Практическое занятие № 15

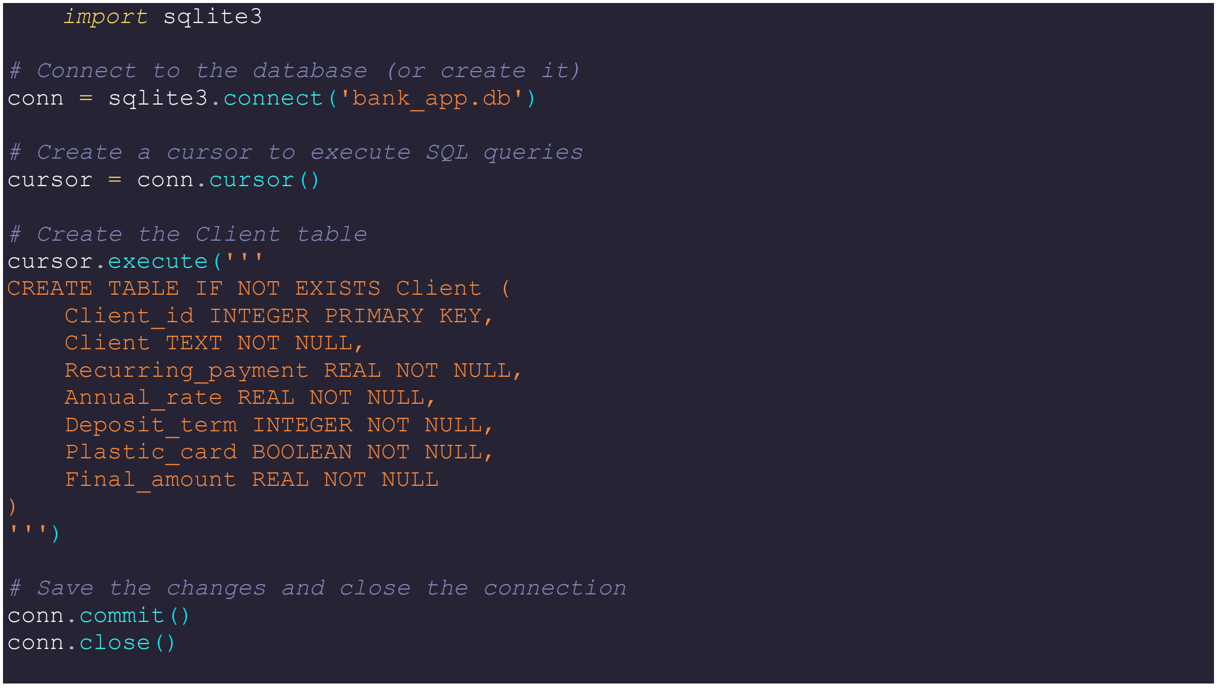
Тема: составление программ для работы с базами данных в IDE PyCharm Community.

Цель: закрепить усвоенные знания, понятия, алгоритмы, основные принципы составления программ, работы с БД в IDE PyCharm Community

