Министерство образования Московской области

Государственное образовательное учреждение высшего образования Московской области

«Государственный гуманитарно-технологический университет»

**Ликино-Дулевский политехнический колледж – филиал ГГТУ**

**КУРСОВОЙ ПРОЕКТ**

Разработка программного модуля «ПО для планирования личного бюджета и контроля финансов. Мои деньги.»

МДК 01.01 «Разработка программных модулей»

**Выполнила:**

Васина Дарья Александровна\_\_\_\_\_\_\_\_ \_\_\_

студентка группы ИСП.20А\_\_\_\_\_\_\_\_\_\_\_\_\_

09.02.07 Информационные системы и программирование)\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

очной формы обучения\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

**Руководитель:**

Кузьмина Елена Евгеньевна­­­­­­­­­­­­­­­­­­\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Оценка

\_\_\_\_\_\_\_\_\_/\_\_\_­­­­­­\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_/

Подпись руководителя

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Ликино-Дулево

2023 год

Оглавление

**[Введение 2](#_Toc25032)**

[Visual Studio Code 5](#_Toc3768)

[3.1 Разработка программного модуля 5](#_Toc21061)

[3.2 Разработка программного модуля 7](#_Toc25714)

[3.3 Текст программы 8](#_Toc9117)

[4.1 Выполнение программного модуля 8](#_Toc8313)

[4.2 Сообщения пользователю 28](#_Toc26742)

**[5.Тестирование программного модуля 30](#_Toc19389)**

[1.1 Модульное тестирование 30](#_Toc19297)

[1.2 Тестовые сценарии 32](#_Toc32378)

**[6. Отладка программного модуля 33](#_Toc974)**

**[Заключение 33](#_Toc14820)**

**Введение**

Личный бюджет - это четкий структурированный план ваших финансов: доходов, расходов, сбережений, долгов и т.д., который должен содержать абсолютно все статьи денежных поступлений и трат. Для чего же нужно составлять этот план? Расписав и структурировав все свои финансы, вам будет намного проще распределить бюджет, спланировать траты, посчитать сколько денег останется и как вы сможете грамотно ими распорядиться (например, инвестировать некоторую сумму или открыть вклад). Выделяют три основных вида личного бюджета: экономный, сбалансированный и расточительный. Они различаются по соотношению процента доходов и расходов. Каждый человек выбирает для себя тот вид, который подходит именно ему, ведь у всех людей разные потребности, привычки, разный характер и другие исходные данные. Но, в любом случае, вести свой бюджет – очень полезное дело. В финансовом плане учитываются цели, сроки планирования (от полугода до нескольких лет), анализируются текущие доходы и расходы, активы и пассивы — все, как в бизнесе, только на уровне одного человека. Учитываются возможности для снижения расходов и увеличения доходной части.

1. **Назначение разработки -** указывается функциональное и эксплуатационное назначение программы, определяется задача, для решения которой разрабатывается программа. Эта программа предназначена для удобст пользования семейного бюджета, контролировать доходы и расходы.

Приложение должно выполнять следующие функции:

1. Добавление доходов, членов семьи, расходы.
2. Редактирование доходов, членов семьи, расходы.
3. Удаление доходов, членов семьи, расходы.
4. Сортировка по имени членов семьи.
5. Поиск по типу расходов.
6. Поиск по типу доходов.
7. Сброс выполняемых операций со списками
8. Вывод в Excel
9. **Требования к программному модулю**
   1. **Требования к функциональным характеристикам**

Программа должна обеспечивать взаимодействие с пользователем, с помощью графического пользовательского интерфейса приложения, который открывается через \*.ехе файл. Программа должна обеспечивать удобный импорт данных в список из базы данных.

Входные данные программы организованы в базе данных Microsoft SQL Server. К этим данным относится не только информация, вводимая клавиатурой, но и фотографии и изображения, хранящиеся в базе в виде двоичных данных переменной длины. При запуске приложения, данные (если они уже были добавлены в базу) буду автоматически загружаться, и можно будет сразу приступать к работе и не тратить драгоценное время на повторное заполнение данных.

Выходные данные организованы в форме списка в соответствии со столбцами в базе данных. Также присутствует возможность печать документа (договора) с подставляемыми данными из списка «Исполнители» в программе Excel.

* 1. **Требования к составу и параметрам технических средств**

Таблица 1. «Состав технических средств и их характеристики»

|  |  |
| --- | --- |
| Процессор | Ядро: Alder Lake;  Частота: 2.5 ГГц и 4.4 ГГц в режиме Turbo;  Сокет: LGA 1700;  Число ядер: 6, потоков 12;  Тепловыделение: 65 Вт;  Технологический процесс: 10 нм; |
| Оперативная Память | Суммарный объем памяти: 6 ГБ,  Тактовая частота: 2133 МГц  Пропускная способность: 28800 Мб/сек |
| Разрешение экрана | 1920 × 1080 |
| Размер экрана | 15 дюймов |
| Устройства ввода | Мышь, клавиатура |
| Операционная система | Windows 10 |
| Устройства вывода | Принтер |
| Жёсткий диск | Объем накопителя: 800 ГБ  Буферная память: 64 Мб  Потребляемая мощность: 5,6 Вт |

* 1. **Требования к информационной и программной совместимости**

Программа которая нам понадобится это:

**Visual Studio Code**

Это редактор кода для разных языков программирования. Он относительно немного весит, гибкий и удобный. В нем можно писать, форматировать и редактировать код на разных языках. VS Code не стоит путать с Visual Studio — это IDE, очень мощная и масштабная, но одновременно с этим тяжеловесная. Названия похожи, потому что у обоих средств разработки один и тот же создатель, но продукты разные по своей сути. VS Code меньше весит, проще в освоении и подходит в том числе для начинающих разработчиков. Редактор кода существует для всех популярных операционных систем: Windows, Linux и macOS. Он бесплатный, в отличие от большинства версий «старшего брата» Visual Studio. С VS Code работают программисты на разных языках. Например, им активно пользуются веб-разработчики, пишущие на HTML/CSS, JavaScript, PHP. Но редактор поддерживает намного большее количество языков: Python, Go, Ruby, C#, TypeScript и так далее. Он работает и с расширениями и фреймворками для популярных языков — например, с React JS и Vue.js, с языками стилей SCSS и LESS, которые дополняют CSS. В теории пользоваться VS Code может разработчик практически на любом из современных языков.

**SQL Server Management Studio (SSMS)** — утилита из Microsoft SQL Server 2005 и более поздних версий для конфигурирования, управления и администрирования всех компонентов Microsoft SQL Server. Утилита включает скриптовый редактор и графическую программу, которая работает с объектами и настройками сервера.

1. **Руководство программиста**
   1. **Разработка программного модуля**

Разработка базы данных началась сформирования следующих диаграмм для проектирования схемы базы данных данной предметной области:

1. Диаграмма прецедентов Use Case:

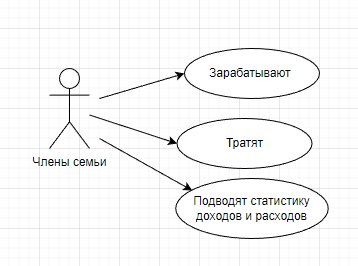


Рис. 1 «Диаграмма прецедентов Use Case»

1. ER – диаграмма («Сущность-связь»):

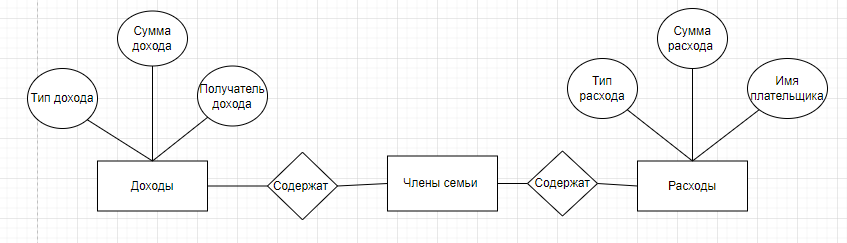


Рис. 2 «ER – диаграмма («Сущность-связь»)»

1. После проектирования диаграмм, была сформирована база данных:

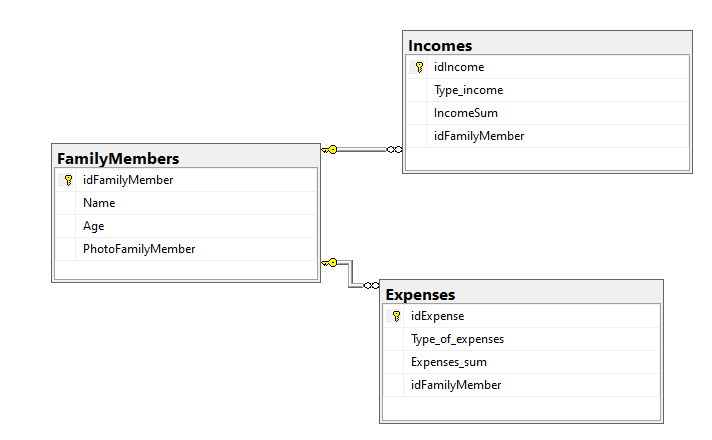


Рис. 3 «Схема данных»

Описание сущностей (таблиц базы данных):

* FamilyMembers – хранит данные об членах семьи.
* Incomes – хранит данные о доходах семьи.
* Expenses– хранит данные о расходах семьи.

Описание атрибутов сущностей (таблиц базы данных):

Таблица 2. «Таблица FamilyMembers»

|  |  |  |  |
| --- | --- | --- | --- |
| Поле | Тип | Атрибуты | Описание |
| IDFamilyMembers | INT | PK NN AI | Уникальный номер исполнителя |
| Name | NVARCHAR(50) |  | Имя члена семьи |
| Age | INT |  | Возраст члена семьи |
| PhotoFamilyMember | IMAGE |  | Фотография (картинка) члена семьи |

Таблица 3. «Таблица Incomes»

|  |  |  |  |
| --- | --- | --- | --- |
| Поле | Тип | Атрибуты | Описание |
| IDIncomes | INT | PK NN AI | Уникальный номер автора |
| Type\_income | NVARCHAR(50) |  | Тип дохода |
| IncomeSum | INT |  | Сумма документа |
| idFamilyMember | INT |  | Семейный бюджет |

Таблица 4. «Таблица Expenses»

|  |  |  |  |
| --- | --- | --- | --- |
| Поле | Тип | Атрибуты | Описание |
| IDExpenses | INT | PK NN AI | Уникальный номер альбома |
| Type\_of\_expenses | NVARCHAR(50) |  | Тип расходов |
| Expenses\_sum | INT |  | Сумма расходов |
| idFamilyMember | INT |  | Семейный бюджет |

## **3.2 Разработка программного модуля**

При создании приложения используется:

1. Языки программирования – C#, SQL.
2. Программное обеспечение:

* Microsoft Visual Studio – интегрированная среда разработки, используемый для разработки компьютерных программ, веб-сайтов. веб-приложений, веб-сервисов и мобильных приложений.
* SQL Server Management Studio – программное приложение для управления базами данных (СУБД).
* Microsoft Office – комплекс программного обеспечения, включающий текстовый редактор, редактор таблиц, редактор презентаций и другие программные продукты.

Для разработки использовалась база данных Microsoft SQL Server и программа Visual Studio 2022. В данной программе был создан проект «Приложение WPF (.Net Framework)» на языке C#.

Для подключения базы данных использовалась технология Entities Framework.

Проект состоит из окна FrameWindow, класса ClassFrame, и подключенной базы данных «FamilyFinances». Содержит следующие страницы для добавления и редактирования информации в списках: PageAddExpenses, PageAddFamilyMembers, PageAddIncomes. Страницы для отображения данных: MainPage, PageExpenses, FamilyMembers, PageIncomes.

**3.3 Текст программы**

Приложение 1.

**4. Руководство пользователя**

## **Выполнение программного модуля**

Для работы с приложение пользователю необходимы на ПК следующие программы: Microsoft SQL Server Management Studio 18, Visual Studio 2022 и Microsoft Word.

Для выполнения работы настольного приложения, пользователю необходимо загрузить базу данных в SQL Server через скрипт или bak файл, при загрузке базы через скрипт пользователь должен назвать базу данных «FamilyFinaces». Следующим шагом является открытие двойным щелчком файла FamilyFinaces.sln, после загрузки откроется программа. Чтобы её запустить, необходимо нажать кнопку «Пуск» на панели инструментов.

При открытии приложения, пользователя приветствует главный экран с общей информацией о приложении. В нижней панели располагаются навигационные кнопки, переходящие на следующую страницу. При нажатии на кнопку, она будет выделяться белым цветом и сбоку слева будет синее выделение. Так пользователю будет понятно на какую именно страницу отображения данных он перешёл.

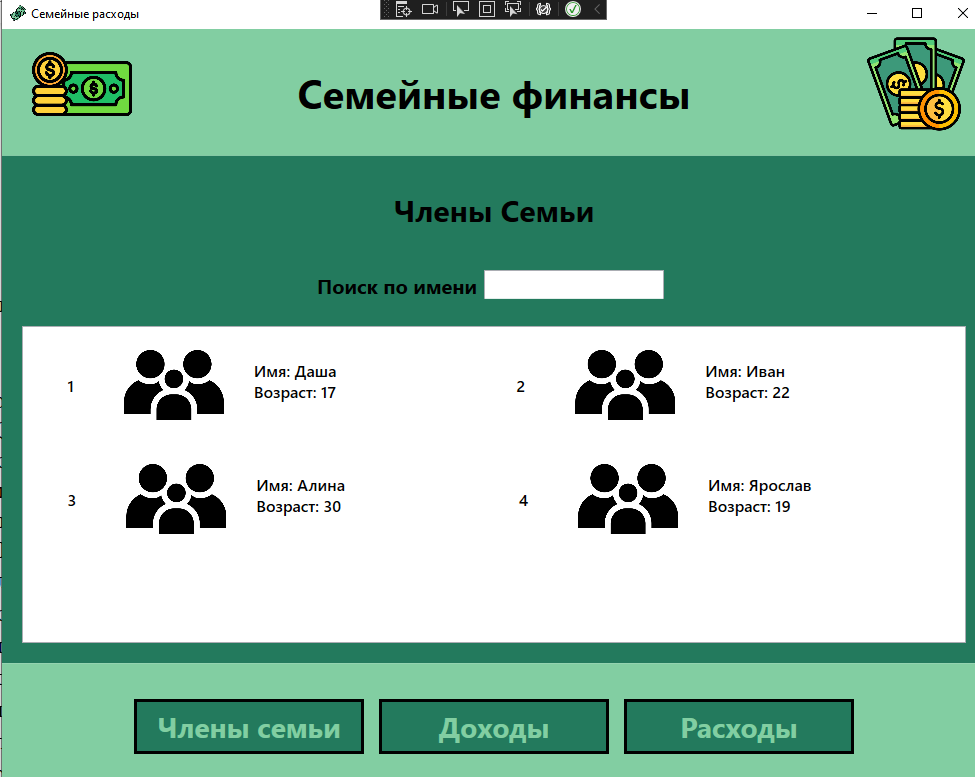


Рис. 4 «Главная страница»

Если пользователю потребовалось добавить данные, то для этого, в контекстном меню, на выбранном элементе из списка требуется нажать «Добавить». После этого пользователь окажется на странице добавления доходов, в которой можно добавить: тип дохода, сумма дохода и получатель дохода.



Рис. 5 «Страница члены семьи с вызовом добавления»

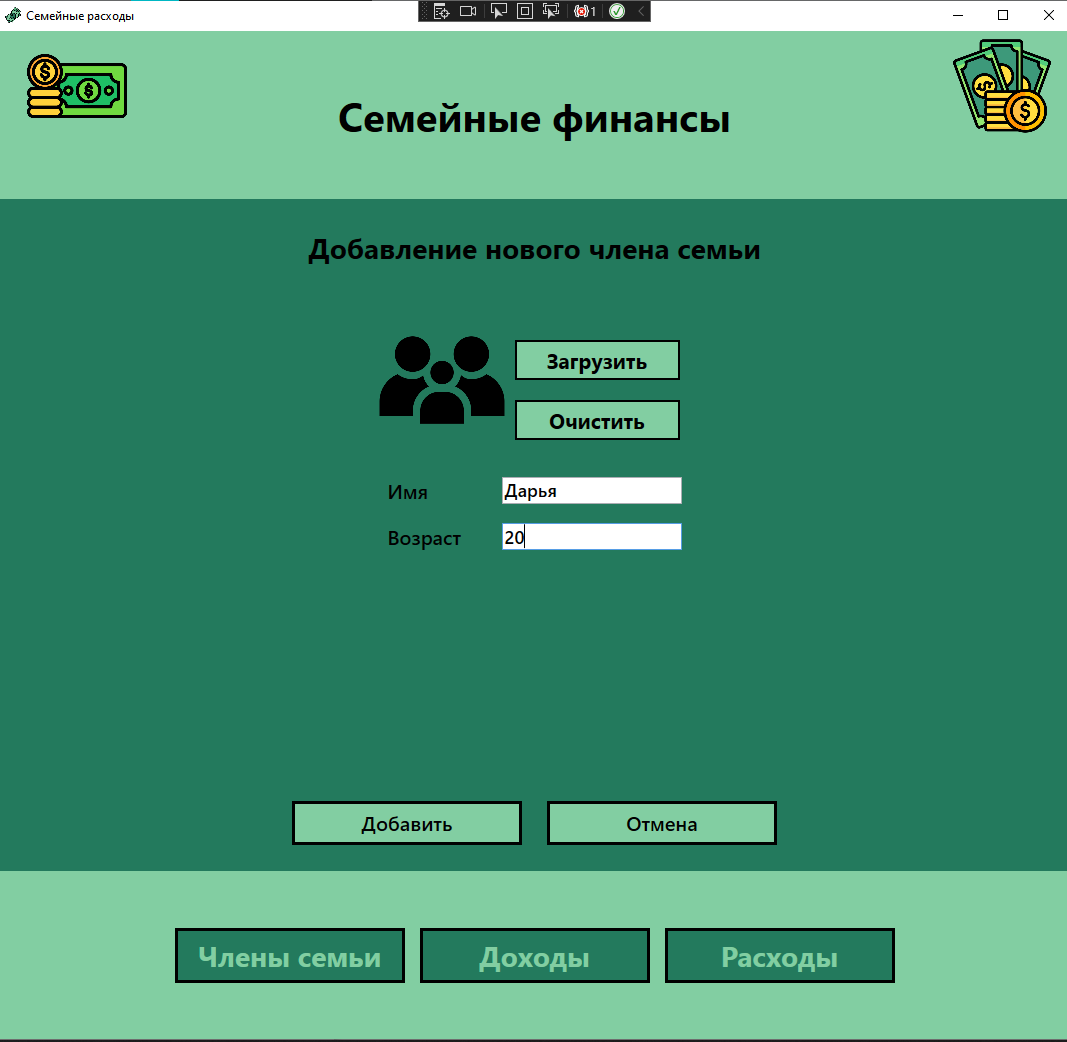


Рис.6 «Страница добавления нового члена семьи»

Если пользователю нужно изменить данные, то для этого, в контекстном меню, на выбранном элементе из списка требуется нажать «Редактировать». После этого пользователь окажется на странице редактирования доходов, в которой можно изменить: тип дохода, сумма дохода и получатель дохода.

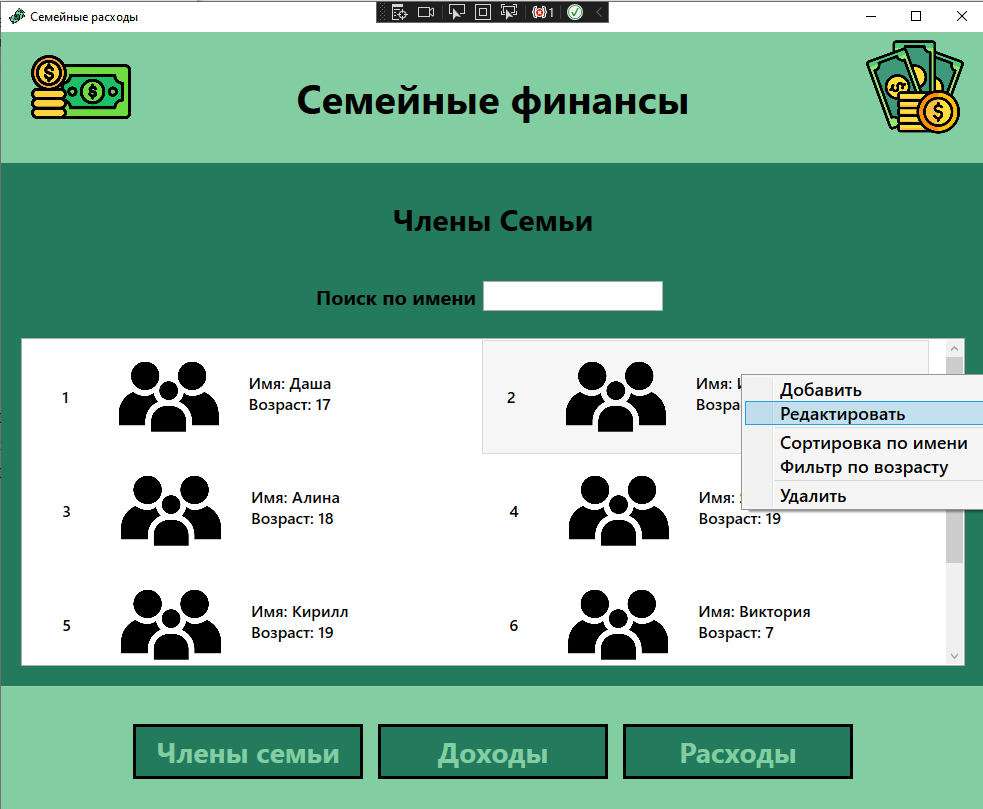


Рис.7 «Странциа члены семьи с вызовом редактирования»

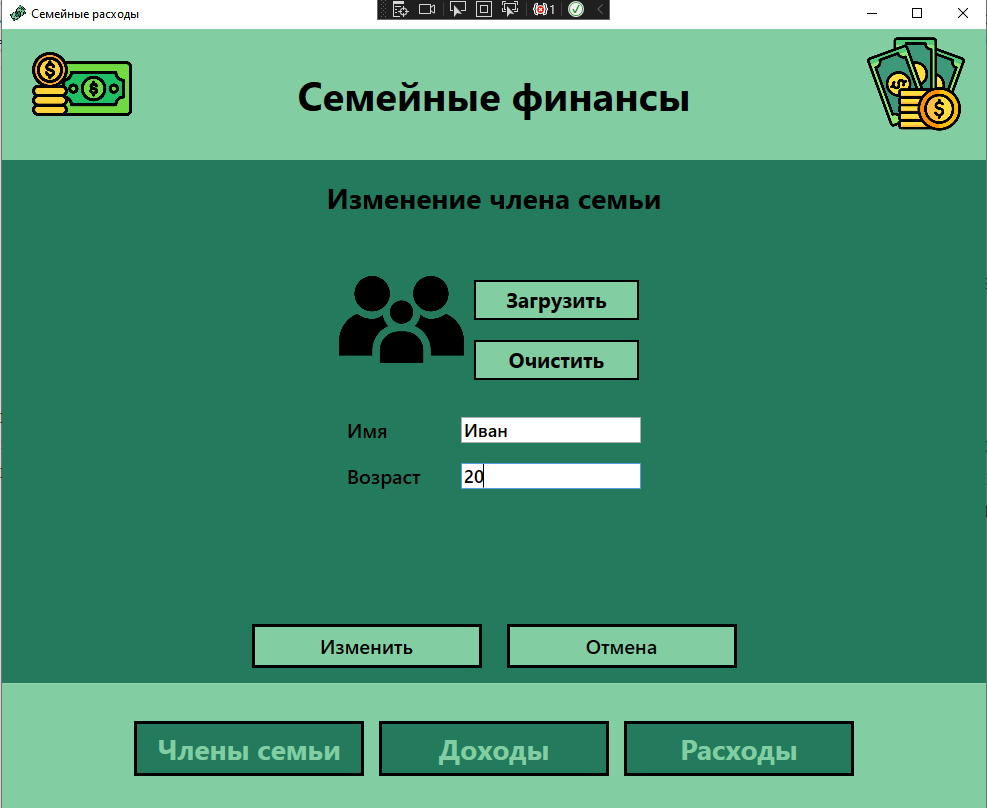


Рис.8 «Страница редактирования члена семьи»

Если следует отсортировать данные в алфавитном порядке или же наоборот, то для этого, в контекстном меню, на выбранном элементе из списка требуется навести курсор на пункт «Сортировка», и выбрать нужную сортировку. После этого пользователь увидит, что данные из списка отсортировались в соответствии с выбранной сортировкой. Если выбранная сортировка больше не нужна, то нужно снова навести курсор в контекстном меню на пункт «Сортировка», а затем нажать на «Сброс».

