



Государственное бюджетное образовательное учреждение высшего образования  
Московской области

**ТЕХНОЛОГИЧЕСКИЙ УНИВЕРСИТЕТ**

---

Колледж космического машиностроения и технологии

## **КУРСОВОЙ ПРОЕКТ**

**По дисциплине «Прикладное программирование»**

**Тема: «Разработка приложения «Семейный бюджет»»**

Выполнил студент

Озерова Анастасия Сергеевна

Группа П1-17

\_\_\_\_\_ (подпись)

Проверил преподаватель

Гусятинер Леонид Борисович

\_\_\_\_\_ (подпись)

\_\_\_\_\_ (Дата сдачи работы)

## Содержание

### Оглавление

Глава 1. Теоретическая часть .....	5
1.1. Описание предметной области .....	5
1.2. Описание существующих разработок .....	7
1.2.1. 1С: Деньги 8 .....	8
1.2.2. Alzex Finance .....	11
Глава 2. Проектная часть .....	15
2.1. Диаграмма прецедентов .....	15
2.2. Выбор инструментов .....	17
2.3. Проектирование сценария .....	18
2.4. Диаграмма классов .....	19
2.5. Описание главного модуля .....	20
2.6. Описание спецификаций к модулям .....	23
2.7. Описание модулей .....	23
2.8. Описание тестовых наборов модулей .....	28
2.9. Описание применения средств отладки .....	41
2.10. Анализ оптимальности использования памяти и быстродействия .....	41
Глава 3. Эксплуатационная часть .....	45
3.1. Руководство оператора .....	45
Аннотация .....	45
1. Назначение программы .....	47
2. Условия выполнения программы .....	47
3. Выполнение программы .....	48

4. Сообщение оператору .....	61
Заключение .....	62
Список литературы и интернет источников.....	63
Приложение 1. Модуль fmain .....	64
Приложение 2. Модуль budget.....	65
Приложение 3. Модуль AboutProgram.....	68
Приложение 4. Модуль dm.....	69
Приложение 5. Модуль money .....	72
Приложение 6. Модуль FamilyBudget.....	73

## **Введение**

Целью данного курсового проекта является создание программы по ведению семейного бюджета. Он рассчитан на упрощение жизни людей, приучения их к учету своих финансовых средств, ведь ведение бюджета позволяет более эффективно и главное сознательно распределять средства.

В настоящее время формирование и расходование семейного бюджета становятся особенно актуальными, так как семейный бюджет является неотъемлемой частью бюджета государства. Он является основой благосостояния всего государства и отражает уровень развития экономики.

Первая глава содержит описание предметной области курсового проекта – определение семейного бюджета, а также разобран готовый пример уже существующей разработки по ведению семейного бюджета.

Во второй главе будут рассмотрены и описаны инструменты и модули разработанной программы, структура программной части и листинги ключевых частей программных модулей.

Третья глава представляет собой руководство оператора.

В заключительной части сделаны основные выводы о курсовом проекте и полученных знаниях при работе с ним с приведенным списком использованных ресурсов, а также приложения с листингами программы.

## **Глава 1. Теоретическая часть**

### **1.1. Описание предметной области**

**Семейный бюджет** — план доходов, расходов и накоплений, описывающий возможности всех членов семьи в определённый период времени.

Семейный бюджет фиксирует доходы и расходы семьи за определенные периоды времени, анализирует траты и оптимизирует расходы. Бюджет позволяет прогнозировать доходы и расходы в будущем, ставить финансовые цели и достигать их, делать накопления.

Формы семейного бюджета:

#### **1 - Общий бюджет**

Также называется совместным бюджетом. Состоит из сбора всех денежных средств членов семьи в один, из которого ведутся расходы на все нужды.

#### **2 - Раздельный бюджет**

Общие семейные траты делятся поровну или пропорционально доходам членов семьи: один платит оплачивает коммунальные услуги, другой покупает корм домашним питомцам и так далее. Оставшиеся деньги тратятся на свои нужды.

#### **3 - Частично общий бюджет**

Большая часть доходов членов семьи (примерно 70%) составляет семейный бюджет. Остальные деньги уходят на собственные нужды.

Есть другой вариант этой формы. Все основные траты берет на себя тот, кто больше зарабатывает в семье, а остальные по усмотрению откладываются для финансовых целей.

Бюджет состоит из статей доходов и статей расходов. Это нужно для того, чтобы в последствии соотносить фактические траты с планируемыми.

Статьи дохода составляют все источники доходов: зарплата, подработки, премии и т.д.

Статьи расходов — это все траты, разделенные на категории. Как правило, траты разделяют на 2 категории: обязательные и необязательные, затем делят их на подкатегории. Количество статей расходов может быть любым.

Плюсы ведения семейного бюджета:

- 1) Контроль денежных средств
- 2) Сознательный выбор. На период ведения семейного бюджета видно, во сколько обходится каждая статья расходов, что дает возможность ее скорректировать – уменьшить или увеличить.
- 3) Минимизируется залезание в долги/кредиты, т.к. можно заранее все рассчитать.
- 4) Проще планировать крупные траты.
- 5) Финансовый вектор. Анализируя свой семейный бюджет, можно увидеть свой подход к финансам и куда он ведет, к процветанию или банкротству, и сделать выводы: найти новый источник заработка, формировать подушку безопасности, инвестировать процент от дохода.

Минусы:

- 1) Отмечание трат и планирование занимает время

## 2)Разногласия в семье из-за распоряжения финансами

### 1.2. Описание существующих разработок

Самые первые приложения и сервисы по учету финансов, как правило, просто заменяли собой старые блокноты: позволяли вести учет не на бумаге, а в компьютере. Однако возможностей у электронного учета больше, вследствие чего стала расширяться функциональность: появилась иерархическая система категорий трат и доходов, напоминания об оплате платежей, оповещения об оплате долга, также некоторые аналитические инструменты, возможность планирования бюджета и совместный учет с другими пользователями. Многие приложения стали кросс-платформенными.

Как правило, основными критериями по выбору приложения учета бюджета являются:

1. Гибкость учета: иерархия категорий затрат, группы счетов, мультивалютность, проекты, контрагенты, регулярные платежи
2. Удобство внесения данных: виджеты, парсинг смс, интеграция с банками, сканирование чеков
3. Отчетность: отчеты по категориям расходов, наглядные отчеты с диаграммами по периодам, экспорт, баланс счетов
4. Аналитические инструменты: прогноз накоплений, прогноз затрат
5. Напоминания о долгах и регулярных платежах
6. Бекап(резервное копирование)
7. Также важна стоимость приложения – желательно, бесплатно

### 1.2.1. 1С: Деньги 8

«1С:Деньги 8» — программный продукт компании «1С». Это программа, предназначенная для учета, анализа и управления личными и семейными финансами.

Данный продукт позволяет учитывать наличные и безналичные деньги в любой валюте и по различным местам хранения — так называемым кошелькам.

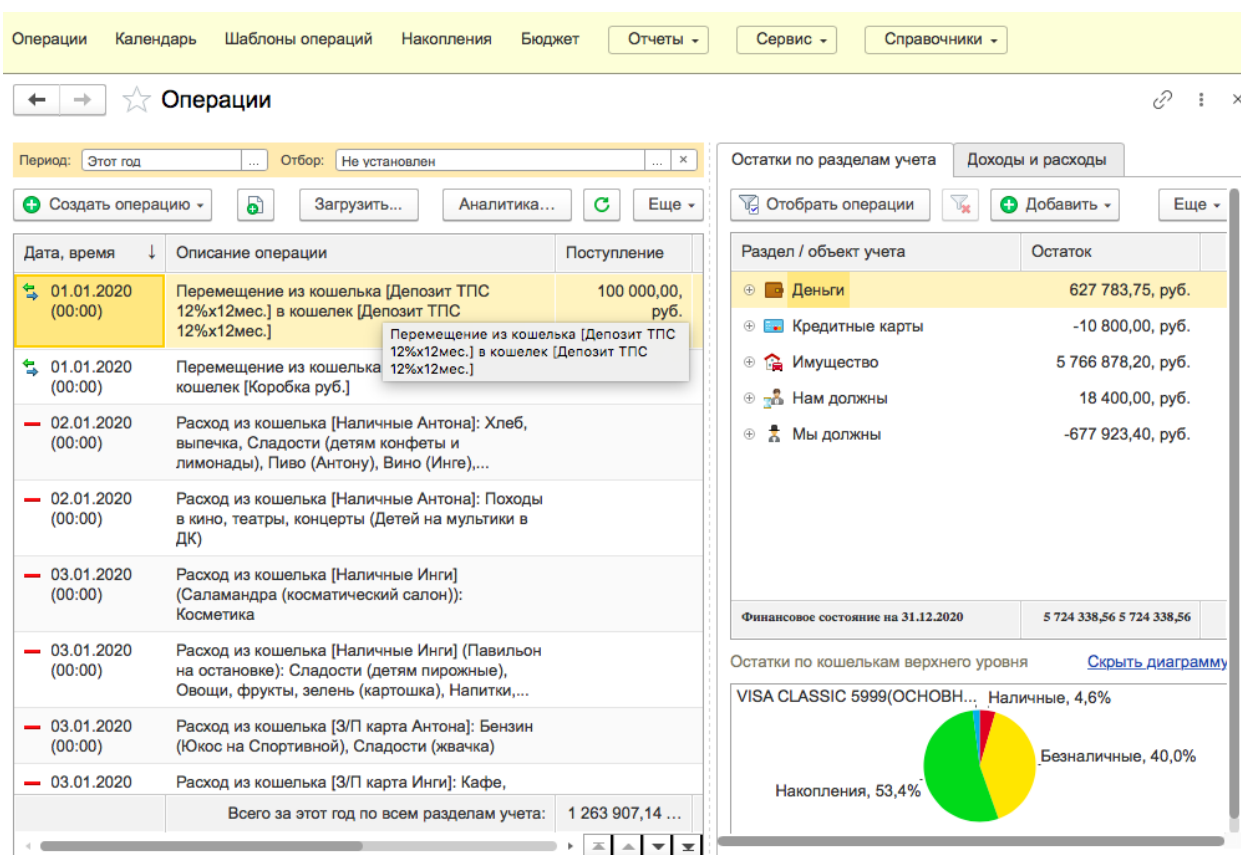


Рис.1 Вкладка «Операции»

При необходимости можно учесть в программе любую нестандартную операцию, например, выполнение работ в долг, погашение выданного займа одновременно имуществом и деньгами и т.д.



Остатки по разделам учета

Доходы и расходы

Отобразить операции

+ Добавить

Еще






Раздел / объект учета	Остаток	
+  Деньги	627 783,75, руб.	
+  Кредитные карты	-10 800,00, руб.	
+  Имущество	5 766 878,20, руб.	
+  Нам должны	18 400,00, руб.	
-  Мы должны	-677 923,40, руб.	
— Взаиморасчеты с родителями	руб.	
— Ипотека в ББК	-677 923,40, руб.	
Финансовое состояние на 31.12.2020	5 724 338,56 5 724 338,56	

Рис.2 Разделы учета

Можно вести учет расходов по статьям и параметрам, чтобы смотреть объем затрат по категориям. Данные по тратам и доходам кошелька можно посмотреть наглядно – в виде диаграммы.

Остатки по кошелькам верхнего уровня

[Скрыть диаграмму](#)

Коробка для общих расх...

Кошельки Антона, 21,3%

Кошельки Инги, 6...

Накопления, 58,4%

Рис.3 Диаграмма кошельков

Еще стоит отметить возможность составления расписания поступлений и платежей — создание шаблонов однотипных финансовых событий. Запланированные операции отображаются в календаре.

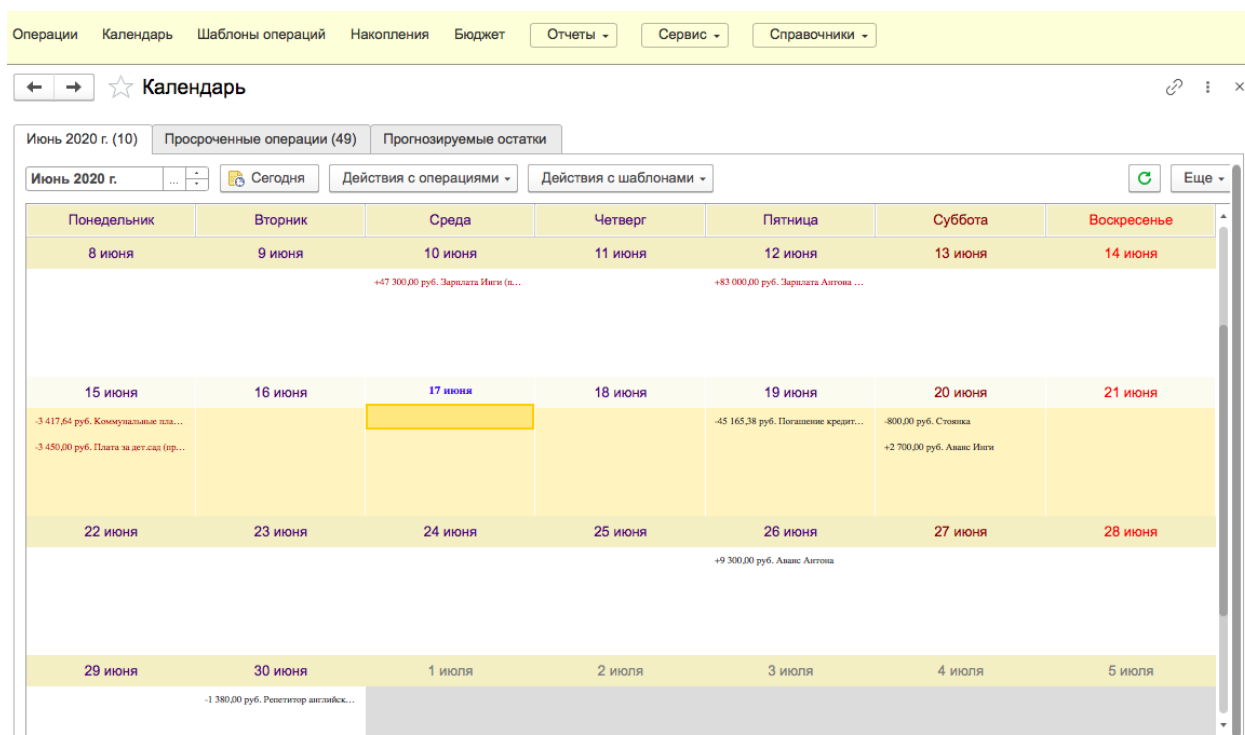


Рис.4 Вкладка «Календарь»

Также есть возможность формирования семейного бюджета, планирования текущих доходов и расходов. Финансовое планирование позволит оптимально распределять расходы, чтобы у семьи всегда имелись средства для нужных покупок и оплаты текущих счетов.

Данная программа также доступна в качестве мобильного приложения iOS и Android.

### 1.2.2. Alzex Finance

Приложение выпускается для мобильных платформ (Android и iOS) и ПК.

Программа поддерживает разные валюты, основные криптовалюты и драгоценные металлы. Можно добавить свои валюты.

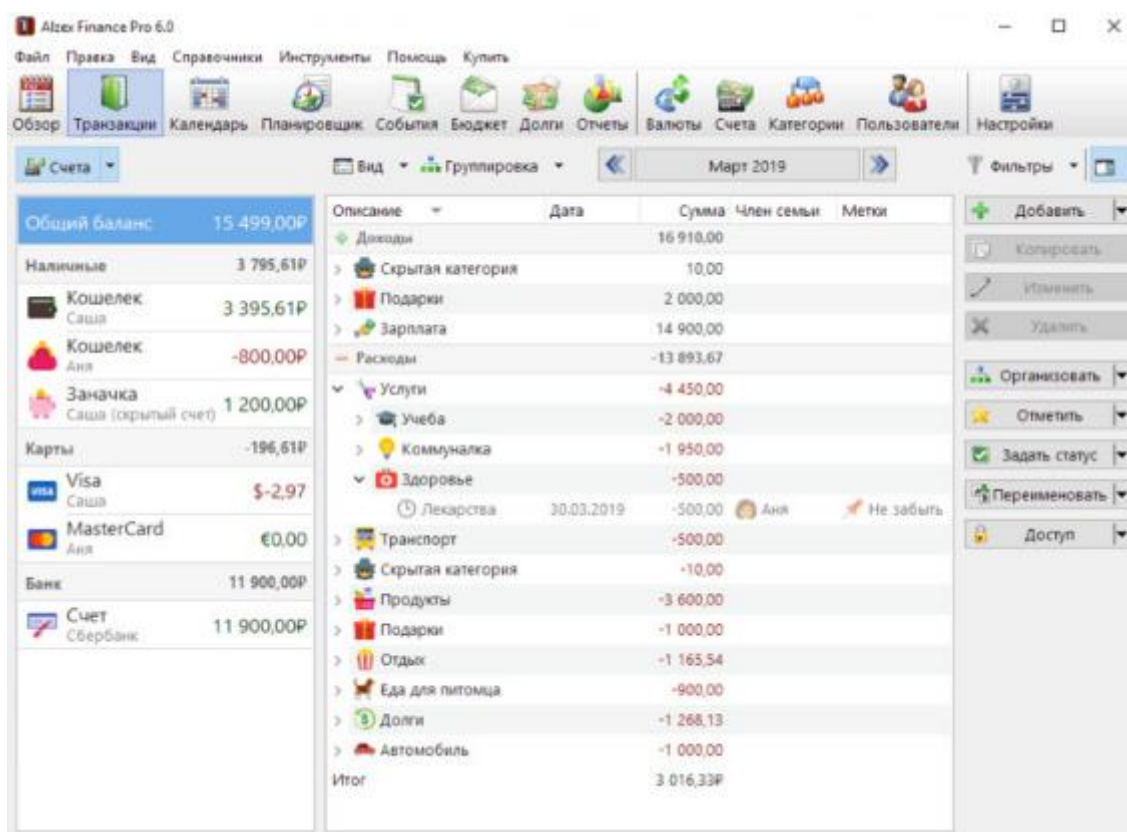
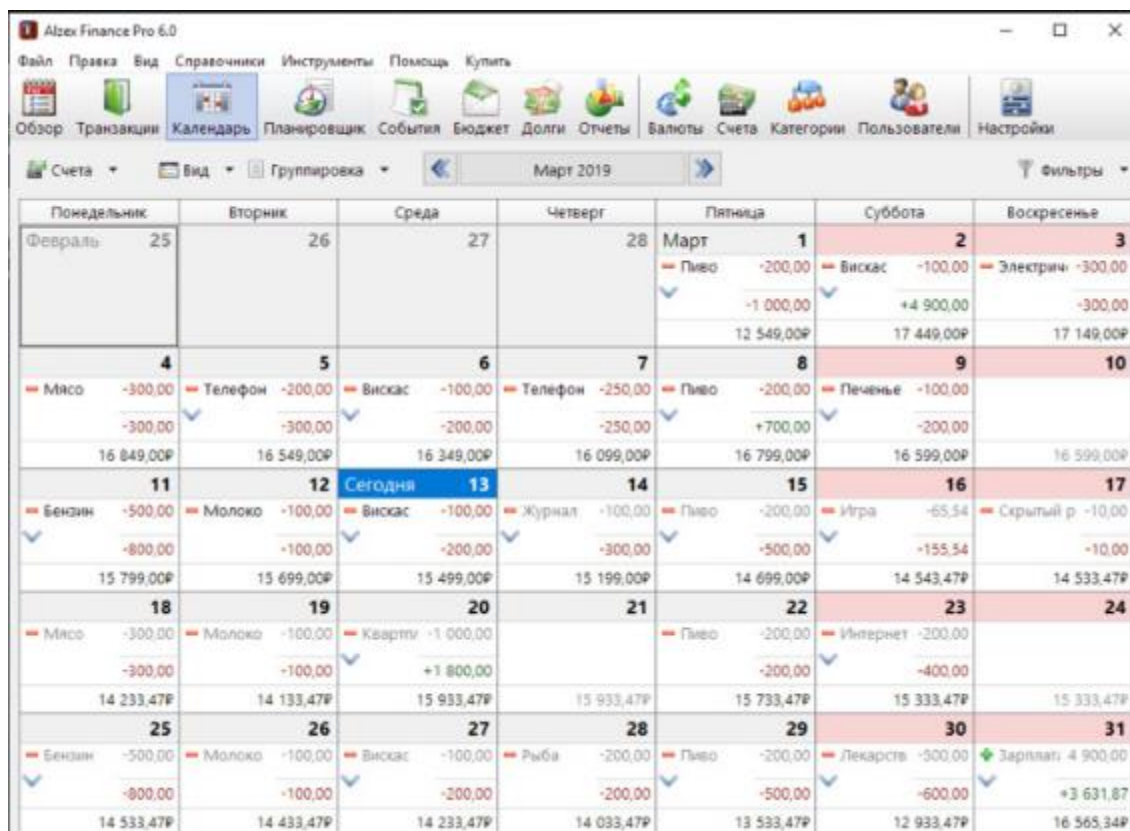


Рис.5 Транзакции

Можно создавать неограниченное количество счетов: кредитные карты, счета в банке, наличные, электронные деньги и так далее. Счета можно

объединять в группы, транзакции распределять по категориям, членам семьи (при совместном использовании), проектам. Планировщик в программе позволяет создавать регулярные транзакции автоматически. Программа учитывает также кредиты и долги.