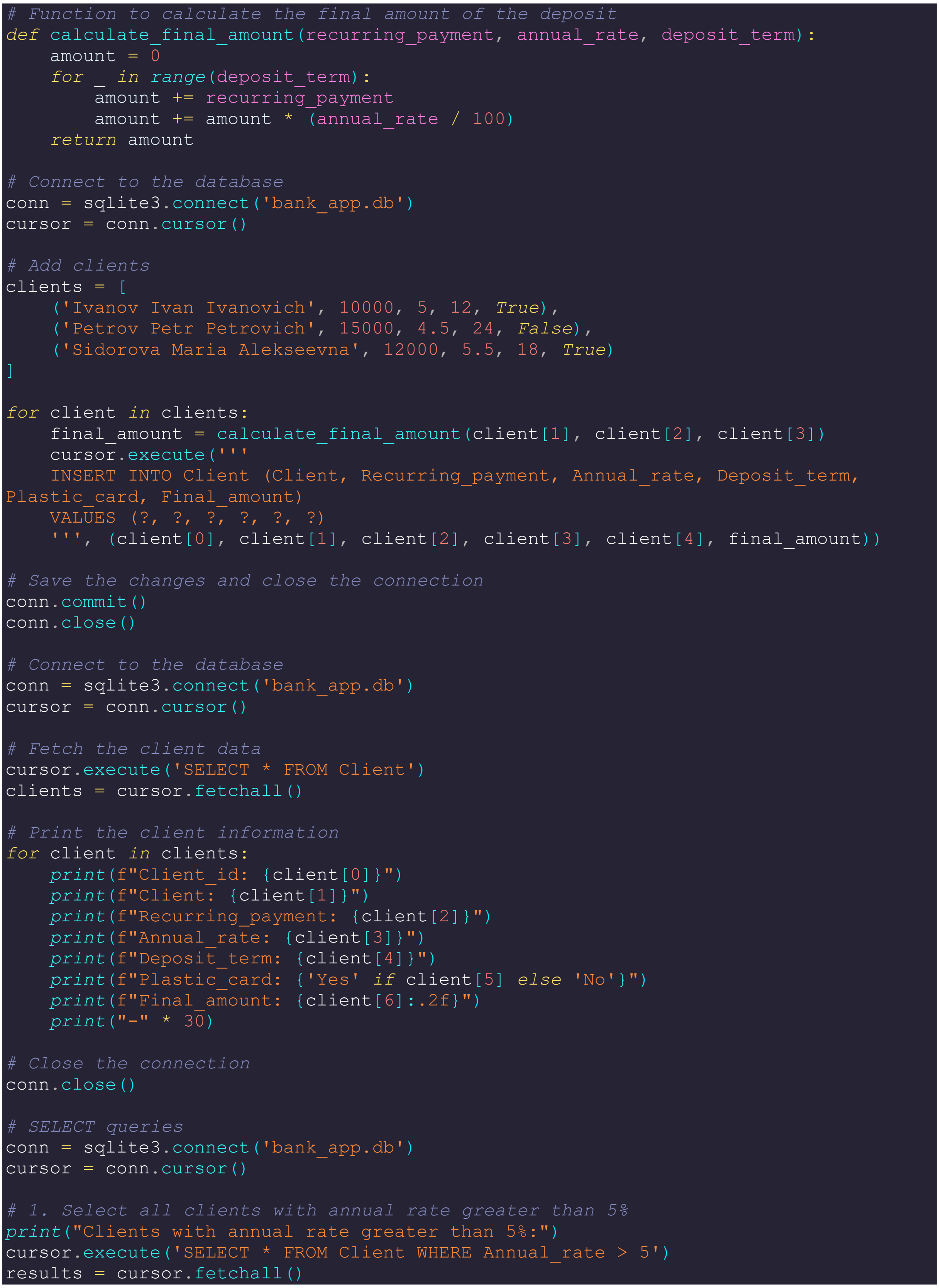
Постановка задачи.

Приложение БАНК для отслеживания накапливаемых на счетах клиентов банка сумм. Таблица Клиент должна содержать следующую информацию: Код клиента, Клиент (Ф.И.О.), Периодический платеж, Годовой %, Срок вклада, Пластиковая карта (логическое поле), Конечная сумма.