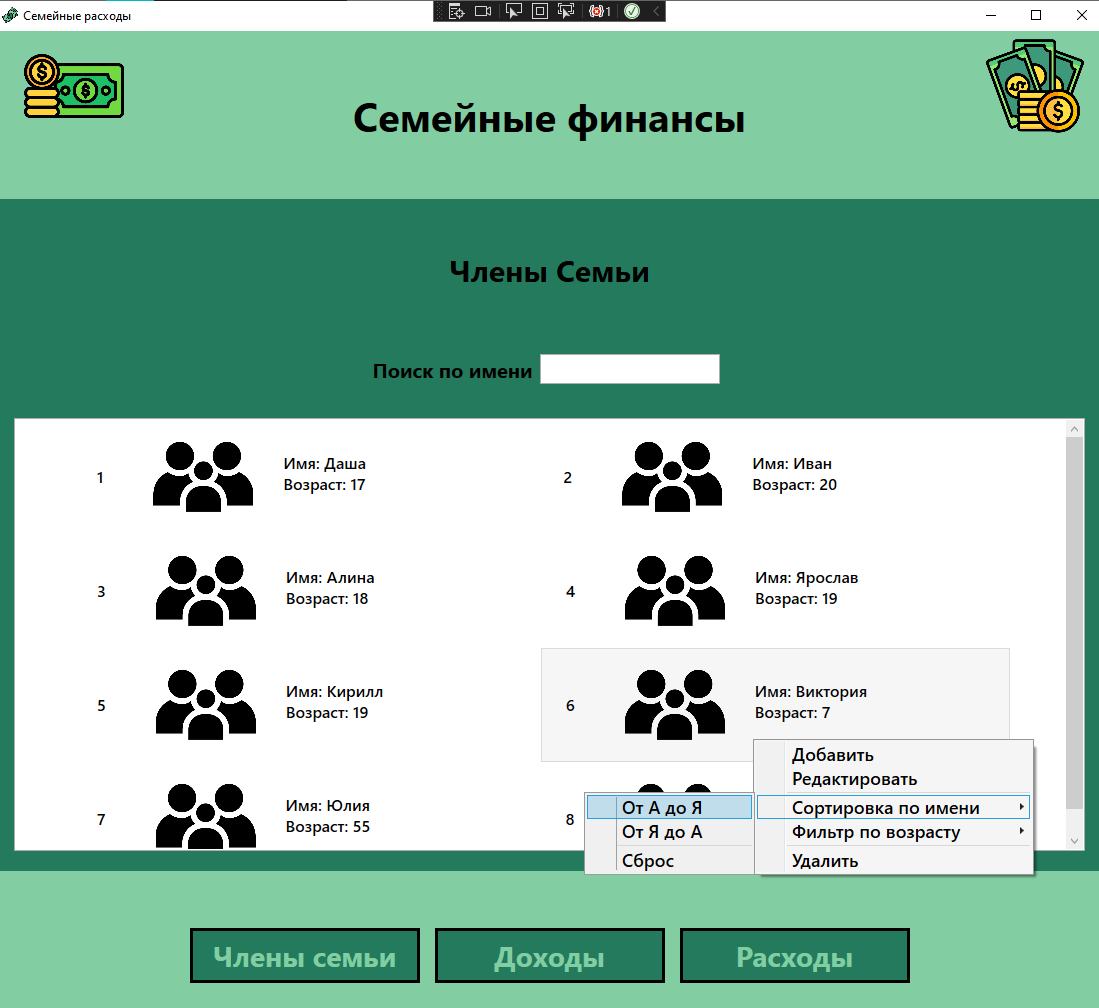


Рис.9 «Страница члены семьи с вызовом сортирования»

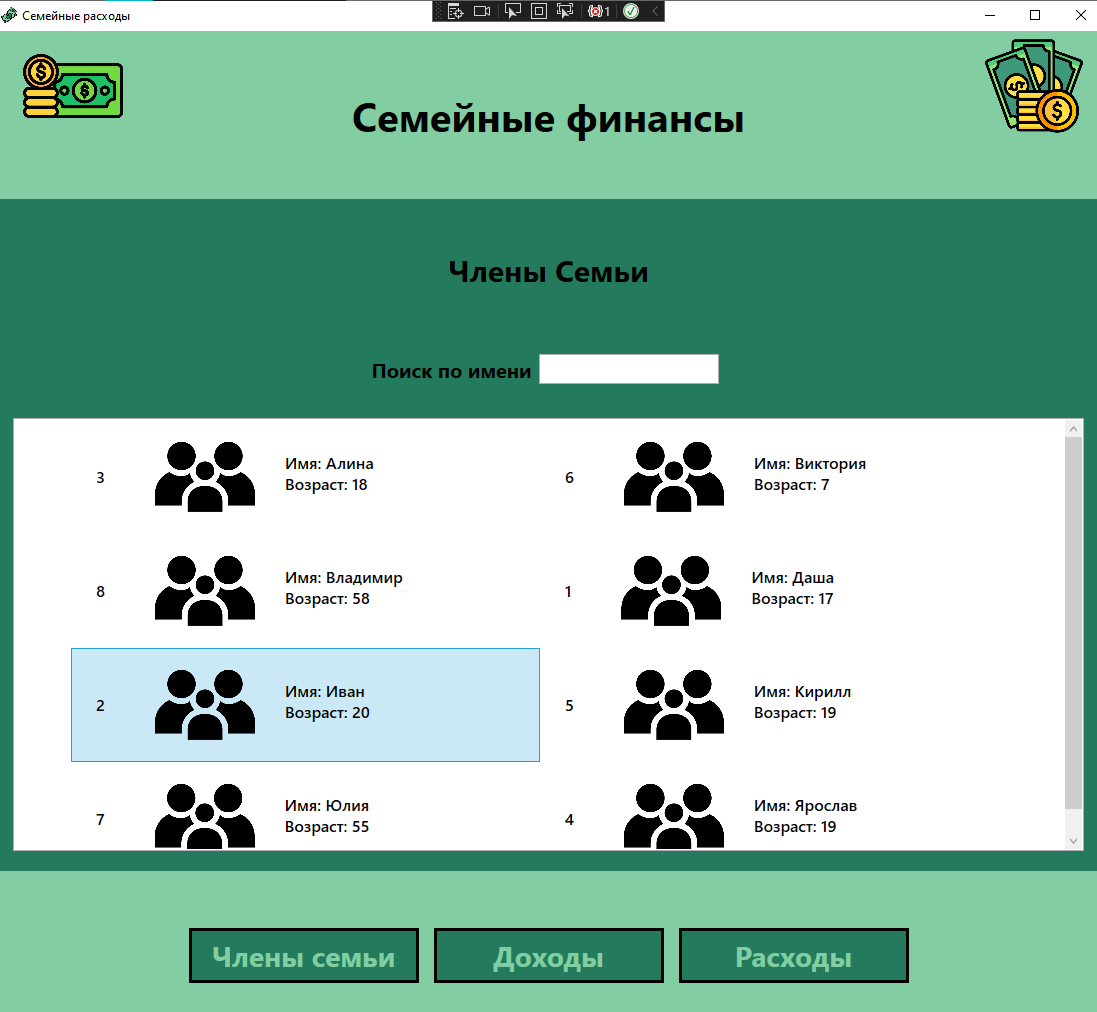


Рис.10 «Страница члены семьи, отсортированных в алфавитном порядке»

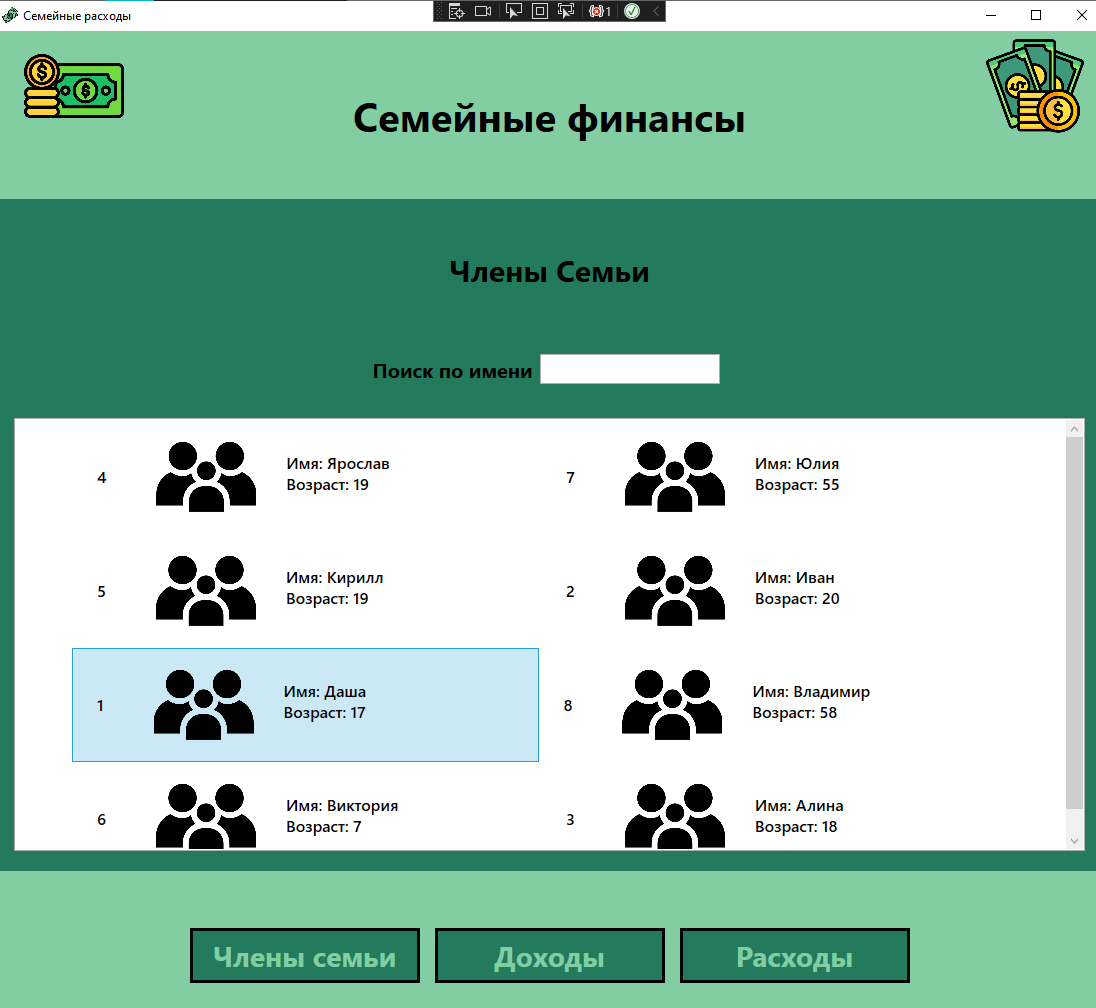


Рис.11 «Страница члены семьи, отсортированных в обратном алфавитном порядке»

Если следует отфильтровать данные младше 18 и старше 18, то для этого, в контекстном меню, на выбранном элементе из списка требуется навести курсор на пункт «Фильтр по возрасту», и выбрать нужную фильтрацию. После этого пользователь увидит, что данные из списка отфильтровались в соответствии с выбранной фильтровкой. Если выбранная фильтровка больше не нужна, то нужно снова навести курсор в контекстном меню на пункт «Фильтр по возрасту», а затем нажать на «Сброс».

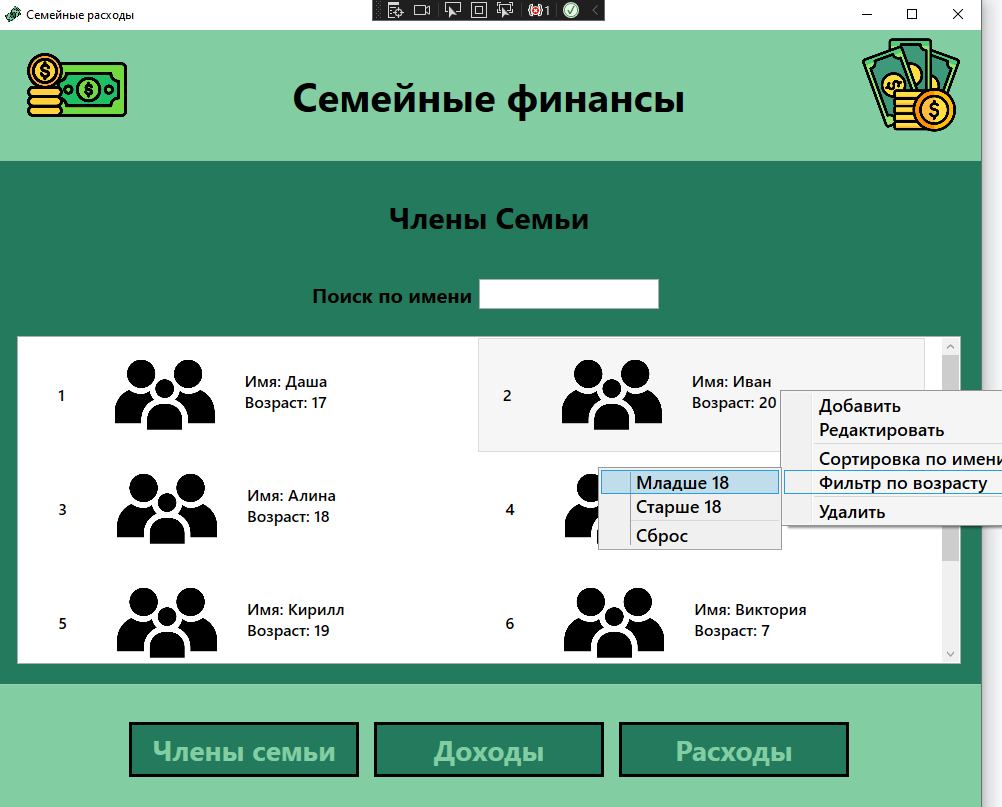


Рис.12 «Страница члены семьи с вызовом фильтр по возрасту»

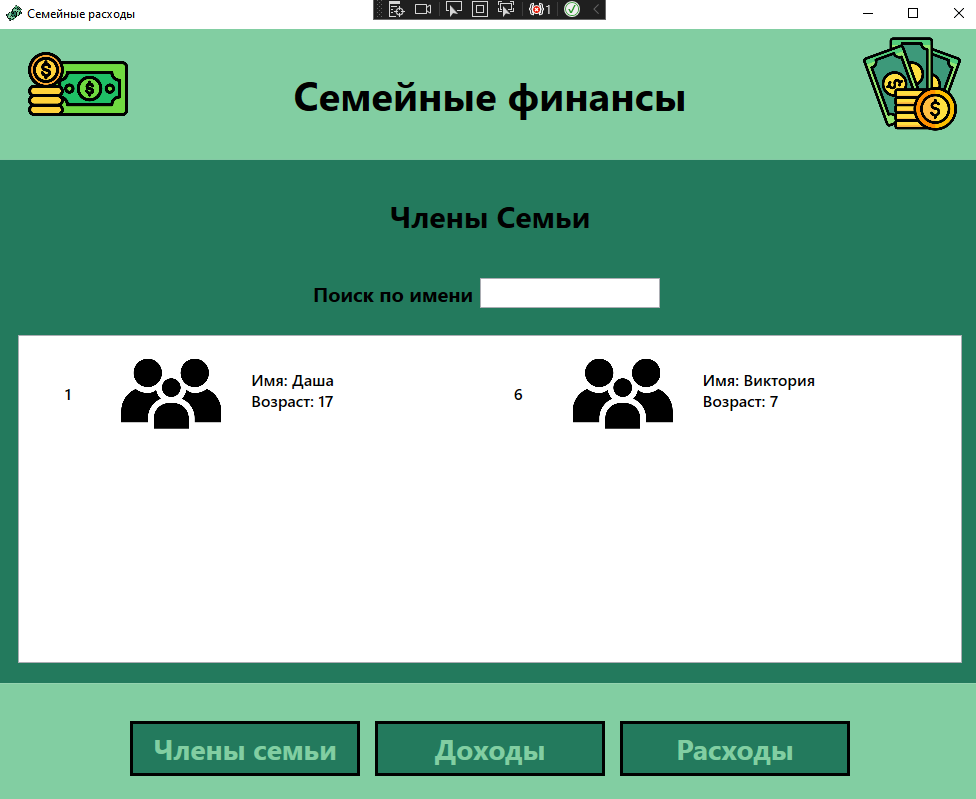


Рис.13 «Страница члены семьи, отфильтрованных младше 18»

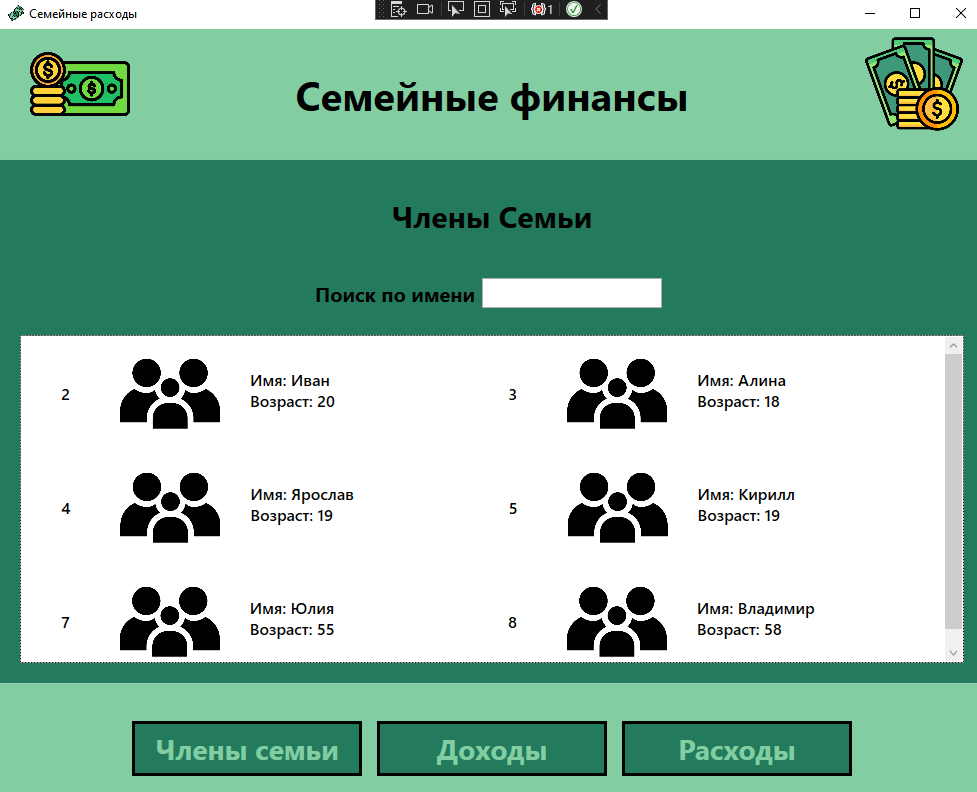


Рис.14 «Страница члены семьи, отфильтрованных старше 18»

При необходимости найти нужного члена семьи, сверху над списком находится поле, внутри которого написано – «поиск», туда нужно вписать одну букву из списка или имя члена семьи, и в списке останется только то, что мы писали в поиске.