Понедельник	Вторник	Среда	Четверг	Пятница	Суббота	Воскресенье
Февраль 25	26	27	28	Март 1	2	3
				Пиво -200,00	Вискас -100,00	Электрич -300,00
				-1 000,00	+4 900,00	-300,00
				12 549,00Р	17 449,00Р	17 149,00Р
4	5	6	7	8	9	10
Мясо -300,00	Телефон -200,00	Вискас -100,00	Телефон -250,00	Пиво -200,00	Печенье -100,00	
-300,00	-300,00	-200,00	-250,00	+700,00	-200,00	
16 849,00Р	16 549,00Р	16 349,00Р	16 099,00Р	16 799,00Р	16 599,00Р	16 599,00Р
11	12	Сегодня 13	14	15	16	17
Бензин -500,00	Молоко -100,00	Вискас -100,00	Журнал -100,00	Пиво -200,00	Игра -65,54	Скрытый р -10,00
-800,00	-100,00	-200,00	-300,00	-500,00	-155,54	-10,00
15 799,00Р	15 699,00Р	15 499,00Р	15 199,00Р	14 699,00Р	14 543,47Р	14 533,47Р
18	19	20	21	22	23	24
Мясо -300,00	Молоко -100,00	Кварты -1 000,00		Пиво -200,00	Интернет -200,00	
-300,00	-100,00	+1 800,00		-200,00	-400,00	
14 233,47Р	14 133,47Р	15 933,47Р	15 933,47Р	15 733,47Р	15 333,47Р	15 333,47Р
25	26	27	28	29	30	31
Бензин -500,00	Молоко -100,00	Вискас -100,00	Рыба -200,00	Пиво -200,00	Лекарств -500,00	Зарплата 4 900,00
-800,00	-100,00	-200,00	-200,00	-500,00	-600,00	+3 631,87
14 533,47Р	14 433,47Р	14 233,47Р	14 033,47Р	13 533,47Р	12 933,47Р	16 565,34Р

Рис.6 Календарь учета

Анализировать состояние своих финансов можно через отчеты:

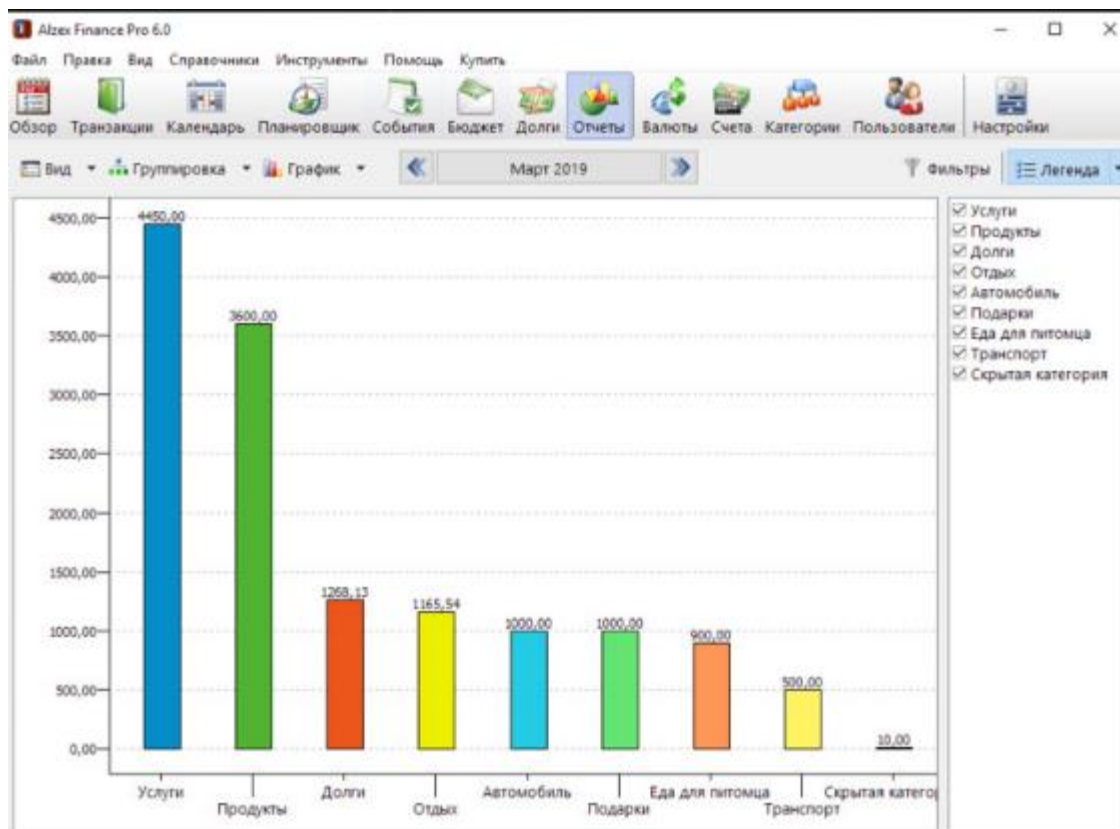


Рис.7 Диаграмма отчета

Alzex Finance поддерживает парсинг СМС от банков(все СМС о транзакциях на карте будут автоматически попадать в список неподтвержденных транзакций), и пользователю останется только указать категорию расходов/доходов и счет. Однако разделить сумму из СМС на несколько категорий нельзя, что является явным минусом.

Данные можно защищать паролем, а также скрывать счета и расходы от других пользователей.

Для мобильных платформ есть бесплатная версия программы с ограничениями, а для Windows — только платная версия с 30-дневным бесплатным периодом для тестирования, стоимостью 990 рублей для персонального использования и 1390 для коммерческого.

Данная программа достаточно удобна в использовании: доступно реализованы финансовые отчеты и транзакции. При возникновении вопросов использования, в программе есть база-образец и справочник.

## Глава 2. Проектная часть

### 2.1. Диаграмма прецедентов

Данная диаграмма отражает варианты использования программы (прецеденты) для распределителя семьи и члена семьи (акторы) с отношениями между элементами диаграммы.

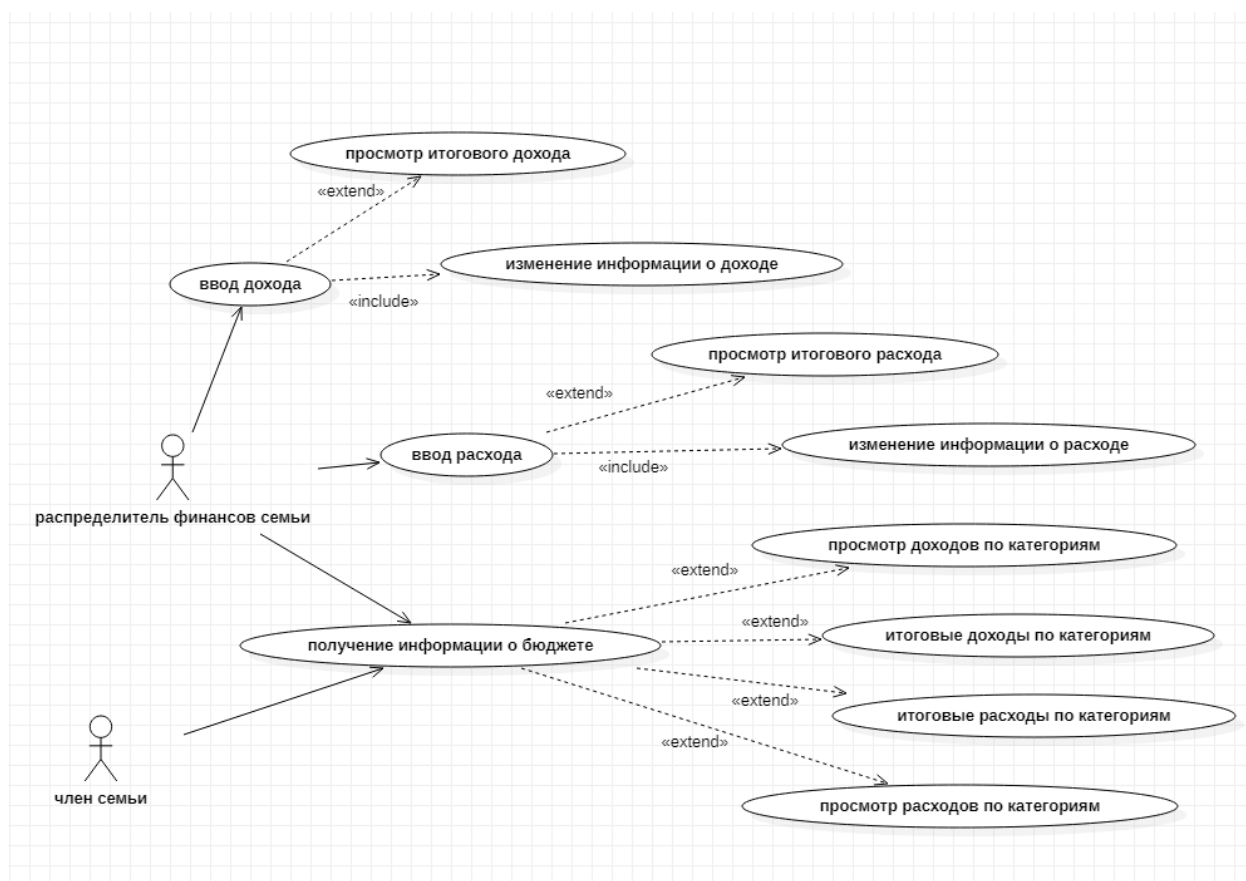


Рис.8 Диаграмма прецедентов

Таблица 1. Описание прецедентов

Прецедент	Краткое описание
Ввод дохода	Запускается распределителем финансов семьи. Позволяет вносить информацию о доходе (член семьи, категория дохода, размер дохода). Включает прецедент «изменение информации о доходе» (удаление, редактирование информации о доходе), расширяемый прецедент «просмотр итогового дохода» (итоговая сумма всех введенных доходов)
Ввод расхода	Запускается распределителем финансов семьи. Позволяет вносить информацию о расходе (член семьи, категория расхода, размер расхода). Включает прецедент «изменение информации о расходе» (удаление, редактирование информации о расходе), расширяемый прецедент «просмотр итогового расхода» (итоговая сумма всех введенных расходов)
Получение информации о бюджете	Может запускаться распределителем финансов и членом семьи. Позволяет посмотреть итоговый состав бюджета. Включает расширяемые прецеденты: «просмотр доходов по категориям» (фильтрация доходов по категории), «просмотр расходов по категориям» (фильтрация расходов по категориям), «итоговые доходы по категориям» (итоговая сумма доходов по каждой категории дохода), «итоговые расходы по категориям» (итоговая сумма расходов по каждой категории расхода)



## 2.2. Выбор инструментов

Выбор инструментов для курсового проекта в первую очередь определялся следующими критериями: простота сопровождения, наличие библиотек для работы с базой данных, скорость разработки, простота разработки графического интерфейса, наличие документации на русском языке. Для приведенных критериев была выбрана степень важности: низкая, средняя и высокая.

Таблица 2. Оценка важности критериев

Критерий	Простота сопровождения	Наличие библиотек для работы с базой данных	Скорость разработки	Простота разработки графического интерфейса	Наличие документации на русском языке
Важность критерия	Высокая	Высокая	Средняя	Средняя	Низкая

Исходя из критериев, было проведено сравнение 3 языков программирования (Java, C++ и Object Pascal), от 0 до 10 баллов за критерий.

Таблица 3. Сравнение

Критерий/Язык программирования	Java	C++	Object Pascal
Простота сопровождения	7	7	8
Наличие библиотек для работы с базой данных	6	4	10
Наличие документации на русском языке	8	8	8
Скорость разработки	7	8	7
Простота разработки графического интерфейса	3	5	10
Итого баллов	31	32	43

По итоговым результатам был выбран язык программирования Object Pascal.

Для дальнейшей работы также в качестве инструментов были выбраны: среда разработки Lazarus[3], СУБД FireBird[4] и инструмент для работы с FireBird IBExpert[5].

### **2.3. Проектирование сценария**

В данном разделе приведен сценарий использования программы распределителем финансов семьи.

Запуская программу, пользователь может выбрать 4 действия: добавить информацию о доходе/расходе, посмотреть информацию о бюджете, посмотреть информацию о программе, либо выйти из программы.

При выборе последнего программа заканчивает свою работу, при выборе же остального пользователю открывается новое окно с дальнейшими действиями.



Рис.9 Сценарий

## 2.4. Диаграмма классов

Данная диаграмма классов отражает используемые в программе таблицы базы данных, характеризующиеся при помощи атрибутов (свойства объектов класса) и операций (функции класса).

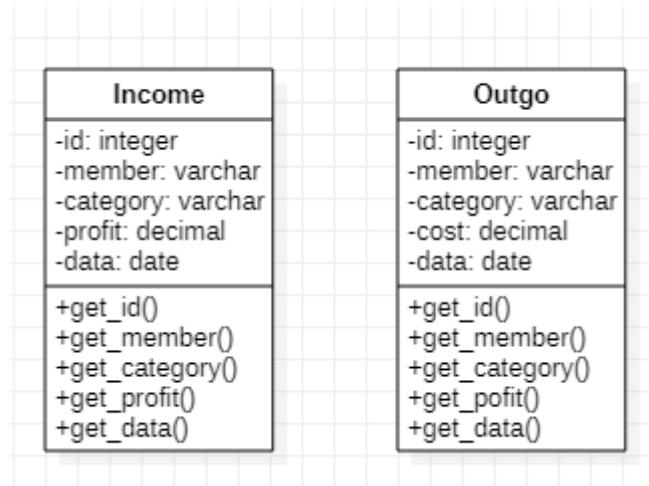


Рис.10 Диаграмма классов

Класс Income – таблица доходов семьи. Содержит ID дохода – ключевое поле, имя члена семьи(member), категория дохода(category), размер дохода(profit) и дата получения дохода (data). Также данный класс содержит методы получения значений полей.

Класс Outgo – таблица расходов семьи. Состоит из ключевого поля ID, имени члена семьи(member), категории расхода(category), размер расхода(cost) и дата получения расхода (data). Также данный класс содержит методы получения значений полей.

## 2.5. Описание главного модуля

Главный модуль программы – модуль основного файла кода проекта (файл FamilyBudget.lpr). Выполняет создание экземпляров форм, содержит список используемых модулей программы, подключает файл ресурсов (Листинг 2.5. Главный модуль).

Исполняемая часть главного модуля находится между инструкциями begin и end, обеспечивающие инициализацию проекта и вывод на экран стартового окна.

Проект начинается с обращения к методу `initialize` глобального объекта `application` и заканчивается после завершения его метода `run`.

### Листинг 2.5. Главный модуль

```
program FamilyBudget;
{$mode objfpc}{$H+}
uses
    {$IFDEF UNIX}{$IFDEF UseCThreads}
    cthreads,
    {$ENDIF}{$ENDIF}
    Interfaces, // this includes the LCL widgetset
    Forms, fmain, money, dm, budget, aboutProgram
    { you can add units after this };
{$R *.res}
begin
    RequireDerivedFormResource:=True;
    Application.Scaled:=True;
    Application.Initialize;
    Application.CreateForm(Tumain, umain); //экземпляр формы umain
    Application.CreateForm(Tumain2, umain2); //экземпляр формы umain2
    Application.CreateForm(Tdmb, dmb); //экземпляр формы dmb
    Application.CreateForm(Tumain3, umain3); //экземпляр формы umain3
    Application.CreateForm(Tumain4, umain4); //экземпляр формы umain4
    Application.Run;
end.
```

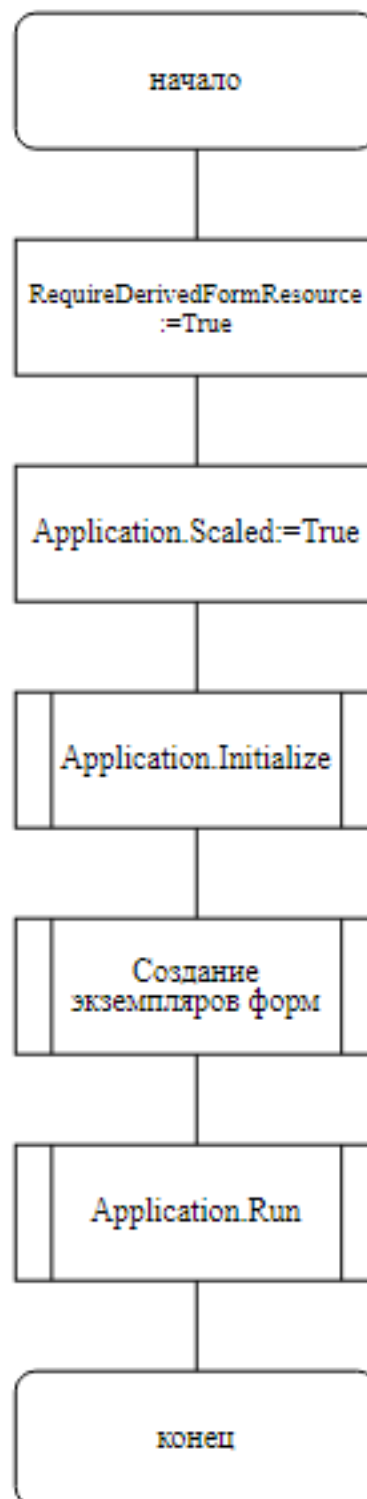


Рис.11 Блок-схема главного модуля

## 2.6. Описание спецификаций к модулям

В данном пункте будут приведены публичные члены модулей программы и в дальнейшем описаны в пункте 2.7.