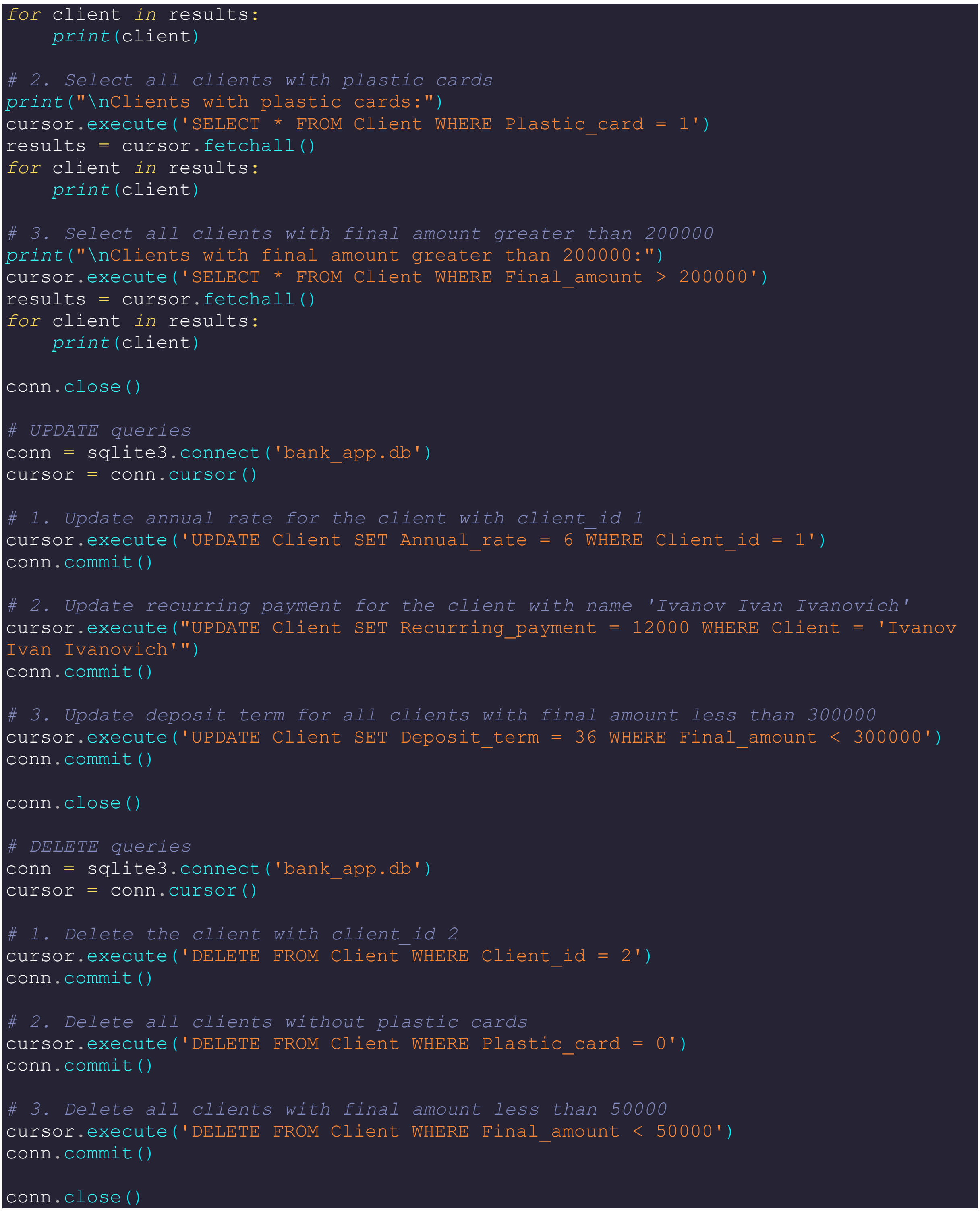
Текст программы:



```
    import sqlite3

# Connect to the database (or create it)
conn = sqlite3.connect('bank_app.db')

# Create a cursor to execute SQL queries
cursor = conn.cursor()

# Create the Client table
cursor.execute('''
CREATE TABLE IF NOT EXISTS Client (
    Client_id INTEGER PRIMARY KEY,
    Client TEXT NOT NULL,
    Recurring_payment REAL NOT NULL,
    Annual_rate REAL NOT NULL,
    Deposit_term INTEGER NOT NULL,
    Plastic_card BOOLEAN NOT NULL,
    Final_amount REAL NOT NULL
)
''')

# Save the changes and close the connection
conn.commit()
conn.close()
```



```
# Function to calculate the final amount of the deposit
def calculate_final_amount(recurring_payment, annual_rate, deposit_term):
    amount = 0
    for _ in range(deposit_term):
        amount += recurring_payment
        amount += amount * (annual_rate / 100)
    return amount

# Connect to the database
conn = sqlite3.connect('bank_app.db')
cursor = conn.cursor()

# Add clients
clients = [
    ('Ivanov Ivan Ivanovich', 10000, 5, 12, True),
    ('Petrov Petr Petrovich', 15000, 4.5, 24, False),
    ('Sidorova Maria Alekseevna', 12000, 5.5, 18, True)
]

for client in clients:
    final_amount = calculate_final_amount(client[1], client[2], client[3])
    cursor.execute('''
    INSERT INTO Client (Client, Recurring_payment, Annual_rate, Deposit_term,
Plastic_card, Final_amount)
    VALUES (?, ?, ?, ?, ?, ?)
    ''', (client[0], client[1], client[2], client[3], client[4], final_amount))

# Save the changes and close the connection
conn.commit()
conn.close()

# Connect to the database
conn = sqlite3.connect('bank_app.db')
cursor = conn.cursor()

# Fetch the client data
cursor.execute('SELECT * FROM Client')
clients = cursor.fetchall()

# Print the client information
for client in clients:
    print(f"Client_id: {client[0]}")
    print(f"Client: {client[1]}")
    print(f"Recurring_payment: {client[2]}")
    print(f"Annual_rate: {client[3]}")
    print(f"Deposit_term: {client[4]}")
    print(f"Plastic_card: {'Yes' if client[5] else 'No'}")
    print(f"Final_amount: {client[6]:.2f}")
    print("-" * 30)

# Close the connection
conn.close()

# SELECT queries
conn = sqlite3.connect('bank_app.db')
cursor = conn.cursor()

# 1. Select all clients with annual rate greater than 5%
print("Clients with annual rate greater than 5%:")
cursor.execute('SELECT * FROM Client WHERE Annual_rate > 5')
results = cursor.fetchall()
```



```
for client in results:
    print(client)

# 2. Select all clients with plastic cards
print("\nClients with plastic cards:")
cursor.execute('SELECT * FROM Client WHERE Plastic_card = 1')
results = cursor.fetchall()
for client in results:
    print(client)

# 3. Select all clients with final amount greater than 200000
print("\nClients with final amount greater than 200000:")
cursor.execute('SELECT * FROM Client WHERE Final_amount > 200000')
results = cursor.fetchall()
for client in results:
    print(client)

conn.close()

# UPDATE queries
conn = sqlite3.connect('bank_app.db')
cursor = conn.cursor()

# 1. Update annual rate for the client with client_id 1
cursor.execute('UPDATE Client SET Annual_rate = 6 WHERE Client_id = 1')
conn.commit()

# 2. Update recurring payment for the client with name 'Ivanov Ivan Ivanovich'
cursor.execute("UPDATE Client SET Recurring_payment = 12000 WHERE Client = 'Ivanov
Ivan Ivanovich'")
conn.commit()

# 3. Update deposit term for all clients with final amount less than 300000
cursor.execute('UPDATE Client SET Deposit_term = 36 WHERE Final_amount < 300000')
conn.commit()

conn.close()

# DELETE queries
conn = sqlite3.connect('bank_app.db')
cursor = conn.cursor()

# 1. Delete the client with client_id 2
cursor.execute('DELETE FROM Client WHERE Client_id = 2')
conn.commit()

# 2. Delete all clients without plastic cards
cursor.execute('DELETE FROM Client WHERE Plastic_card = 0')
conn.commit()

# 3. Delete all clients with final amount less than 50000
cursor.execute('DELETE FROM Client WHERE Final_amount < 50000')
conn.commit()

conn.close()
```