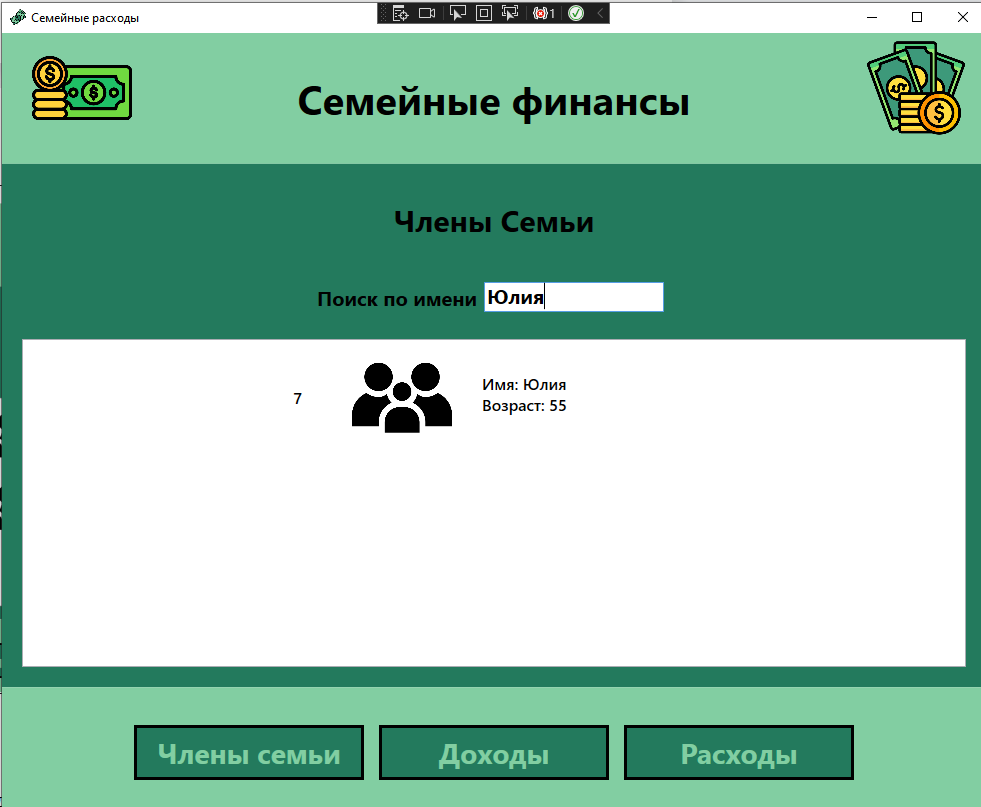


Рис. 15 «Страница члены семьи с использованием поля поиска»

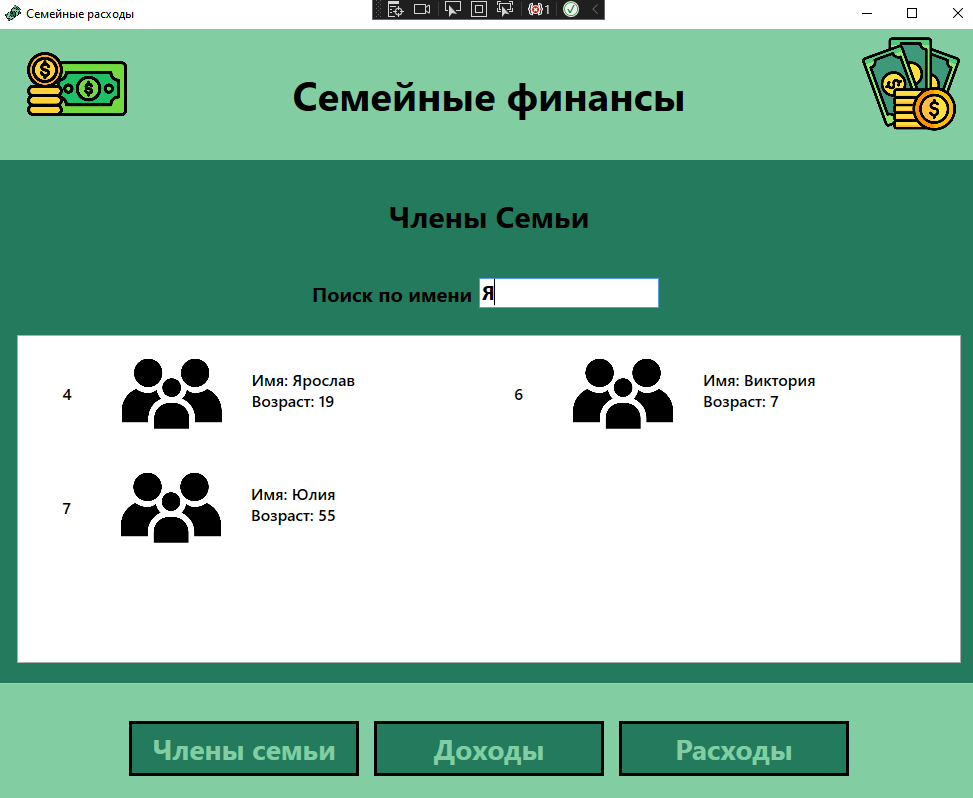


Рис. 16 «Страница членов семьи с использованием поля поиска»

С помощью функции удаления можно удалить доход по различным причинам. Для этого необходимо в контекстном меню выделенного элемента списка выбрать пункт «Удалить»

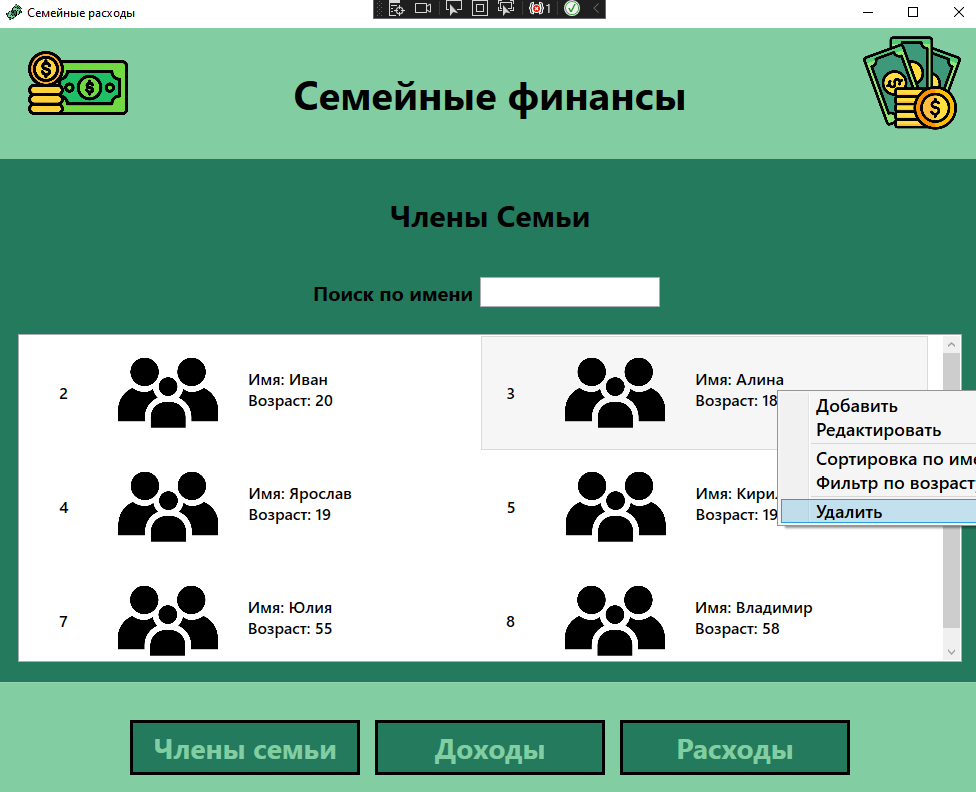


Рис. 17 «Страница члены семьи с вызовом удаления»

После нажатия на одну из кнопок, пользователь попадает на страницу, на которой есть список с заполненными данными. Элементы управления для взаимодействия с информацией располагаются в контекстном меню (чтобы его открыть нужно нажать ПКМ, по элементу списка), а также функции для редактирования, и удаления информации из списка.

При нажатии на кнопку «Доходы» откроется страница с соответствующим списком.

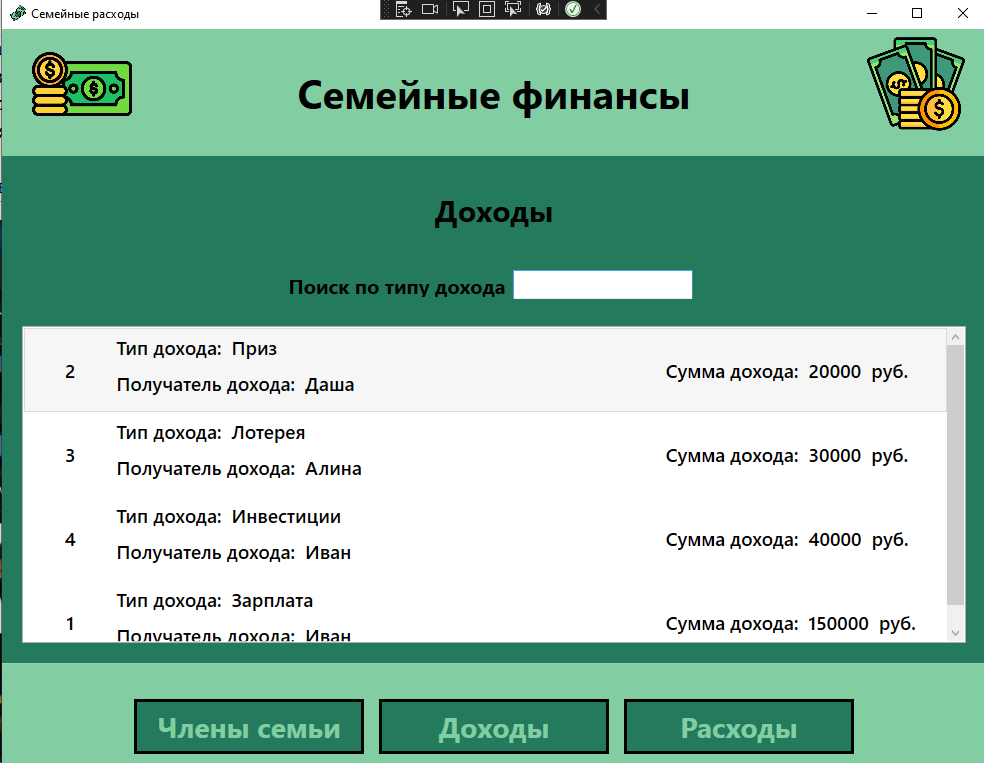


Рис. 18«Страница доходов»

Если пользователю потребовалось добавить данные, то для этого, в контекстном меню, на выбранном элементе из списка требуется нажать «Добавить». После этого пользователь окажется на странице добавления доходов, в которой можно добавить: тип дохода, сумма дохода и получатель дохода.

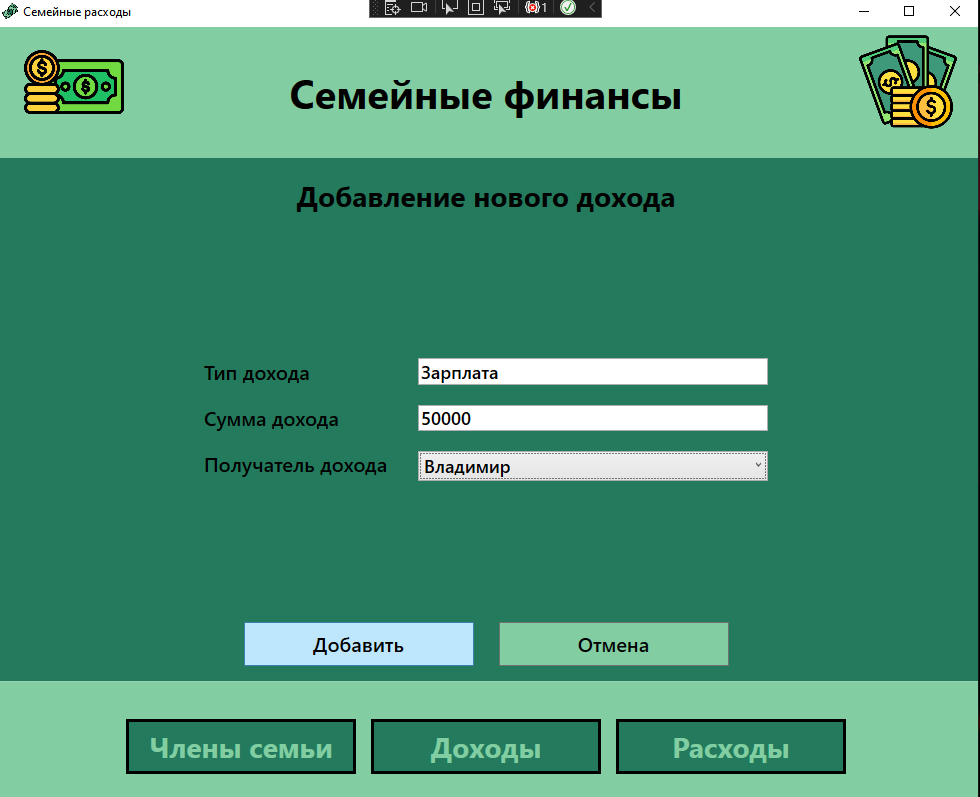


Рис. 19 «Страница добавления доходов»

Если пользователю нужно изменить данные, то для этого, в контекстном меню, на выбранном элементе из списка требуется нажать «Редактировать». После этого пользователь окажется на странице редактирования доходов, в которой можно изменить: тип дохода, сумма дохода и получатель дохода.

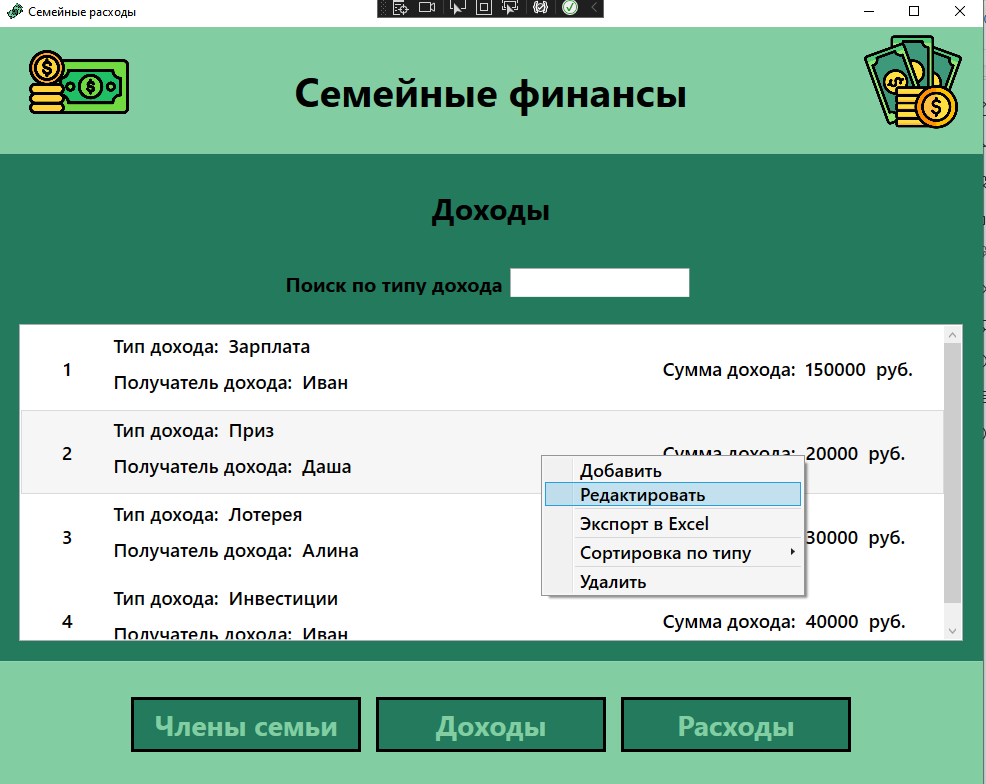


Рис. 20 «Страница исполнителей с вызовом редактирования»

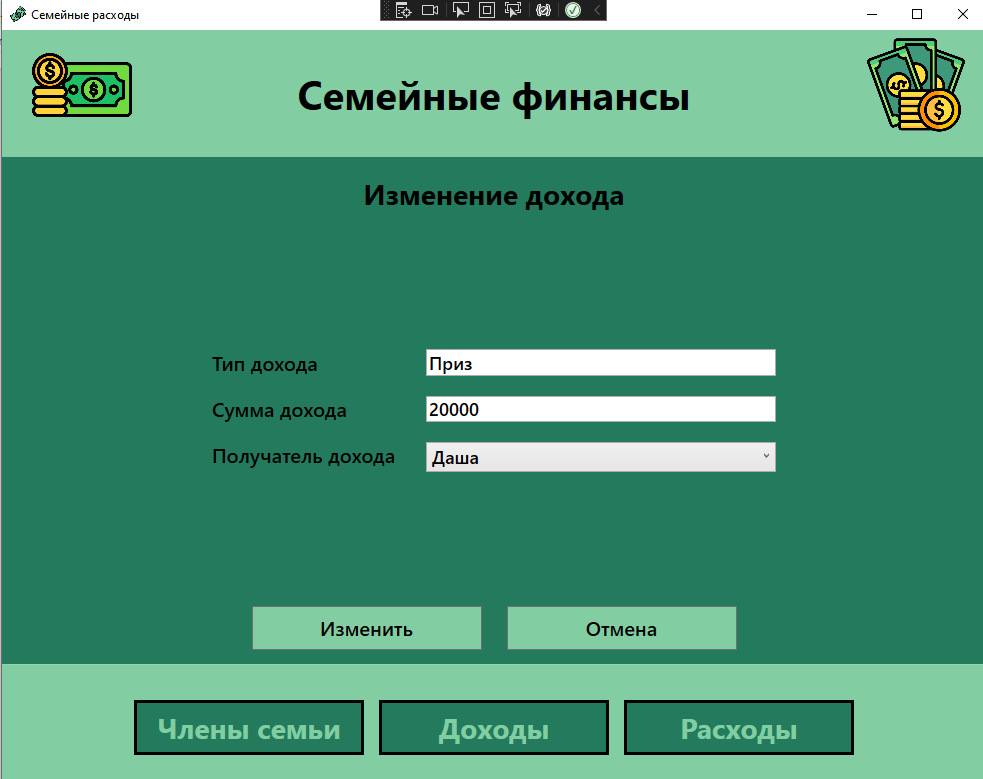


Рис. 21 «Страница редактирования дохода»

Если следует отсортировать данные в алфавитном порядке или же наоборот, то для этого, в контекстном меню, на выбранном элементе из списка требуется навести курсор на пункт «Сортировка», и выбрать нужную сортировку. После этого пользователь увидит, что данные из списка отсортировались в соответствии с выбранной сортировкой. Если выбранная сортировка больше не нужна, то нужно снова навести курсор в контекстном меню на пункт «Сортировка», а затем нажать на «Сброс».

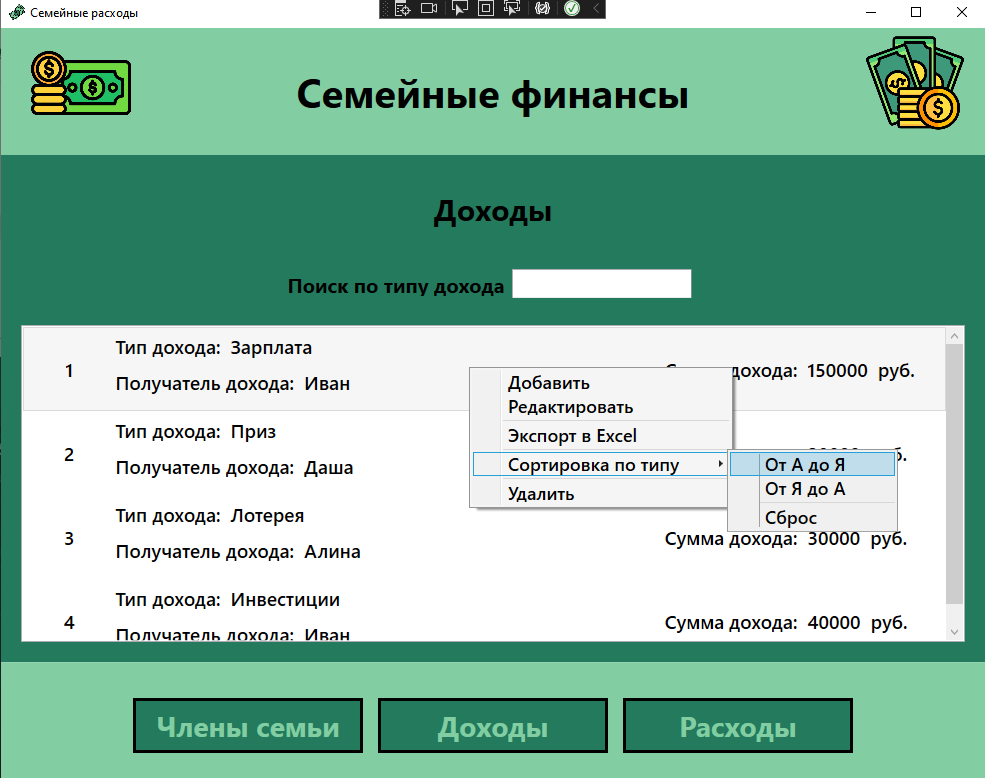


Рис.22 «Страница доходов с вызовом сортировки»

