**Министерство науки и высшего образования Российской Федерации**

ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ АВТОНОМНОЕ ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ

**НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ УНИВЕРСИТЕТ ИТМО**

**ITMO University**

**Отчет**

**По дисциплине** Программирование

**Тема работы** Разработка программы для контроля собственных денежных средств с графическим интерфейсом

**Обучающийся** Новиков Николай Викторович

**Факультет** факультет инфокоммуникационных технологий

**Группа** К3123

**Направление подготовки** 11.03.02 Инфокоммуникационные технологии и системы связи

**Образовательная программа** Программирование в инфокоммуникационных системах

**Обучающийся**  Новиков Н. В.

(дата) (подпись) (Ф.И.О.)

**Руководитель**  Казанова П.П.

(дата) (подпись) (Ф.И.О.)

# СОДЕРЖАНИЕ

[СОДЕРЖАНИЕ 2](#_Toc158670081)

[ВВЕДЕНИЕ 3](#_Toc158670082)

[1. Создание проекта 4](#_Toc158670083)

[1.1 Файловая структура проекта и используемые модули 4](#_Toc158670084)

[1.2 Подключение модулей и файлов 4](#_Toc158670085)

[1.3 Создание дизайна системы 5](#_Toc158670086)

[2. Функционал программы 6](#_Toc158670087)

[2.1 Добавление товара в корзину. 6](#_Toc158670088)

[2.2 Удаление продукта 7](#_Toc158670089)

[2.3 Фильтрация 7](#_Toc158670090)

[3. Работа с базой данных 7](#_Toc158670091)

[3.1 Создание и подключение к базе данных 7](#_Toc158670092)

[3.2 Добавление и удаление продукта на уровне БД 8](#_Toc158670093)

[4. Диаграммы и схемы 8](#_Toc158670094)

[4.1 Диаграмма классов 8](#_Toc158670095)

[4.2 Схема базы данных 9](#_Toc158670096)

[ЗАКЛЮЧЕНИЕ 10](#_Toc158670097)

# ВВЕДЕНИЕ

Целью проекта было создание программы для контроля собственных денежных средств с графическим интерфейсом. Программа должна была содержать в себе такие функции как: Добавление трат, просмотр общего баланса, фильтрация по категории и цене.

Предварительно был проведен анализ предметной области и требований. Выяснилось, что наиболее схожим ПО является корзина в заметках, которой пользуются, в большинстве своем, люди, не понимающие программирования. Поэтому приоритетной целью всего проекта было создание простого и понятного приложения для корзины.

# 1. Создание проекта

## 1.1 Файловая структура проекта и используемые модули

Файловая структура проекта (Рисунок 1) представляет из себя несколько исполняемых файлов питона, главным из который является main.py, вместе с папкой для иконок, базой данных и файлами дизайна интерфейса

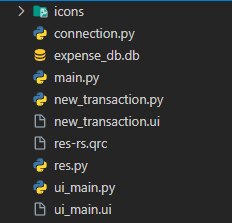


Рисунок - Файловая структура проекта

## 1.2 Подключение модулей и файлов

Так как по сравнению с консольным приложением, в графическом приложении зависимостей и библиотек будет куда больше. В данной программе вся графическая часть была построена на основе библиотеки PySide6, которая имеет крайне большой спектр возможностей. В main.py, помимо PySide6, были подключены также и другие библиотеки (Рисунок 2), такие как sys для работы с оболочкой приложения. В качестве модулей выступил файл подключения к базе данных.

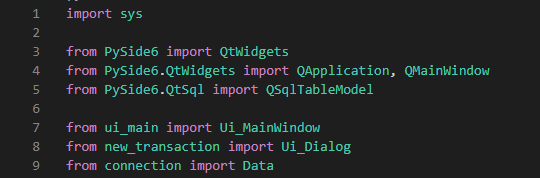


Рисунок - Модули основного файла

## 1.3 Создание дизайна системы

Перед началом разработки было необходимо создать статичный дизайн программы для дальнейшей работы. В качестве графического редактора PySide6 имеет инструмент QtDesigner. В нем был создан, как главный экран приложения (Рисунок 3), так и модальное окно, в котором будет происходить добавление новых покупок (Рисунок 4).

