**Разработка программных модулей Windows Forms**

Отчёт по лабораторной работе

Студента группы ИСП9-21-2

Щербаков Тимофей Андреевич

**Тема: Разработка приложения для учета данных по расходам (SQLite)**

**Цель работы**: разработка приложения для учета данных расходов пользователя, используя локальную базу данных SQLite.

**Задачи**:

1. Создание макетов форм

2. Разработка класса с БД

3. Отображение расходов за 30 дней в старте, а также последние расходы

4. Добавление расходов

5. Поиск расходов по фильтрам и без фильтров

**Разработка приложения**

Первой задачей являлось создание макетов интерфейса всех форм приложения.

На рисунке 1 показана форма старта приложения.

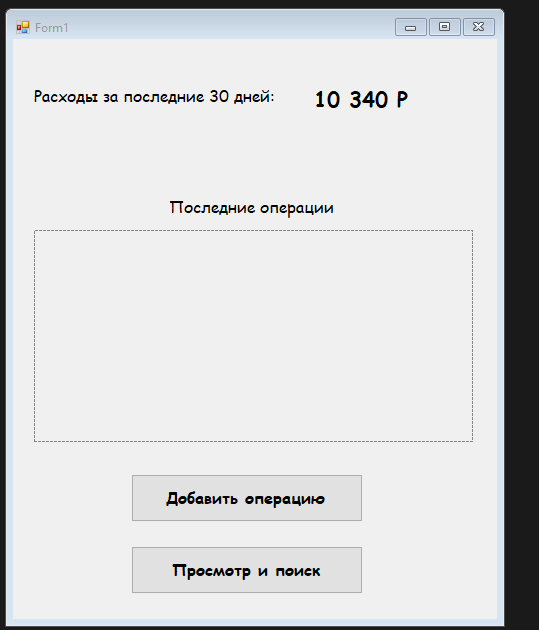


Рисунок 1 - Макет старта приложения

На рисунке 2 показана форма добавления расхода пользователем.

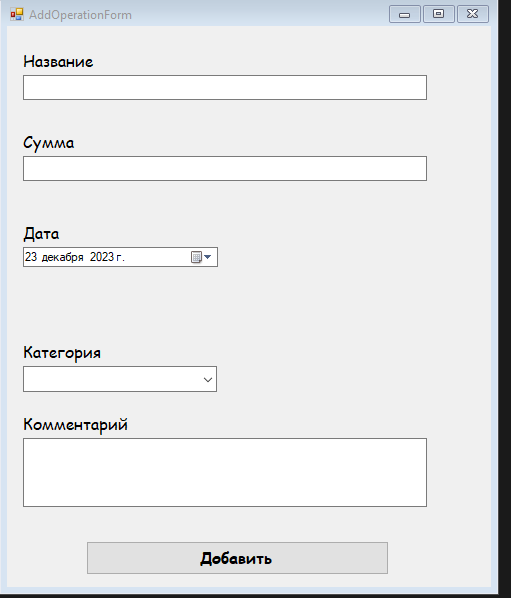


Рисунок 2 - Макет добавления записи расхода

На рисунке 3 показана форма поиска каких-либо расходов.