1. Публичные объекты модуля форм: dmb (форма дата-модуля с компонентами для связи с базой данных), umain (главная форма меню), umain2 (форма для работы с доходами/расходами), umain3 (форма отображения бюджета семьи), umain4 (форма информации о программе).
2. Публичные процедуры.

В дата-модуле: процедура DataModuleCreate – подключение базы данных к программе и работа с нужными компонентами.

В модулях форм: обработчики событий OnClick, OnCreate, OnChange (для работы с формами), процедура loadtable (создание выпадающего списка в компоненте ComboBox).

## 2.7. Описание модулей

В процессе написания программы были разработаны следующие модули:

1. Дата-модуль (dm).

Отвечает за подключение базы данных и обеспечивает доступ к компонентам.

Дата-модуль запускается автоматически при запуске программы.

В реализации модуля описана процедура открытия необходимых компонентов базы данных при запуске, запуск транзакции (Листинг 2.7.1. Дата-модуль). Полный код дата - модуля представлен в листинге приложения 4.

**Листинг 2.7.1. Дата-модуль**

```
procedure Tdmb.DataModuleCreate(Sender: TObject);    //дата-модуль
begin
    if(opendialog1.Execute) then
    begin
        ibdb.DatabaseName :=
ExtractFileDir(OpenDialog1.FileName)+'\BQ.FDB'; //выбор пути бд
        ibdb.Connected:=true;
    end;
    ibtr.StartTransaction; //транзакция
    ibIncome.Open; // таблица доходов
    ibOutgo.Open; // таблица расходов
    ibqIncome.open; // запрос к таблице доходов для комбобокса
    ibqOutgo.open; // запрос к таблице расходов для комбобокса
    ibqOutgoSum.Open; // запрос для общей суммы расходов
    ibqIncomeSum.Open; // запрос для общей суммы расходов
    ibqIncomeCat.Open; // запрос для суммы доходов по каждой категории
дохода
    ibqOutgoCat.Open; // запрос для суммы расходов по каждой категории
расхода
    end;
```



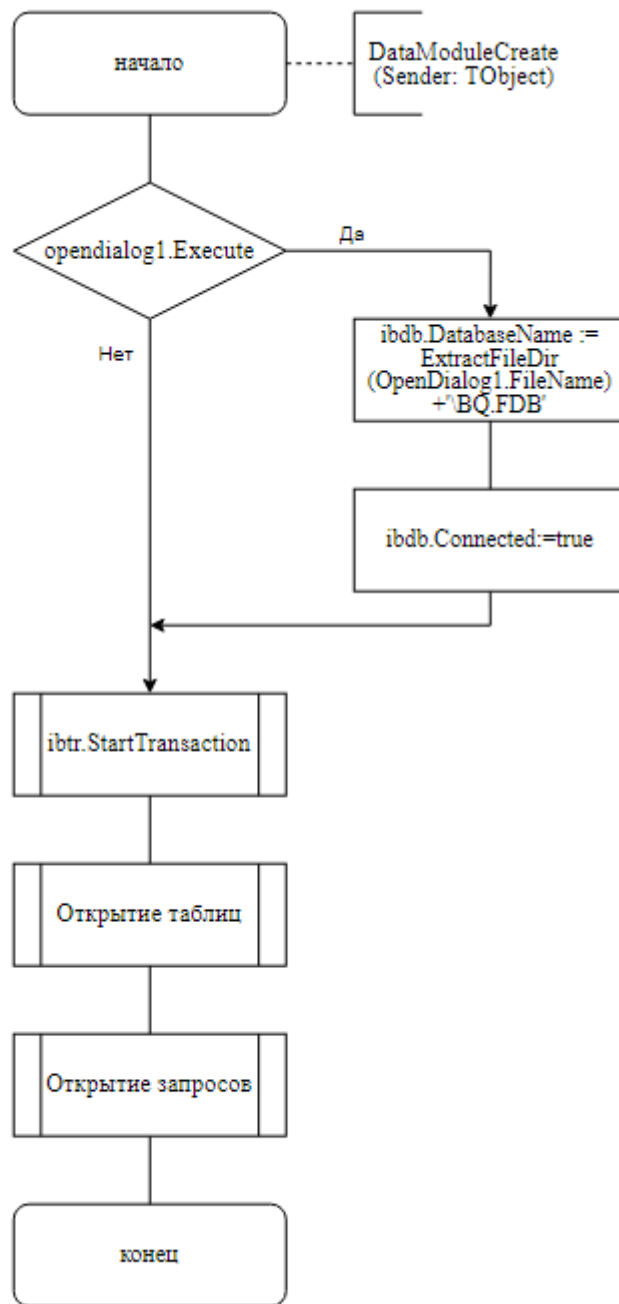


Рис.12 Блок-схема процедуры DataModuleCreate

2. Модули работы с таблицами и форм проекта. Представляют графическое представление программы, работу с компонентами базы данных при помощи навигационной панели и отражение результатов sql-запросов: по фильтрации категорий доходов/расходов, итоговой сумме доходов/расходов, а также по подсчету итоговой суммы каждой категории для доходов и расходов. Также включает в себя процедуру loadtable, создающую выпадающий список для

компонента ComboBox (Листинг 2.7.1. Создание выпадающего списка), и впоследствии при выборе значения из списка меняет скрипт соответствующего запроса и выводит фильтрацию по выбранному значению категории. Полный листинг модулей представлен в приложениях 1, 2, 3, 5.

### **Листинг 2.7.1. Создание выпадающего списка**

```
procedure Tmain3.loadtable(tablename : string; cmb : TCombobox);
// создаем комбобокс с выпадающим списком
begin
dm.dmb.ibq.close(); //закрыть запрос
if not (dm.dmb.ibq.Transaction.Active) then
dm.dmb.ibq.Transaction.StartTransaction;
//запустить транзакцию , если неактивна
dm.dmb.ibq.SQL.Text := 'select distinct category from ' + tablename;
//запрос на список неповторяющихся категорий
dm.dmb.ibq.open; //открыть запрос
while not (dm.dmb.ibq.eof) do //пока не конец запроса
begin
cmb.items.add(dm.dmb.ibq.FieldName('CATEGORY').AsString);
//берем название поля и добавляем в свойство items
dm.dmb.ibq.next; //сдвиг
end;
end;
```

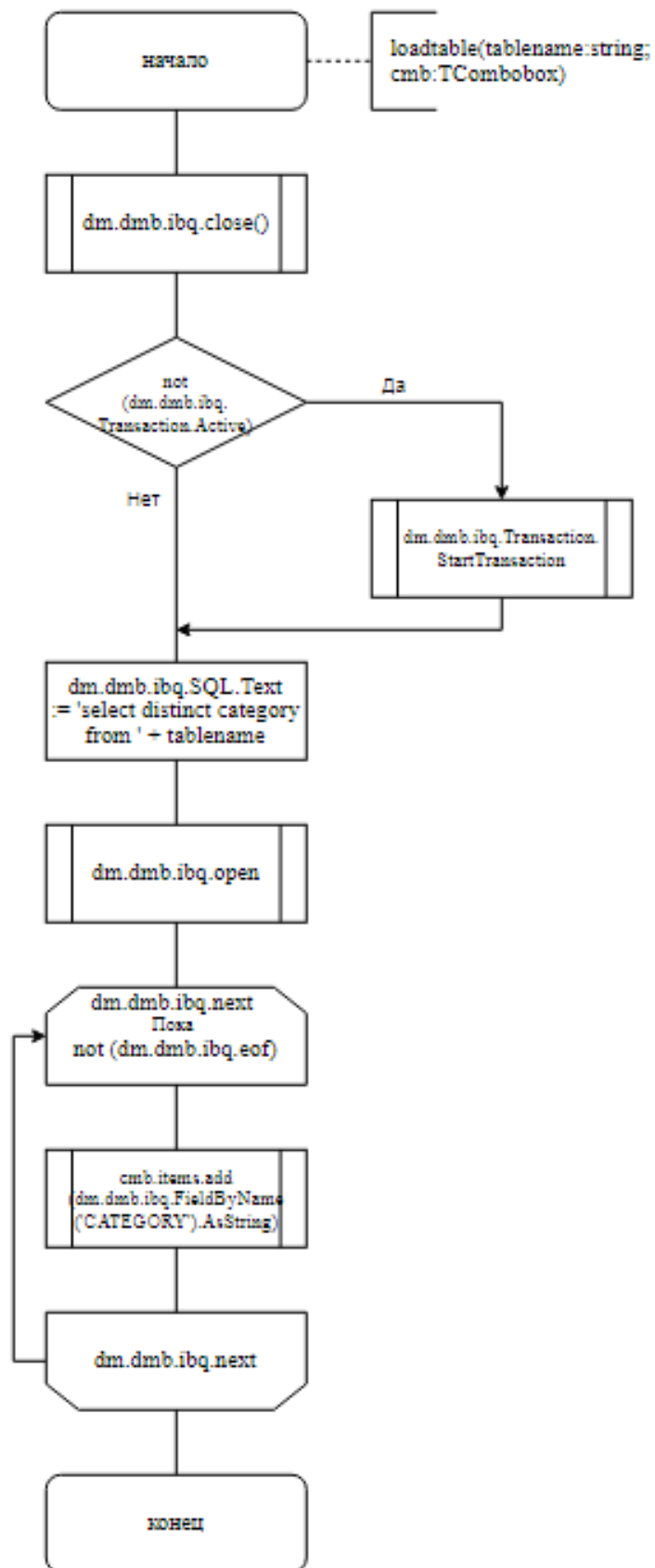


Рис.13 Блок-схема процедуры loadtable

3. Модуль основного файла кода проекта (FamilyBudget.lpr). Реализует запуск программы и связь со всеми модулями. Свойства, назначение и разбор описаны в п. 2.5.

## 2.8. Описание тестовых наборов модулей

В данном пункте будут продемонстрированы результаты тестирования методом «черного ящика».

Тест 1. Переход в окно «Доходы/расходы»

Действия: Нажать на кнопку «Добавить доход/расход»

Ожидаемый результат: Переход в окно «Доходы/расходы»

Результат

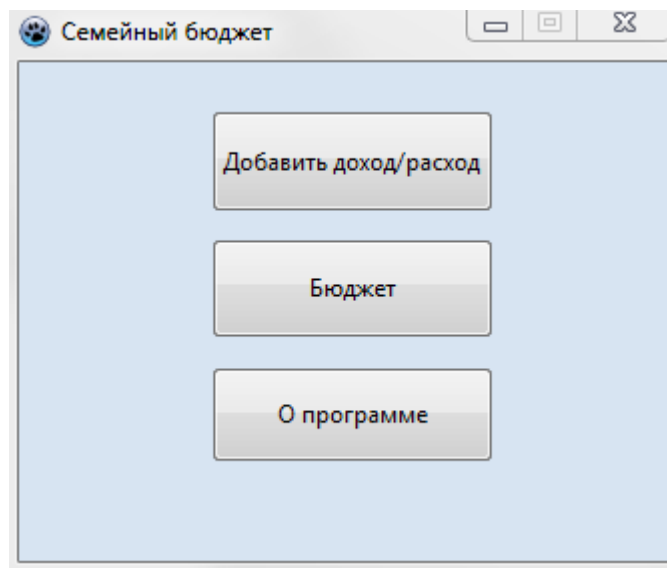


Рис.14 Результат до перехода

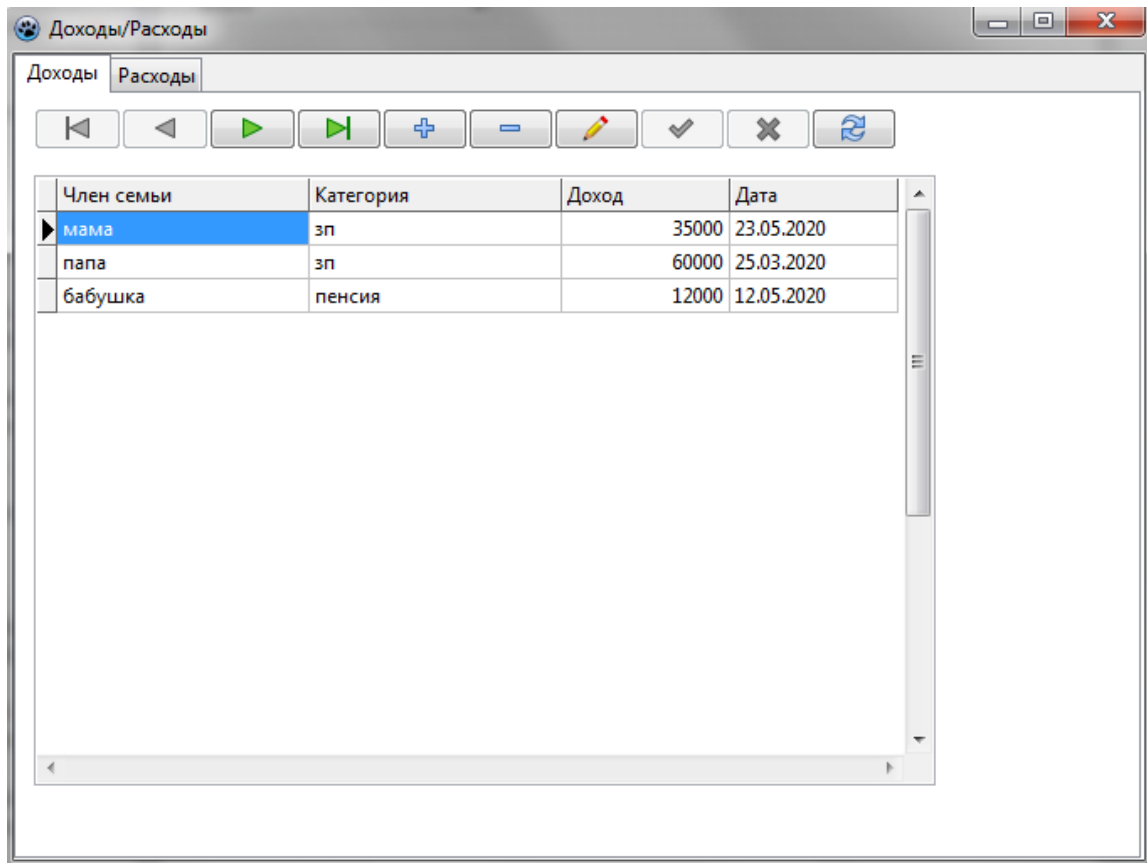


Рис.15 Результат после перехода

Тест 2. Смена вкладки

Действия: Открыть другую вкладку

Ожидаемый результат: Смена одной вкладки на другую

Результат

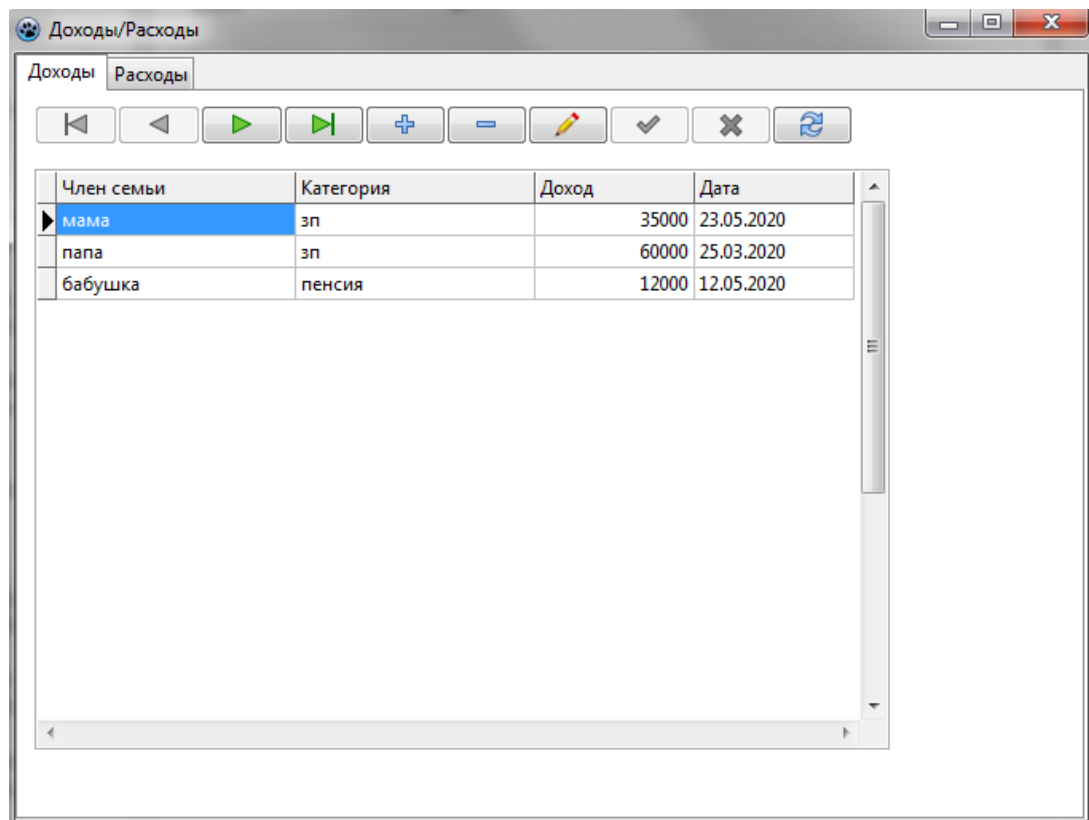


Рис.16 Результат до смены вкладки

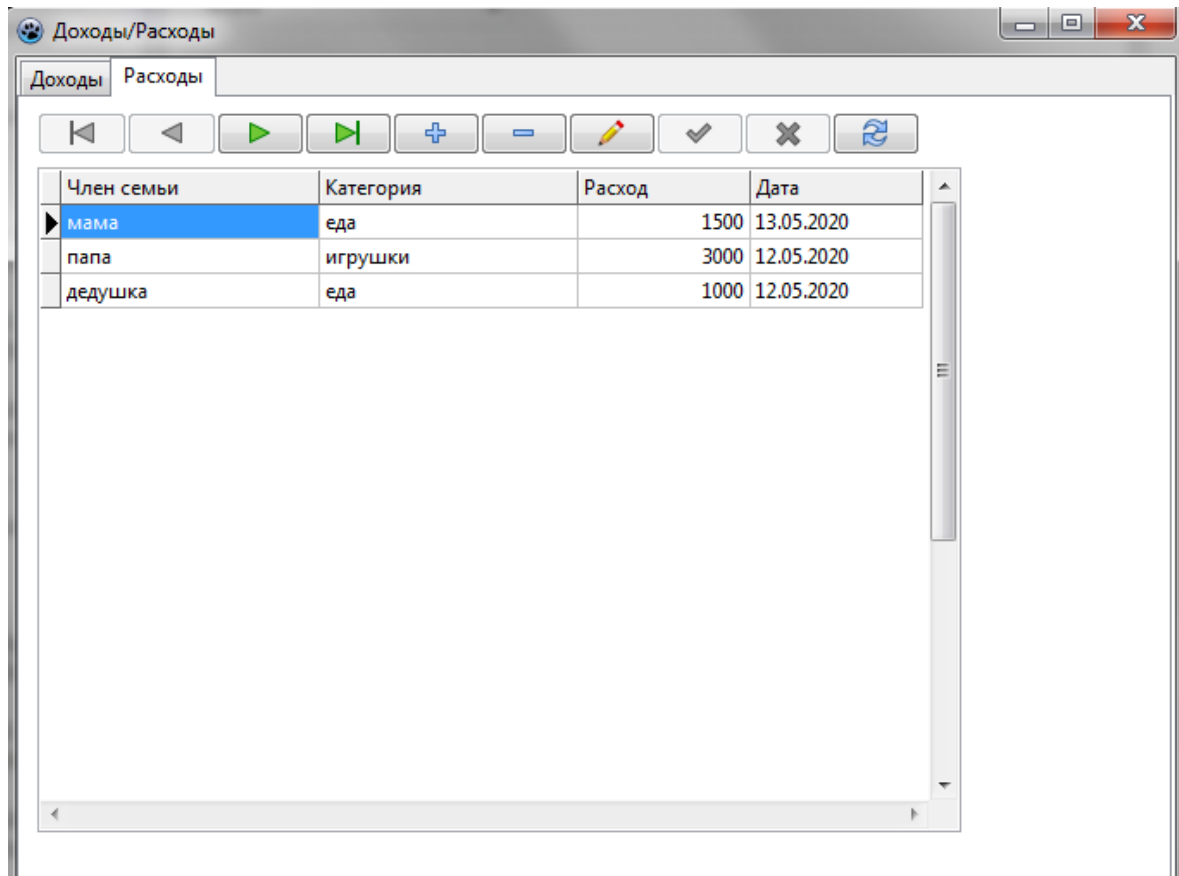


Рис.17 Результат после смены вкладки

Тест 3. Переход в окно «Бюджет»