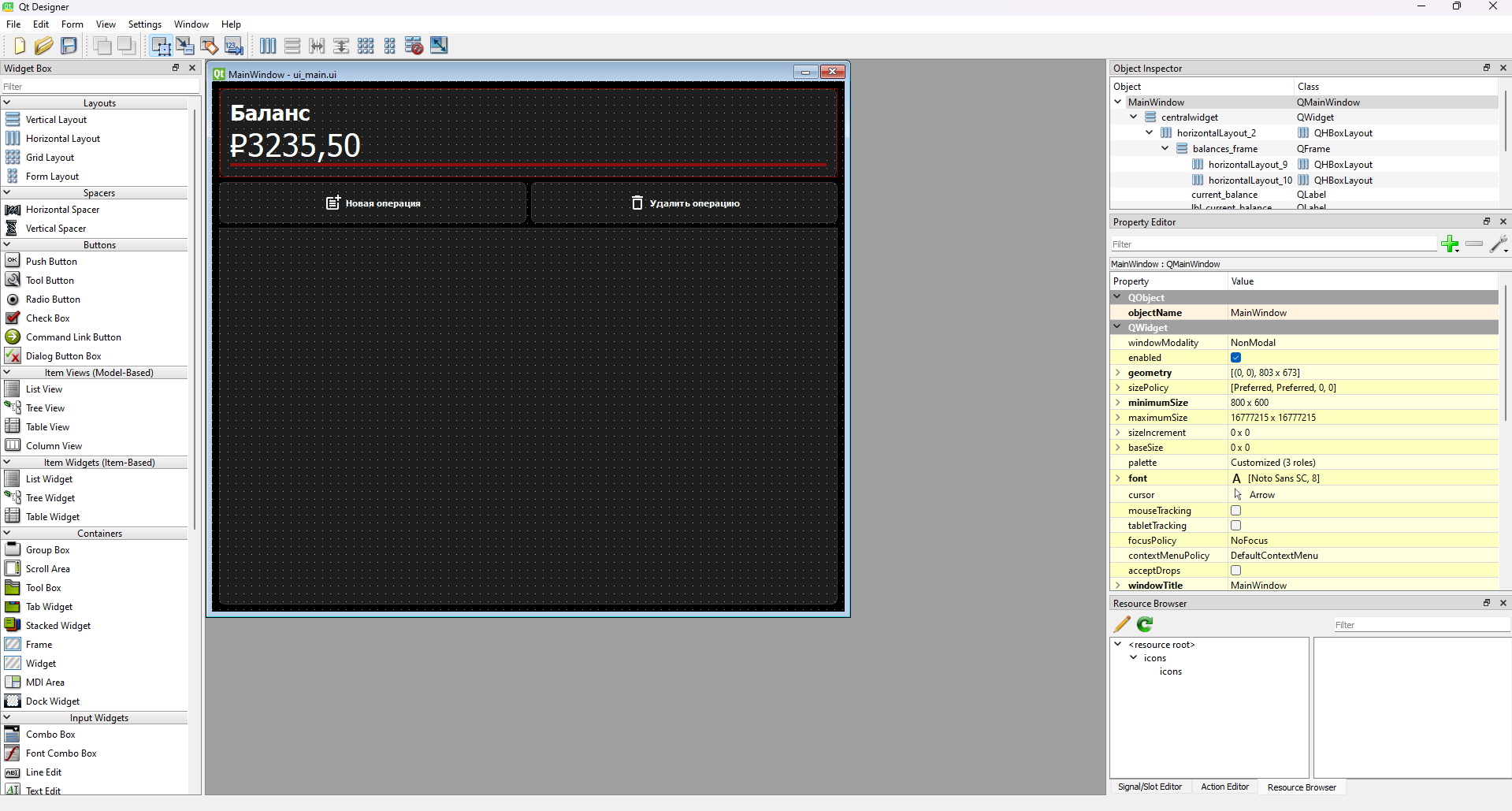


Рисунок - Основной экран

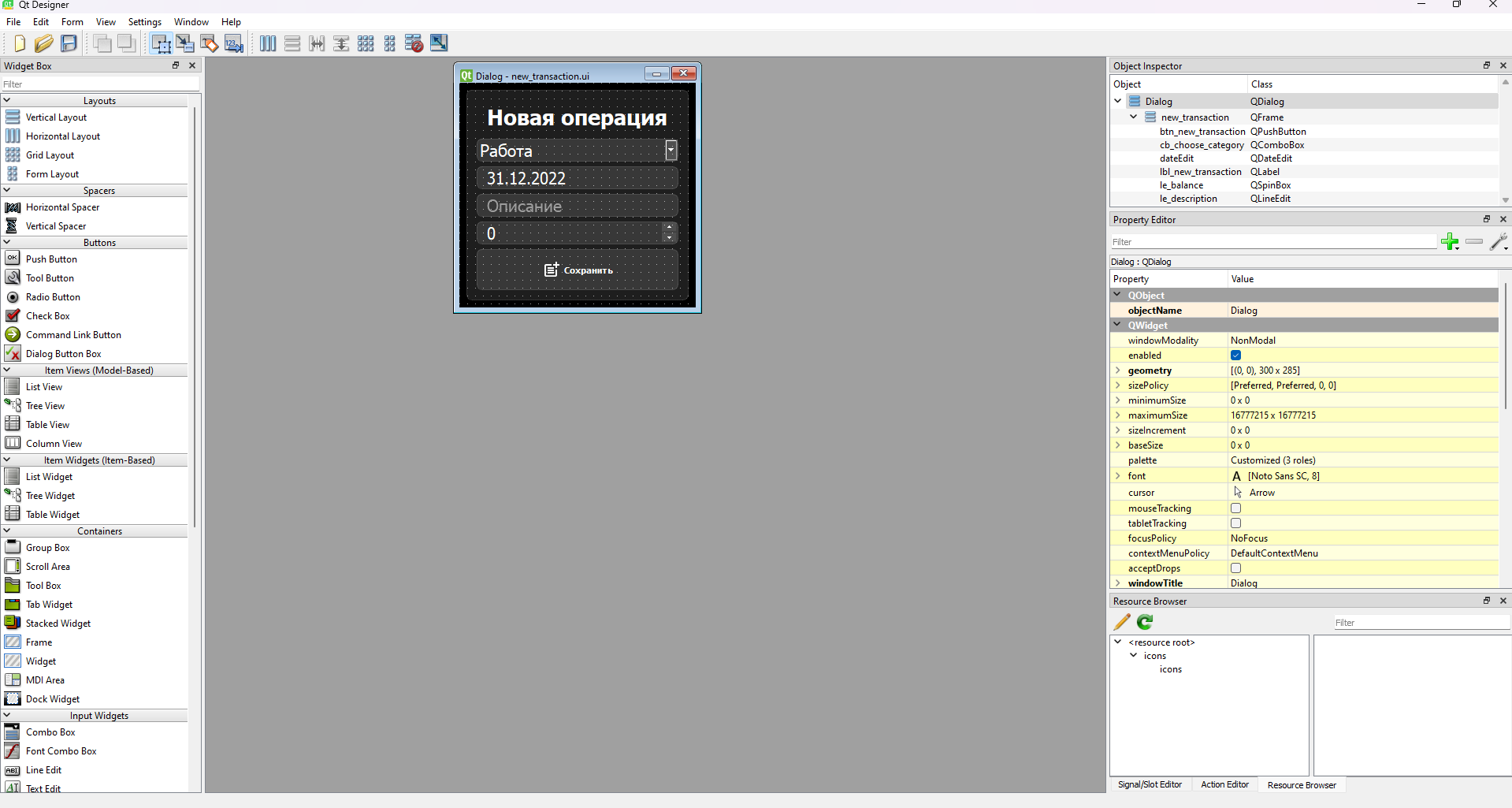


Рисунок - модальное окно

В дальнейшем эти файлы, с помощью PySide6, были преобразованы в файлы с расширением .py (new\_transaction.py и ui\_main.py).

