексеевич Претендент', 'Акционер', 'акции', 10000.0)
Выберите действие (удалить - d, редактировать - e, поиск - s, выход - q): q
(1, 'Пётр Первый Великий', 'Царь', 'Оружие', 1000.0)
(2, 'Кирилл Алексеевич Претендент', 'Акционер', 'акции', 10000.0)

Process finished with exit code 0
```

**Вывод:** Я закрепил знания, понятия, основные принципы работы базами данных в IDE PyCharm Community.