Действия: Нажать на кнопку «Бюджет»

Ожидаемый результат: Переход в окно «Бюджет»

Результат

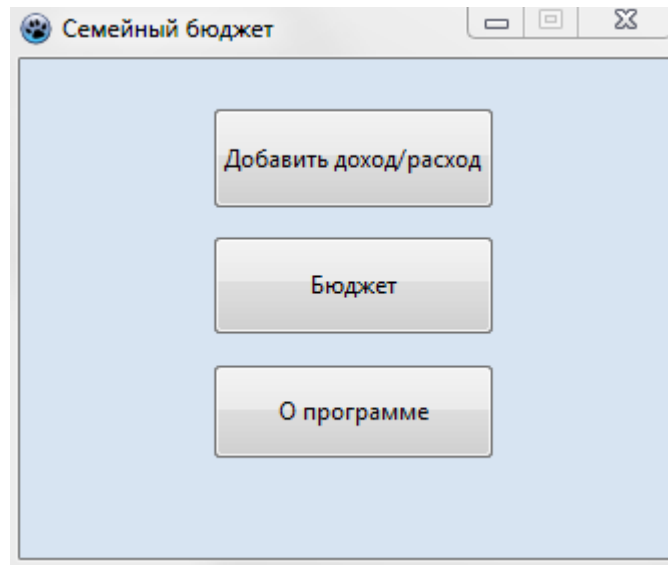


Рис.18 Результат до перехода



Бюджет

Доходы Расходы

Фильтр доходов по категориям :

Всего

Член семьи	Категория	Доход	Дата
▶ мама	зп	35000	23.05.2020
папа	зп	60000	25.03.2020
бабушка	пенсия	12000	12.05.2020

Итого доходов : 107000

Итоговые доходы по категориям :

Категория	Итог
▶ зп	95000
пенсия	12000

Рис.19 Результат после перехода

Тест 4. Переход в окно «О программе»

Действия: Нажать на кнопку «О программе»

Ожидаемый результат: Переход в окно «О программе»

Результат

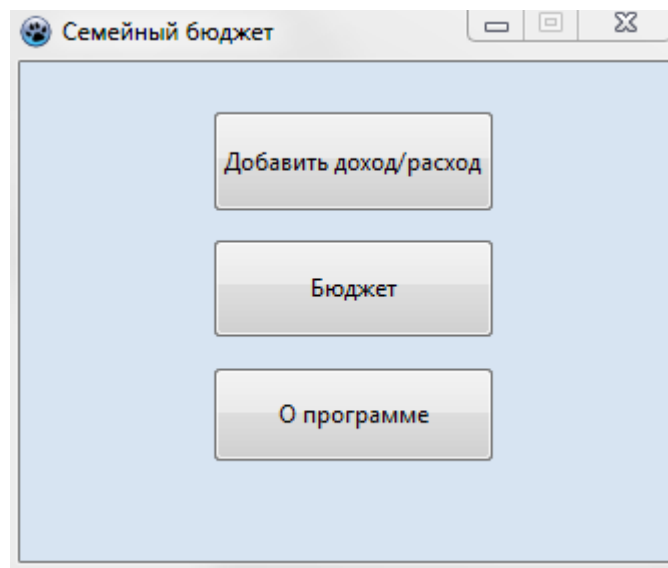


Рис.20 Результат до перехода

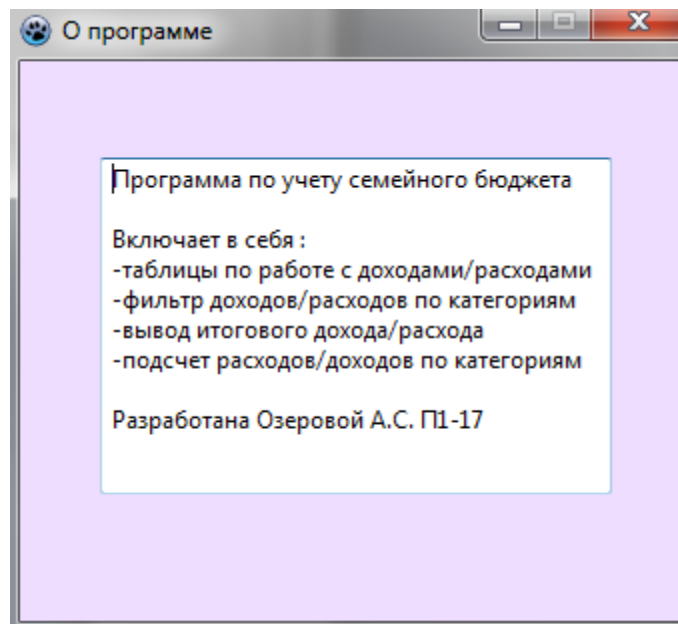


Рис.21 Результат после перехода

## Тест 5. Добавление позиций в таблицу

Действия:

- 1) Нажать на кнопку добавления на панели навигации
- 2) Ввести информацию в новой строке таблицы
- 3) Нажать на кнопку обновления таблицы

Ожидаемый результат: добавление новой позиции

Результат

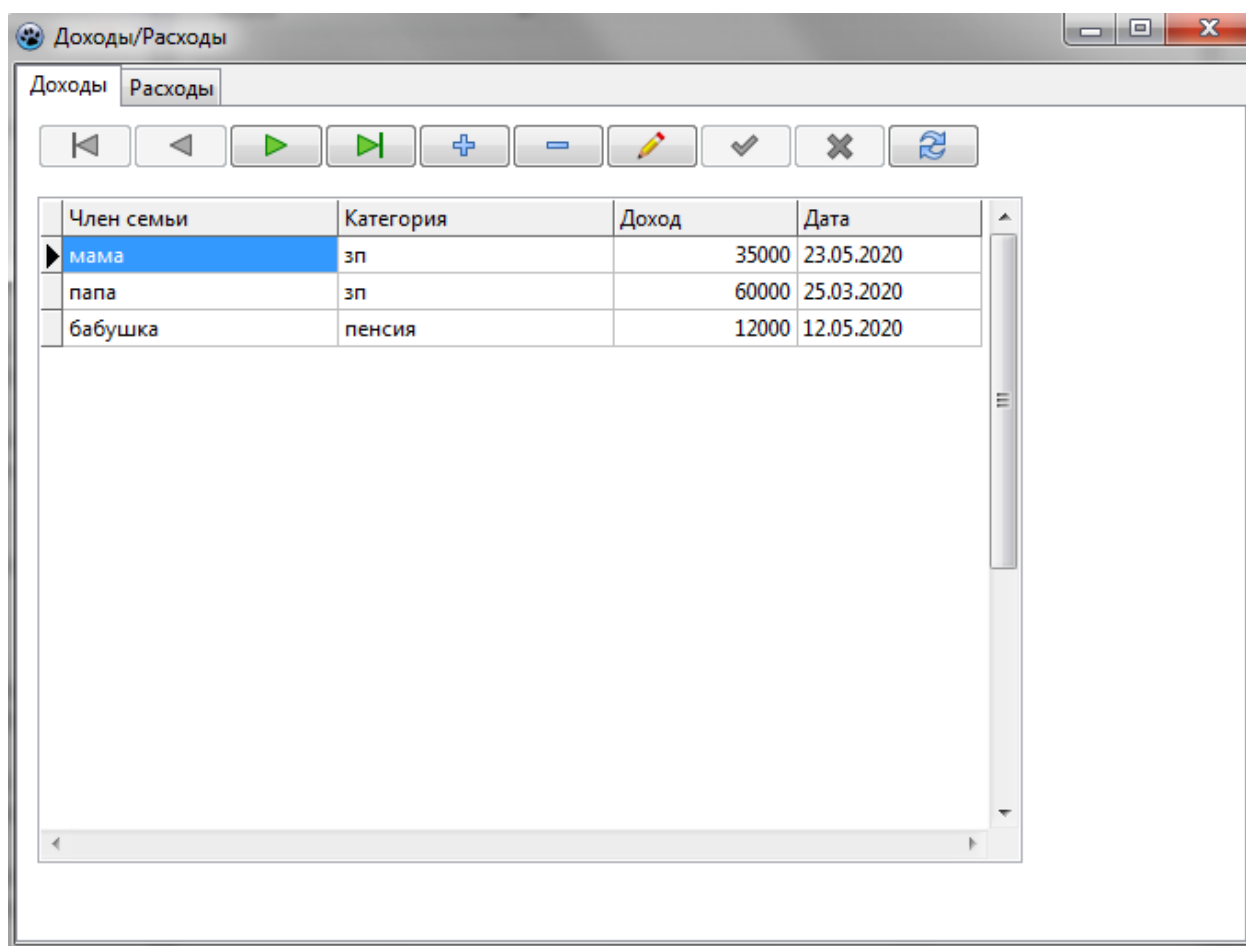


Рис.22 Результат до добавления

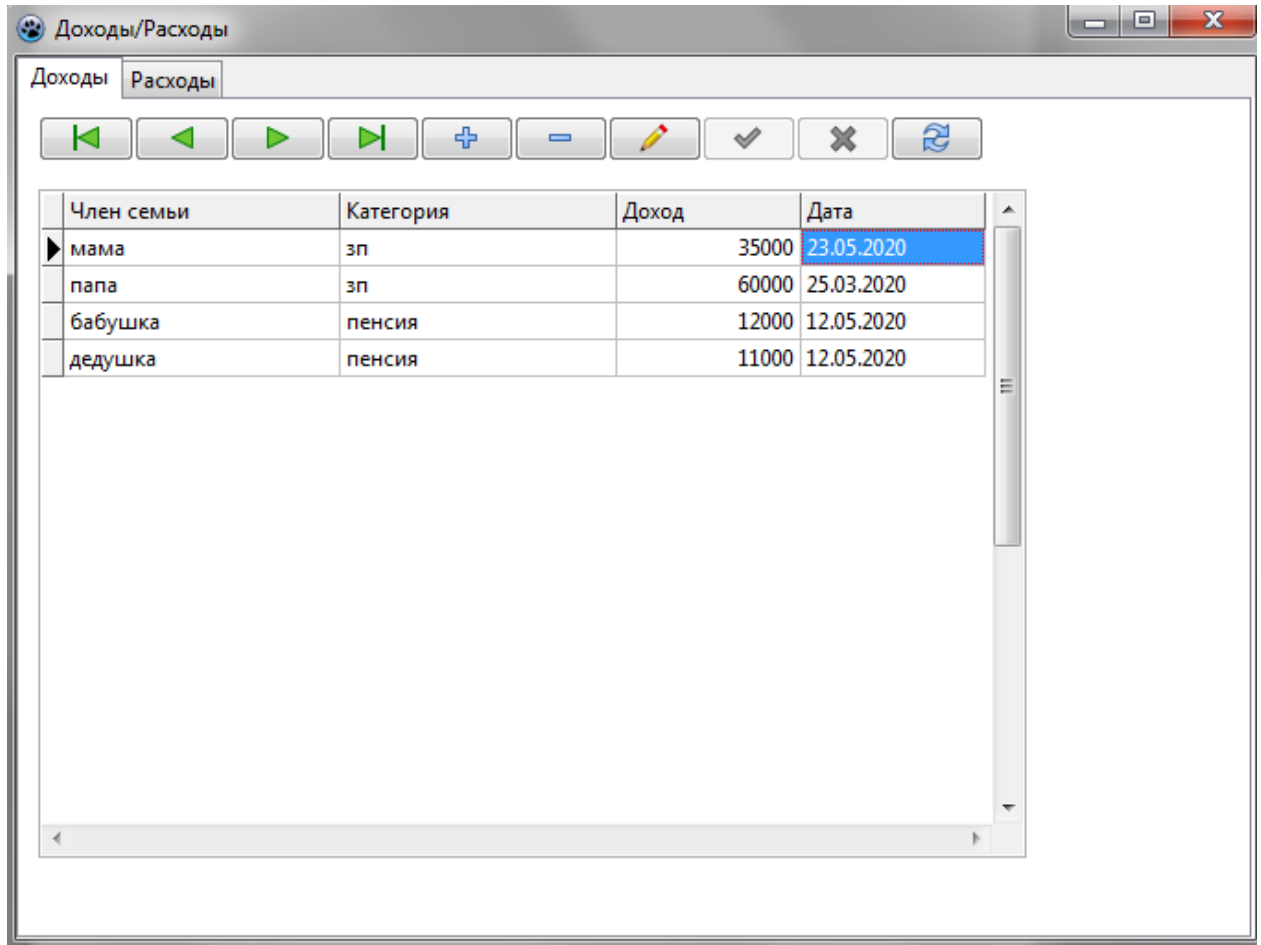


Рис.23 Результат после добавления

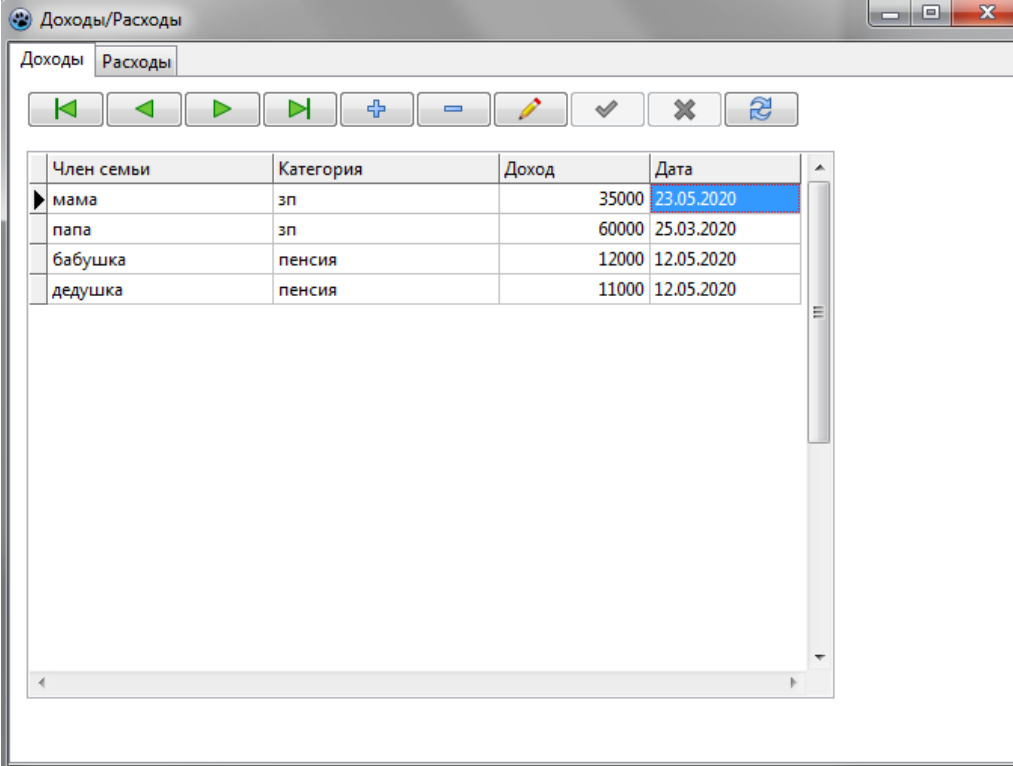
## Тест 6. Удаление позиции в таблице

Действия:

- 1) Нажать на поле для удаления
- 2) Нажать на кнопку удаления на панели навигации
- 3) Согласиться с удалением позиции в всплывающем окне