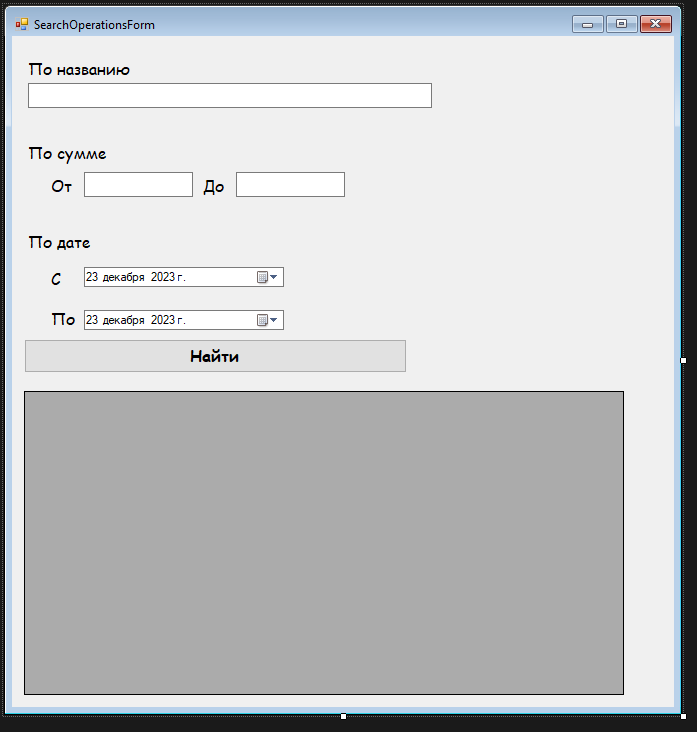


Рисунок 3 - Поиск расходов

Второй задачей является разработка класса с БД (ExpensesTrackerDatabase) для хранения данных расходов. Таблица должна содержать в себе название расхода, сумму, категорию, дату и комментарий.

Запрос на создание БД показан на рисунке 4.

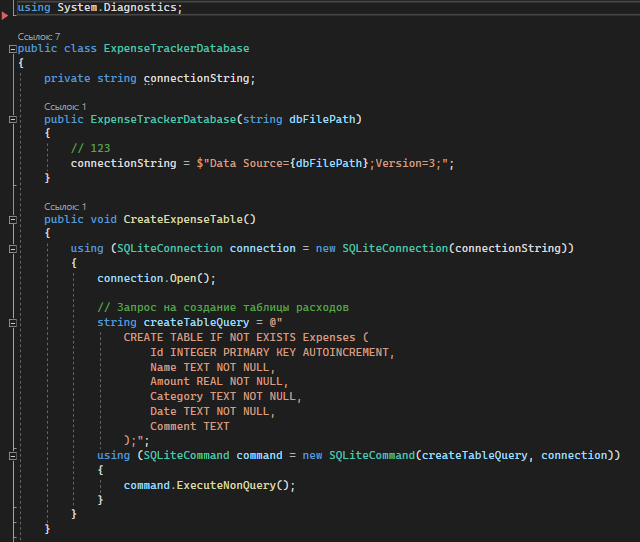


Рисунок 4 - Создание таблицы

На рисунке 5 отображена функция в классе, которая добавляет новый ряд в БД, т.е. новую запись расходов. Она принимает в себя все атрибуты таблицы, и добавляет их.

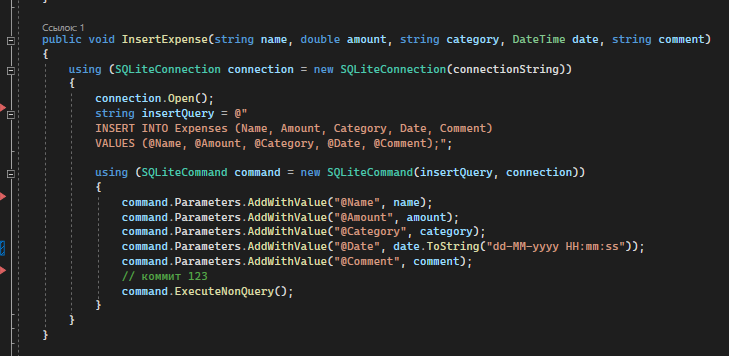


Рисунок 5 - Добавление записи

Для поиска по фильтрам, создана функция GetDataTableWithFilters, которая возвращает DataTable, для будущего отображения этих данных в элементе-таблице DataGridView.

В старте функции, создается запрос. Он создается в зависимости от переменных. Например, если название пусто – оно не учитывается при запросе, и так далее. Создание запроса показано на рисунке 6.

