

## Задание 1

Разработать приложение для планирования расходов.

Функции приложения:

1. Просмотр/добавление/изменение/удаление расходов: наименование, категория, сумма в рублях с точностью до копеек, дата.
2. Отображение общего кол-ва записей и общей суммы расходов.
3. Возможность фильтровать расходы по датам.

Категории:

1. Продукты.
2. Коммунальные услуги.
3. Транспорт.
4. Другое.

Требования:

1. Использовать WPF.
2. Список расходов реализовать в виде таблицы.
3. При использовании сторонних библиотек подготовить инструкцию по их установке.
4. Результат сохранить в архив \*.zip или репозиторий git на GitHub или GitLab.

Будет плюсом:

1. Хранение записей в локальной БД, напр. SDF или SQLite.
2. Отображение общего кол-ва записей и общей суммы расходов внизу самой таблицы.

## Задание 2

Разработать приложение для ведения коллекции.

Функции приложения:

1. Просмотр/добавление/изменение/удаление категорий и входящих в них элементов коллекции. Категории имеют наименование, а элементы - наименование и дату добавления.
2. Сортировка по наименованию.

Пример коллекции:

1. Монеты:
  - a. СССР:
    - i. 1 копейка, 01.01.2020.
    - ii. 1 рубль, 02.01.2020.
  - b. Россия:
    - i. 1 копейка, 03.01.2020.
    - ii. 1 рубль, 04.01.2020.

2. Купюры:

а. СССР:

- i. 1 рубль, 05.01.2020.
- ii. 3 рубля, 06.01.2020.

б. Россия:

- i. 10 рублей, 07.01.2020.
- ii. 50 рублей, 08.01.2020.

Требования:

- 1. Использовать WPF.
- 2. Коллекцию реализовать в виде дерева/Tree List.
- 3. Кол-во уровней вложенности категорий может быть произвольным, достаточно поддержки до 5 уровней вложенности.
- 4. При использовании сторонних библиотек подготовить инструкцию по их установке.
- 5. Результат сохранить в архив \*.zip или репозиторий git на GitHub или GitLab.

Будет плюсом:

- 1. Хранение записей в локальной БД, напр. SDF или SQLite.
- 2. Визуальное отличие категорий от элементов, напр.: иконка, цвет.