Министерство науки и высшего образования Российской Федерации

Федеральное государственное автономное образовательное учреждение высшего образования



Факультет информационных технологий

Кафедра «Инфокогнитивные технологии»

Образовательная программа (профиль):

«Веб-технологии»

Пояснительная записка

За 1 семестр

на тему: «Задание на выполнение курсового проекта»

по дисциплине: «Основы программирования»

|  |  |  |  |
| --- | --- | --- | --- |
|  |  | Выполнил:  Студент Группа  Дата | Тимшин Александр Денисович  231-323  10.09.25 |
|  |  | Принял:  Старший преподаватель | Красникова Ирина Николаевна |

Москва

2025

**Цель работы**

Целью выполнения курсового проекта является разработка десктопного приложения для учёта личных расходов пользователя с возможностью выбора категории расходов и способа оплаты. Проект направлен на закрепление навыков объектно-ориентированного программирования на языке Python, работы с графическими пользовательскими интерфейсами с использованием библиотеки tkinter, а также создания и использования реляционной базы данных на основе SQLite.

**Формулировка задания**

Разработать программное обеспечение, которое позволяет пользователю:

* добавлять расходы, указывая сумму, категорию и способ оплаты;
* сохранять данные о расходах в базу данных;
* просматривать историю расходов;
* генерировать ежемесячный отчет с разбивкой по категориям и способам оплаты;
* получать уведомления о перерасходе по определённым категориям.

**Теоретическая часть**

Программирование на языке Python позволяет быстро разрабатывать приложения различной сложности благодаря лаконичному синтаксису и широкому выбору библиотек. Для создания графического интерфейса была использована стандартная библиотека tkinter, обеспечивающая создание оконных приложений с элементами управления, такими как текстовые поля, кнопки и выпадающие списки.

Хранение данных реализовано с использованием встроенной реляционной базы данных SQLite. Это облегчает работу с данными, позволяет организовать их в таблицы и поддерживать связи между ними через первичные и внешние ключи.

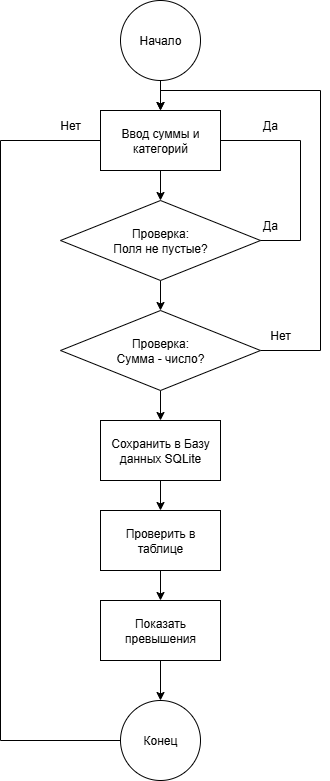
Архитектура базы данных разработана таким образом, чтобы обеспечить нормализацию данных: расходы привязываются к пользователю, категории и способу оплаты. Это позволяет гибко анализировать данные и строить отчеты по различным параметрам.

Для представления отчётной информации была реализована функция подсчёта расходов по категориям. Программа считывает данные из файла, фильтрует их по текущему месяцу и подсчитывает суммы по каждой категории. Уведомления о перерасходе основаны на заранее установленных пороговых значениях.

**Алгоритм работы программы**

1. При запуске приложения автоматически создаются все необходимые таблицы в базе данных, если они отсутствуют.
2. Пользователь вводит сумму расхода, категорию и выбирает способ оплаты.
3. Программа проверяет корректность введённых данных.
4. В случае корректности данные сохраняются в базу данных.
5. Расход отображается в таблице на экране.
6. При добавлении расходов производится проверка превышения лимита по категориям.
7. Пользователь может сгенерировать отчёт за текущий месяц.
8. Все расходы сохраняются в файл базы данных expenses.db.

Блок-схема алгоритма работы представлена на рисунке 1.



1 Блок-схема алгоритма

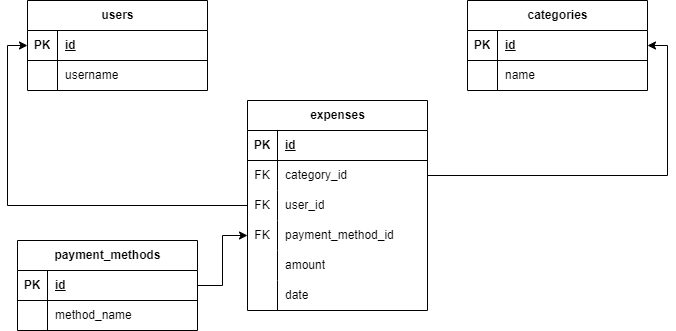
**ER-диаграмма базы данных**

База данных состоит из четырёх таблиц:

* **users** (id, username)
* **categories** (id, name)
* **payment\_methods** (id, method\_name)
* **expenses** (id, user\_id, category\_id, payment\_method\_id, amount, date)

Связи:

* expenses.user\_id → users.id
* expenses.category\_id → categories.id
* expenses.payment\_method\_id → payment\_methods.id



1ER-диаграмма

**Логическая модель базы данных**

|  |  |  |  |
| --- | --- | --- | --- |
| **Таблица** | **Поле** | **Тип данных** | **Ограничения** |
| **users** | id | INTEGER | PRIMARY KEY AUTOINCREMENT |
|  | username | TEXT | NOT NULL, UNIQUE |
| **categories** | id | INTEGER | PRIMARY KEY AUTOINCREMENT |
|  | name | TEXT | NOT NULL, UNIQUE |
| **payment\_methods** | id | INTEGER | PRIMARY KEY AUTOINCREMENT |
|  | method\_name | TEXT | NOT NULL, UNIQUE |
| **expenses** | id | INTEGER | PRIMARY KEY AUTOINCREMENT |
|  | user\_id | INTEGER | NOT NULL, FOREIGN KEY REFERENCES users(id) |
|  | category\_id | INTEGER | NOT NULL, FOREIGN KEY REFERENCES categories(id) |
|  | payment\_method\_id | INTEGER | NOT NULL, FOREIGN KEY REFERENCES payment\_methods(id) |
|  | amount | REAL | NOT NULL |
|  | date | TEXT | NOT NULL (формат YYYY-MM-DD HH:MM:SS) |

**Листинг программы**

Программа состоит из нескольких модулей:

* main.py — основной модуль с графическим интерфейсом;
* database.py — работа с базой данных (создание таблиц, добавление и получение данных);
* reports.py — генерация ежемесячного отчёта;
* notifications.py — проверка превышения по категориям.

Ниже приведены основные части кода (фрагменты):

**Форма ввода и таблица:**

self.amount\_entry = tk.Entry(root)

self.category\_entry = tk.Entry(root)

self.expenses\_tree = ttk.Treeview(root, columns=("Дата", "Сумма", "Категория"), show="headings")

**Добавление расходов:**

def add\_expense(self):

amount = float(self.amount\_entry.get())

category = self.category\_entry.get()

with open(FILENAME, "a", newline="", encoding="utf-8") as f:

writer = csv.writer(f)

writer.writerow([date, amount, category])

**Генерация отчета:**

def generate\_monthly\_report():

for row in reader:

if row[0].startswith(current\_month):

expenses\_by\_category[category] += float(row[1])

**Проверка уведомлений:**

if spending[category] > limit:

messages.append(f"Ты тратишь слишком много на {category.lower()}")

*Полный исходный код размещён в каталоге проекта.*

**Результаты работы программы**

В результате выполнения проекта разработано десктопное приложение для учёта расходов пользователя, включающее:

* ввод данных о расходах с категорией и способом оплаты;
* сохранение данных в базу данных;
* просмотр истории расходов;
* формирование текстового отчета;
* систему уведомлений о перерасходах.

Программа прошла тестирование на различных данных и показала корректную работу.

Пример сформированного отчёта:

**==================================================**

**📊 ОТЧЕТ ЗА СЕНТЯБРЬ 2025**

**==================================================**

**💰 ОБЩАЯ СТАТИСТИКА:**

**Всего расходов: 123324.00 ₽**

**Количество транзакций: 3**

**Средний чек: 41108.00 ₽**

**📂 РАСХОДЫ ПО КАТЕГОРИЯМ:**

**Кварплата: 123123.00 ₽ (99.8%)**

**Тест: 201.00 ₽ (0.2%)**

**💳 РАСХОДЫ ПО СПОСОБАМ ОПЛАТЫ:**

**📱 Онлайн-платеж: 123123.00 ₽ (99.8%)**

**💳 Тест: 201.00 ₽ (0.2%)**

**==================================================**

**Отчет сгенерирован: 10.09.2025 17:42:10**

**==================================================**

**Выводы**

Разработка приложения для отслеживания расходов позволила закрепить навыки программирования на языке Python, освоить работу с графическим интерфейсом и реализовать простую систему хранения и анализа пользовательских данных. Работа была выполнена в соответствии с техническим заданием, поставленные цели достигнуты.

Созданный продукт может быть использован как основа для более сложных решений, например, с использованием баз данных, графиков и сетевого доступа.

Github:

Я выполнял работу один

**Список литературы**

1. Лутц М. «Изучаем Python». 5-е издание. – СПб.: Питер, 2019.
2. Документация Python – <https://docs.python.org/3/>
3. Tkinter Tutorial – https://tkdocs.com/tutorial/
4. Официальная документация по модулю csv – <https://docs.python.org/3/library/csv.html>
5. ГОСТ 7.32-2017. Отчет о научно-исследовательской работе. Структура и правила оформления.