Ожидаемый результат: удаление позиции

Результат



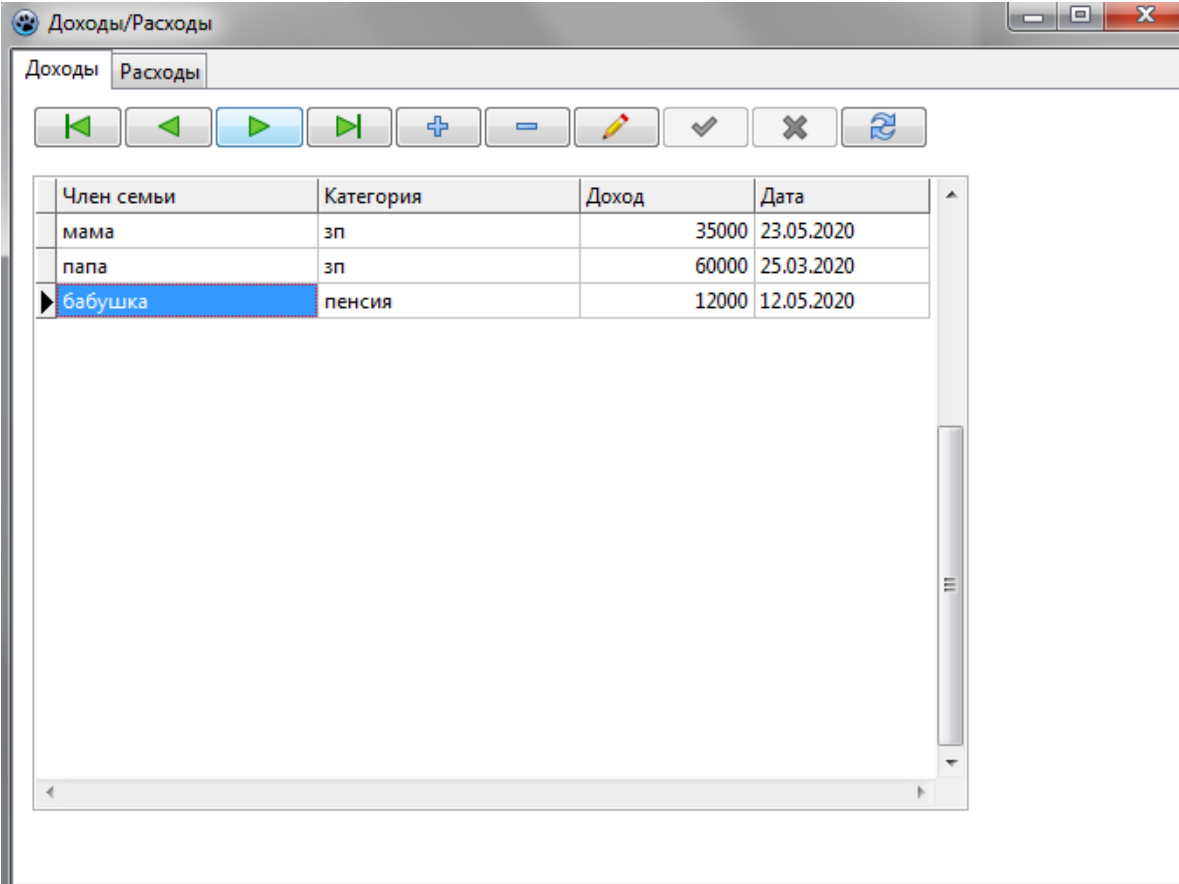
Доходы/Расходы

Доходы Расходы

⏮ ⏪ ⏩ ⏭ + - ✎ ✓ ✕ ↺

Член семьи	Категория	Доход	Дата
▶ мама	зп	35000	23.05.2020
папа	зп	60000	25.03.2020
бабушка	пенсия	12000	12.05.2020
дедушка	пенсия	11000	12.05.2020

Рис.24 Результат до удаления



Доходы/Расходы

Доходы Расходы

⏮ ⏪ ⏩ ⏭ + - ✎ ✓ ✕ ↺

Член семьи	Категория	Доход	Дата
▶ мама	зп	35000	23.05.2020
папа	зп	60000	25.03.2020
▶ бабушка	пенсия	12000	12.05.2020

Рис.25 Результат после удаления

## Тест 7. Редактирование позиции в таблице

Действия:

- 1) Нажать на поле для редактирования
- 1) Нажать на кнопку редактирования на панели навигации
- 2) Редактировать позицию
- 3) Нажать на кнопку обновления таблицы

Ожидаемый результат: редактирование позиции

Результат

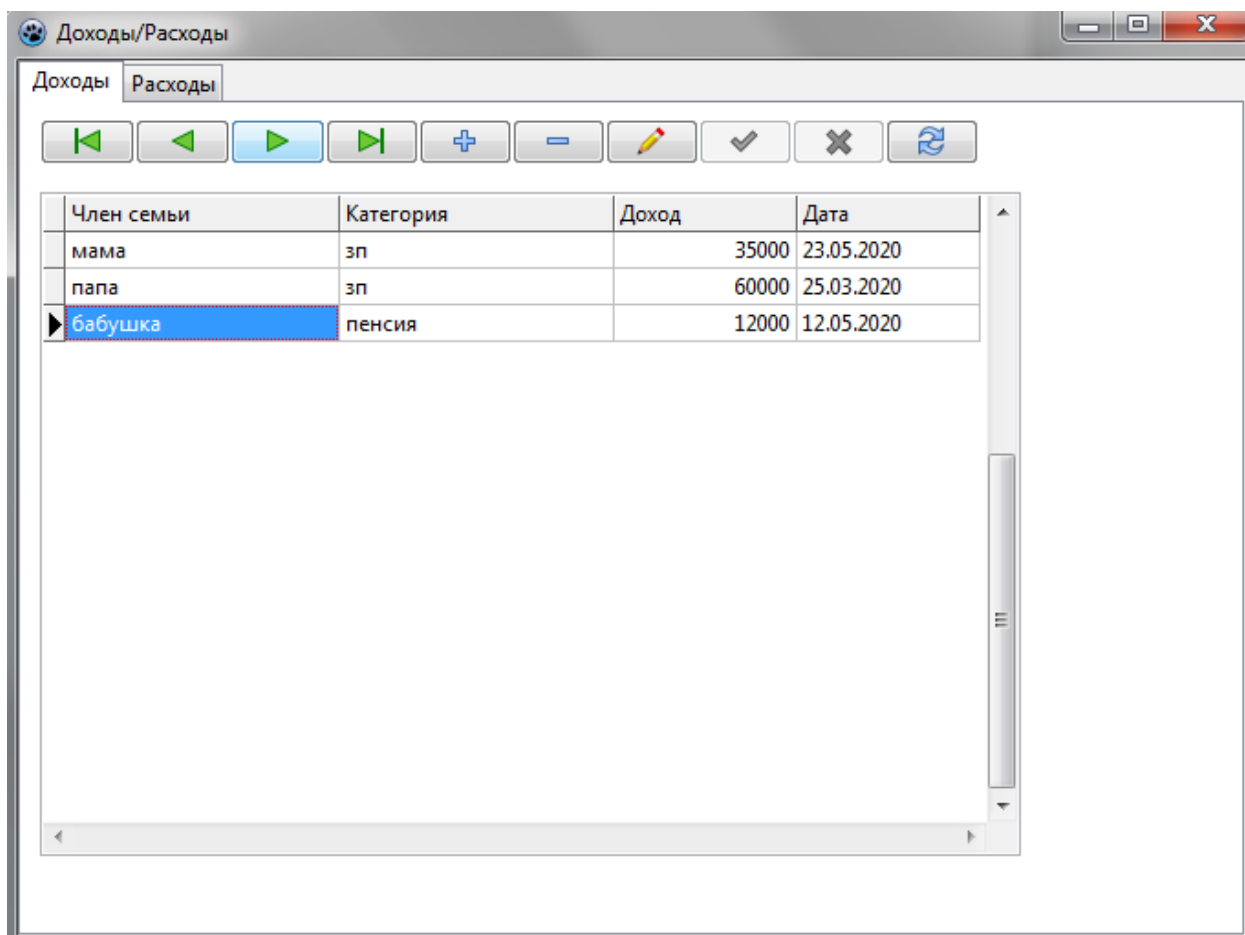


Рис.26 Результат до редактирования

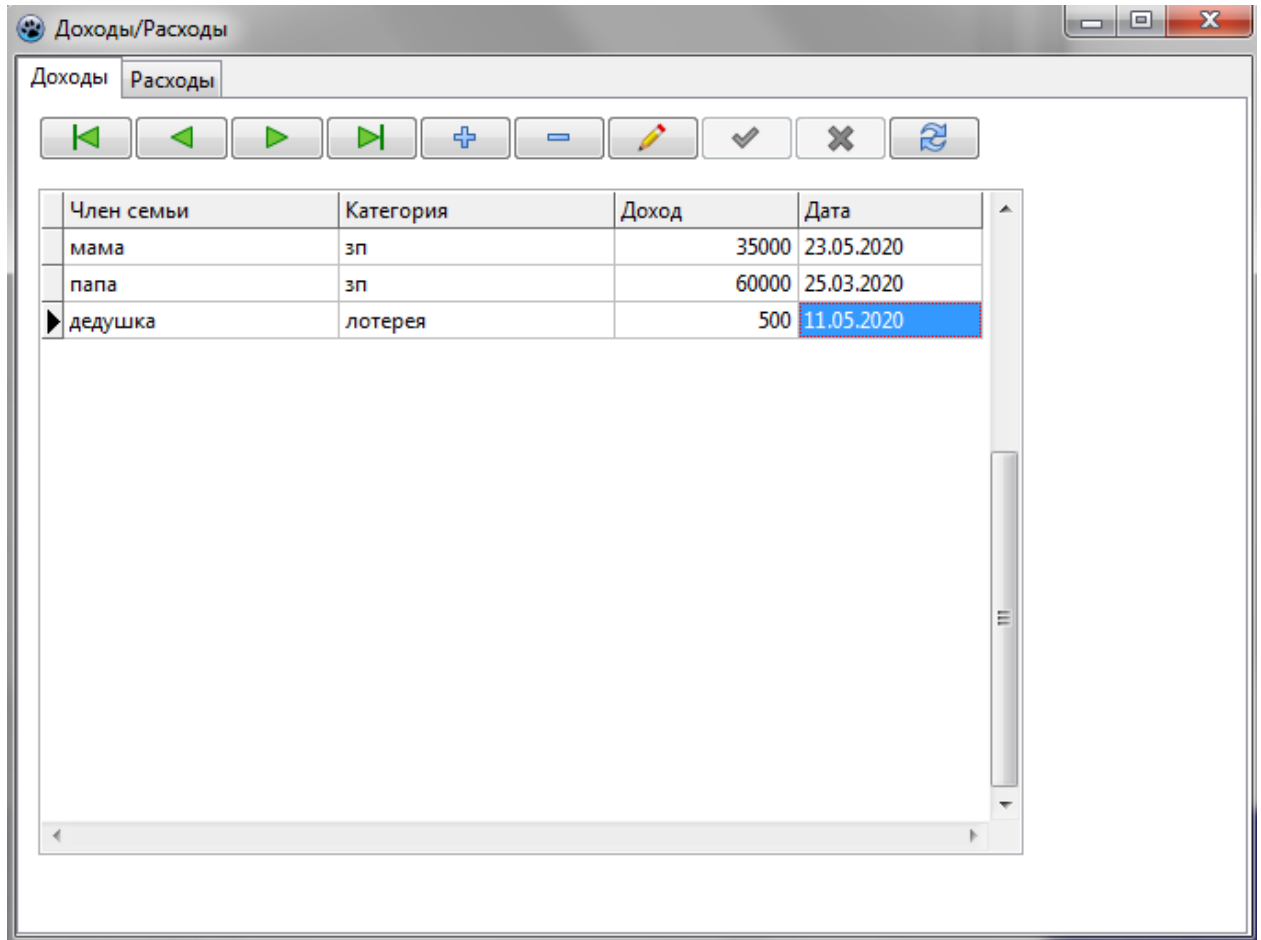


Рис.27 Результат после редактирования

## Тест 8. Фильтр по категориям

Действия:

- 1) Нажать на выпадающий список
- 2) Выбрать другую категорию для фильтра.

Ожидаемый результат: фильтр таблицы по выбранной категории

Результат

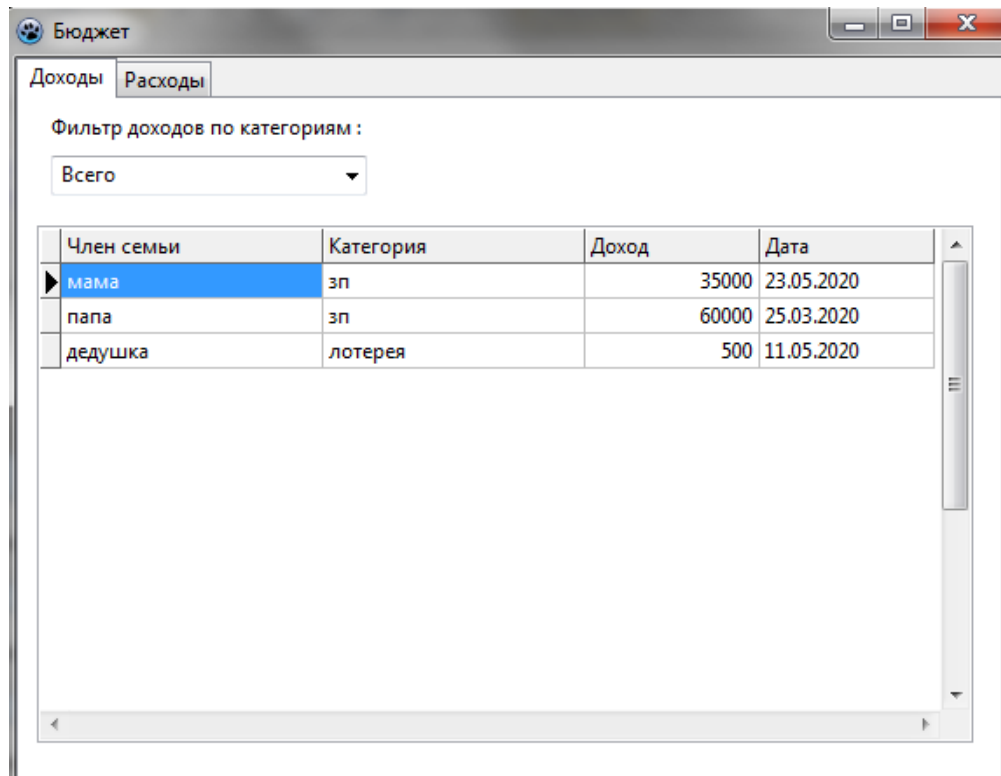


Рис.28 Результат до смены фильтра

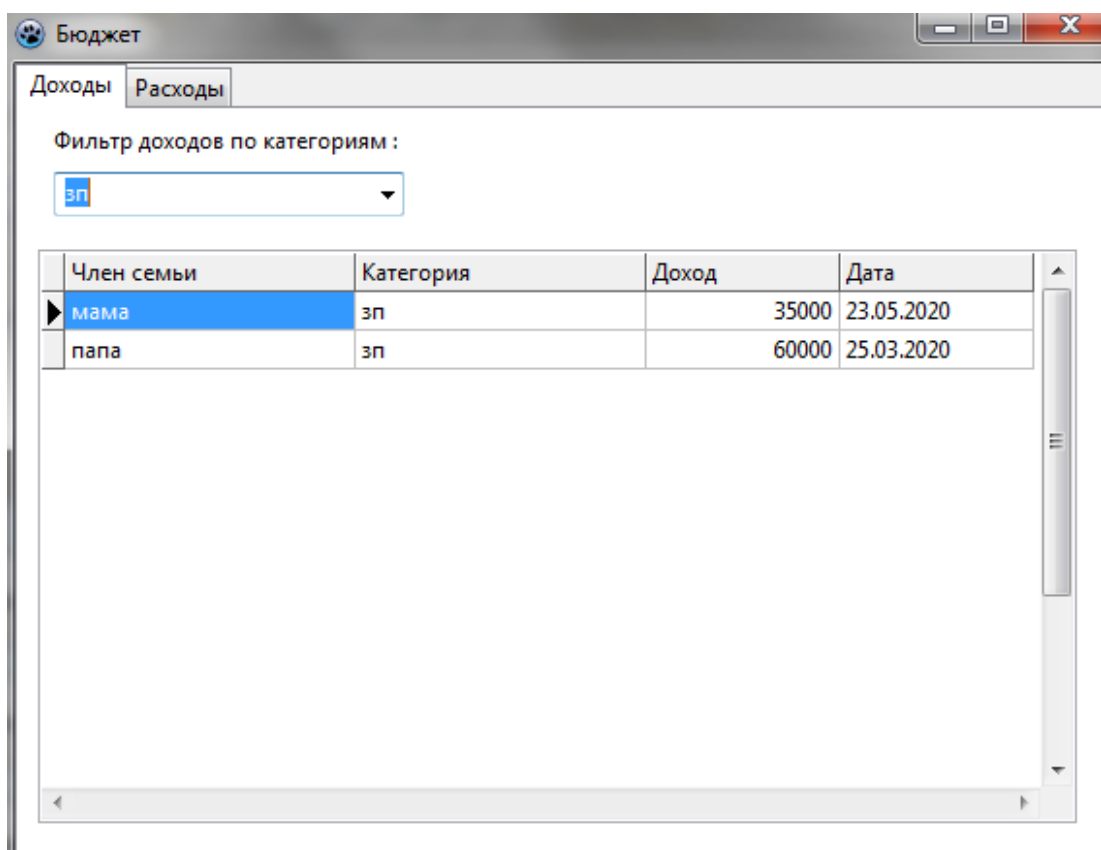


Рис.29 Результат после смены фильтра



## 2.9. Описание применения средств отладки

Данный раздел показывает умение работать со средствами отладки.

В ходе написания проекта при попытке запустить программу было получено следующее сообщение (Рис.30)

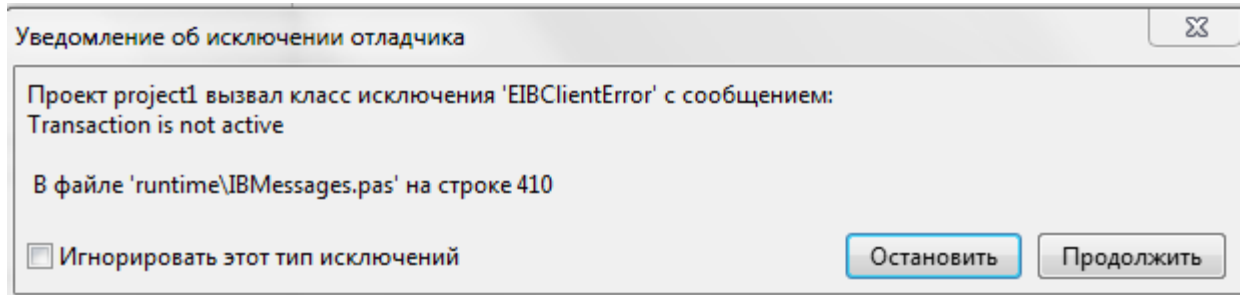


Рис.30 Результат до отладки

Решением данной ошибки был запуск транзакции базы данных. Повторный запуск программы после отладки ошибок не выявил (Рис.31)

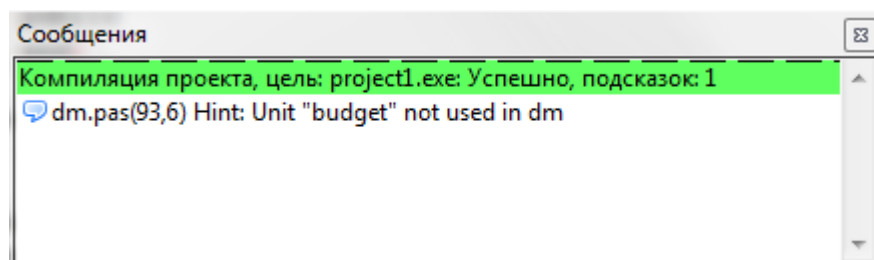


Рис.31 Результат после отладки

## 2.10. Анализ оптимальности использования памяти и быстродействия

В данном разделе будет проведен анализ оптимальности использования памяти и быстродействия программы.

Размер файла FamilyBudget.exe достаточно большой (Рис.32). Было решено оптимизировать компиляцию, тем самым уменьшив размер файла.

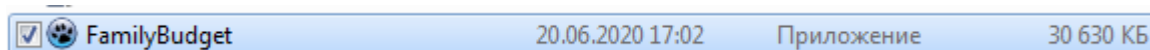


Рис.32 Размер файла до оптимизации

Предпринятые действия:

- 1) Для уменьшения размера файла необходимо отключить «Генерировать отладочную информацию» и включить «вырезать символы из исполняемого файла» в пункте «Отладка».