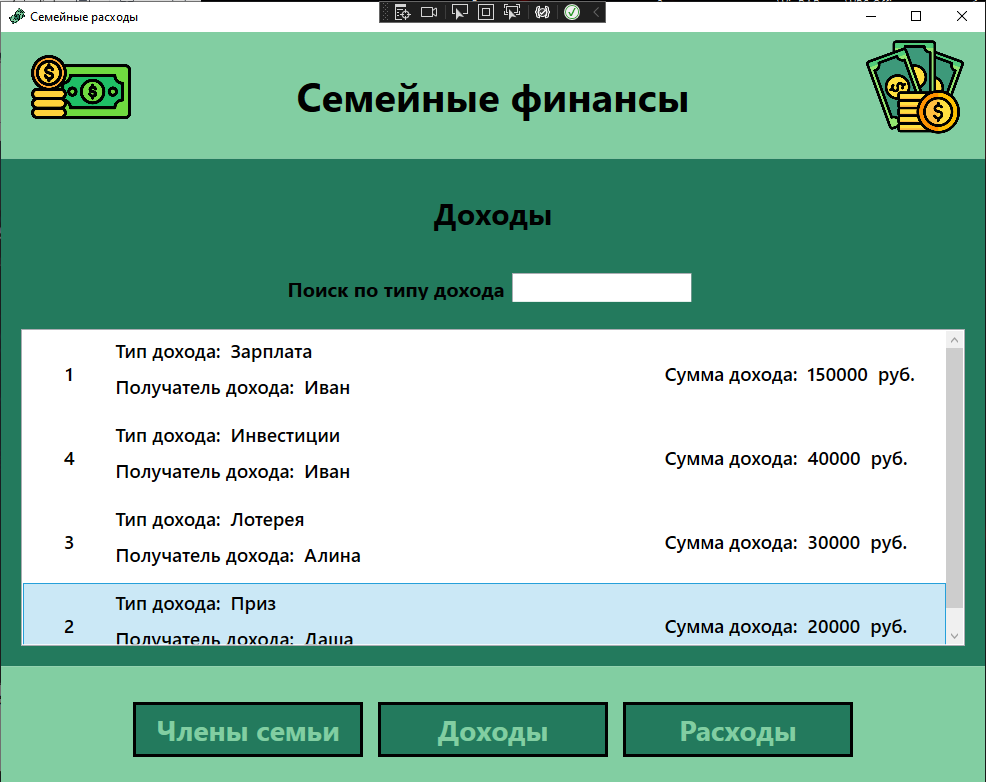


Рис. 23 «Страница доходов, отсортированных в алфавитном порядке»

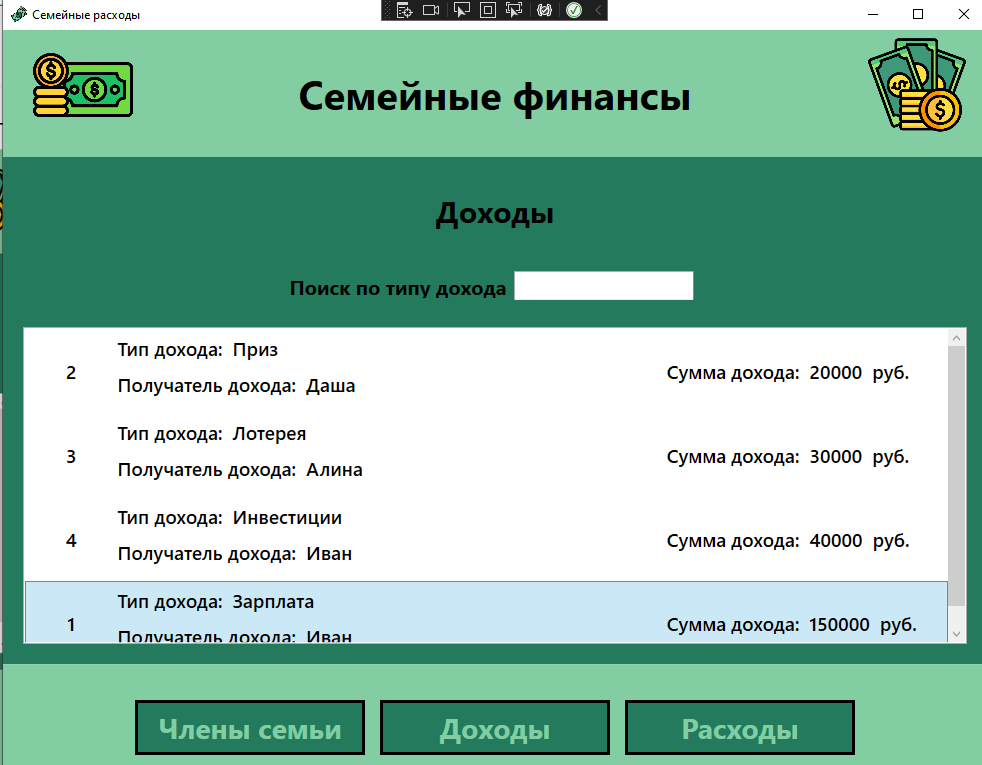


Рис. 24 «Страница доходов, отсортированных в обратном алфавитном порядке»

При необходимости найти нужный доход семьи, сверху над списком находится поле, внутри которого написано – «поиск», туда нужно вписать одну букву из списка или название типа доходов, и в списке останется только то, что мы списали в поиске.

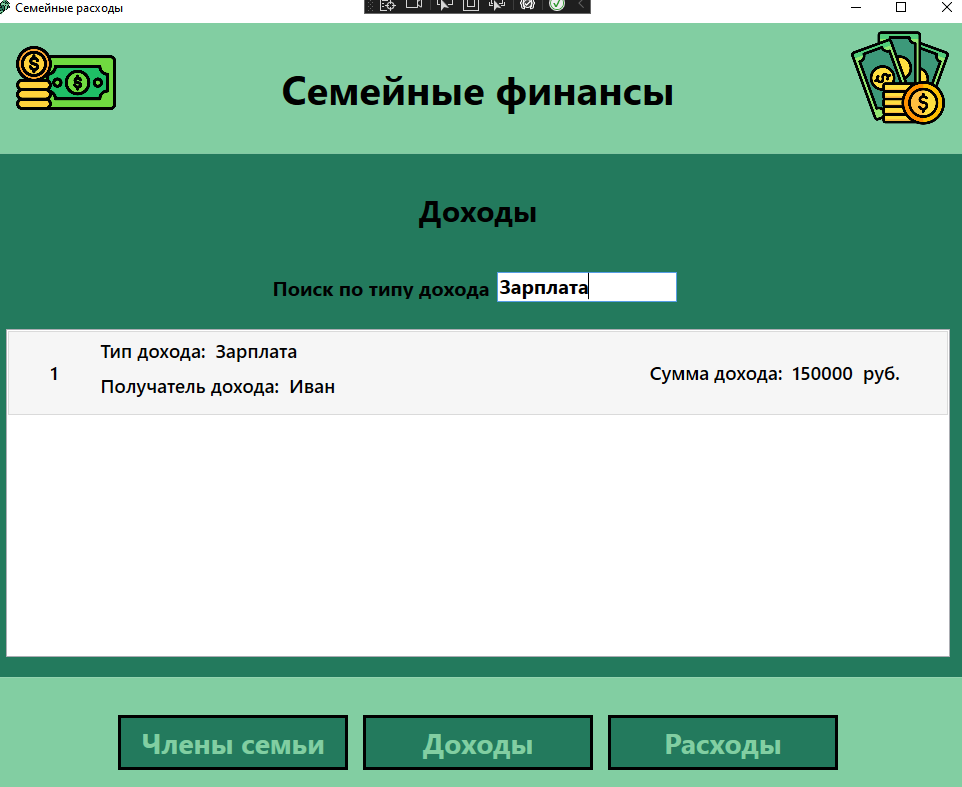


Рис. 25 «Страница доходов с использованием поля поиска»

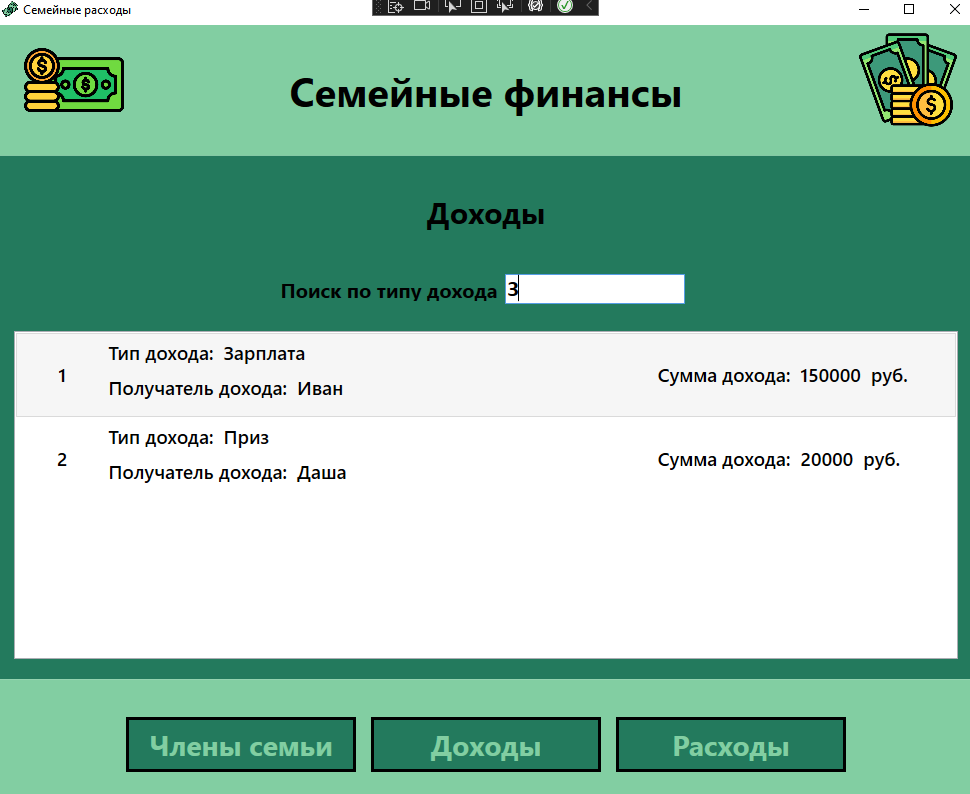


Рис. 26 «Страница доходов с использованием поля поиска»

Для вывода договора о контролимости доходов, необходимо на выбранном элементе списке в контекстном меню кликнуть на «Экспорт в Excel»

Откроется документ «Договор.xlsx» в котором находиться шаблон печати договора о концерте исполнителя

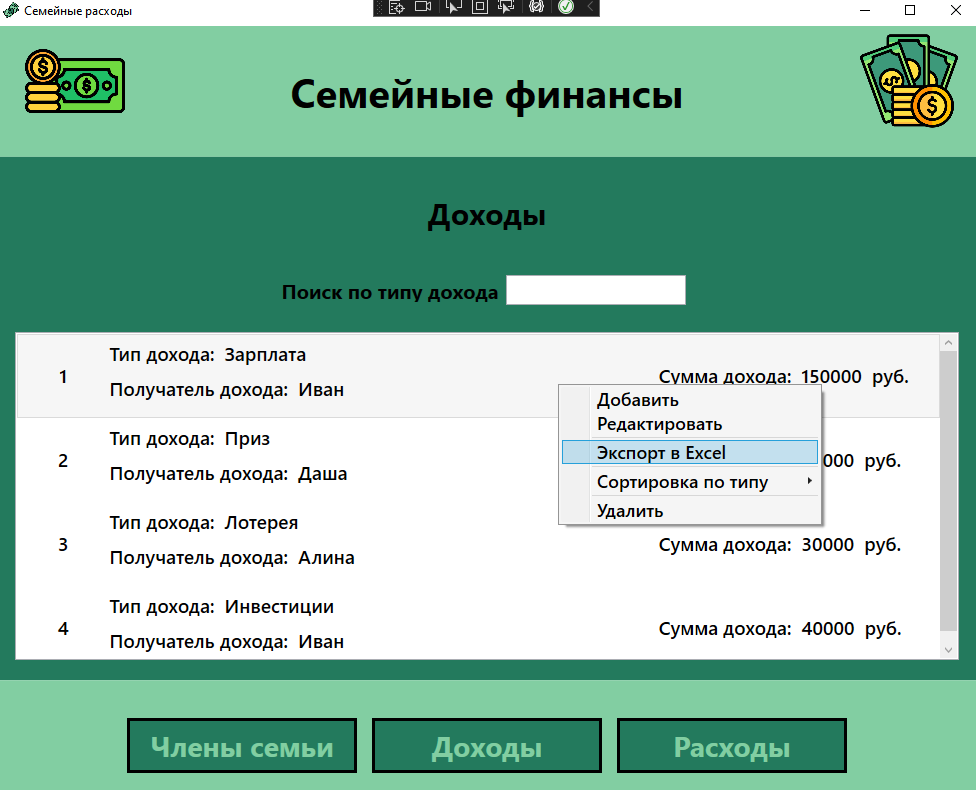


Рис. 27 «Страница доходы с использованием контекстного меню»



Рис. 28 «Вывод документа на печать»

С помощью функции удаления можно удалить доход по различным причинам. Для этого необходимо в контекстном меню выделенного элемента списка выбрать пункт«Удалить».

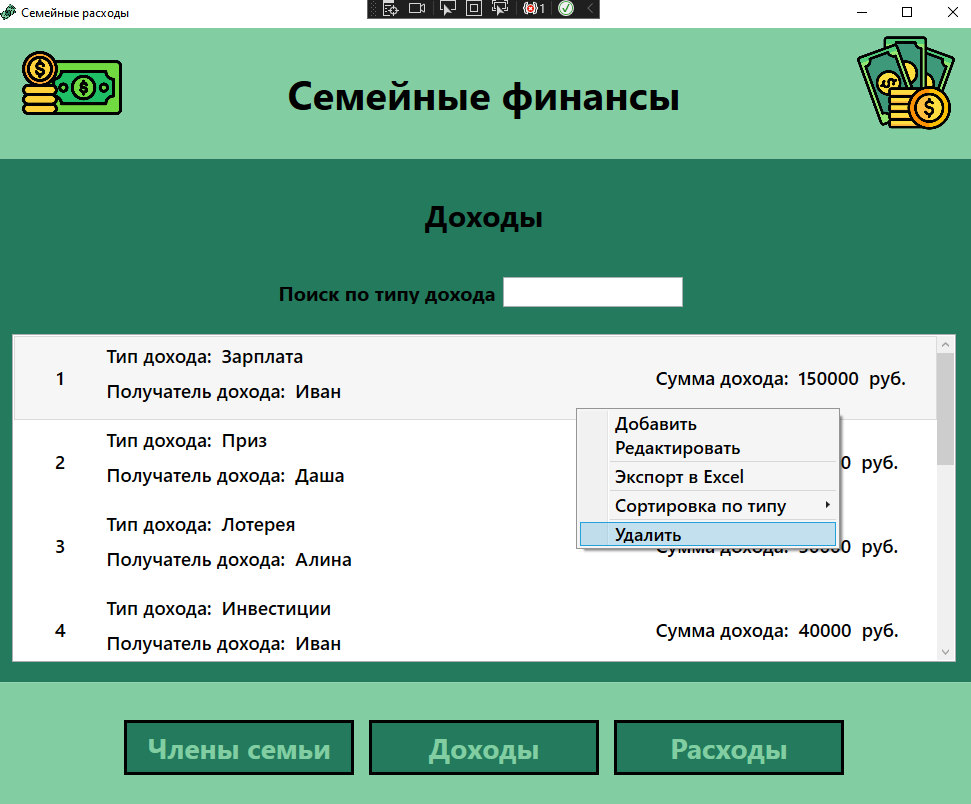


Рис. 29 «Страница доходов с вызовом функции удаления»

При нажатии на кнопку «Расходы», откроется соответствующий список

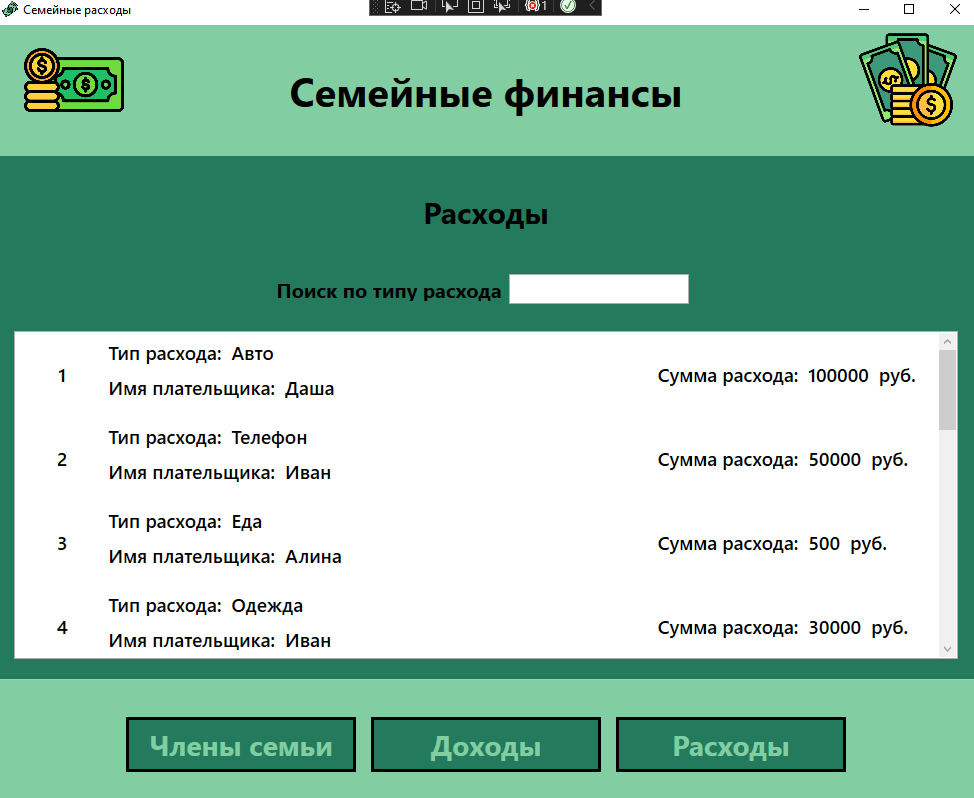


Рис. 30 «Страница расходов»

Если пользователь захочет добавить данные расходов, то для этого, в контекстном меню, на выбранном элементе из списка требуется нажать «Добавить». После этого пользователь окажется на странице добавление нового расхода, в которой можно добавить: тип расхода, сумма дохода, имя плательщика.

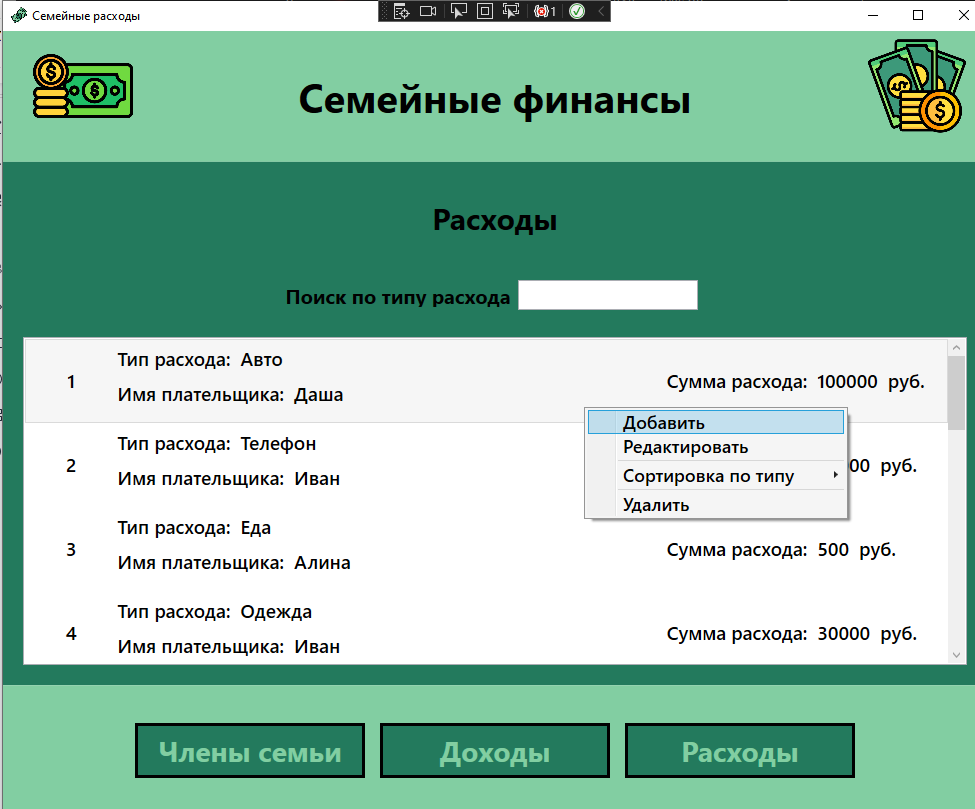


Рис. 31 «Страница доходов с вызовом добавления»

Если пользователь захочет редактировать данные расходов, то для этого, в контекстном меню, на выбранном элементе из списка требуется нажать «Редактировать». После этого пользователь окажется на странице редактирования нового расхода, в которой можно добавить: тип расхода, сумма дохода, имя плательщика.

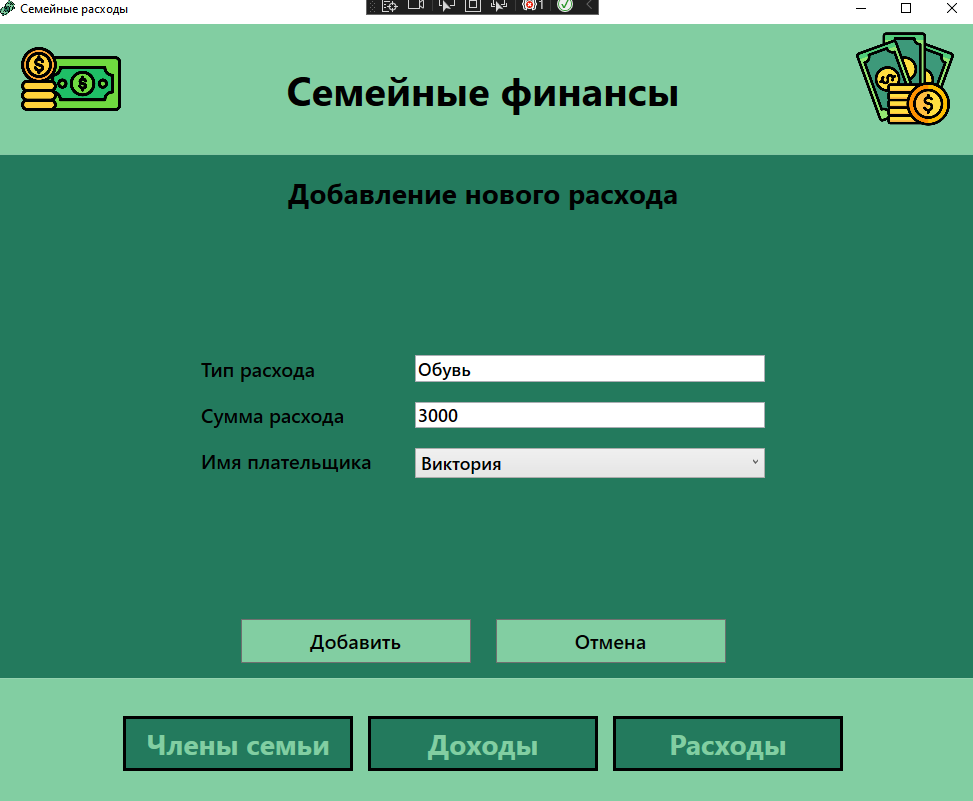


Рис. 32 «Страница добавления расходов»

Если пользователь захочет редактировать данные расходов, то для этого, в контекстном меню, на выбранном элементе из списка требуется нажать «Редактировать». После этого пользователь окажется на странице редактирования нового расхода, в которой можно добавить: тип расхода, сумма дохода, имя плательщика.

