Code:

import sqlite3

from datetime import datetime

# Подключение к базе данных

def connect\_db():

    return sqlite3.connect("бухгалтерия.db")

# Создание таблиц в базе данных

def create\_tables():

    conn = connect\_db()

    cursor = conn.cursor()

    # Создание таблицы счетов

    cursor.execute('''

        CREATE TABLE IF NOT EXISTS счета (

            id INTEGER PRIMARY KEY AUTOINCREMENT,

            название TEXT NOT NULL,

            тип TEXT NOT NULL,

            баланс REAL DEFAULT 0

        )

    ''')

    # Создание таблицы транзакций

    cursor.execute('''

        CREATE TABLE IF NOT EXISTS транзакции (

            id INTEGER PRIMARY KEY AUTOINCREMENT,

            дебет\_счет INTEGER,

            кредит\_счет INTEGER,

            сумма REAL NOT NULL,

            дата TEXT NOT NULL,

            описание TEXT,

            FOREIGN KEY(дебет\_счет) REFERENCES счета(id),

            FOREIGN KEY(кредит\_счет) REFERENCES счета(id)

        )

    ''')

    conn.commit()

    conn.close()

# Добавление нового счета

def add\_account(name, acc\_type):

    conn = connect\_db()

    cursor = conn.cursor()

    cursor.execute('''

        INSERT INTO счета (название, тип)

        VALUES (?, ?)

    ''', (name, acc\_type))

    conn.commit()

    conn.close()

    print(f"Счет '{name}' успешно добавлен.")

# Добавление транзакции

def add\_transaction(debit\_account, credit\_account, amount, description=""):

    conn = connect\_db()

    cursor = conn.cursor()

    date = datetime.now().strftime("%Y-%m-%d %H:%M:%S")

    # Добавление транзакции

    cursor.execute('''

        INSERT INTO транзакции (дебет\_счет, кредит\_счет, сумма, дата, описание)

        VALUES (?, ?, ?, ?, ?)

    ''', (debit\_account, credit\_account, amount, date, description))

    # Обновление баланса счетов

    cursor.execute('UPDATE счета SET баланс = баланс - ? WHERE id = ?', (amount, debit\_account))

    cursor.execute('UPDATE счета SET баланс = баланс + ? WHERE id = ?', (amount, credit\_account))

    conn.commit()

    conn.close()

    print("Транзакция успешно добавлена.")

# Генерация финансового отчета

def generate\_report():

    conn = connect\_db()

    cursor = conn.cursor()

    # Получение балансов счетов

    cursor.execute('SELECT id, название, тип, баланс FROM счета')

    accounts = cursor.fetchall()

    print("\n--- Финансовый отчет ---")

    for acc in accounts:

        print(f"Счет: {acc[1]} ({acc[2]}) - Баланс: {acc[3]:,.2f} руб.")

    print("-----------------------")

    conn.close()

# Меню программы

def menu():

    while True:

        print("\n--- Простая бухгалтерская система ---")

        print("1. Создать таблицы")

        print("2. Добавить новый счет")

        print("3. Добавить транзакцию")

        print("4. Сгенерировать финансовый отчет")

        print("5. Выйти")

        choice = input("Выберите опцию: ")

        if choice == "1":

            create\_tables()

            print("Таблицы в базе данных успешно созданы.")

        elif choice == "2":

            name = input("Введите название счета: ")

            acc\_type = input("Введите тип счета (Актив/Пассив): ")

            add\_account(name, acc\_type)

        elif choice == "3":

            debit\_account = int(input("Введите ID дебетового счета: "))

            credit\_account = int(input("Введите ID кредитового счета: "))

            amount = float(input("Введите сумму транзакции: "))

            description = input("Введите описание транзакции (необязательно): ")

            add\_transaction(debit\_account, credit\_account, amount, description)

        elif choice == "4":

            generate\_report()

        elif choice == "5":

            print("Выход из программы. До свидания!")

            break

        else:

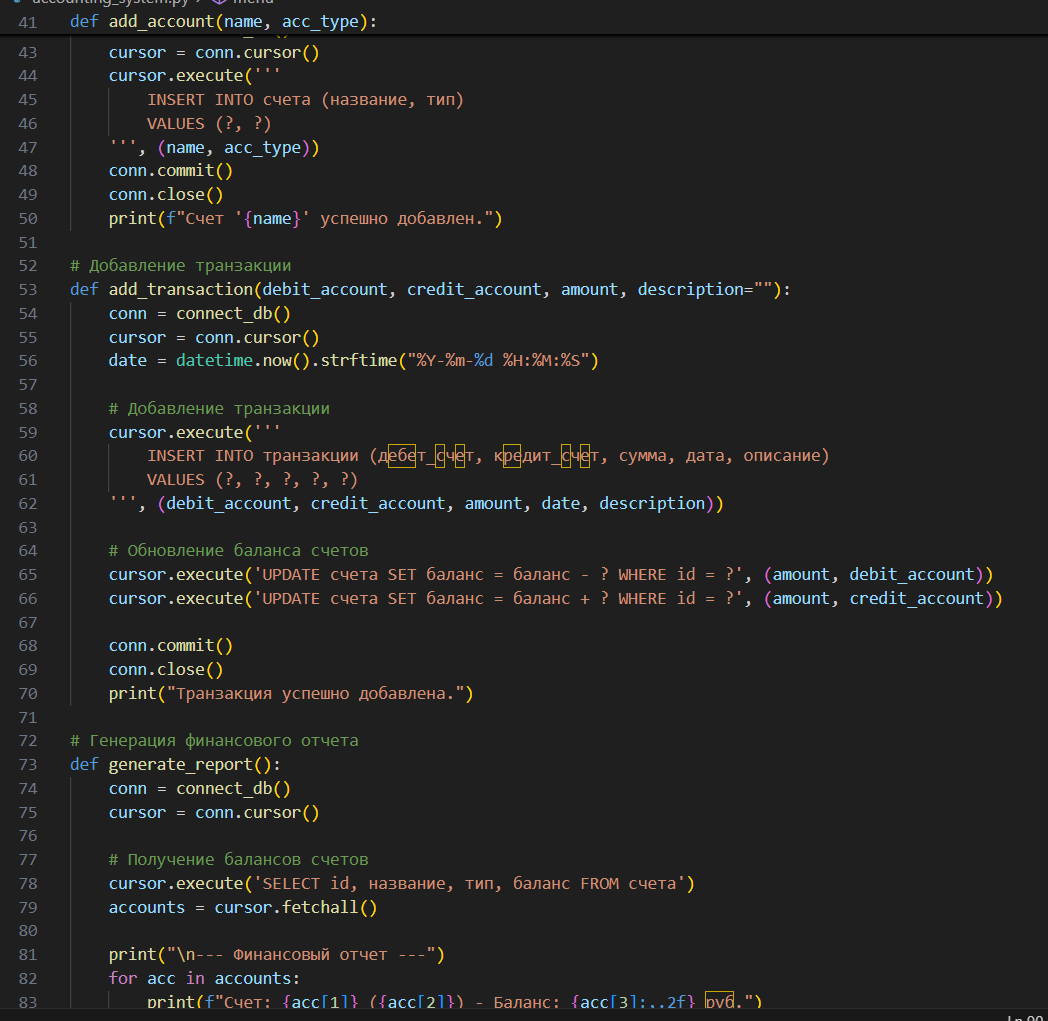
            print("Неверный выбор. Попробуйте снова.")