Протокол работы программы:

Client_id: 1

```
Client: Ivanov Ivan Ivanovich
Recurring_payment: 10000.0
Annual_rate: 5.0
Deposit_term: 12
Plastic_card: Yes
Final_amount: 167129.83
------------------------------
Client_id: 2
Client: Petrov Petr Petrovich
Recurring_payment: 15000.0
Annual_rate: 4.5
Deposit_term: 24
Plastic_card: No
Final_amount: 653478.15
------------------------------
Client_id: 3
Client: Sidorova Maria Alekseevna
Recurring_payment: 12000.0
Annual_rate: 5.5
Deposit_term: 18
Plastic_card: Yes
Final_amount: 373232.05
------------------------------
Clients with annual rate greater than 5%:
(3, 'Sidorova Maria Alekseevna', 12000.0, 5.5, 18, 1, 373232.05320549925)

Clients with plastic cards:
(1, 'Ivanov Ivan Ivanovich', 10000.0, 5.0, 12, 1, 167129.82846464711)
(3, 'Sidorova Maria Alekseevna', 12000.0, 5.5, 18, 1, 373232.05320549925)
```

```
Clients with final amount greater than 200000:
(2, 'Petrov Petr Petrovich', 15000.0, 4.5, 24, 0, 653478.1521790329)
(3, 'Sidorova Maria Alekseevna', 12000.0, 5.5, 18, 1, 373232.05320549925)

Process finished with exit code 0
```

В ходе работы я закрепил усвоенные знания, понятия, алгоритмы, основные принципы составления программ, работы с БД в IDE PyCharm Community Готовые программные коды выложены на GitHub.