**Пояснительная записка к проекту**

**«Бюджетный ассистент»**

Подготовил: Пугач МВ

Назначение приложения

Данное приложение разработано с целью демонстрация программной модели, позволяющей вести учёт доходов и расходов одного человека либо группы. В ходе разработки приложения были поставлены задачи освоить работу с фреймворком PyQt5, библиотекой sqlite и понять взаимодействие между инструментами языка Python и структурой данных, реализуемой SQlite.

Структура данных и модель работы

Приложение состоит из двух частей: базы данных и программного интерфейса взаимодействия с базой данных.

База данных состоит из трёх таблиц, соединённых связями один-ко-многим

(рис.1).

Operations

(Типы операций)

Categories

(Категории операций)

Actions

(Протокол операций)

1 ко многим

1 ко многим

1 ко многим

Рис.1 – схема данных

Таблица Operations имеет 1 атрибут – op\_name (строка), описывающий название категории, к значениям атрибута применено ограничение уникальности.

Таблица Categories имеет 2 атрибута: - cat\_name (строка), описывающий название категории, к значениям применено ограничение уникальности, и optype (строка) являющийся внешним ключом в таблицу Operations для группировки категорий по типам операций.

Таблица Actions содержит следующие атрибуты: id, action\_datetime, optype, category, amount, comment.

Атрибут id (целочисленный) является суррогатным первичным ключом и позволяет уникально идентифицировать каждую операцию в таблице.

Атрибут action\_datetime (дата-время) содержит дату и время операции с точностью до секунды, подразумевается, что пользователь не совершит более одной операции в секунду.

Атрибут optype (строка) является внешним ключом в таблицу Operations и позволяет группировать операции по то типу.

Атрибут category (строка) является внешним ключом в таблицу Categories и позволяет группировать операции по категориям.

Атрибут amount (действительное число) описывает сумму, использованную в операции.

Атрибут comment (строка) позволяет оставить заметку к операции.

Программный интерфейс состоит из нескольких форм и классов, реализующих (совместно с базой) паттерн model-view-controller: моделью выступает БД, представлением – интерфейс в формах, а контроллером – остальные классы.

Рис.2 – структура программы

Главная

Форма

Настройки

Редактор

Базы данных

Environment

DB-dialer

База данных

ОС и внешние файлы

Главная форма реализует большую часть функционала взаимодействия с БД. Из неё можно попасть в форму настроек и в форму редактора, а также вызвать отчёты и справку.

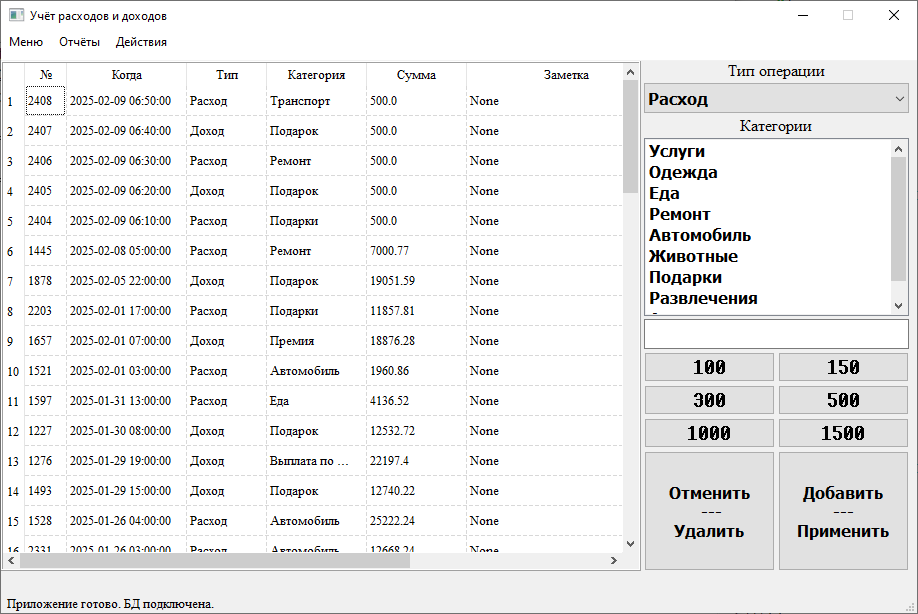


Рис.3 – главная форма

Форма настроек позволяет сконфигурировать внутренние переменные программы, выбрать БД и значения для кнопок предустановленных значений.

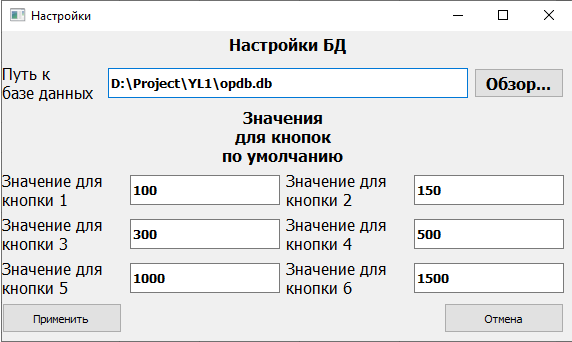


Рис.4 – форма настроек

Форма редактора данных позволяет отредактировать категории операций в БД.

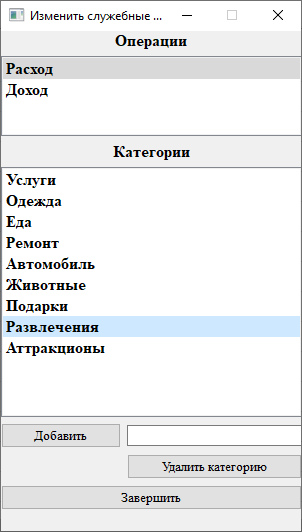


Рис.5 Форма редактора

По мере взаимодействия с программой пользователь может встретить различные сообщения и уведомления (рис.6)

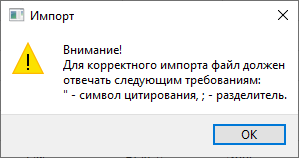
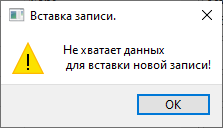
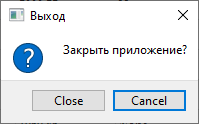
  

Рис.6 – примеры сообщений программы

Класс Environment не имеет графического интерфейса и реализует взаимодействие программы с ОС и файлами. Класс DB-dialer реализует взаимодействие с БД (запросы, создание и анализ, формирование отчётов).

Сценарии использования

Для начала работы с приложением необходимо его просто запустить. Основная часть работы происходит на главной форме.

Для добавления новой записи в таблицу операций необходимо выполнить следующие действия: выбрать тип операции, выбрать категорию, ввести сумму (либо нажать клавишу «дежурной» суммы), нажать клавишу «Добавить \ Применить».

Для редактирования записи в таблице нужно дважды кликнуть в ячейку редактирования, ввести новое значение и нажать клавишу «Добавить \ Применить».

Для удаления записи нужно единожды кликнуть на индекс строки в таблице и нажать клавишу «Отменить \ Удалить».

Для вызова формы настроек, редактора БД или формирования отчётов нужно воспользоваться меню программы в верхней части окна.

Планы на будущее

На основе краткой эксплуатации и обзора существующих решений в будущих версиях будет не лишним добавить:

* Алгоритм порционной загрузки записей, срабатывающий по мере прокручивания таблицы на главной форме;
* Асинхронное обращение к БД;
* Разделение БД на локальный кэш и библиотеку данных для ускорения операций чтения записи и реализации многопользовательского доступа.