# 2. Функционал программы

## 2.1 Добавление товара в корзину.

Одной из основных функций программы является добавление продукта в корзину. Оно реализовано следующим образом (Рисунок 5).

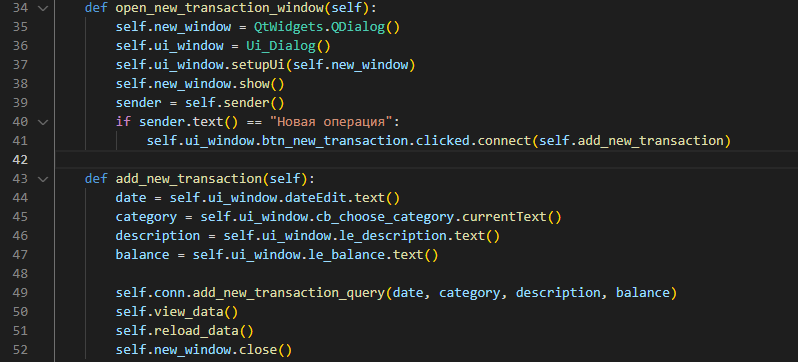


Рисунок - Добавление нового товара

При нажатии на кнопку “Новая операция” вызывается открытие модального окна, в котором пользователь заполняет такие поля как: Категория товара, Дата приобретения, Описание товара и его стоимость. В стоимость пользователь не сможет ввести ничего, кроме числа, так как в QtDesigner для чисел есть специальное поле, которое ограничивает вводимые данные на уровне моделирования системы. После чего следуют стандартные функции обновления экрана и показа таблицы (Рисунок 6). Баланс также обновляется автоматически после каждого действия пользователя.

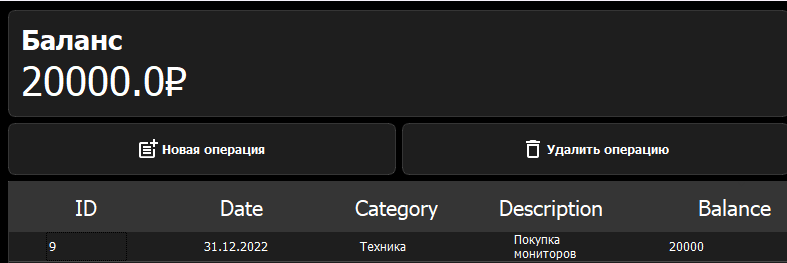


Рисунок - Экран с введенными данными

## 2.2 Удаление продукта

Удаление продукта происходит при клике пользователя по нужной кнопке. Функция ищет в строке id данного продукта и вызывает функцию удаления товара на уровне базы данных.

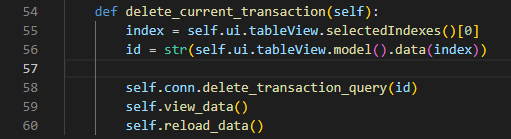


Рисунок - Функция удаления продукта

## 2.3 Фильтрация

В программе было необходимо также реализовать фильтрацию товара по категории и цене. Благодаря PySide6 данная задача стала крайне простой в реализации. При выводе полей к ним присоединялась функция сортировки по нужному полю. Условно нажав на кнопку “Balance”, можно отсортировать продукты по балансу в обе стороны, так же работает и для всех кнопок.