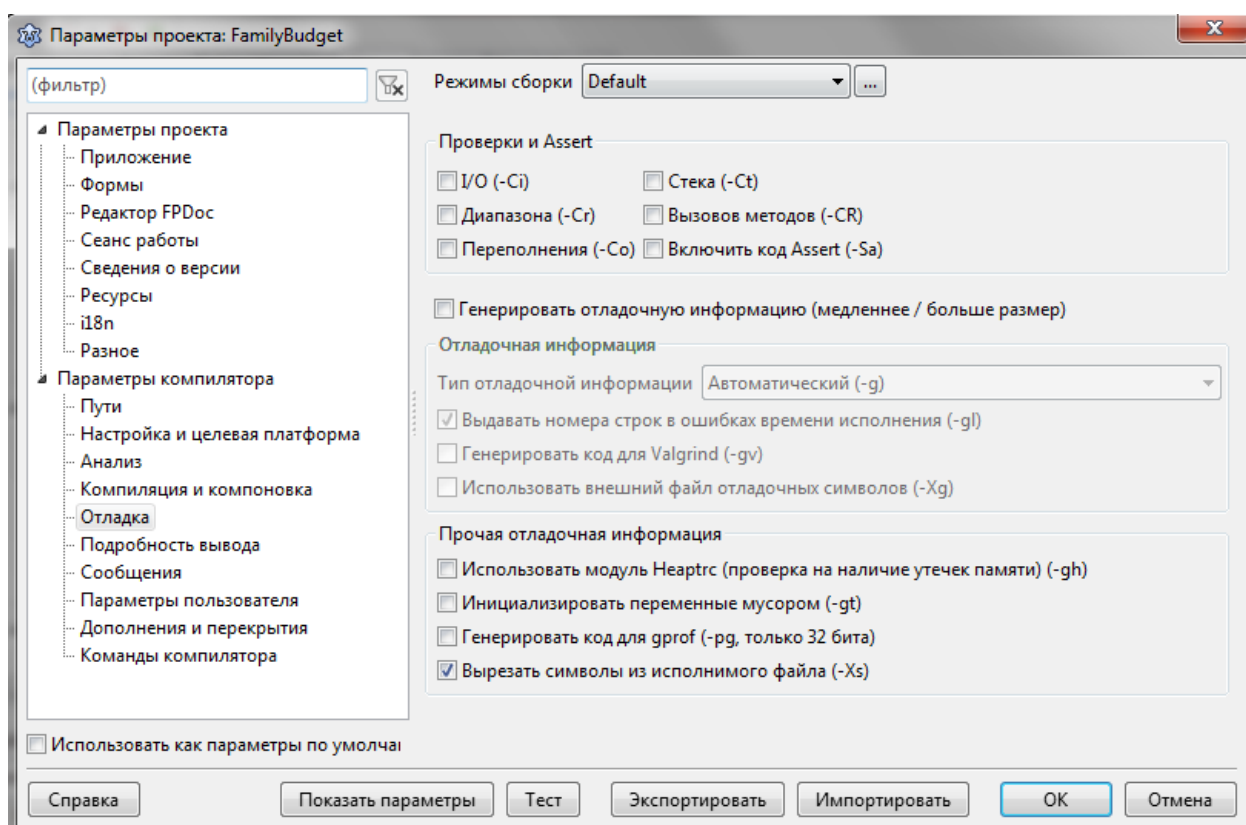


Рис.33 Пункт «Отладка»

- 2) Так же необходимо включить «Умная компоновка» в разделе «стиль модуля» и «Компоновка» пункта «Компиляция и компоновка».

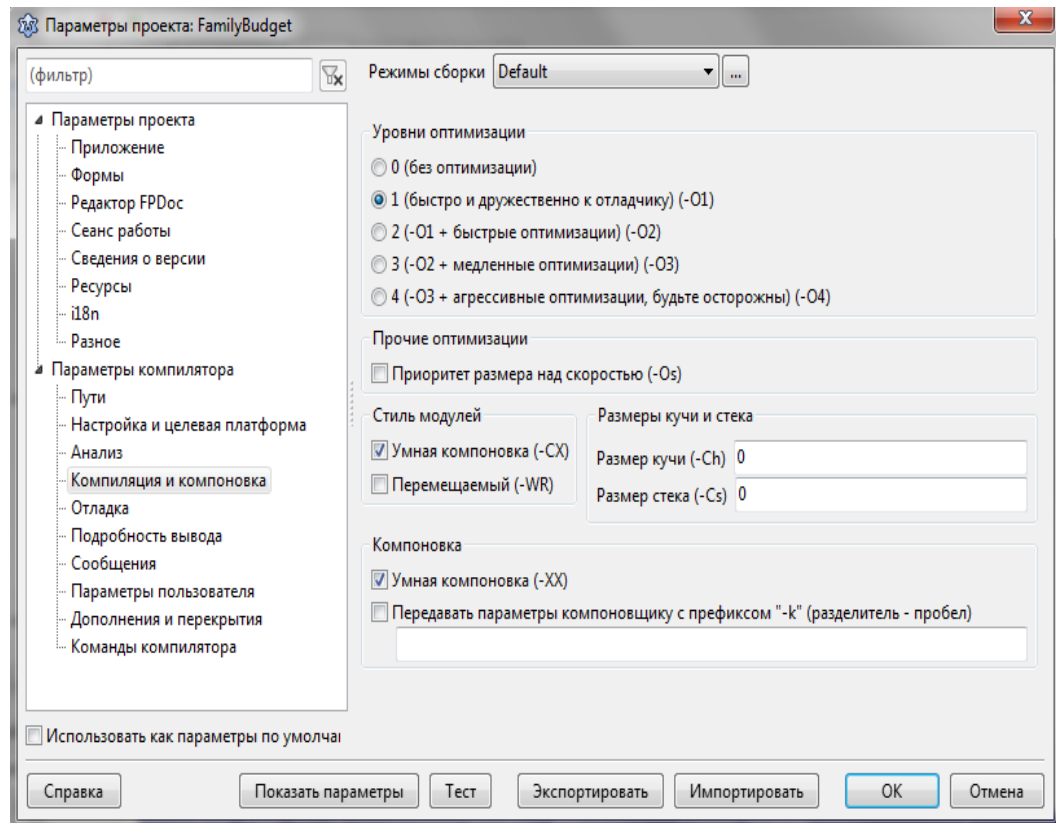


Рис.34 Пункт «Компиляция и компоновка»

- 3) Для уменьшения памяти можно убрать стандартный значок в пункте «Приложение».

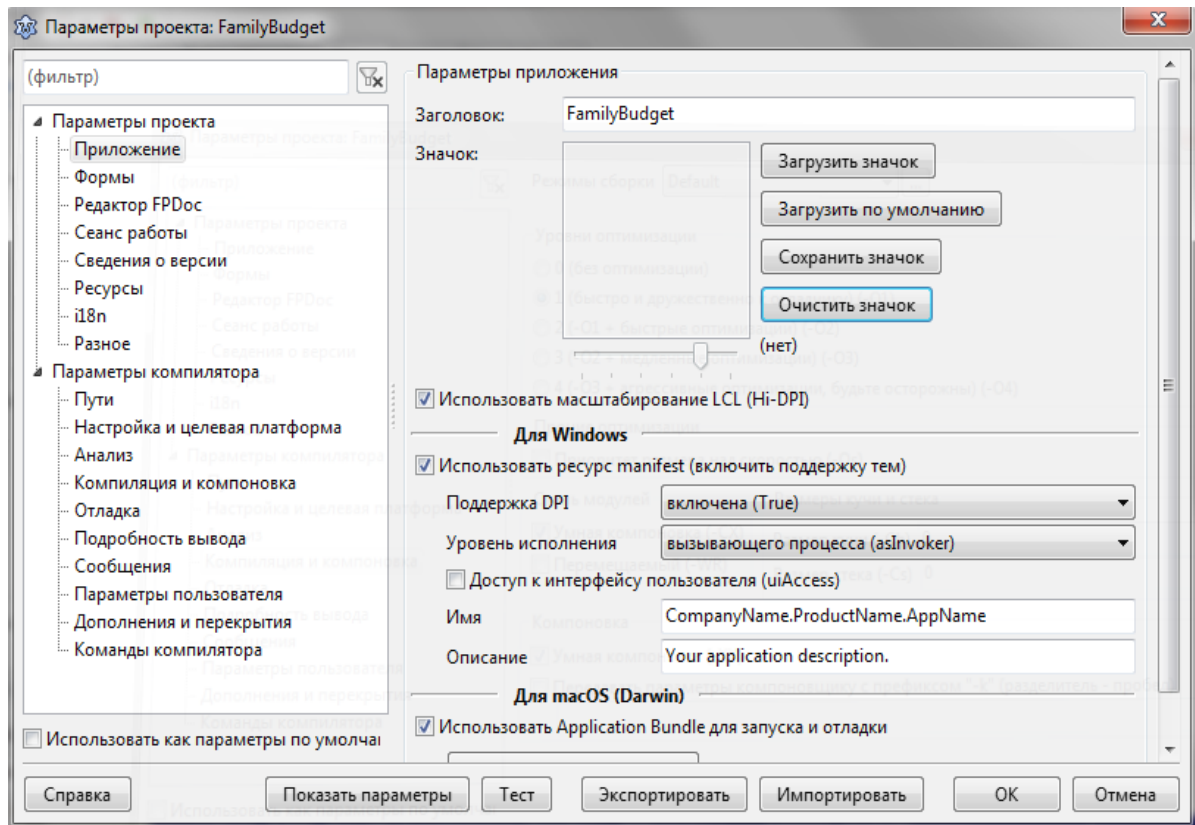


Рис.35 Пункт «Приложение»

После проведенной оптимизации компиляции можно заметить существенную разницу в размере файла. До оптимизации размер exe-файла составлял 30630 КБ, а после 4203 КБ(Рис. 36)


	FamilyBudget	22.06.2020 12:30	Приложение	4 203 КБ
---	--------------	------------------	------------	----------

Рис.36 Размер файла после оптимизации

## **Глава 3. Эксплуатационная часть**

### **3.1. Руководство оператора**

#### **Аннотация**

В данной главе приведено руководство оператора по применению и эксплуатации программы «FamilyBudget», предназначенной для учета ведения семейного бюджета.

В разделе «Назначение программы» указаны сведения о назначении программы и информация, достаточная для понимания функций программы и ее эксплуатации.

Раздел «Условия выполнения программы» содержит условия, необходимые для выполнения программы (минимальный состав аппаратных и программных средств и т.п.).

В разделе «Выполнение программы» приведена последовательность действий оператора, обеспечивающих загрузку, запуск, выполнение и завершение программы, приведено описание функций, формата и возможных вариантов команд, с помощью которых оператор осуществляет загрузку и управляет выполнением программы, а также ответы программы на эти команды.

В разделе «Сообщения оператору» приведены тексты сообщений, выдаваемых в ходе выполнения программы, описание их содержания и соответствующие действия оператора (действия оператора в случае сбоя, возможности повторного запуска программы и т.п.).

Оформление программного документа «Руководство оператора» произведено по требованиям ЕСПД (ГОСТ 19.101-77 <sup>1)</sup>, ГОСТ 19.103-77 <sup>2)</sup>, ГОСТ 19.104-

---

<sup>1)</sup> ГОСТ 19.101-77 ЕСПД. Виды программ и программных документов

78\* <sup>3)</sup>, ГОСТ 19.105-78\* <sup>4)</sup>, ГОСТ 19.106-78\* <sup>5)</sup>, ГОСТ 19.505-79\* <sup>6)</sup>, ГОСТ 19.604-78\* <sup>7)</sup>).

---

<sup>2)</sup> ГОСТ 19.103-77 ЕСПД. Обозначение программ и программных документов

<sup>3)</sup> ГОСТ 19.104-78\* ЕСПД. Основные надписи

<sup>4)</sup> ГОСТ 19.105-78\* ЕСПД. Общие требования к программным документам

<sup>5)</sup> ГОСТ 19.106-78\* ЕСПД. Общие требования к программным документам, выполненным печатным способом

<sup>6)</sup> ГОСТ 19.505-79\* ЕСПД. Руководство оператора. Требования к содержанию и оформлению

<sup>7)</sup> ГОСТ 19.604-78\* ЕСПД. Правила внесения изменений в программные документы, выполненные печатным способом

## **1. Назначение программы**

### **1.1. Функциональное назначение программы**

Функциональным назначением программы является ведения учета семейного бюджета.

### **1.2. Эксплуатационное назначение программы**

Программное обеспечение «FamilyBudget» предназначено для использования в учебных целях по изучению работы баз данных в среде разработки Lazarus.

### **1.3. Состав функций**

- 1) Функция добавления элемента в таблицу
- 2) Функция редактирования элемента в таблице
- 3) Функция удаления элемента в таблице
- 4) Функция фильтрации таблицы по выбранному критерию
- 5) Функция подсчета итоговых доходов семьи
- 6) Функция подсчета итоговых расходов семьи
- 7) Функция выбора файла для открытия
- 8) Функция открытия окна таблиц
- 9) Функция подсчета итоговых расходов семьи по категориям
- 10) Функция подсчета итоговых доходов семьи по категориям

## **2. Условия выполнения программы**

### **2.1. Минимальные системные требования**

ОС: Windows Vista/7/8/10 x 32 bit

Процессор: 2.0 GHz Dual Core Processor

Оперативная память: 1 GB ОЗУ

DirectX: Версии 9.0с

Место на диске: 750 MB

## 2.2. Требование к персоналу (пользователю)

Конечный пользователь программы должен обладать практическими навыками работы с графическим пользовательским интерфейсом операционной системы.

## 3. Выполнение программы

### 3.1. Запуск программы

Запустите программу «FamilyBudget» и ждите загрузки окна для выбора файла базы данных

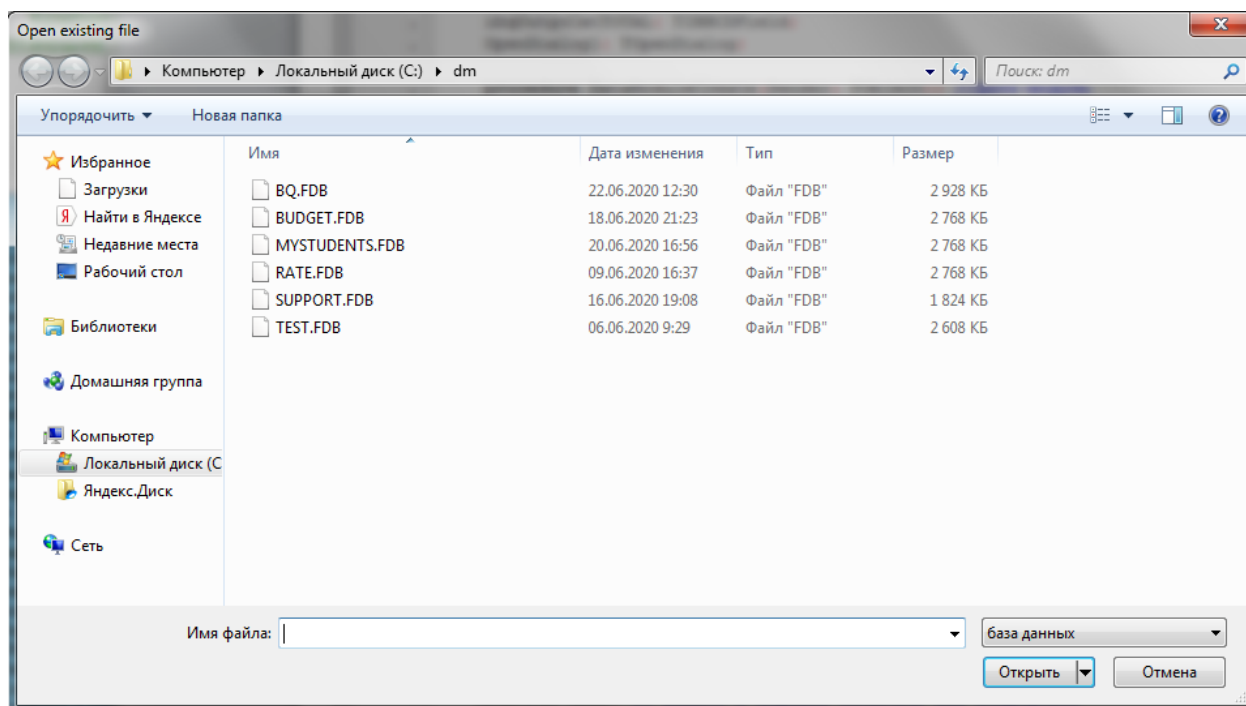


Рис.37 Выбор файла

После выбора соответствующего файла откроется окно главного меню программы с выбором кнопок:

«Добавить доход/расход» - открытие окна с вкладками, содержащими таблицы заполнения доходов и расходов.

«Бюджет» - открытие окна с вкладками, содержащими информацию о бюджете.



«О программе» - открытие окна с информацией о программе.

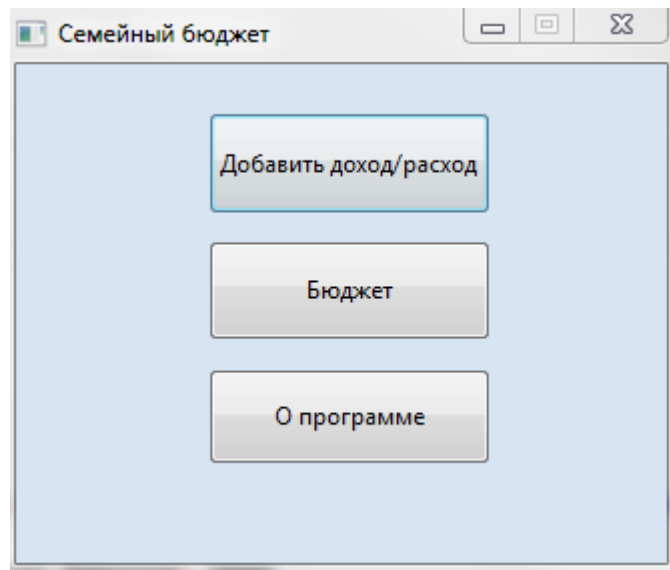


Рис.38 Главное меню

### **3.2. Выполнение функций программы**

#### **3.2.1. Выполнение функции открытия окна таблиц**

Выберите пункт меню «Добавить доход/расход».