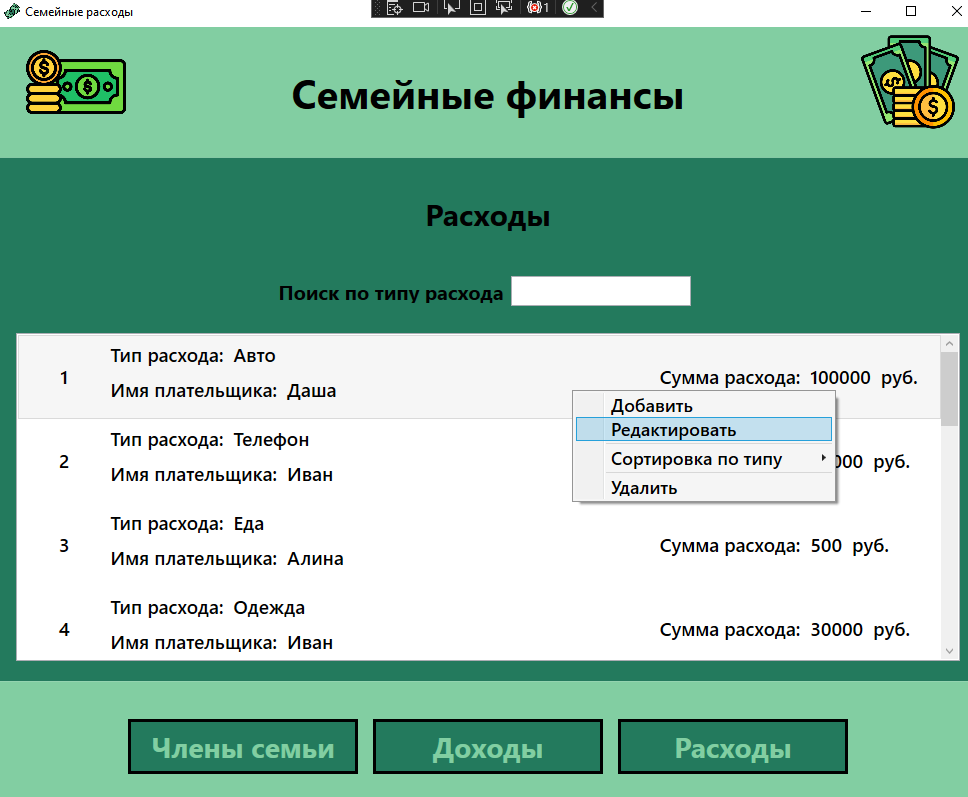


Рис. 33 «Страница расходов с вызовом редактирования»

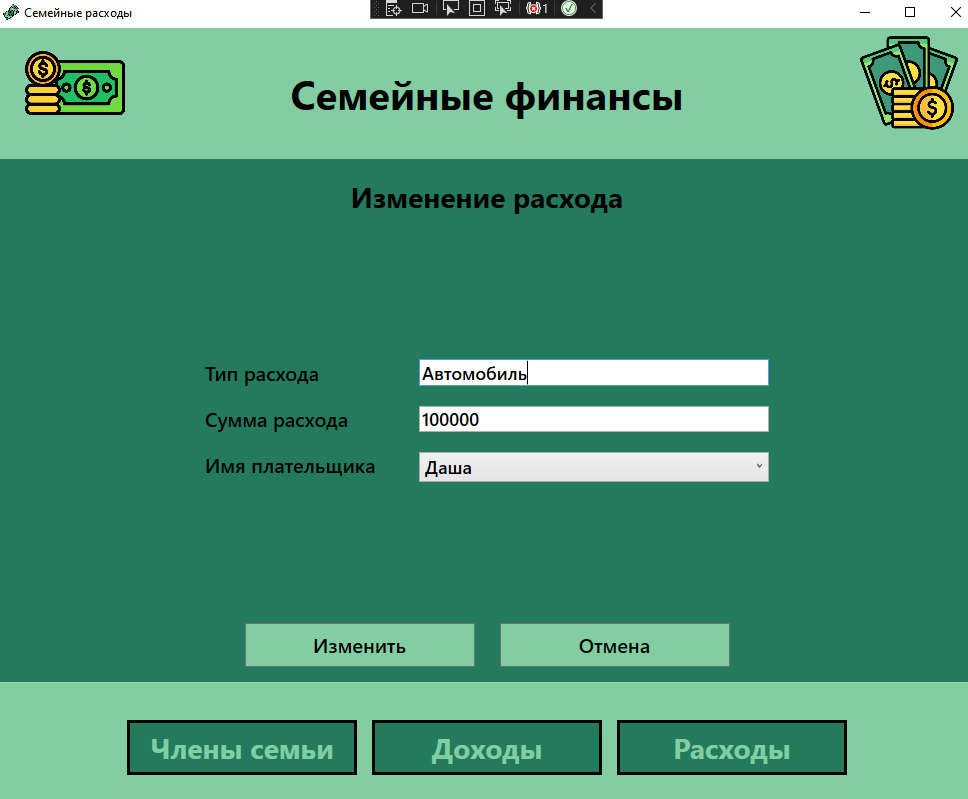


Рис. 34 «Страница редактирования расходов редактирования»

Если следует отсортировать данные в алфавитном порядке или же наоборот, то для этого, в контекстном меню, на выбранном элементе из списка требуется навести курсор на пункт «Сортировка», и выбрать нужную сортировку. После этого пользователь увидит, что данные из списка отсортировались в соответствии с выбранной сортировкой. Если выбранная сортировка больше не нужна, то нужно снова навести курсор в контекстном меню на пункт «Сортировка», а затем нажать на «Сброс».

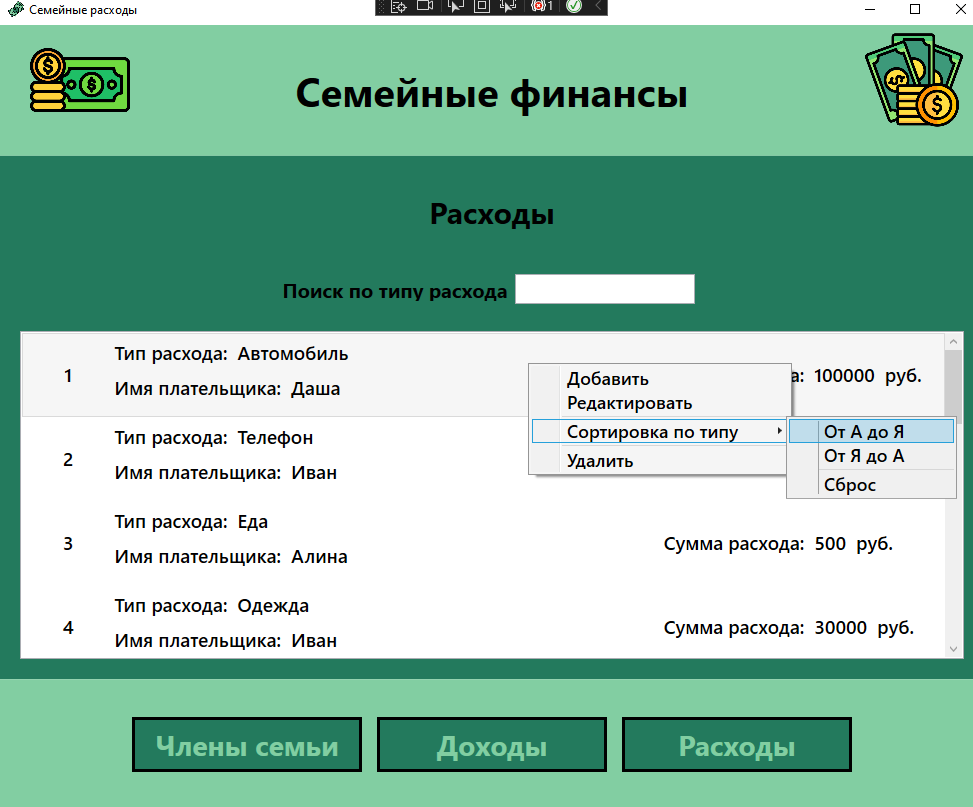


Рис. 35 «Страница расходов с вызовом сортировки»

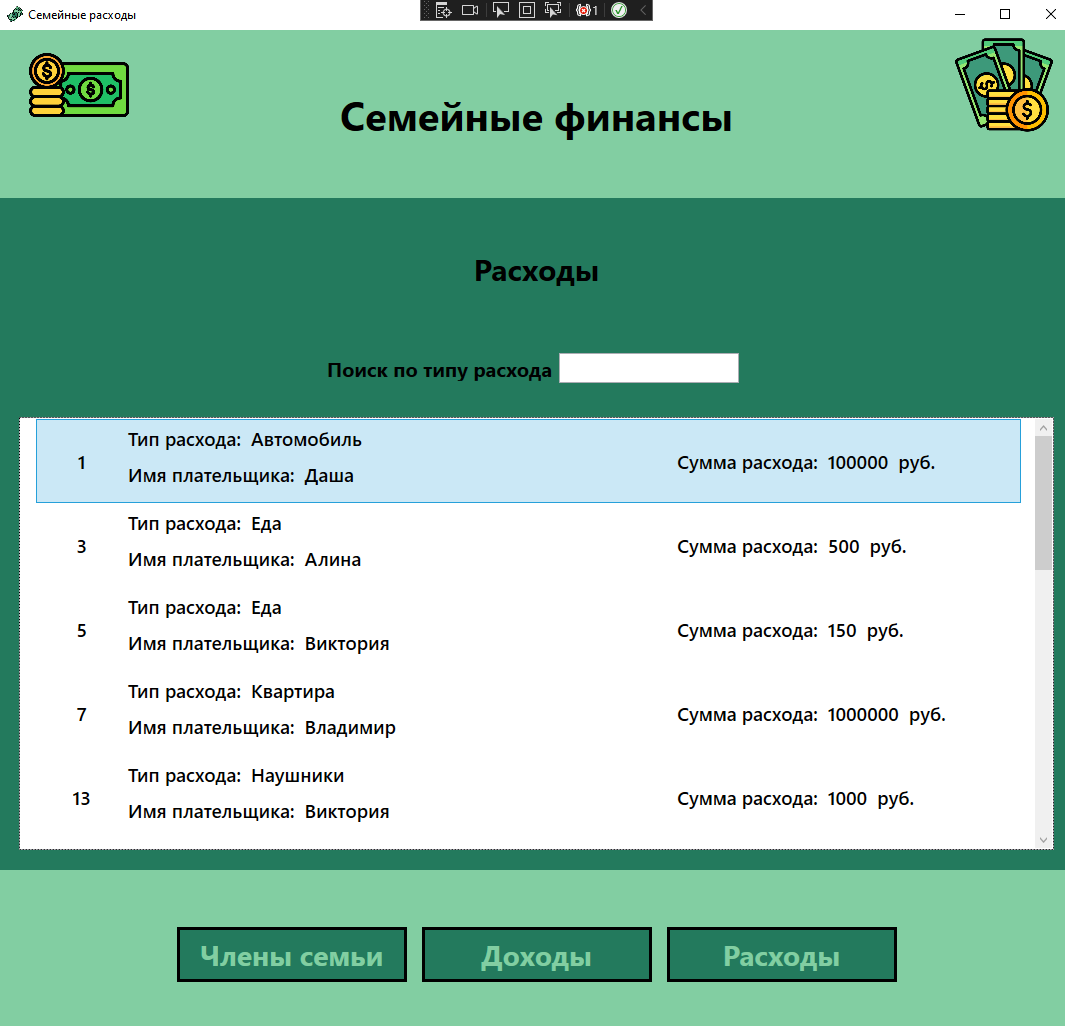


Рис. 36 «Страница расходов, отсортированных в алфавитном порядке

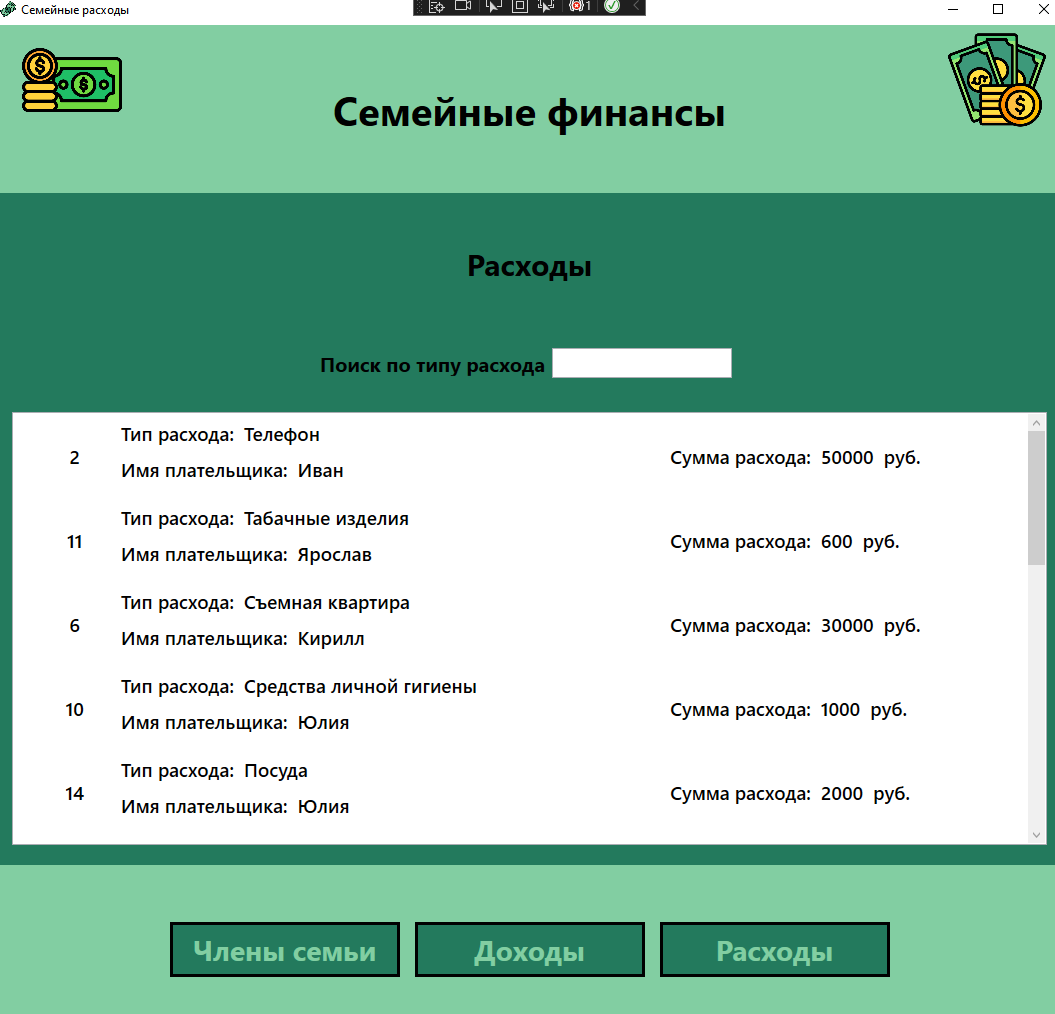


Рис. 37 «Страница расходов, отсортированных в обратном алфавитном порядке»

При необходимости найти нужный расход семьи, сверху над списком находится поле, внутри которого написано – «поиск», туда нужно вписать одну букву из списка или название типа доходов, и в списке останется только то, что мы писали в поиске.

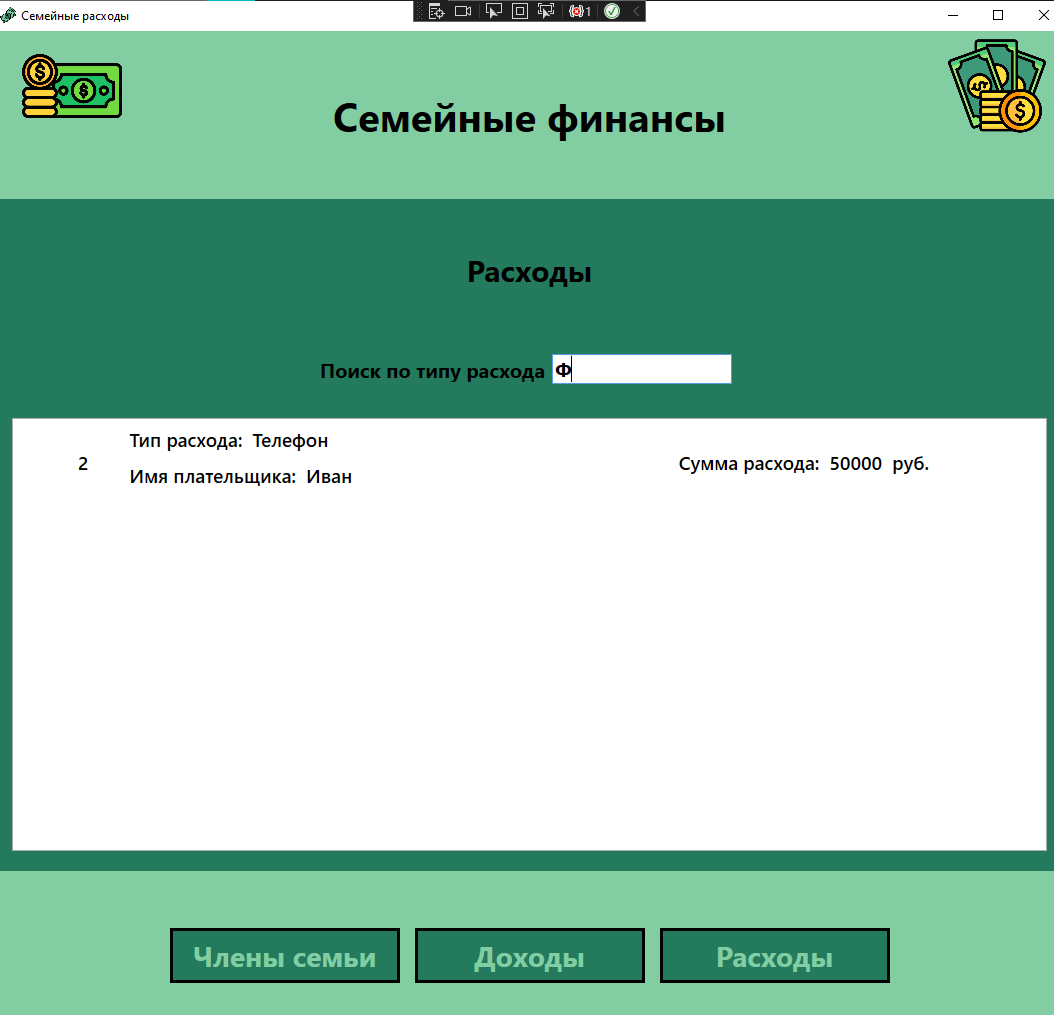


Рис. 38 «Страница авторов с использованием поле поиска»

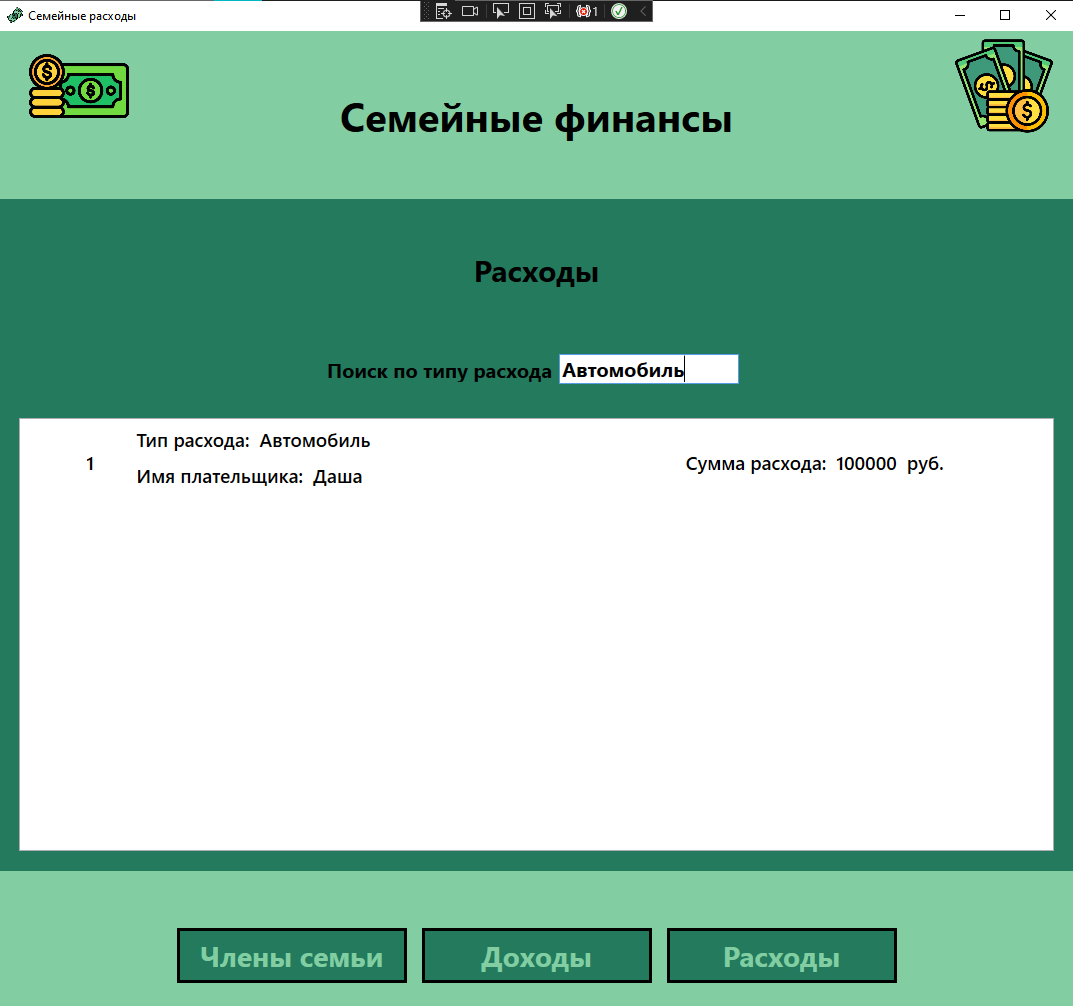


Рис. 39 «Страница авторов с использованием поле поиска»

С помощью функции удаления можно удалить доход по различным причинам. Для этого необходимо в контекстном меню выделенного элемента списка выбрать пункт «Удалить».