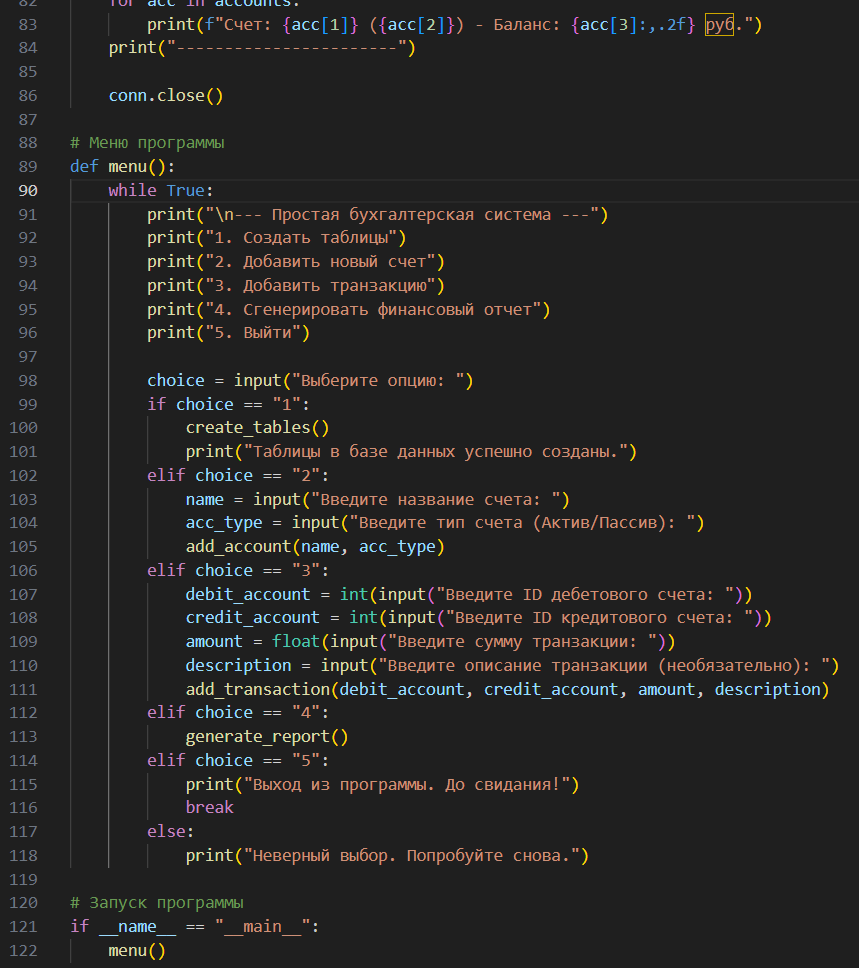
# Запуск программы

if \_\_name\_\_ == "\_\_main\_\_":

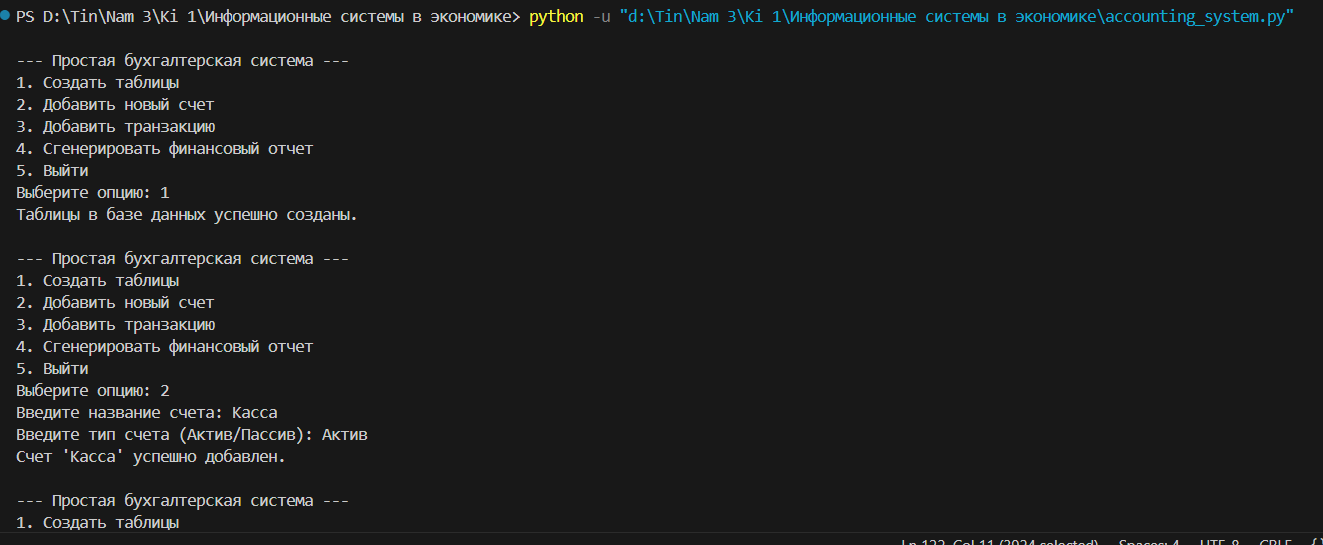
    menu()