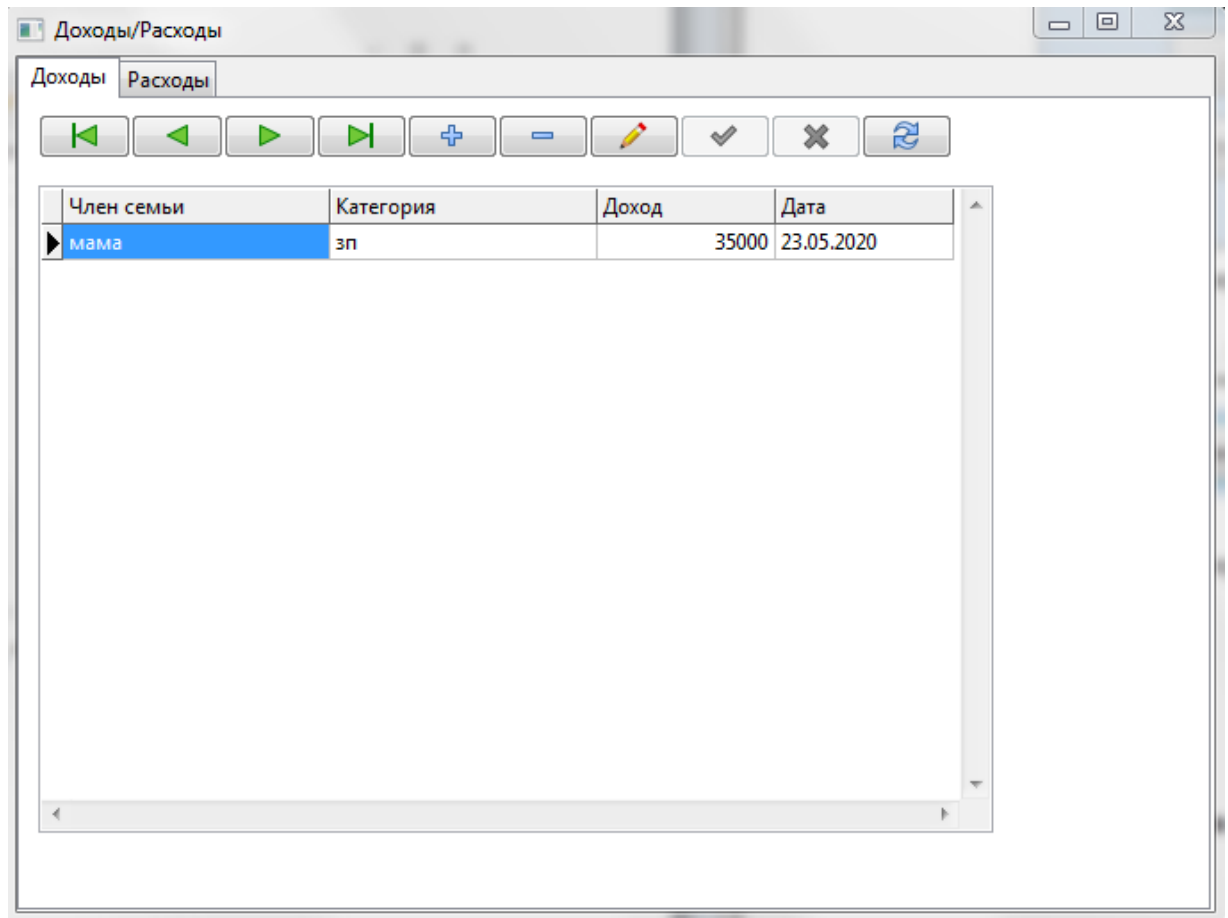


Рис.39 Окно «Доходы/Расходы»

### 3.2.2. Функция добавления элемента в таблицу

На панели навигации (расположена над таблицей) выберите кнопку добавления (insert) и заполните новую строку данными. После заполнения нажмите на кнопку подтверждения (post) и обновления (refresh).

Доходы/Расходы

Доходы Расходы

Член семьи Категория Доход Дата

мама	зп	35000	23.05.2020
------	----	-------	------------

Рис.40 Добавление элемента в таблицу

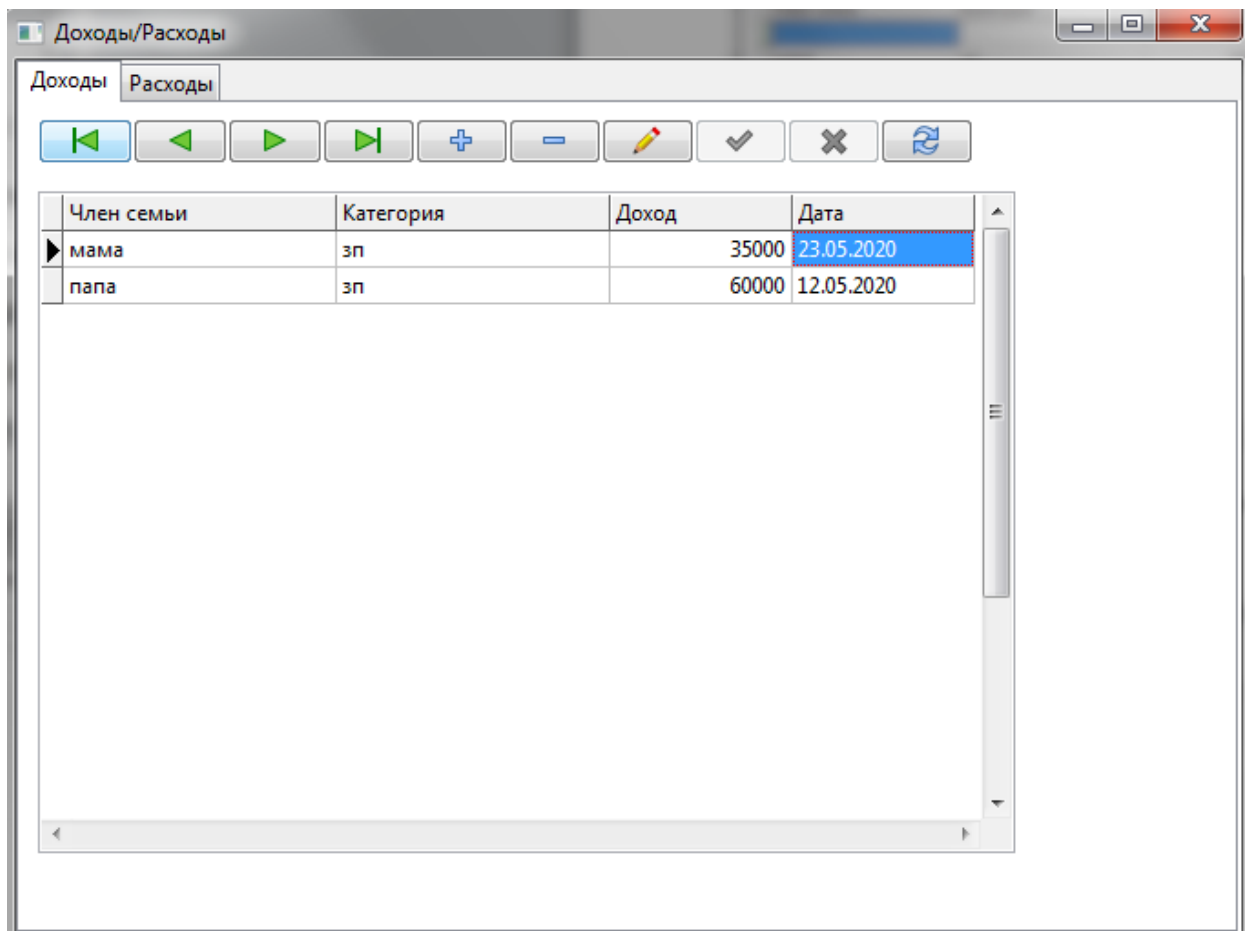


Рис.41 Элемент добавлен в таблицу

### 3.2.3. Функция удаления элемента из таблицы.

На панели навигации, используя кнопки передвижения, выберите необходимую запись и нажмите кнопку удаления (delete). В появившемся диалоговом окне нажмите кнопку ОК. Обновите таблицу (refresh).

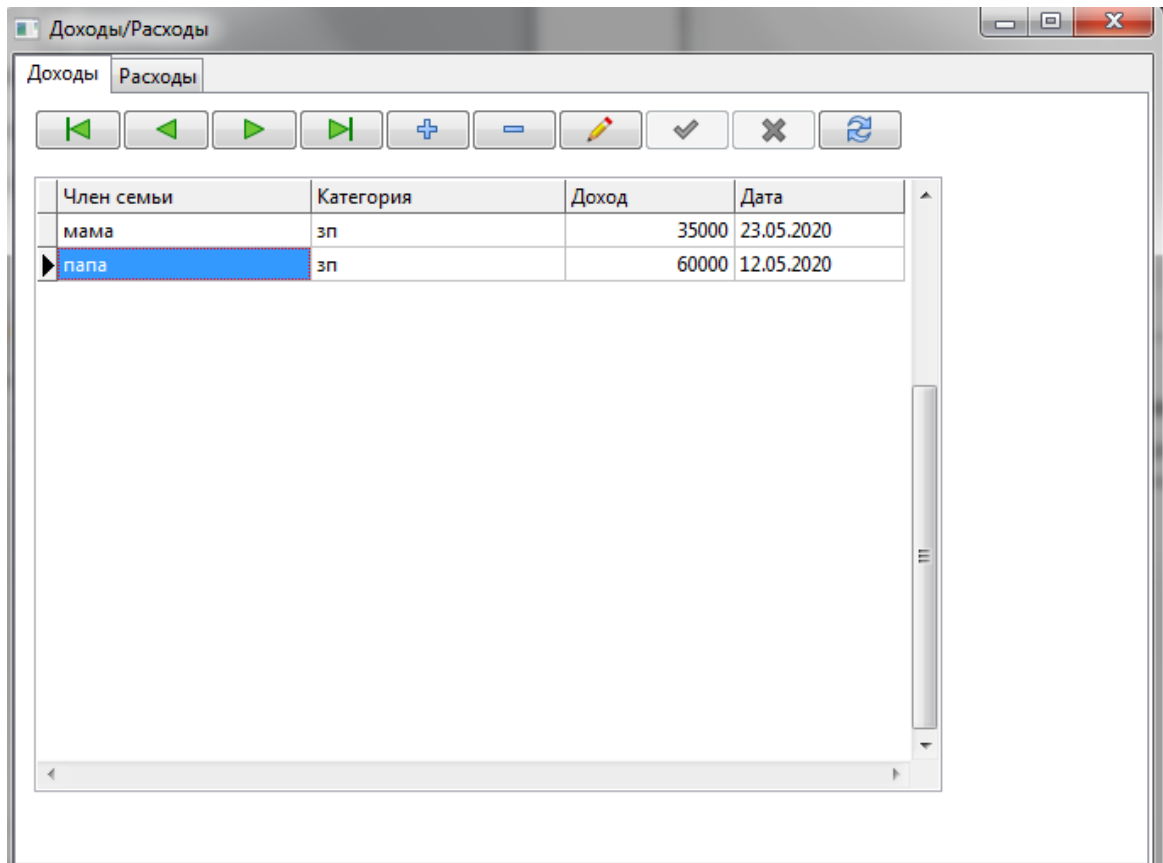


Рис.42 Выбор записи для удаления

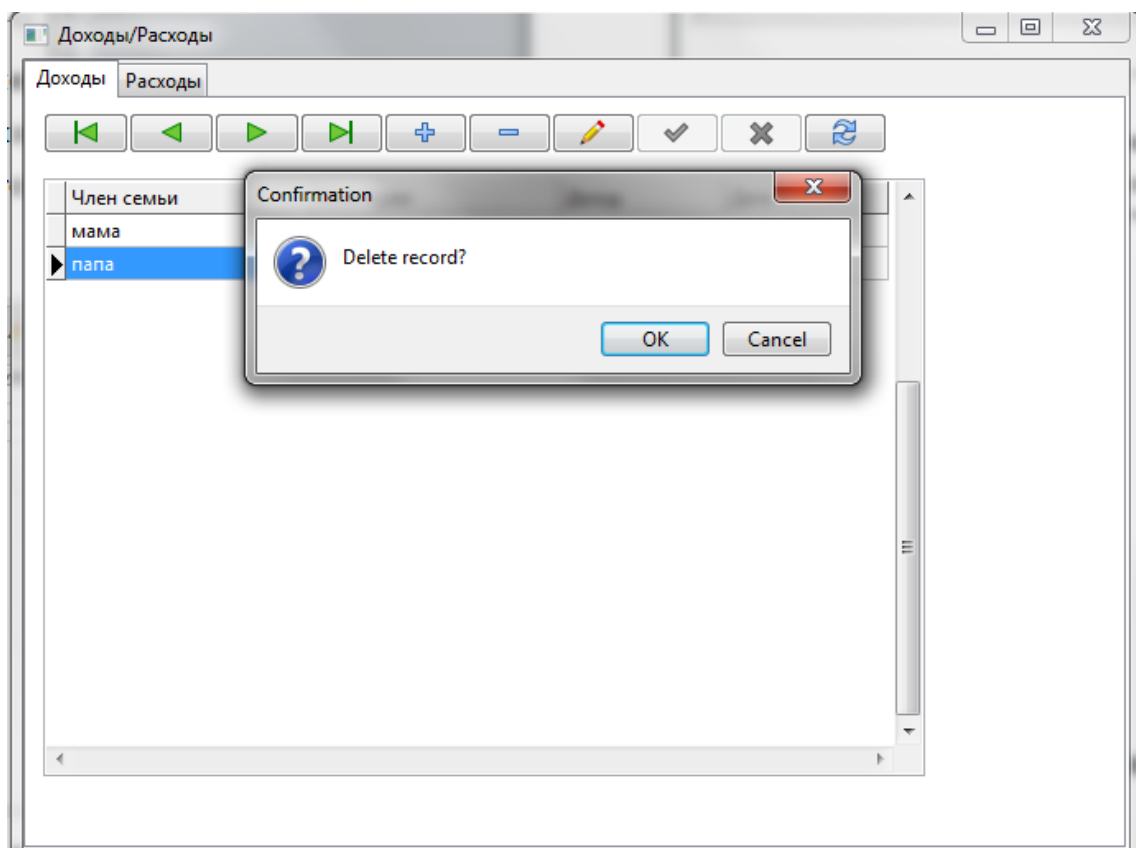


Рис.43 Подтверждение удаления

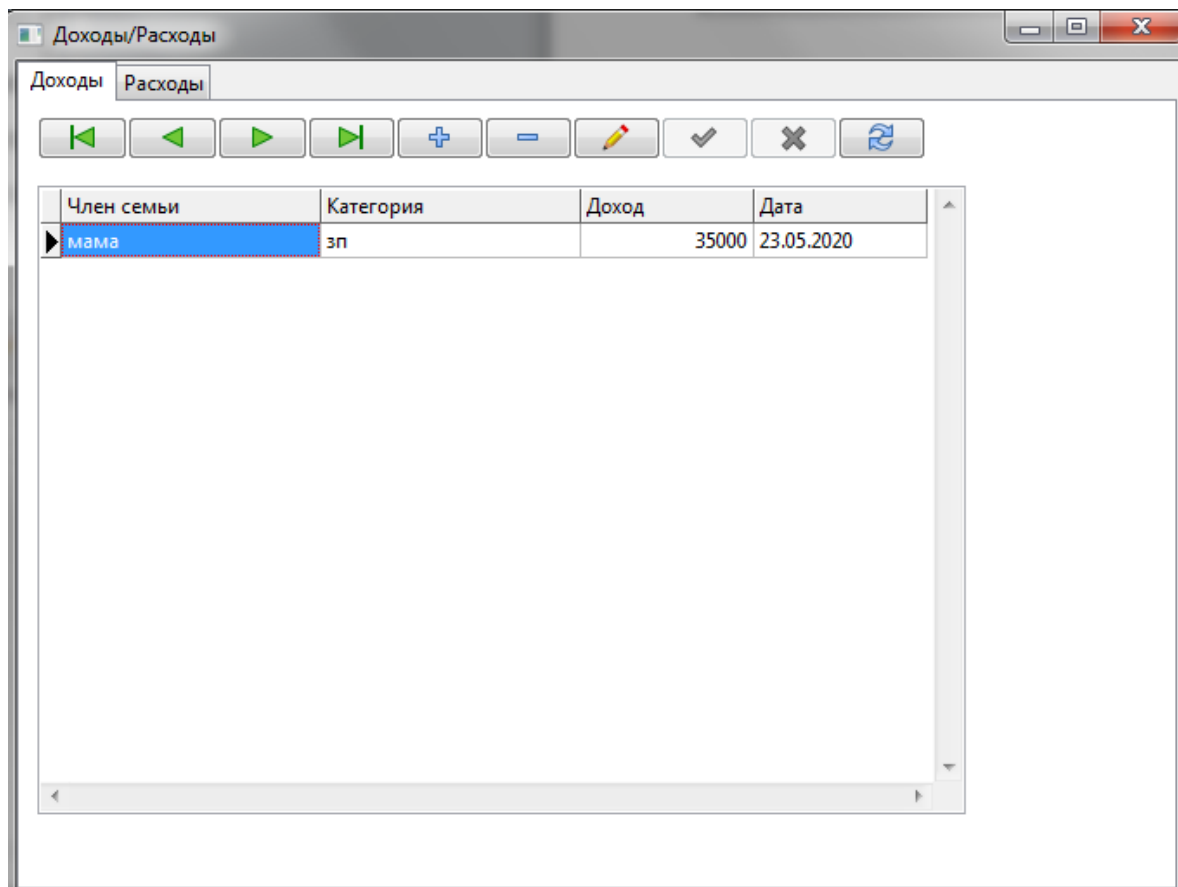


Рис.44 Удаление записи

### 3.2.4. Функция редактирования элемента в таблице

На панели навигации, используя кнопки передвижения, выберите необходимую запись и нажмите кнопку редактирования (edit). После редактирования записи нажмите кнопку подтверждения (post) и обновите таблицу (refresh).