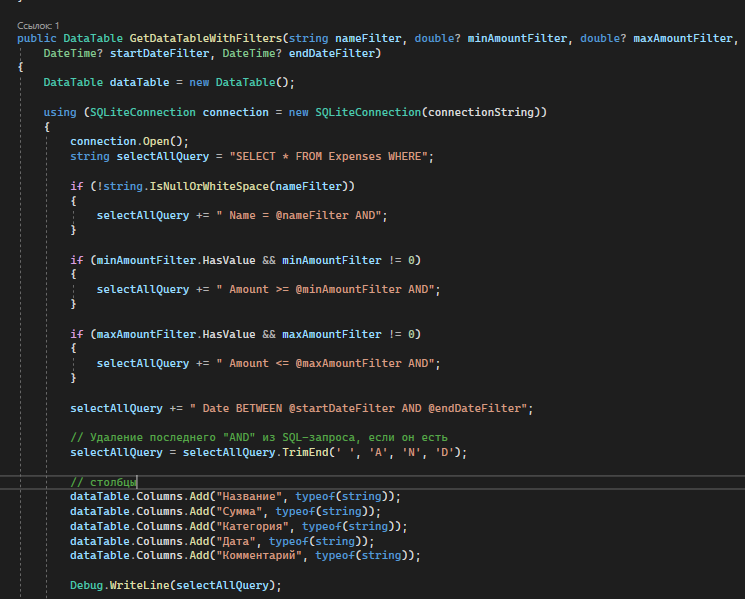


Рисунок 6 - Формирование запроса по фильтрам

Далее, подставляются параметры в зависимости от фильтра, в переменные запроса. На рисунке 7 показано подставление параметров и само выполнение запроса, возвращающее DataTable.

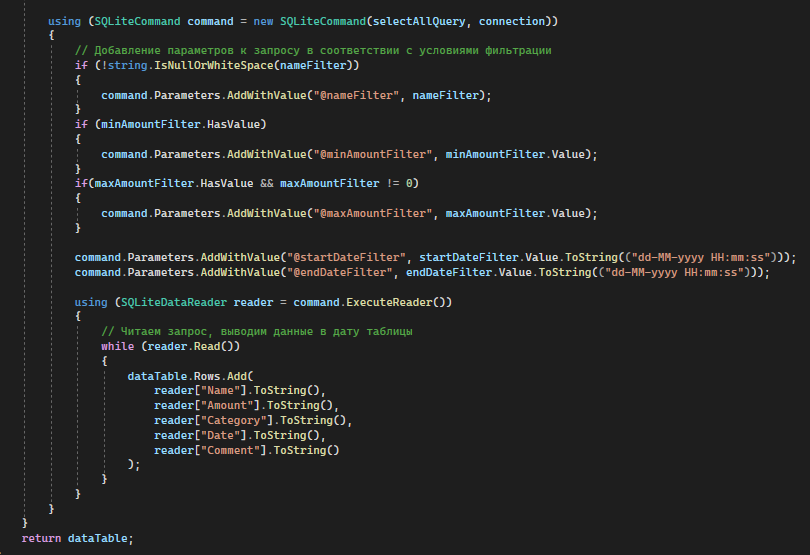


Рисунок 7 - Выполнение запроса

Для получения последних операций расходов, используется функция GetLastNExpenses. Она принимает n-е количество записей, которые необходимо посмотреть. После создания запроса и его выполнения, при чтении данные добавляются в список в виде “название – сумма”. В конце чтения функция возвращает список n-ых расходов. Функция показана на рисунке 8.



Рисунок 8 - Получение n расходов

Для получения суммы, в виде расходов за последние 30 дней, используется функция CalculateExpensesLast30Days. Она получает текущую дату, а также стартовую дату, равную текущей дате – 30 дней. После чего выполняется SQL-запрос на чтение данных и эти данные суммируются к переменной totalExpenses. После чего функция возвращает данную переменную. Функция показана на рисунке 9.

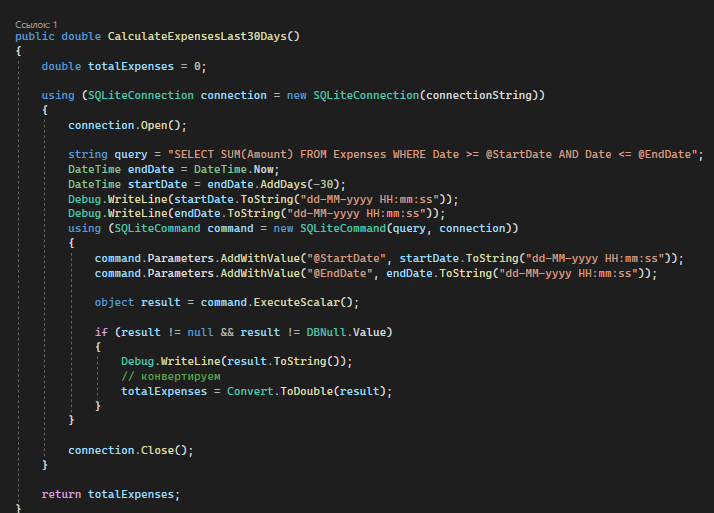


Рисунок 9 - Функция для подсчета расходов за 30 дней

После чего, все эти функции используются в классах форм. Функция, указанная на рисунке 9 используется в основной форме с данными. Также, там используется функция для подсчета последних n расходов. Класс формы показан на рисунке 10.