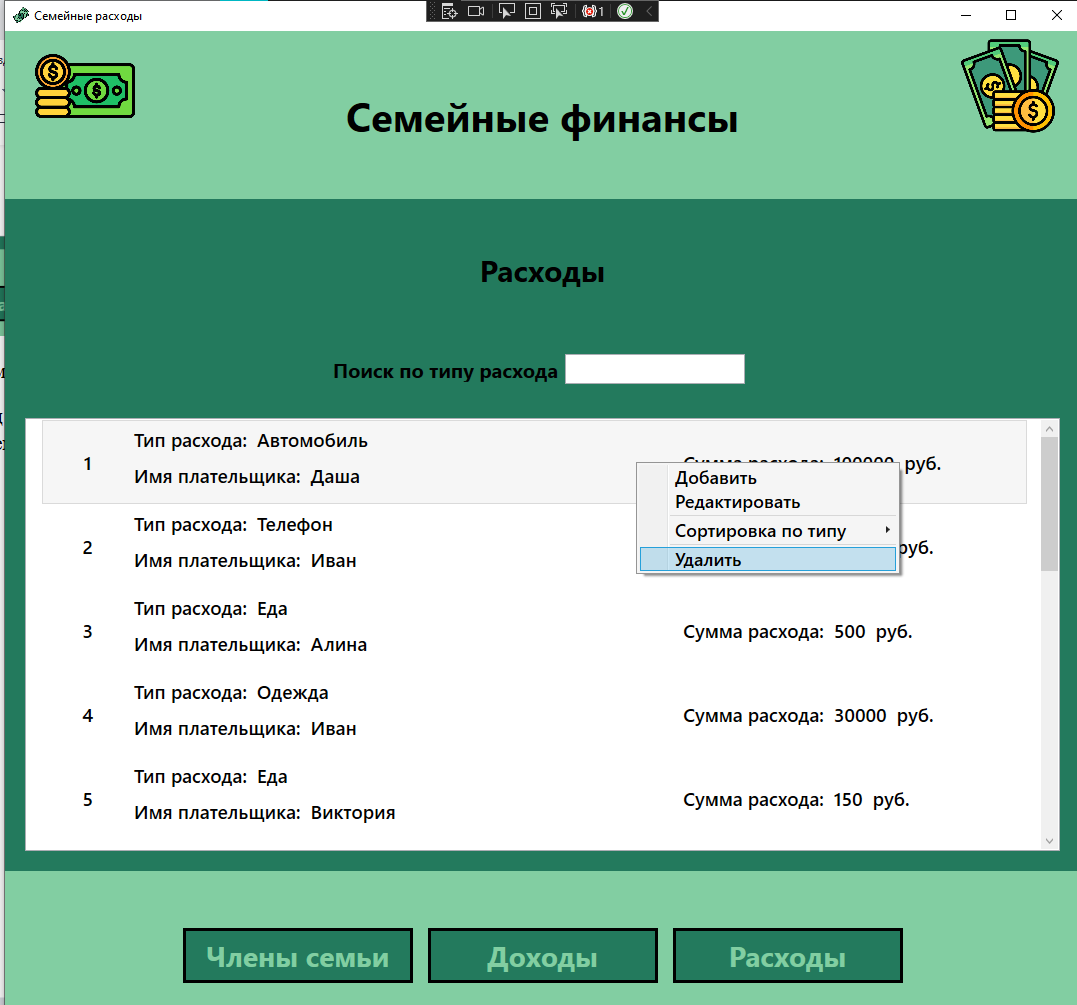


Рис. 40 «Страница расходов с вызовом функции удаления»

Для завершения работы приложения нужно нажать на символ крестика, который находится в верхнем правом углу окна.

## **Сообщения пользователю**

При добавлении и редактировании данных, если обязательные поля для ввода остаются пустыми и пользователь произведёт добавление или сохранение данных, то приложение выведет сообщение с перечислением незаполненных обязательных полей.

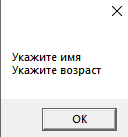


Рис. 41 «Сообщение при добавлении нового члена семьи с незаполненными полями»

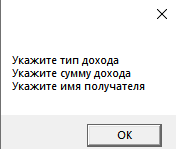


Рис. 42 «Сообщение при добавлении дохода с незаполненными полями»

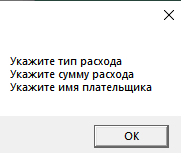


Рис. 43 «Сообщение при добавлении расхода с незаполненными полями»

При добавлении новой записи в любом списке появляется окно с уведомлением, что новая запись добавлена.

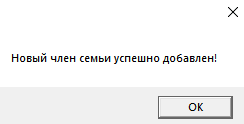


Рис. 44 «Сообщение об успешном добавлении нового члена семьи»

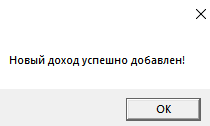


Рис. 45 «Сообщение об успешном добавлении дохода»

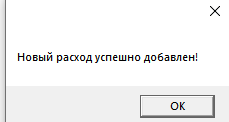


Рис. 46 «Сообщение об успешном добавлении расхода»

При редактировании записи в любом списке появляется окно с уведомлением, что новая запись изменена.

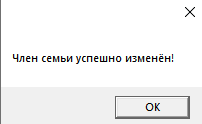


Рис. 47 «Сообщение об успешном изменении члена семьи»

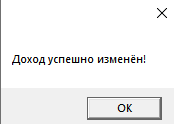


Рис. 48 «Сообщение об успешном изменении дохода»

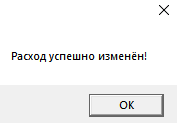


Рис. 49 «Сообщение об успешном изменении расхода»

Когда пользователь собирается удалить запись, появляется предупреждающее сообщение с подтверждением удаления записи с номером.

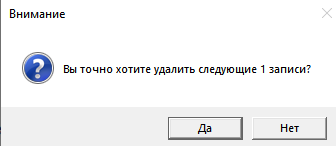


Рис. 50 «Сообщение об уточнении удаления записи с номером»

При подтверждении об удалении появится уведомление, о выполненной операции.

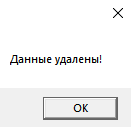


Рис. 51 «Сообщение об успешном удалении данных»

# **5.Тестирование программного модуля**



## **Модульное тестирование**

Библиотека классов с методом подсчета суммарной продолжительности альбома по номеру в зависимости от продолжительностей песен, которые входят в этот альбом

namespace FamilyFinances

{

public class FamilyFinances

{

public string Method\_DurationIncomeOnId(int idIncome)

{

return TimeSpan.FromSeconds((int)CatalogEntities.GetContext().TypeIncome.Where(x => x.idIncome == idIncome).Sum(x => x.Duration)).ToString();

}

}

}

Текст модульного тестирования:

namespace MusicCalatogTest

{

[TestClass]

public class UnitTest\_MusicCatalog

{

MCLibrary lib = new MCLibrary();

[TestMethod]

public void Test\_DurationAlbum1()

{

string expected\_d = "00:22:39";

string actual\_d = lib.Method\_DurationAlbumOnId(1);

Assert.AreEqual(expected\_d, actual\_d);

}

[TestMethod]

public void Test\_DurationAlbum2()

{

string expected\_d = "00:40:25";

string actual\_d = lib.Method\_DurationAlbumOnId(2);

Assert.AreEqual(expected\_d, actual\_d);

}

[TestMethod]

public void Test\_DurationAlbum3()

{

string expected\_d = "00:45:06";

string actual\_d = lib.Method\_DurationAlbumOnId(3);

Assert.AreEqual(expected\_d, actual\_d);

}

[TestMethod]

public void Test\_DurationAlbum4()

{

string expected\_d = "00:51:57";

string actual\_d = lib.Method\_DurationAlbumOnId(4);

Assert.AreEqual(expected\_d, actual\_d);

}

[TestMethod]

public void Test\_DurationAlbum5()

{

string expected\_d = "00:12:34";

string actual\_d = lib.Method\_DurationAlbumOnId(5);

Assert.AreEqual(expected\_d, actual\_d);

}

[TestMethod]

public void Test\_DurationAlbum6()

{

string expected\_d = "00:44:28";

string actual\_d = lib.Method\_DurationAlbumOnId(6);

Assert.AreEqual(expected\_d, actual\_d);

}

[TestMethod]

public void Test\_DurationAlbum7()

{

string expected\_d = "00:45:29";

string actual\_d = lib.Method\_DurationAlbumOnId(7);

Assert.AreEqual(expected\_d, actual\_d);

}

}

}

## Тестовые сценарии

Тестовый сценарий №1:

|  |  |
| --- | --- |
| **Поле** | **Описание** |
| Тестовый пример # | 1 |
| Приоритет тестирования  (Низкий/Средний/Высокий) | Низкий |
| Заголовок/название теста | Сортировка доход |
| Краткое изложение теста | Список доходов должен от сортироваться по их наименованиям |
| Этапы теста | 1. Запустите программу через Visual Studio.  2. Попадаете в главное меню. Оттуда переходите на страницу «Доходы».  3. Затем нажмите в контекстном меню пункт «Сортировка» |
| Тестовые данные | - |
| Ожидаемый результат | Если будет использоваться сортировка, то данные должны быть отсортированы в списке |
| Фактический результат | Сортировка применилась некорректно |
| Статус | Незачет |
| Предварительное условие | Исправьте код, чтобы сортировка была выполнена |
| Постусловие | - |
| Примечания/комментари**и** | Данный способ сортировки не работал, т.к. возможно в коде был неправильно указан параметр для сортировки. |

Тестовый сценарий №2:

|  |  |
| --- | --- |
| **Поле** | **Описание** |
| Тестовый пример # | 2 |
| Приоритет тестирования | Средний |
| Заголовок/название теста | Поиск расхода по наименованию |
| Краткое изложение теста | Должен вывестись расход, который указан в поле поиска |
| Этапы теста | 1. Запустите программу через Visual Studio.  2. Попадаете в главное меню. Оттуда переходите на страницу «Расходы».  3. Затем в поле поиска вводите исполнителя |
| Тестовые данные | Наименование: Еда |
| Ожидаемый результат | Список расходов, который указан в поле поиска, |
| Фактический результат | Расход указанный в поле поиска выведен в список расходы |
| Статус | Зачет |
| Предварительное условие | - |
| Постусловие | - |

# **Отладка программного модуля**

Код сортировки песен, работающий некорректно:

private void MenuAddExpense\_Click(object sender, RoutedEventArgs e)

{

Classes.ClassFrame.frmObj.Navigate(new PageAddExpenses(null));

}

private void MenuEditExpense\_Click(object sender, RoutedEventArgs e)

{

Classes.ClassFrame.frmObj.Navigate(new PageAddExpenses((Expenses)LTVIExpenses.SelectedItem))

}

Исправленный код сортировки песен, работающий корректно:

private void MenuAddExpense\_Click(object sender, RoutedEventArgs e)

{

Classes.ClassFrame.frmObj.Navigate(new PageAddExpenses(null));

}

private void MenuEditExpense\_Click(object sender, RoutedEventArgs e)

{

Classes.ClassFrame.frmObj.Navigate(new PageAddExpenses((Expenses)LTVIExpenses.SelectedItem));

}

# **Заключение**

В результате выполнения поставленной задачи разработана программа, предназначена для того, чтобы молодые или взрослые семьи научились ввести и контролировать семейный бюджет, а то есть доходы и расходы семьи. Я бы хотела ещё добавить в эту программу более интересный интерфейс с визуальными эффектами, чтобы радовал глаз. Для обеспечения уверенной работы приложения выполнены тестовые сценарии. Документирование позволило рассмотреть программный модуль с точки зрения программиста и пользователя.

В процессе разработки данной программы, освоила новые знания и навыки в области информационных технологий, изучила новые и полезные методики и технологии, которые я успешно применила в ходе этой работы.

**Литература:**

1. Брюс Лоусон, Реми Шарп - “Изучаем HTML5”, **Издательство:** Питер, **Год:**2012
2. Кристофер Шмитт, Кайл Симпсон - "HTML5: Рецепты программирования", **Издательство:**Питер Год 2012
3. Эстель Вейл - "HTML5: Разработка приложений для мобильных устройств", Издательство: Питер, Год: 2015
4. Дженнифер Нидерст Роббинс - "HTML”, Издательство: Вильямс, Год: 2019
5. Джон Дакетт - "HTML и CSS.” Издательство: Эксмо, Год:2019
6. https://doskol.narod.ru/FILES/HTML.pdf
7. <https://developer.mozilla.org/ru/docs/Learn/Getting_started_with_the_web/CSS_basics>
8. Великий Фаберже. Искусство ювелиров придворной фирмы. - Москва: **Машиностроение, 2019**. - 144 c.
9. Культин Никита Борисович Основы программирования в Microsoft Visual C# 2010; БХВ-Петербург - М., 2011. - **384**c.

10.Джозеф Албахари C# 5.0. Справочник. Полное описание языка; Диалектика / Вильямс - М., 2014. - **545**c.

11.https://media.halvacard.ru/finance/lichnyy-byudzhet/

Приложение 1

PageAddExpenses.xaml

<Page x:Class="FamilyFinances.Pages.PageAddExpenses"

xmlns="http://schemas.microsoft.com/winfx/2006/xaml/presentation"

xmlns:x="http://schemas.microsoft.com/winfx/2006/xaml"

xmlns:mc="http://schemas.openxmlformats.org/markup-compatibility/2006"

xmlns:d="http://schemas.microsoft.com/expression/blend/2008"

xmlns:local="clr-namespace:FamilyFinances.Pages"

mc:Ignorable="d"

d:DesignHeight="450" d:DesignWidth="800"

Title="PageAddExpenses">

<Grid>

<Grid.RowDefinitions>

<RowDefinition Height="1\*"/>

<RowDefinition Height="5\*"/>

<RowDefinition Height="1\*"/>

</Grid.RowDefinitions>

<StackPanel Orientation="Vertical" Grid.Row="0" HorizontalAlignment="Center" VerticalAlignment="Center">

<TextBlock Text="Добавление нового расхода" Name="TitletxExpense" FontSize="28" FontWeight="Bold" Foreground="Black"/>

</StackPanel>

<StackPanel Grid.Row="1" Orientation="Vertical" Width="600" Margin="15" VerticalAlignment="Center">

<StackPanel Orientation="Horizontal" HorizontalAlignment="Center">

<TextBlock Text="Тип расхода" Width="200" FontSize="20" Margin="7" FontWeight="DemiBold" Foreground="Black"/>

<TextBox Text="{Binding Type\_of\_expenses}" Width="350" FontSize="18" Margin="7" FontWeight="DemiBold"/>

</StackPanel>

<Separator Opacity="0" Height="1.5"/>

<StackPanel Orientation="Horizontal" HorizontalAlignment="Center">

<TextBlock Text="Сумма расхода" Width="200" FontSize="20" Margin="7" FontWeight="DemiBold" Foreground="Black"/>

<TextBox Text="{Binding Expenses\_sum}" Width="350" FontSize="18" Margin="7" FontWeight="DemiBold"/>

</StackPanel>

<Separator Opacity="0" Height="1.5"/>

<StackPanel Orientation="Horizontal" HorizontalAlignment="Center">

<TextBlock Text="Имя плательщика" Width="200" FontSize="20" Margin="7,7,7,7" FontWeight="DemiBold" Foreground="Black"/>

<ComboBox Name="CmbName" SelectedValue="{Binding idFamilyMember}" Width="350" FontSize="18" Margin="7" FontWeight="DemiBold" />

</StackPanel>

</StackPanel>

<StackPanel Grid.Row="2" Orientation="Vertical" VerticalAlignment="Center" HorizontalAlignment="Center">

<StackPanel Orientation="Horizontal" Grid.Row="2" >

<Button Width="230" Height="44" Content="Добавить" VerticalAlignment="Center" FontSize="20" FontWeight="DemiBold" Name="BtnAddExpense" Click="BtnAddExpense\_Click" Margin="25,0" Background="#82cea2" Foreground="Black"/>

<Button Width="230" Height="44" Content="Отмена" VerticalAlignment="Center" FontSize="20" FontWeight="DemiBold" Name="BtnCancelExpense" Click="BtnCancelExpense\_Click" Background="#82cea2" Foreground="Black" Margin="0,0,25,0"/>

</StackPanel>

</StackPanel>

</Grid>

</Page>

PageAddFamilyMembers.xaml.cs

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

using System.Windows;

using System.Windows.Controls;

using System.Windows.Data;

using System.Windows.Documents;

using System.Windows.Input;

using System.Windows.Media;

using System.Windows.Media.Imaging;

using System.Windows.Navigation;

using System.Windows.Shapes;

using FamilyFinances.Classes;

namespace FamilyFinances.Pages

{

/// <summary>

/// Логика взаимодействия для PageAddExpenses.xaml

/// </summary>

public partial class PageAddExpenses : Page

{

private Expenses \_currentExpenses = new Expenses();

public PageAddExpenses(Expenses selectedExpense)

{

InitializeComponent();

if (selectedExpense != null)

{

\_currentExpenses = selectedExpense;

TitletxExpense.Text = "Изменение расхода";

BtnAddExpense.Content = "Изменить";

}

DataContext = \_currentExpenses;

CmbName.ItemsSource = FamilyFinancesEntities.GetContext().FamilyMembers.ToList();

CmbName.SelectedValuePath = "idFamilyMember";

CmbName.DisplayMemberPath = "Name";

}

private void BtnAddExpense\_Click(object sender, RoutedEventArgs e)

{

StringBuilder error = new StringBuilder();

if (string.IsNullOrWhiteSpace(\_currentExpenses.Type\_of\_expenses)) error.AppendLine("Укажите тип расхода");

if (string.IsNullOrWhiteSpace(Convert.ToString(\_currentExpenses.Expenses\_sum))) error.AppendLine("Укажите сумму расхода");

if (string.IsNullOrWhiteSpace(Convert.ToString(\_currentExpenses.idFamilyMember))) error.AppendLine("Укажите имя плательщика");

if (error.Length > 0)

{

MessageBox.Show(error.ToString());

return;

}

if (\_currentExpenses.idExpense == 0)

{

FamilyFinancesEntities.GetContext().Expenses.Add(\_currentExpenses);

try

{

FamilyFinancesEntities.GetContext().SaveChanges();

Classes.ClassFrame.frmObj.Navigate(new PageExpenses());

MessageBox.Show("Новый расход успешно добавлен!");

}

catch (Exception ex)

{

MessageBox.Show(ex.Message.ToString());

}

}

else

{

try

{

FamilyFinancesEntities.GetContext().SaveChanges();

Classes.ClassFrame.frmObj.Navigate(new PageExpenses());

MessageBox.Show("Расход успешно изменён!");

}

catch (Exception ex)

{

MessageBox.Show(ex.Message.ToString());

}

}

}

private void BtnCancelExpense\_Click(object sender, RoutedEventArgs e)

{

Classes.ClassFrame.frmObj.Navigate(new PageExpenses());

}

}

}

PageAddFamilyMembers.xaml

<Page x:Class="FamilyFinances.Pages.PageAddFamilyMembers"

xmlns="http://schemas.microsoft.com/winfx/2006/xaml/presentation"

xmlns:x="http://schemas.microsoft.com/winfx/2006/xaml"

xmlns:mc="http://schemas.openxmlformats.org/markup-compatibility/2006"

xmlns:d="http://schemas.microsoft.com/expression/blend/2008"

xmlns:local="clr-namespace:FamilyFinances.Pages"

mc:Ignorable="d"

d:DesignHeight="450" d:DesignWidth="800"

Title="PageAddFamilyMembers">

<Grid>

<Grid.RowDefinitions>

<RowDefinition Height="1\*"/>

<RowDefinition Height="5\*"/>

<RowDefinition Height="1\*"/>

</Grid.RowDefinitions>

<StackPanel Orientation="Vertical" Grid.Row="0" HorizontalAlignment="Center" VerticalAlignment="Center">

<TextBlock Text="Добавление нового члена семьи" Name="TitletxtFamilyMember" FontSize="28" FontWeight="Bold" Foreground="Black"/>

</StackPanel>

<StackPanel Grid.Row="1" Orientation="Vertical" Width="600" Margin="15">

<StackPanel Orientation="Horizontal" HorizontalAlignment="Center">

<Image Name="ImageFamilyMember" Width="125" Height="125" Source="{Binding PhotoFamilyMember, TargetNullValue={StaticResource UnknownFamilyMember}}" Stretch="UniformToFill"/>

<StackPanel Margin="0,20,0,0">

<Button Content="Загрузить" Width="165" Height="40" Foreground="Black" FontWeight="Bold" BorderThickness="2" BorderBrush="Black" Background="#82cea2" FontSize="21" Margin="10" Click="LoadImage"/>

<Button Content="Очистить" Width="165" Height="40" Foreground="Black" FontWeight="Bold" BorderThickness="2" BorderBrush="Black" Background="#82cea2" FontSize="21" Margin="10" Click="ClearImage"/>