# 3. Работа с базой данных

## 3.1 Создание и подключение к базе данных

Все товары, которые были добавлены пользователем, должны где-то храниться. PySide6 предоставляет свой модуль для работы с базами данных. Алгоритм создания и подключения к базе данных представлен на Рисунке 8

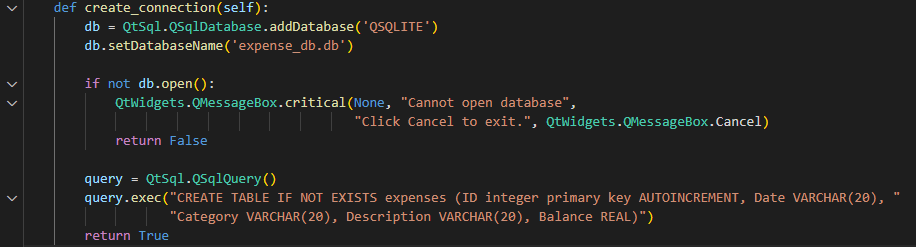


Рисунок - Создание и подключение к БД

Здесь создается база данных с именем (expence\_db.db), а после всевозможных проверок создается таблица с названием expenses со всеми полями из модального окна.

## 3.2 Добавление и удаление продукта на уровне БД

Как было выше сказано, то функция удаления и добавления хоть и вызывается в главном файле, но все вычисления и запросы выполняются непосредственно в файле с БД (Рисунок 9).

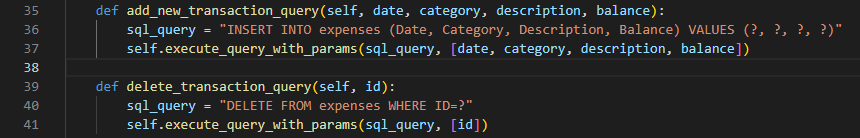


Рисунок - Удаление и добавление товара

В функции передаются необходимые данные удаленного/добавленного товара и после на их основе составляется SQL запрос к базе данных.

# 4. Диаграммы и схемы

## 4.1 Диаграмма классов

В программе существует два класса – ExpenceTracker (основной класс всей программы) и Data (отвечает за взаимодействие с БД). Они являются независимыми друг от друга. Диаграмма классов представлена на Рисунке 10

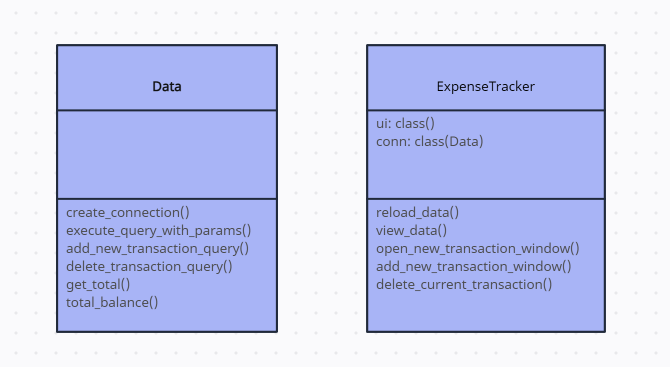


Рисунок - диаграмма классов

## 4.2 Схема базы данных

База данных, используемая проекта, имеет в себе лишь одну таблицу, хранящую в себе данные о товаре, так как в данном приложении больше ничего и не надо. Схема базы данных представлена на Рисунке 11

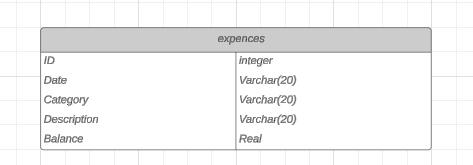


Рисунок - Схема базы данных

# ЗАКЛЮЧЕНИЕ

С помощью нескольких модулей python`а удалось создать программу для контроля собственных денежных средств с графическим интерфейсом. Были реализованы такие функции как: добавление продукта в корзину, просмотр всего записанного, просмотр по категории, распределение по стоимости, удаление добавленных записей.