Рисунок 10 - Основная форма

В старте формы создается класс с БД, после чего он передается в другие формы. Для вывода информации по последним операциям, создаются новые Label, количество которых равно количеству элементов списка прочитанных операций.

При добавлении расхода в форме AddOperationForm, используется кнопка Добавить. При нажатии, она проверяет все текста, особенно текст с суммой. Если текст суммы содержит в себе символы букв, какие либо знаки, ничего не добавится.

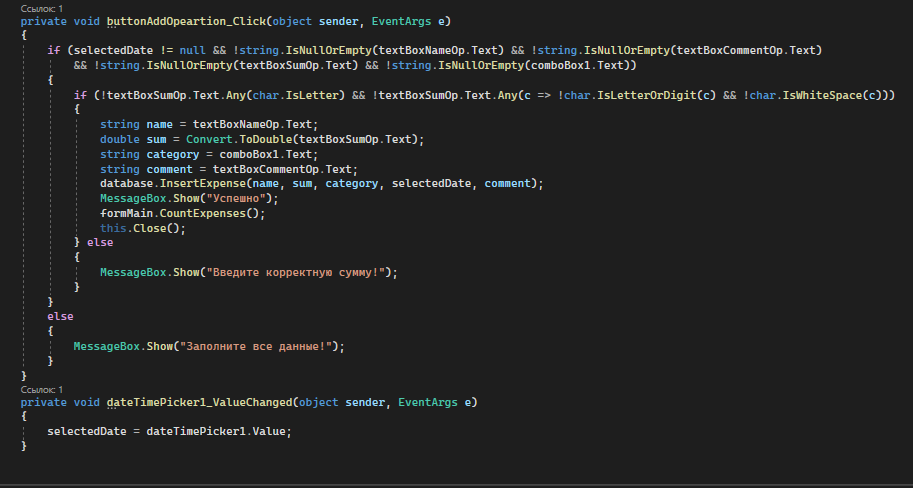


Рисунок 11 - Добавление операции в классе AddOperationForm

На рисунке 12 изображен класс SearchOperationForm. Данный класс в старте загружает столбцы в таблицу DataGridView. А нажатие на кнопку Найти выполняет поиск в классе с БД по фильтрам.

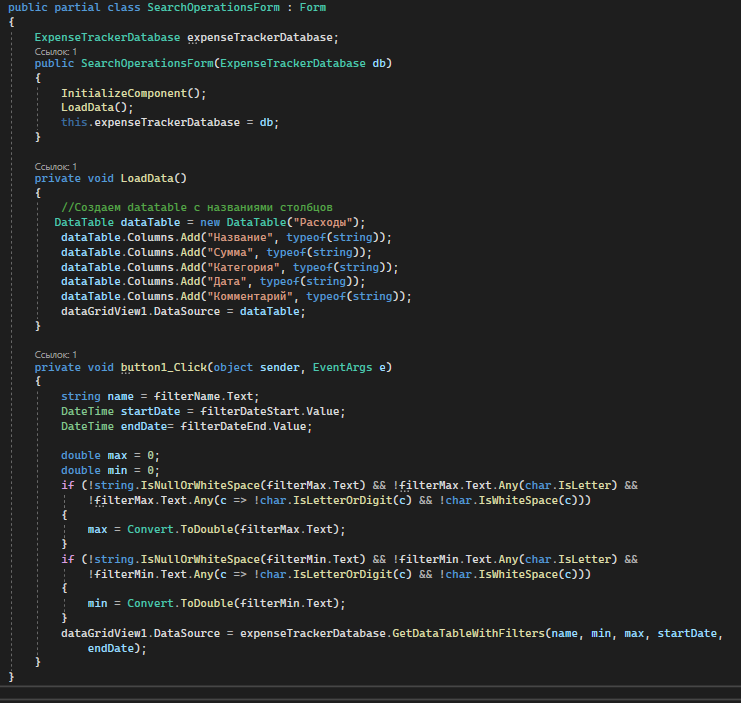


Рисунок 12 - Класс формы SearchOperationsForm