</StackPanel>

</StackPanel>

<StackPanel Orientation="Horizontal" HorizontalAlignment="Center" Margin="0,20,0,0">

<TextBlock Text="Имя" Width="100" FontSize="20" Margin="7" FontWeight="DemiBold" Foreground="Black"/>

<TextBox Name="TxtPhone" Text="{Binding Name}" Width="180" FontSize="18" Margin="7" FontWeight="DemiBold"/>

</StackPanel>

<Separator Opacity="0" Height="1.5"/>

<StackPanel Orientation="Horizontal" HorizontalAlignment="Center">

<TextBlock Text="Возраст" Width="100" FontSize="20" Margin="7" FontWeight="DemiBold" Foreground="Black"/>

<TextBox Name="TxtRating" Text="{Binding Age}" Width="180" FontSize="18" Margin="7" FontWeight="DemiBold"/>

</StackPanel>

</StackPanel>

<StackPanel Grid.Row="2" Orientation="Vertical" VerticalAlignment="Center" HorizontalAlignment="Center">

<StackPanel Orientation="Horizontal" Grid.Row="2" >

<Button Width="230" Height="44" Content="Добавить" VerticalAlignment="Center" FontSize="20" FontWeight="DemiBold" Name="BtnAddFamilyMember" Click="BtnAddFamilyMember\_Click" Margin="25,0" Background="#82cea2" Foreground="Black" BorderBrush="Black" BorderThickness="3"/>

<Button Width="230" Height="44" Content="Отмена" VerticalAlignment="Center" FontSize="20" FontWeight="DemiBold" Name="BtnCancelFamilyMember" Click="BtnCancelFamilyMember\_Click" Background="#82cea2" Foreground="Black" Margin="0,0,25,0" BorderBrush="Black" BorderThickness="3"/>

</StackPanel>

</StackPanel>

</Grid>

</Page>

PageAddFmailyMembres.xaml.cs

using FamilyFinances.Classes;

using Microsoft.Win32;

using System;

using System.Collections.Generic;

using System.IO;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

using System.Windows;

using System.Windows.Controls;

using System.Windows.Data;

using System.Windows.Documents;

using System.Windows.Input;

using System.Windows.Media;

using System.Windows.Media.Imaging;

using System.Windows.Navigation;

using System.Windows.Shapes;

namespace FamilyFinances.Pages

{

/// <summary>

/// Логика взаимодействия для PageAddFamilyMembers.xaml

/// </summary>

public partial class PageAddFamilyMembers : Page

{

string imgLoc = "пусто";

private FamilyMembers \_currentFamilyMembers = new FamilyMembers();

public PageAddFamilyMembers(FamilyMembers selectedFamilyMember)

{

InitializeComponent();

if (selectedFamilyMember != null)

{

\_currentFamilyMembers = selectedFamilyMember;

TitletxtFamilyMember.Text = "Изменение члена семьи";

BtnAddFamilyMember.Content = "Изменить";

}

DataContext = \_currentFamilyMembers;

}

private void BtnAddFamilyMember\_Click(object sender, RoutedEventArgs e)

{

StringBuilder error = new StringBuilder();

if (string.IsNullOrWhiteSpace(\_currentFamilyMembers.Name)) error.AppendLine("Укажите имя");

if (string.IsNullOrWhiteSpace(Convert.ToString(\_currentFamilyMembers.Age))) error.AppendLine("Укажите возраст");

if (error.Length > 0)

{

MessageBox.Show(error.ToString());

return;

}

if (\_currentFamilyMembers.idFamilyMember == 0)

{

FamilyFinancesEntities.GetContext().FamilyMembers.Add(\_currentFamilyMembers);

try

{

FamilyFinancesEntities.GetContext().SaveChanges();

Classes.ClassFrame.frmObj.Navigate(new PageFamilyMembers());

MessageBox.Show("Новый член семьи успешно добавлен!");

}

catch (Exception ex)

{

MessageBox.Show(ex.Message.ToString());

}

}

else

{

try

{

if (imgLoc != "пусто" && imgLoc != "очистить")

{

byte[] img = null;

FileStream fs = new FileStream(imgLoc, FileMode.Open, FileAccess.Read);

BinaryReader br = new BinaryReader(fs);

img = br.ReadBytes((int)fs.Length);

\_currentFamilyMembers.PhotoFamilyMember = img;

}

if (imgLoc == "очистить") \_currentFamilyMembers.PhotoFamilyMember = null;

FamilyFinancesEntities.GetContext().SaveChanges();

Classes.ClassFrame.frmObj.Navigate(new PageFamilyMembers());

MessageBox.Show("Член семьи успешно изменён!");

}

catch (Exception ex)

{

MessageBox.Show(ex.Message.ToString());

}

}

}

private void BtnCancelFamilyMember\_Click(object sender, RoutedEventArgs e)

{

Classes.ClassFrame.frmObj.Navigate(new PageFamilyMembers());

}

private void LoadImage(object sender, RoutedEventArgs e)

{

try

{

OpenFileDialog dlg = new OpenFileDialog

{

Filter = "JPG Files (\*.jpg)|\*.jpg|PNG Files (\*.png)|\*.png|GIF Files (\*.gif)|\*.gif|All Files (\*.\*)|\*.\*"

};

if (dlg.ShowDialog() == true)

{

imgLoc = dlg.FileName.ToString();

ImageFamilyMember.Source = new BitmapImage(new Uri(imgLoc));

}

}

catch (Exception ex)

{

MessageBox.Show(ex.Message);

}

}

private void ClearImage(object sender, RoutedEventArgs e)

{

ImageFamilyMember.Source = (ImageSource)ImageFamilyMember.FindResource("UnknownFamilyMember");

imgLoc = "очистить";

}

}

}

PageAddIncomes.xaml

Page x:Class="FamilyFinances.Pages.PageAddIncomes"

xmlns="http://schemas.microsoft.com/winfx/2006/xaml/presentation"

xmlns:x="http://schemas.microsoft.com/winfx/2006/xaml"

xmlns:mc="http://schemas.openxmlformats.org/markup-compatibility/2006"

xmlns:d="http://schemas.microsoft.com/expression/blend/2008"

xmlns:local="clr-namespace:FamilyFinances.Pages"

mc:Ignorable="d"

d:DesignHeight="450" d:DesignWidth="800"

Title="PageAddIncomes">

<Grid>

<Grid.RowDefinitions>

<RowDefinition Height="1\*"/>

<RowDefinition Height="5\*"/>

<RowDefinition Height="1\*"/>

</Grid.RowDefinitions>

<StackPanel Orientation="Vertical" Grid.Row="0" HorizontalAlignment="Center" VerticalAlignment="Center">

<TextBlock Text="Добавление нового дохода" Name="TitletxIncome" FontSize="28" FontWeight="Bold" Foreground="Black"/>

</StackPanel>

<StackPanel Grid.Row="1" Orientation="Vertical" Width="600" Margin="15" VerticalAlignment="Center">

<StackPanel Orientation="Horizontal" HorizontalAlignment="Center">

<TextBlock Text="Тип дохода" Width="200" FontSize="20" Margin="7" FontWeight="DemiBold" Foreground="Black"/>

<TextBox Text="{Binding Type\_income}" Width="350" FontSize="18" Margin="7" FontWeight="DemiBold"/>

</StackPanel>

<Separator Opacity="0" Height="1.5"/>

<StackPanel Orientation="Horizontal" HorizontalAlignment="Center">

<TextBlock Text="Сумма дохода" Width="200" FontSize="20" Margin="7" FontWeight="DemiBold" Foreground="Black"/>

<TextBox Text="{Binding IncomeSum}" Width="350" FontSize="18" Margin="7" FontWeight="DemiBold"/>

</StackPanel>

<Separator Opacity="0" Height="1.5"/>

<StackPanel Orientation="Horizontal" HorizontalAlignment="Center">

<TextBlock Text="Получатель дохода" Width="200" FontSize="20" Margin="7,7,7,7" FontWeight="DemiBold" Foreground="Black"/>

<ComboBox Name="CmbName" SelectedValue="{Binding idFamilyMember}" Width="350" FontSize="18" Margin="7" FontWeight="DemiBold" />

</StackPanel>

</StackPanel>

<StackPanel Grid.Row="2" Orientation="Vertical" VerticalAlignment="Center" HorizontalAlignment="Center">

<StackPanel Orientation="Horizontal" Grid.Row="2" >

<Button Width="230" Height="44" Content="Добавить" VerticalAlignment="Center" FontSize="20" FontWeight="DemiBold" Name="BtnAddIncome" Click="BtnAddIncome\_Click" Margin="25,0" Background="#82cea2" Foreground="Black"/>

<Button Width="230" Height="44" Content="Отмена" VerticalAlignment="Center" FontSize="20" FontWeight="DemiBold" Name="BtnCancelIncome" Click="BtnCancelIncome\_Click" Background="#82cea2" Foreground="Black" Margin="0,0,25,0"/>

</StackPanel>

</StackPanel>

</Grid>

</Page>

PageAddIncomes.xaml.cs

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

using System.Windows;

using System.Windows.Controls;

using System.Windows.Data;

using System.Windows.Documents;

using System.Windows.Input;

using System.Windows.Media;

using System.Windows.Media.Imaging;

using System.Windows.Navigation;

using System.Windows.Shapes;

using FamilyFinances.Classes;

namespace FamilyFinances.Pages

{

/// <summary>

/// Логика взаимодействия для PageAddIncomes.xaml

/// </summary>

public partial class PageAddIncomes : Page

{

private Incomes \_currentIncomes = new Incomes();

public PageAddIncomes(Incomes selectedIncome)

{

InitializeComponent();

if (selectedIncome != null)

{

\_currentIncomes = selectedIncome;

TitletxIncome.Text = "Изменение дохода";

BtnAddIncome.Content = "Изменить";

}

DataContext = \_currentIncomes;

CmbName.ItemsSource = FamilyFinancesEntities.GetContext().FamilyMembers.ToList();

CmbName.SelectedValuePath = "idFamilyMember";

CmbName.DisplayMemberPath = "Name";

}

private void BtnAddIncome\_Click(object sender, RoutedEventArgs e)

{

StringBuilder error = new StringBuilder();

if (string.IsNullOrWhiteSpace(\_currentIncomes.Type\_income)) error.AppendLine("Укажите тип дохода");

if (string.IsNullOrWhiteSpace(Convert.ToString(\_currentIncomes.IncomeSum))) error.AppendLine("Укажите сумму дохода");

if (string.IsNullOrWhiteSpace(Convert.ToString(\_currentIncomes.idFamilyMember))) error.AppendLine("Укажите имя получателя");

if (error.Length > 0)

{

MessageBox.Show(error.ToString());

return;

}

if (\_currentIncomes.idIncome == 0)

{

FamilyFinancesEntities.GetContext().Incomes.Add(\_currentIncomes);

try

{

FamilyFinancesEntities.GetContext().SaveChanges();

Classes.ClassFrame.frmObj.Navigate(new PageIncomes());

MessageBox.Show("Новый доход успешно добавлен!");

}

catch (Exception ex)

{

MessageBox.Show(ex.Message.ToString());

}

}

else

{

try

{

FamilyFinancesEntities.GetContext().SaveChanges();

Classes.ClassFrame.frmObj.Navigate(new PageIncomes());

MessageBox.Show("Доход успешно изменён!");

}

catch (Exception ex)

{

MessageBox.Show(ex.Message.ToString());

}

}

}

private void BtnCancelIncome\_Click(object sender, RoutedEventArgs e)

{

Classes.ClassFrame.frmObj.Navigate(new PageIncomes());

}

}

}

PageExpenses.xaml

Page x:Class="FamilyFinances.Pages.PageExpenses"

xmlns="http://schemas.microsoft.com/winfx/2006/xaml/presentation"

xmlns:x="http://schemas.microsoft.com/winfx/2006/xaml"

xmlns:mc="http://schemas.openxmlformats.org/markup-compatibility/2006"

xmlns:d="http://schemas.microsoft.com/expression/blend/2008"

xmlns:local="clr-namespace:FamilyFinances.Pages"

mc:Ignorable="d"

d:DesignHeight="750" d:DesignWidth="800"

Title="PageExpenses">

<Grid>

<Grid.RowDefinitions>

<RowDefinition Height="1.5\*"/>

<RowDefinition Height="0.6\*"/>

<RowDefinition Height="5\*"/>

</Grid.RowDefinitions>

<StackPanel Orientation="Horizontal" Grid.Row="1" HorizontalAlignment="Center" VerticalAlignment="Center">

<TextBlock Text="Поиск по типу расхода" FontSize = "20" FontWeight="Bold" Foreground="Black" Height="25" />

<TextBox Height="30" Width="180" FontSize="20" Foreground="Black" FontWeight="Bold" Margin="7" Name="SearchExpense" TextChanged="SearchExpense\_TextChanged" />

</StackPanel>

<StackPanel Orientation="Vertical" Grid.Row="0" HorizontalAlignment="Center" VerticalAlignment="Center">

<TextBlock Text="Расходы" FontSize="30" FontWeight="Bold" Foreground="Black"/>

</StackPanel>

<ListView Name="LTVIExpenses" Grid.Row="2" ScrollViewer.HorizontalScrollBarVisibility="Disabled"

Width="auto"

FontSize="16" FontWeight="DemiBold"

Foreground="Black" Margin="20">

<ListView.ContextMenu>

<ContextMenu>

<MenuItem Header="Добавить" Name="MenuAddExpense" Click="MenuAddExpense\_Click" FontSize="18" FontWeight="DemiBold" Foreground="Black" />

<MenuItem Header="Редактировать" Name="MenuEditExpense" Click="MenuEditExpense\_Click" FontSize="18" FontWeight="DemiBold" Foreground="Black" />

<Separator/>

<MenuItem Header="Сортировка по типу" FontSize="18" FontWeight="DemiBold" Foreground="Black" >

<MenuItem Header="От А до Я" Name="MenuSortTypeV1" Click="MenuSortTypeV1\_Click" FontSize="18" FontWeight="DemiBold" Foreground="Black" />

<MenuItem Header="От Я до А" Name="MenuSortTypeV2" Click="MenuSortTypeV2\_Click" FontSize="18" FontWeight="DemiBold" Foreground="Black" />

<Separator/>

<MenuItem Header="Cброс" Name="MenuSortСlear" Click="MenuSortСlear\_Click" FontSize="18" FontWeight="DemiBold" Foreground="Black" />

</MenuItem>

<Separator/>

<MenuItem Header="Удалить" Name="MenuDelExpense" Click="MenuDelExpense\_Click" FontSize="18" FontWeight="DemiBold" Foreground="Black" />

</ContextMenu>

</ListView.ContextMenu>

<ListView.ItemsPanel>

<ItemsPanelTemplate>

<WrapPanel Orientation="Horizontal" HorizontalAlignment="Center" />

</ItemsPanelTemplate>

</ListView.ItemsPanel>

<ListView.ItemTemplate>

<DataTemplate>

<Grid Width="975" Height="80">

<Grid.RowDefinitions>

<RowDefinition Height="0.5\*"/>

<RowDefinition Height="3\*"/>

</Grid.RowDefinitions>

<Grid.ColumnDefinitions>

<ColumnDefinition Width="0.6\*"/>

<ColumnDefinition Width="4\*"/>

<ColumnDefinition Width="2.5\*"/>

</Grid.ColumnDefinitions>

<StackPanel Grid.Row="0" Grid.Column="0" Grid.RowSpan="2" HorizontalAlignment="Center" VerticalAlignment="Center">

<Label Content="{Binding idExpense}" FontSize="19" Foreground="Black" VerticalAlignment="Center" HorizontalAlignment="Left" />

</StackPanel>

<StackPanel Grid.Row="0" Grid.RowSpan="2" Grid.Column="1" Orientation="Horizontal" HorizontalAlignment="Left">

<StackPanel Orientation="Vertical" HorizontalAlignment="Left">

<StackPanel Orientation="Horizontal">

<Label Content="Тип расхода:" FontSize="19" Foreground="Black"/>

<Label Content="{Binding Type\_of\_expenses}" FontSize="19" Foreground="Black"/>

</StackPanel>

<StackPanel Orientation="Horizontal">

<Label Content="Имя плательщика:" FontSize="19" Foreground="Black" VerticalAlignment="Center"/>

<Label Content="{Binding FamilyMembers.Name}" FontSize="19" Foreground="Black"/>

</StackPanel>

</StackPanel>

</StackPanel>

<StackPanel Grid.Row="0" Grid.RowSpan="2" Grid.Column="2" Orientation="Horizontal" HorizontalAlignment="Left">

<StackPanel Orientation="Horizontal" HorizontalAlignment="Center" VerticalAlignment="Center">

<Label Content="Сумма расхода:" FontSize="19" Foreground="Black"/>

<Label Content="{Binding Expenses\_sum}" FontSize="19" Foreground="Black" />

<Label Content="руб." FontSize="19" Foreground="Black"/>

</StackPanel>

</StackPanel>

</Grid>

</DataTemplate>

</ListView.ItemTemplate>

</ListView>

</Grid>

</Page>

PageExpenses.xaml.cs

using FamilyFinances.Classes;

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

using System.Windows;

using System.Windows.Controls;

using System.Windows.Data;

using System.Windows.Documents;

using System.Windows.Input;

using System.Windows.Media;

using System.Windows.Media.Imaging;

using System.Windows.Navigation;

using System.Windows.Shapes;

namespace FamilyFinances.Pages

{

/// <summary>

/// Логика взаимодействия для PageExpenses.xaml

/// </summary>

public partial class PageExpenses : Page

{

public PageExpenses()

{

InitializeComponent();

LTVIExpenses.ItemsSource = FamilyFinancesEntities.GetContext().Expenses.ToList();

}

private void MenuAddExpense\_Click(object sender, RoutedEventArgs e)

{

Classes.ClassFrame.frmObj.Navigate(new PageAddExpenses(null));

}

private void MenuEditExpense\_Click(object sender, RoutedEventArgs e)

{

Classes.ClassFrame.frmObj.Navigate(new PageAddExpenses((Expenses)LTVIExpenses.SelectedItem));

}

private void MenuSortTypeV1\_Click(object sender, RoutedEventArgs e)

{

LTVIExpenses.ItemsSource = FamilyFinancesEntities.GetContext().Expenses.OrderBy(x => x.Type\_of\_expenses).ToList();

}

private void MenuSortTypeV2\_Click(object sender, RoutedEventArgs e)

{

LTVIExpenses.ItemsSource = FamilyFinancesEntities.GetContext().Expenses.OrderByDescending(x => x.Type\_of\_expenses).ToList();

}

private void MenuSortСlear\_Click(object sender, RoutedEventArgs e)

{

LTVIExpenses.ItemsSource = FamilyFinancesEntities.GetContext().Expenses.ToList();

}

private void MenuDelExpense\_Click(object sender, RoutedEventArgs e)

{

var expenseForRemoving = LTVIExpenses.SelectedItems.Cast<Expenses>().ToList();

if (MessageBox.Show($"Вы точно хотите удалить следующие {expenseForRemoving.Count()} записи?", "Внимание", MessageBoxButton.YesNo, MessageBoxImage.Question) == MessageBoxResult.Yes)

{

try

{

FamilyFinancesEntities.GetContext().Expenses.RemoveRange(expenseForRemoving);

FamilyFinancesEntities.GetContext().SaveChanges();

MessageBox.Show("Данные удалены!");

LTVIExpenses.ItemsSource = FamilyFinancesEntities.GetContext().Expenses.ToList();

}

catch (Exception ex)

{

MessageBox.Show(ex.Message.ToString());

}

}

}

private void SearchExpense\_TextChanged(object sender, TextChangedEventArgs e)

{

if (LTVIExpenses.ItemsSource != null)

{

LTVIExpenses.ItemsSource = FamilyFinancesEntities.GetContext().Expenses.Where(x => x.Type\_of\_expenses.ToLower().Contains(SearchExpense.Text.ToLower())).ToList();

}

if (SearchExpense.Text.Count() == 0) LTVIExpenses.ItemsSource = FamilyFinancesEntities.GetContext().Expenses.ToList();

}

}

}

PageFamilyMembers.xaml

<Page x:Class="FamilyFinances.Pages.PageFamilyMembers"

xmlns="http://schemas.microsoft.com/winfx/2006/xaml/presentation"

xmlns:x="http://schemas.microsoft.com/winfx/2006/xaml"

xmlns:mc="http://schemas.openxmlformats.org/markup-compatibility/2006"

xmlns:d="http://schemas.microsoft.com/expression/blend/2008"

xmlns:local="clr-namespace:FamilyFinances.Pages"

mc:Ignorable="d"

d:DesignHeight="750" d:DesignWidth="750"

Title="PageFamilyMembers">

<Grid>

<Grid.RowDefinitions>

<RowDefinition Height="1.5\*"/>

<RowDefinition Height="0.6\*"/>

<RowDefinition Height="5\*"/>

</Grid.RowDefinitions>

<StackPanel Orientation="Horizontal" Grid.Row="1" HorizontalAlignment="Center" VerticalAlignment="Center">

<TextBlock Text="Поиск по имени" FontSize = "20" FontWeight="Bold" Foreground="Black" Height="25" />

<TextBox Height="30" Width="180" FontSize="20" Foreground="Black" FontWeight="Bold" Margin="7" Name="SearchFamilyMember" TextChanged="SearchFamilyMember\_TextChanged" />

</StackPanel>

<StackPanel Orientation="Vertical" Grid.Row="0" HorizontalAlignment="Center" VerticalAlignment="Center">

<TextBlock Text="Члены Семьи" FontSize="30" FontWeight="Bold" Foreground="Black"/>

</StackPanel>

<ListView Name="LTVFamilyMembers" Grid.Row="2" ScrollViewer.HorizontalScrollBarVisibility="Disabled"

Width="auto"

FontSize="16" FontWeight="DemiBold"

Foreground="Black" Margin="20">

<ListView.ContextMenu>

<ContextMenu>

<MenuItem Header="Добавить" Name="MenuAddFamilyMember" Click="MenuAddFamilyMember\_Click" FontSize="18" FontWeight="DemiBold" Foreground="Black" />