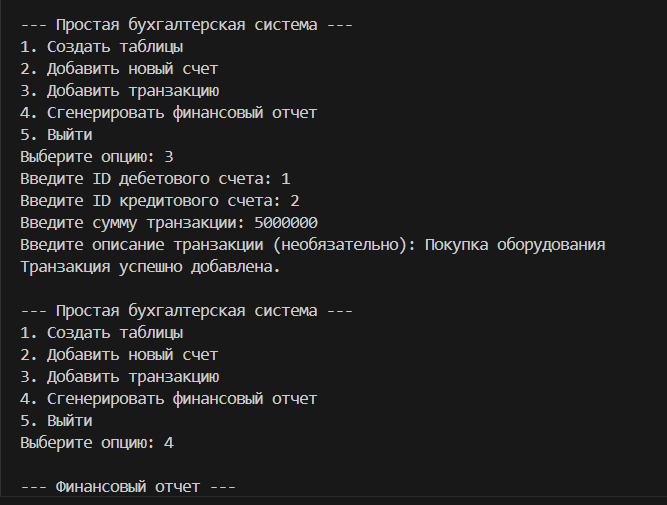


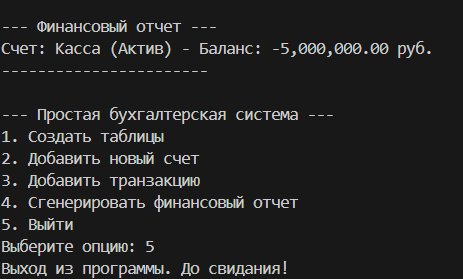




Результат:







Эта программа представляет собой простую бухгалтерскую систему, написанную на Python с использованием SQLite в качестве базы данных для управления информацией о счетах и транзакциях. Программа включает основные функции, такие как создание таблиц базы данных, добавление счетов, запись транзакций и генерация финансового отчета.

При запуске программа подключается к базе данных через функцию connect\_db, используя файл бухгалтерия.db. Если этот файл отсутствует, SQLite автоматически его создаст. Функция создания таблиц (create\_tables) создает две основные таблицы: таблицу счета, которая хранит информацию о счетах (ID, название, тип счета, например, "Актив" или "Пассив", и баланс по умолчанию равен 0), и таблицу транзакции, которая записывает финансовые операции между счетами (включая ID дебетового и кредитового счета, сумму, дату и описание).

Пользователь может добавить новый счет через функцию add\_account, указав название и тип счета. Транзакции записываются с помощью функции add\_transaction, где с дебетового счета списываются деньги, а на кредитовый счет зачисляются. Каждая транзакция сохраняется в таблице транзакции, и баланс соответствующих счетов обновляется мгновенно.

Финансовый отчет создается с помощью функции generate\_report, которая извлекает информацию из таблицы счета и отображает список счетов с их балансами в денежном формате. Вся система управляется через интерактивное меню, где пользователь может выбрать опции, такие как создание таблиц, добавление счетов, добавление транзакций или просмотр финансового отчета. Программа обеспечивает гибкость и удобство использования, позволяя эффективно управлять финансами.

Результат:

Результат показывает, что вы выполнили несколько операций в бухгалтерской программе. Сначала вы выбрали "1. Создать таблицы", и программа успешно создала две таблицы в базе данных: таблицу счета для хранения информации о счетах и таблицу транзакции для хранения информации о транзакциях. Затем вы выбрали "2. Добавить новый счет", ввели название счета "Касса" и тип счета "Актив", и этот счет был успешно добавлен в базу данных с начальным балансом 0,00 руб..

После этого вы выбрали "3. Добавить транзакцию", совершили транзакцию с ID дебетового счета = 1 (счет "Касса"), ID кредитового счета = 2, сумма транзакции составила 5,000,000 руб., и описание транзакции — "Покупка оборудования". Программа успешно зарегистрировала эту транзакцию, но так как ID кредитового счета = 2 не существует в базе данных, деньги не были зачислены на другой счет, а только списаны с счета "Касса". Поэтому баланс счета "Касса" стал -5,000,000.00 руб..

Когда вы выбрали "4. Сгенерировать финансовый отчет", программа вывела финансовый отчет с единственным счетом: "Касса (Актив)", с балансом -5,000,000.00 руб.. Этот результат показывает, что транзакция уменьшила баланс счета "Касса", но не обновила счет получателя, так как ID получающего счета был неверным. В конце, когда вы выбрали "5. Выйти", программа завершила свою работу и вышла.