<MenuItem Header="Редактировать" Name="MenuEditFamilyMember" Click="MenuEditFamilyMember\_Click" FontSize="18" FontWeight="DemiBold" Foreground="Black" />

<Separator/>

<MenuItem Header="Сортировка по имени" FontSize="18" FontWeight="DemiBold" Foreground="Black" >

<MenuItem Header="От А до Я" Name="MenuSortNameV1" Click="MenuSortNameV1\_Click" FontSize="18" FontWeight="DemiBold" Foreground="Black" />

<MenuItem Header="От Я до А" Name="MenuSortNameV2" Click="MenuSortNameV2\_Click" FontSize="18" FontWeight="DemiBold" Foreground="Black" />

<Separator/>

<MenuItem Header="Cброс" Name="MenuSortСlear" Click="MenuSortСlear\_Click" FontSize="18" FontWeight="DemiBold" Foreground="Black" />

</MenuItem>

<MenuItem Header="Фильтр по возрасту" FontSize="18" FontWeight="DemiBold" Foreground="Black" >

<MenuItem Header="Младше 18" Name="MenuFilterAge1" Click="MenuFilterAge1\_Click" FontSize="18" FontWeight="DemiBold" Foreground="Black" />

<MenuItem Header="Старше 18 " Name="MenuFilterAge2" Click="MenuFilterAge2\_Click" FontSize="18" FontWeight="DemiBold" Foreground="Black" />

<Separator/>

<MenuItem Header="Сброс" Name="MenuFilterClear" Click="MenuFilterClear\_Click" FontSize="18" FontWeight="DemiBold" Foreground="Black" />

</MenuItem>

<Separator/>

<MenuItem Header="Удалить" Name="MenuDelFamilyMember" Click="MenuDelFamilyMember\_Click" FontSize="18" FontWeight="DemiBold" Foreground="Black" />

</ContextMenu>

</ListView.ContextMenu>

<ListView.ItemsPanel>

<ItemsPanelTemplate>

<WrapPanel Orientation="Horizontal" HorizontalAlignment="Center" />

</ItemsPanelTemplate>

</ListView.ItemsPanel>

<ListView.ItemTemplate>

<DataTemplate>

<Grid>

<Grid.ColumnDefinitions>

<ColumnDefinition Width="1\*"/>

<ColumnDefinition Width="2\*"/>

<ColumnDefinition Width="1\*"/>

</Grid.ColumnDefinitions>

<Image Grid.Column="1" Width="100" Margin="30,5,30,5" Source="{Binding PhotoFamilyMember, TargetNullValue={StaticResource UnknownFamilyMember}}"/>

<TextBlock Text="{Binding idFamilyMember}" Grid.Column="0" Margin="20,0,20,0" VerticalAlignment="Center"/>

<StackPanel Grid.Column="2" Orientation="Vertical" Width="250" Margin="0,30,0,0">

<StackPanel Orientation="Horizontal" VerticalAlignment="Center">

<TextBlock Text = "Имя: "/>

<TextBlock Text="{Binding Name}"/>

</StackPanel>

<StackPanel Orientation="Horizontal" VerticalAlignment="Center">

<TextBlock Text = "Возраст: "/>

<TextBlock Text="{Binding Age}" />

</StackPanel>

</StackPanel>

</Grid>

</DataTemplate>

</ListView.ItemTemplate>

</ListView>

</Grid>

</Page>

PageFamilyMembers.xaml.cs

using FamilyFinances.Classes;

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

using System.Windows;

using System.Windows.Controls;

using System.Windows.Data;

using System.Windows.Documents;

using System.Windows.Input;

using System.Windows.Media;

using System.Windows.Media.Imaging;

using System.Windows.Navigation;

using System.Windows.Shapes;

namespace FamilyFinances.Pages

{

/// <summary>

/// Логика взаимодействия для PageFamilyMembers.xaml

/// </summary>

public partial class PageFamilyMembers : Page

{

public PageFamilyMembers()

{

InitializeComponent();

LTVFamilyMembers.ItemsSource = FamilyFinancesEntities.GetContext().FamilyMembers.ToList();

}

private void MenuAddFamilyMember\_Click(object sender, RoutedEventArgs e)

{

Classes.ClassFrame.frmObj.Navigate(new PageAddFamilyMembers(null));

}

private void MenuEditFamilyMember\_Click(object sender, RoutedEventArgs e)

{

Classes.ClassFrame.frmObj.Navigate(new PageAddFamilyMembers((FamilyMembers)LTVFamilyMembers.SelectedItem));

}

private void MenuSortNameV1\_Click(object sender, RoutedEventArgs e)

{

LTVFamilyMembers.ItemsSource = FamilyFinancesEntities.GetContext().FamilyMembers.OrderBy(x => x.Name).ToList();

}

private void MenuSortNameV2\_Click(object sender, RoutedEventArgs e)

{

LTVFamilyMembers.ItemsSource = FamilyFinancesEntities.GetContext().FamilyMembers.OrderByDescending(x => x.Name).ToList();

}

private void MenuSortСlear\_Click(object sender, RoutedEventArgs e)

{

LTVFamilyMembers.ItemsSource = FamilyFinancesEntities.GetContext().FamilyMembers.ToList();

}

private void MenuFilterAge1\_Click(object sender, RoutedEventArgs e)

{

LTVFamilyMembers.ItemsSource = FamilyFinancesEntities.GetContext().FamilyMembers.Where(x => x.Age <= 17).ToList();

}

private void MenuFilterAge2\_Click(object sender, RoutedEventArgs e)

{

LTVFamilyMembers.ItemsSource = FamilyFinancesEntities.GetContext().FamilyMembers.Where(x => x.Age >= 18).ToList();

}

private void MenuFilterClear\_Click(object sender, RoutedEventArgs e)

{

LTVFamilyMembers.ItemsSource = FamilyFinancesEntities.GetContext().FamilyMembers.ToList();

}

private void MenuDelFamilyMember\_Click(object sender, RoutedEventArgs e)

{

var familyMemberForRemoving = LTVFamilyMembers.SelectedItems.Cast<FamilyMembers>().ToList();

if (MessageBox.Show($"Вы точно хотите удалить следующие {familyMemberForRemoving.Count()} записи?", "Внимание", MessageBoxButton.YesNo, MessageBoxImage.Question) == MessageBoxResult.Yes)

{

try

{

FamilyFinancesEntities.GetContext().FamilyMembers.RemoveRange(familyMemberForRemoving);

FamilyFinancesEntities.GetContext().SaveChanges();

MessageBox.Show("Данные удалены!");

LTVFamilyMembers.ItemsSource = FamilyFinancesEntities.GetContext().FamilyMembers.ToList();

}

catch (Exception ex)

{

MessageBox.Show(ex.Message.ToString());

}

}

}

private void SearchFamilyMember\_TextChanged(object sender, TextChangedEventArgs e)

{

if (LTVFamilyMembers.ItemsSource != null)

{

LTVFamilyMembers.ItemsSource = FamilyFinancesEntities.GetContext().FamilyMembers.Where(x => x.Name.ToLower().Contains(SearchFamilyMember.Text.ToLower())).ToList();

}

if (SearchFamilyMember.Text.Count() == 0) LTVFamilyMembers.ItemsSource = FamilyFinancesEntities.GetContext().FamilyMembers.ToList();

}

}

}

PageIncomes.xaml

<Page x:Class="FamilyFinances.Pages.PageIncomes"

xmlns="http://schemas.microsoft.com/winfx/2006/xaml/presentation"

xmlns:x="http://schemas.microsoft.com/winfx/2006/xaml"

xmlns:mc="http://schemas.openxmlformats.org/markup-compatibility/2006"

xmlns:d="http://schemas.microsoft.com/expression/blend/2008"

xmlns:local="clr-namespace:FamilyFinances.Pages"

mc:Ignorable="d"

d:DesignHeight="750" d:DesignWidth="750"

Title="PageIncomes">

<Grid>

<Grid.RowDefinitions>

<RowDefinition Height="1.5\*"/>

<RowDefinition Height="0.6\*"/>

<RowDefinition Height="5\*"/>

</Grid.RowDefinitions>

<StackPanel Orientation="Horizontal" Grid.Row="1" HorizontalAlignment="Center" VerticalAlignment="Center">

<TextBlock Text="Поиск по типу дохода" FontSize = "20" FontWeight="Bold" Foreground="Black" Height="25" />

<TextBox Height="30" Width="180" FontSize="20" Foreground="Black" FontWeight="Bold" Margin="7" Name="SearchIncome" TextChanged="SearchIncome\_TextChanged" />

</StackPanel>

<StackPanel Orientation="Vertical" Grid.Row="0" HorizontalAlignment="Center" VerticalAlignment="Center">

<TextBlock Text="Доходы" FontSize="30" FontWeight="Bold" Foreground="Black"/>

</StackPanel>

<ListView Name="LTVIncomes" Grid.Row="2" ScrollViewer.HorizontalScrollBarVisibility="Disabled"

Width="auto"

FontSize="16" FontWeight="DemiBold"

Foreground="Black" Margin="20">

<ListView.ContextMenu>

<ContextMenu>

<MenuItem Header="Добавить" Name="MenuAddIncome" Click="MenuAddIncome\_Click" FontSize="18" FontWeight="DemiBold" Foreground="Black" />

<MenuItem Header="Редактировать" Name="MenuEditIncome" Click="MenuEditIncome\_Click" FontSize="18" FontWeight="DemiBold" Foreground="Black" />

<Separator/>

<MenuItem Header="Экспорт в Excel" Name="MenuExcelIncome" Click="MenuExcelIncome\_Click" FontSize="18" FontWeight="DemiBold" Foreground="Black" />

<Separator/>

<MenuItem Header="Сортировка по типу" FontSize="18" FontWeight="DemiBold" Foreground="Black" >

<MenuItem Header="От А до Я" Name="MenuSortTypeV1" Click="MenuSortTypeV1\_Click" FontSize="18" FontWeight="DemiBold" Foreground="Black" />

<MenuItem Header="От Я до А" Name="MenuSortTypeV2" Click="MenuSortTypeV2\_Click" FontSize="18" FontWeight="DemiBold" Foreground="Black" />

<Separator/>

<MenuItem Header="Cброс" Name="MenuSortСlear" Click="MenuSortСlear\_Click" FontSize="18" FontWeight="DemiBold" Foreground="Black" />

</MenuItem>

<Separator/>

<MenuItem Header="Удалить" Name="MenuDelIncome" Click="MenuDelIncome\_Click" FontSize="18" FontWeight="DemiBold" Foreground="Black" />

</ContextMenu>

</ListView.ContextMenu>

<ListView.ItemsPanel>

<ItemsPanelTemplate>

<WrapPanel Orientation="Horizontal" HorizontalAlignment="Center" />

</ItemsPanelTemplate>

</ListView.ItemsPanel>

<ListView.ItemTemplate>

<DataTemplate>

<Grid Width="975" Height="80">

<Grid.RowDefinitions>

<RowDefinition Height="0.5\*"/>

<RowDefinition Height="3\*"/>

</Grid.RowDefinitions>

<Grid.ColumnDefinitions>

<ColumnDefinition Width="0.6\*"/>

<ColumnDefinition Width="4\*"/>

<ColumnDefinition Width="2.5\*"/>

</Grid.ColumnDefinitions>

<StackPanel Grid.Row="0" Grid.Column="0" Grid.RowSpan="2" HorizontalAlignment="Center" VerticalAlignment="Center">

<Label Content="{Binding idIncome}" FontSize="19" Foreground="Black" VerticalAlignment="Center" HorizontalAlignment="Left" />

</StackPanel>

<StackPanel Grid.Row="0" Grid.RowSpan="2" Grid.Column="1" Orientation="Horizontal" HorizontalAlignment="Left">

<StackPanel Orientation="Vertical" HorizontalAlignment="Left">

<StackPanel Orientation="Horizontal">

<Label Content="Тип дохода:" FontSize="19" Foreground="Black"/>

<Label Content="{Binding Type\_income}" FontSize="19" Foreground="Black"/>

</StackPanel>

<StackPanel Orientation="Horizontal">

<Label Content="Получатель дохода:" FontSize="19" Foreground="Black" VerticalAlignment="Center"/>

<Label Content="{Binding FamilyMembers.Name}" FontSize="19" Foreground="Black"/>

</StackPanel>

</StackPanel>

</StackPanel>

<StackPanel Grid.Row="0" Grid.RowSpan="2" Grid.Column="2" Orientation="Horizontal" HorizontalAlignment="Left">

<StackPanel Orientation="Horizontal" HorizontalAlignment="Center" VerticalAlignment="Center">

<Label Content="Сумма дохода:" FontSize="19" Foreground="Black"/>

<Label Content="{Binding IncomeSum}" FontSize="19" Foreground="Black" />

<Label Content="руб." FontSize="19" Foreground="Black"/>

</StackPanel>

</StackPanel>

</Grid>

</DataTemplate>

</ListView.ItemTemplate>

</ListView>

</Grid>

</Page>

PageIncomes.xaml.cs

using FamilyFinances.Classes;

using System;

using System.Collections.Generic;

using System.IO;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

using System.Windows;

using System.Windows.Controls;

using System.Windows.Data;

using System.Windows.Documents;

using System.Windows.Input;

using System.Windows.Media;

using System.Windows.Media.Imaging;

using System.Windows.Navigation;

using System.Windows.Shapes;

using Excel = Microsoft.Office.Interop.Excel;

namespace FamilyFinances.Pages

{

/// <summary>

/// Логика взаимодействия для PageIncomes.xaml

/// </summary>

public partial class PageIncomes : Page

{

public PageIncomes()

{

InitializeComponent();

LTVIncomes.ItemsSource = FamilyFinancesEntities.GetContext().Incomes.ToList();

}

private void MenuAddIncome\_Click(object sender, RoutedEventArgs e)

{

Classes.ClassFrame.frmObj.Navigate(new PageAddIncomes(null));

}

private void MenuEditIncome\_Click(object sender, RoutedEventArgs e)

{

Classes.ClassFrame.frmObj.Navigate(new PageAddIncomes((Incomes)LTVIncomes.SelectedItem));

}

private void MenuExcelIncome\_Click(object sender, RoutedEventArgs e)

{

Excel.Application excelApp = new Excel.Application();

Excel.Workbook wb = excelApp.Workbooks.Open($"{Directory.GetCurrentDirectory()}\\Доходы.xlsx");

Excel.Worksheet ws = (Excel.Worksheet)wb.Worksheets[1];

int IndexRows = 1;

ws.Cells[1][IndexRows] = "№";

ws.Cells[2][IndexRows] = "Тип дохода";

ws.Cells[3][IndexRows] = "Получатель дохода";

ws.Cells[4][IndexRows] = "Сумма дохода";

Excel.Range headerRange = ws.Range[ws.Cells[1][1], ws.Cells[4][1]];

headerRange.HorizontalAlignment = Excel.XlHAlign.xlHAlignCenter;

headerRange.Font.Bold = true;

headerRange.Borders.LineStyle = Excel.XlLineStyle.xlContinuous;

headerRange.Borders.Weight = Excel.XlBorderWeight.xlThin;

List<Incomes> printItems = new List<Incomes>();

for (int i = 0; i < LTVIncomes.Items.Count; i++) printItems.Add((Incomes)LTVIncomes.Items[i]);

foreach (var item in printItems)

{

ws.Cells[1][IndexRows + 1] = IndexRows;

ws.Cells[2][IndexRows + 1] = item.Type\_income;

ws.Cells[3][IndexRows + 1] = item.IncomeSum;

ws.Cells[4][IndexRows + 1] = item.FamilyMembers.Name;

IndexRows++;

Excel.Range bodyRange = ws.Range[ws.Cells[1][IndexRows], ws.Cells[4][IndexRows]];

bodyRange.HorizontalAlignment = Excel.XlHAlign.xlHAlignCenter;

bodyRange.Borders.LineStyle = Excel.XlLineStyle.xlContinuous;

bodyRange.Borders.Weight = Excel.XlBorderWeight.xlThin;

Excel.Range numbersRange = ws.Range[ws.Cells[2, 1], ws.Cells[IndexRows, 1]];

numbersRange.Font.Italic = true;

}

ws.Columns.AutoFit();

ws.Rows.AutoFit();

excelApp.Visible = true;

}

private void MenuSortTypeV1\_Click(object sender, RoutedEventArgs e)

{

LTVIncomes.ItemsSource = FamilyFinancesEntities.GetContext().Incomes.OrderBy(x => x.Type\_income).ToList();

}

private void MenuSortTypeV2\_Click(object sender, RoutedEventArgs e)

{

LTVIncomes.ItemsSource = FamilyFinancesEntities.GetContext().Incomes.OrderByDescending(x => x.Type\_income).ToList();

}

private void MenuSortСlear\_Click(object sender, RoutedEventArgs e)

{

LTVIncomes.ItemsSource = FamilyFinancesEntities.GetContext().Incomes.ToList();

}

private void MenuDelIncome\_Click(object sender, RoutedEventArgs e)

{

var incomeForRemoving = LTVIncomes.SelectedItems.Cast<Incomes>().ToList();

if (MessageBox.Show($"Вы точно хотите удалить следующие {incomeForRemoving.Count()} записи?", "Внимание", MessageBoxButton.YesNo, MessageBoxImage.Question) == MessageBoxResult.Yes)

{

try

{

FamilyFinancesEntities.GetContext().Incomes.RemoveRange(incomeForRemoving);

FamilyFinancesEntities.GetContext().SaveChanges();

MessageBox.Show("Данные удалены!");

LTVIncomes.ItemsSource = FamilyFinancesEntities.GetContext().Incomes.ToList();

}

catch (Exception ex)

{

MessageBox.Show(ex.Message.ToString());

}

}

}

private void SearchIncome\_TextChanged(object sender, TextChangedEventArgs e)

{

if (LTVIncomes.ItemsSource != null)

{

LTVIncomes.ItemsSource = FamilyFinancesEntities.GetContext().Incomes.Where(x => x.Type\_income.ToLower().Contains(SearchIncome.Text.ToLower())).ToList();

}

if (SearchIncome.Text.Count() == 0) LTVIncomes.ItemsSource = FamilyFinancesEntities.GetContext().Incomes.ToList();

}

}

}

MainWindow.xaml

<Window x:Class="FamilyFinances.MainWindow"

xmlns="http://schemas.microsoft.com/winfx/2006/xaml/presentation"

xmlns:x="http://schemas.microsoft.com/winfx/2006/xaml"

xmlns:d="http://schemas.microsoft.com/expression/blend/2008"

xmlns:mc="http://schemas.openxmlformats.org/markup-compatibility/2006"

xmlns:local="clr-namespace:FamilyFinances"

mc:Ignorable="d"

Title="Семейные расходы" Height="800" Width="1000" WindowStartupLocation="CenterScreen">

<Grid>

<Grid.RowDefinitions>

<RowDefinition Height="1\*"/>

<RowDefinition Height="4\*"/>

<RowDefinition Height="1\*"/>

</Grid.RowDefinitions>

<StackPanel Grid.Row="0" Background="#82cea2">

<Image Width="100" Height="100" HorizontalAlignment="Left" Margin="30,5,0,5" Source="C:\Users\ПК\Desktop\01-05-2023\_02-06-56 (1)\FamilyFinances\FamilyFinances\FamilyFinances\Resources\Money1.jpg"/>

</StackPanel>

<StackPanel Grid.Row="0">

<Image Width="100" Height="100" HorizontalAlignment="Right" Margin="0,5,20,5" Source="C:\Users\ПК\Desktop\01-05-2023\_02-06-56 (1)\FamilyFinances\FamilyFinances\FamilyFinances\Resources\Money2.jpg"/>

</StackPanel>

<StackPanel Grid.Row="0" VerticalAlignment="Center" HorizontalAlignment="Center">

<Label Content="Семейные финансы" FontSize="40" FontWeight="Bold" />

</StackPanel>

<Frame Grid.Row="1" Name="FrmMain" Background="#237a5d" NavigationUIVisibility="Hidden"/>

<StackPanel Grid.Row="2" Orientation="Horizontal" Background="#82cea2"/>

<StackPanel Grid.Row="2" HorizontalAlignment="Center" VerticalAlignment="Center">

<StackPanel Orientation="Horizontal">

<Button Name="BTNFamilyMembers" Click="BTNFamilyMembers\_Click" Content="Члены семьи" Width="230" Height="55" FontSize="28" FontWeight="Bold" Background="#237a5d" Foreground="#82cea2" BorderBrush="Black" BorderThickness="3"/>

<Button Name="BTNIncomes" Click="BTNIncomes\_Click" Content="Доходы" Width="230" Height="55" FontSize="28" FontWeight="Bold" Background="#237a5d" Foreground="#82cea2" BorderBrush="Black" BorderThickness="3" Margin="15,0,0,0"/>

<Button Name="BTNExpenses" Click="BTNExpenses\_Click" Content="Расходы" Width="230" Height="55" FontSize="28" FontWeight="Bold" Background="#237a5d" Foreground="#82cea2" BorderBrush="Black" BorderThickness="3" Margin="15,0,0,0"/>

</StackPanel>

</StackPanel>

</Grid>

</Window>

MainWindow.xaml.cs

using FamilyFinances.Classes;

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

using System.Windows;

using System.Windows.Controls;

using System.Windows.Data;

using System.Windows.Documents;

using System.Windows.Input;

using System.Windows.Media;

using System.Windows.Media.Imaging;

using System.Windows.Navigation;

using System.Windows.Shapes;

using FamilyFinances.Pages;

namespace FamilyFinances

{

/// <summary>

/// Логика взаимодействия для MainWindow.xaml

/// </summary>

public partial class MainWindow : Window

{

public MainWindow()

{

InitializeComponent();

ClassFrame.frmObj = FrmMain;

FrmMain.Navigate(new Pages.PageFamilyMembers());

}

private void BTNFamilyMembers\_Click(object sender, RoutedEventArgs e)

{

Classes.ClassFrame.frmObj.Navigate(new PageFamilyMembers());

}

private void BTNIncomes\_Click(object sender, RoutedEventArgs e)

{

Classes.ClassFrame.frmObj.Navigate(new PageIncomes());

}

private void BTNExpenses\_Click(object sender, RoutedEventArgs e)

{

Classes.ClassFrame.frmObj.Navigate(new PageExpenses());

}

}

}