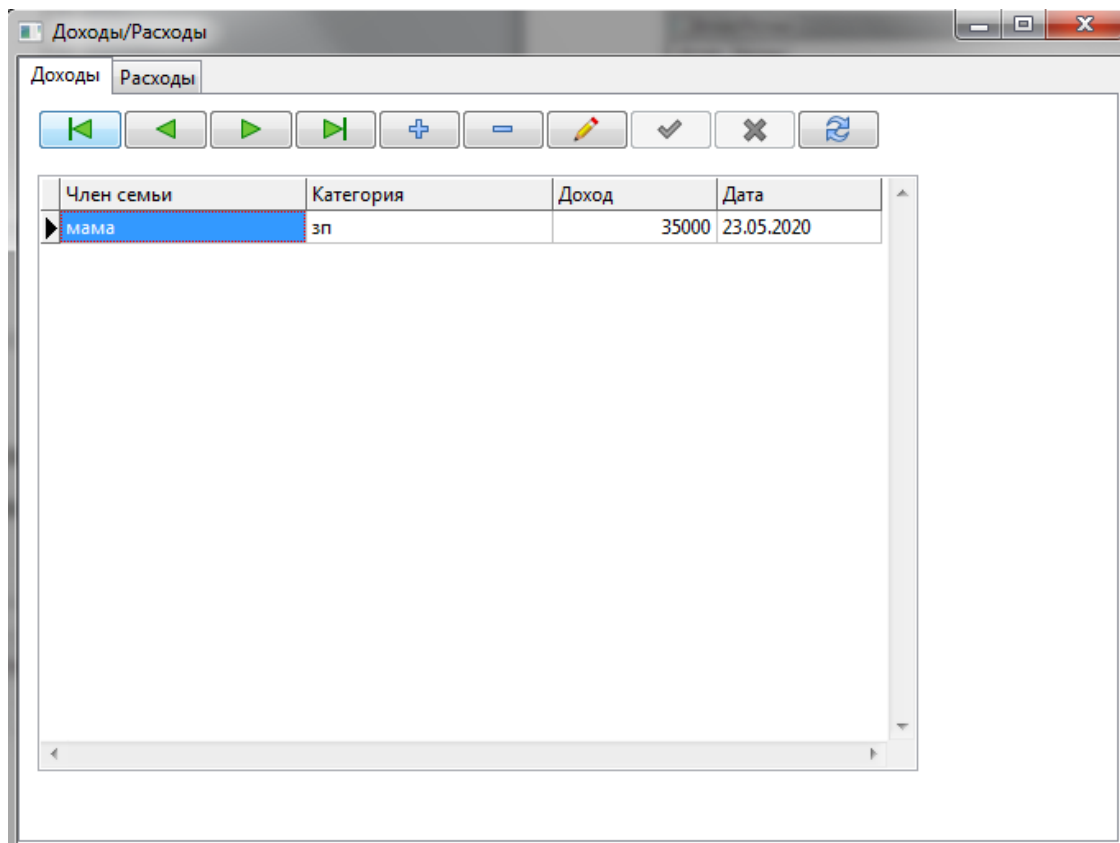


Рис.45 Выбор записи для редактирования

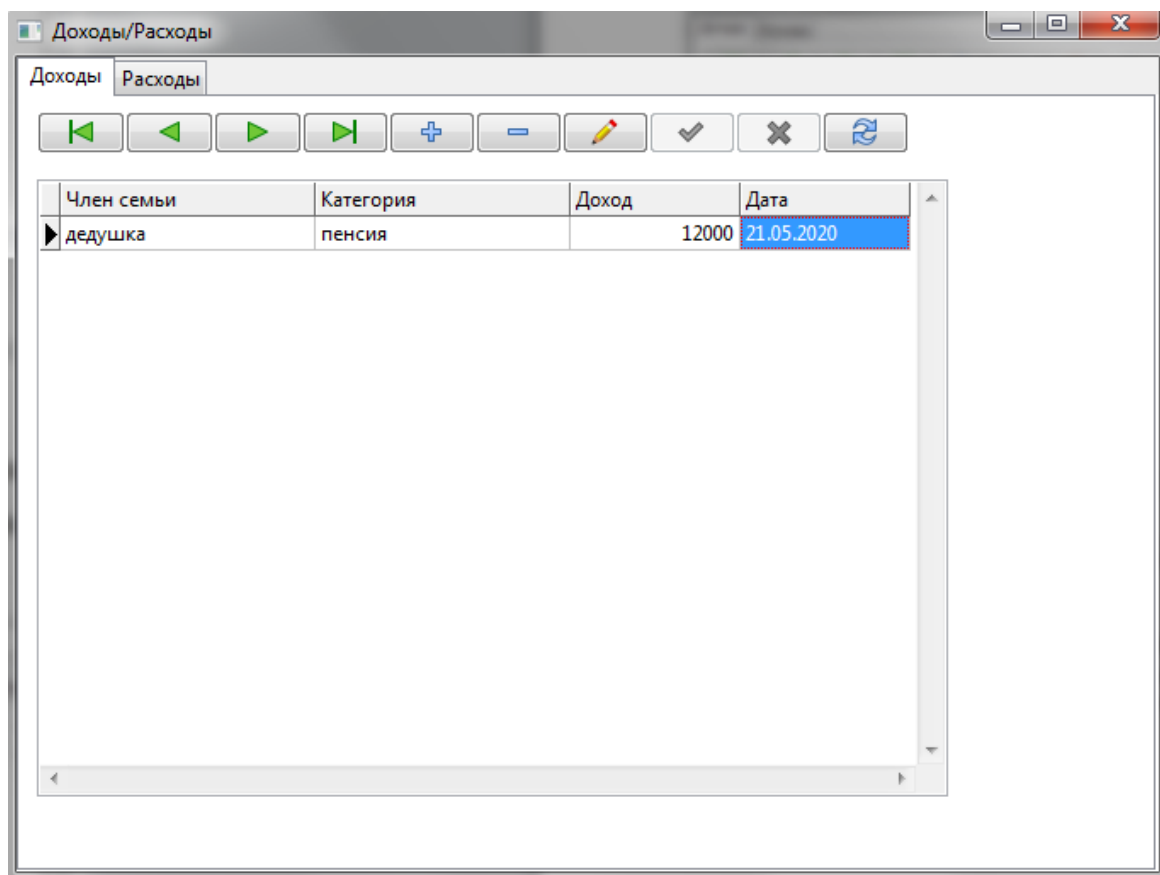


Рис.46 Запись отредактирована

### 3.2.5. Функция фильтрации таблицы по выбранному критерию

В главном меню программы нажмите кнопку «Бюджет» и ждите открытия соответствующего окна. Нажмите на поле под надписью «Фильтр доходов по категориям» и выберите категорию. Произойдет фильтрация таблицы по выбранной категории.

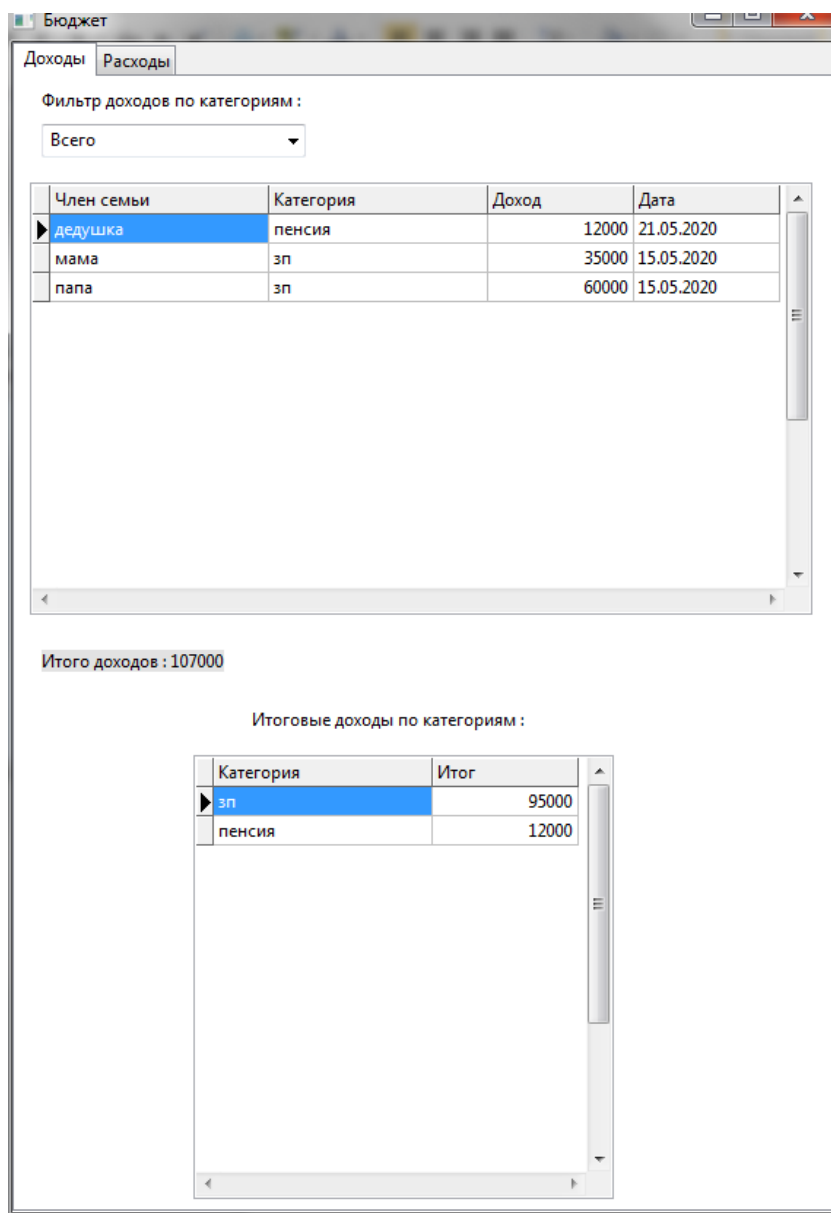


Рис.47 Окно «Бюджет»



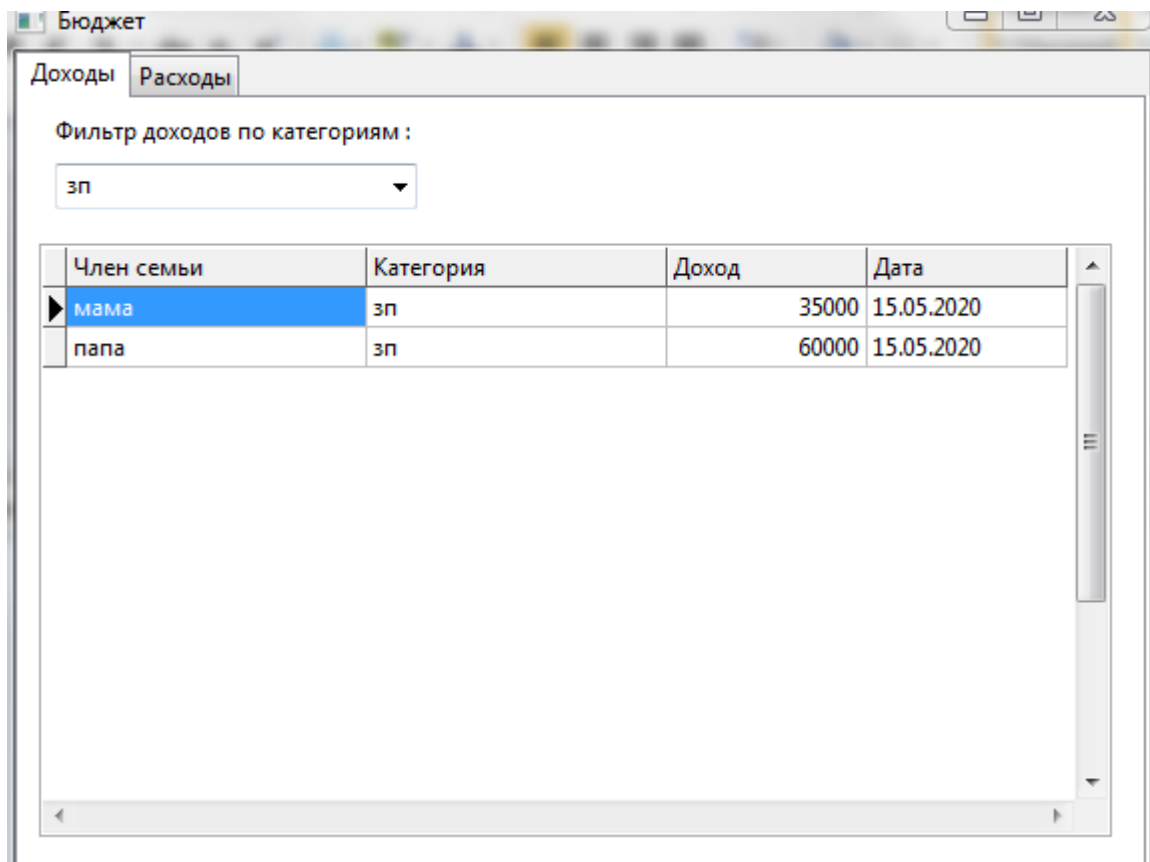


Рис.48 Фильтр по категории «зп»

### 3.2.6. Функция подсчета итоговых доходов семьи

В окне «Бюджет» выберите вкладку «Доходы». В поле «Итого доходов» можно посмотреть текущую сумму всех доходов семьи.

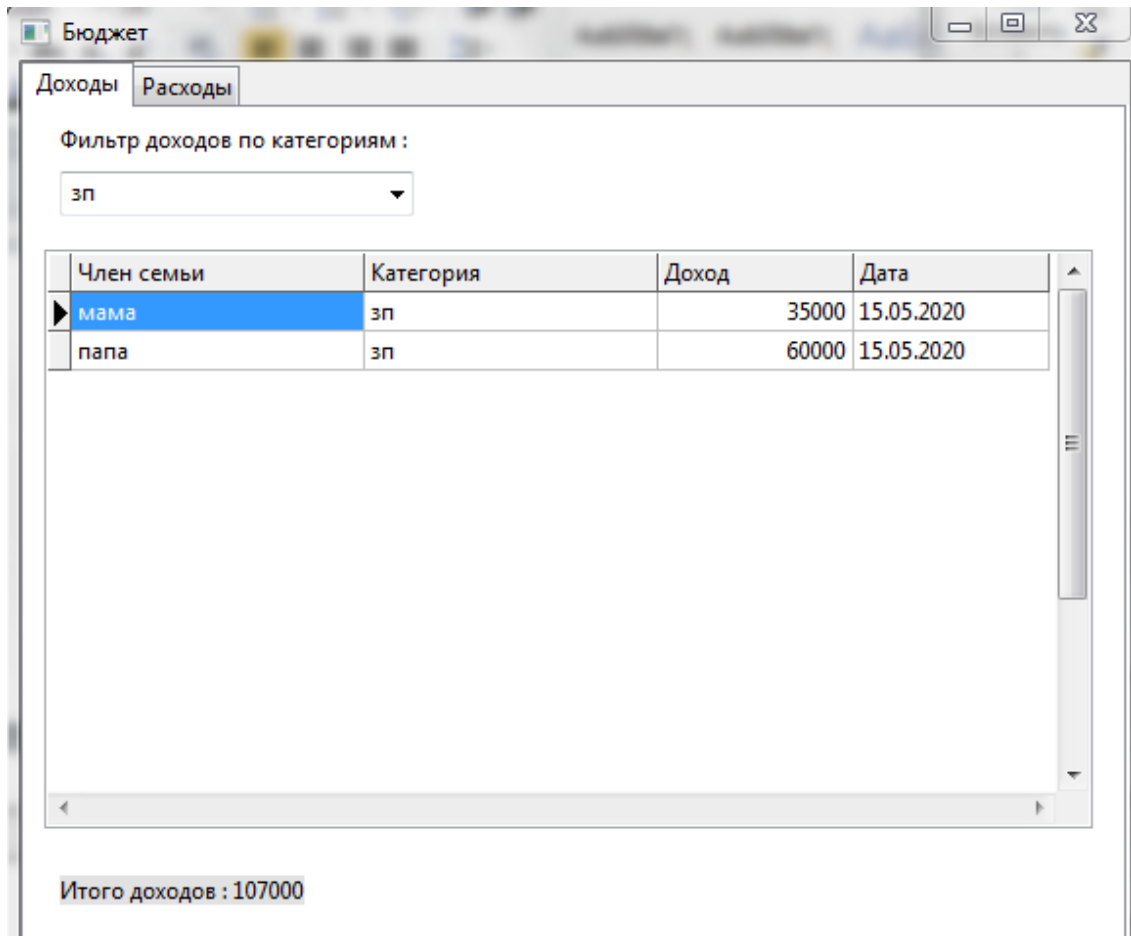


Рис.49 Текущая сумма всех доходов

### 3.2.7. Функция подсчета итоговых расходов семьи

В окне «Бюджет» выберите вкладку «Расходы». В поле «Итого расходов» можно посмотреть текущую сумму всех расходов семьи.

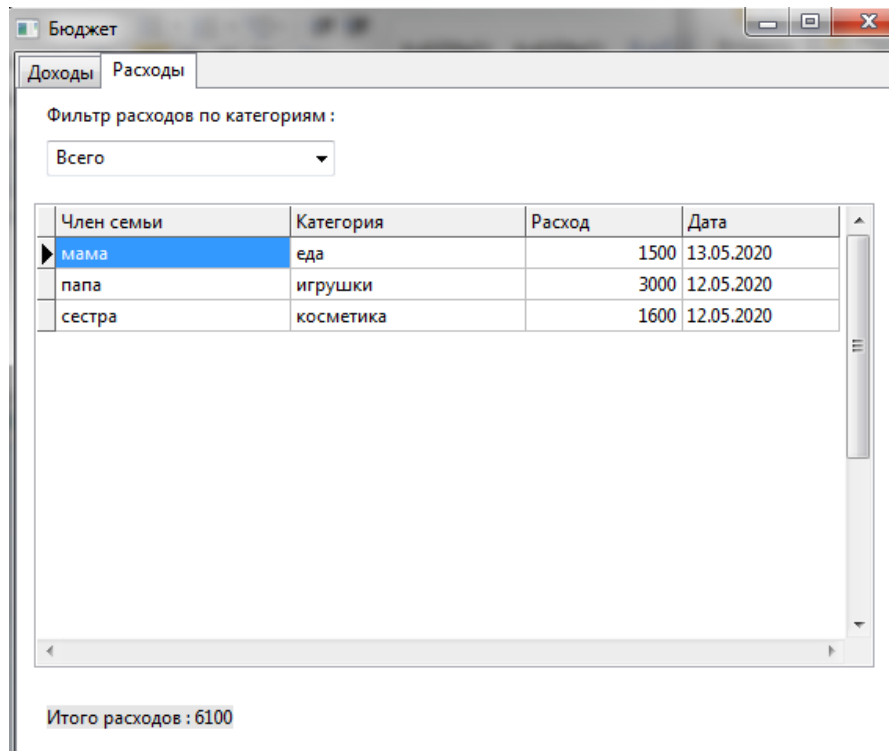


Рис.50 Текущая сумма всех расходов

### 3.2.8. Функция подсчета итоговых доходов семьи по категориям

В окне «Бюджет» выберите вкладку «Доходы». Под надписью «Итоговые доходы по категориям» расположена таблица с категориями и итоговыми доходами по категориям.

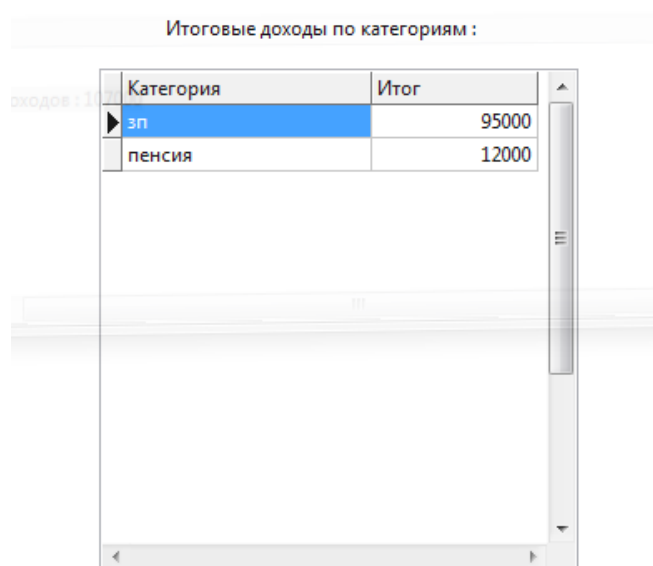


Рис.51 Итоговые доходы по категориям

### 3.2.9. Функция подсчета итоговых расходов семьи по категориям

В окне «Бюджет» выберите вкладку «Расходы». Под надписью «Итоговые расходы по категориям» расположена таблица с категориями и итоговыми расходами по категориям.

Итоговые расходы по категориям :

Категория	Итог
еда	1500
игрушки	3000
косметика	1600

Рис.52 Итоговые расходы по категориям

### 3.3. Завершение работы программы

Для завершения работы программы нажмите на крестик в верхнем правом углу главного меню.

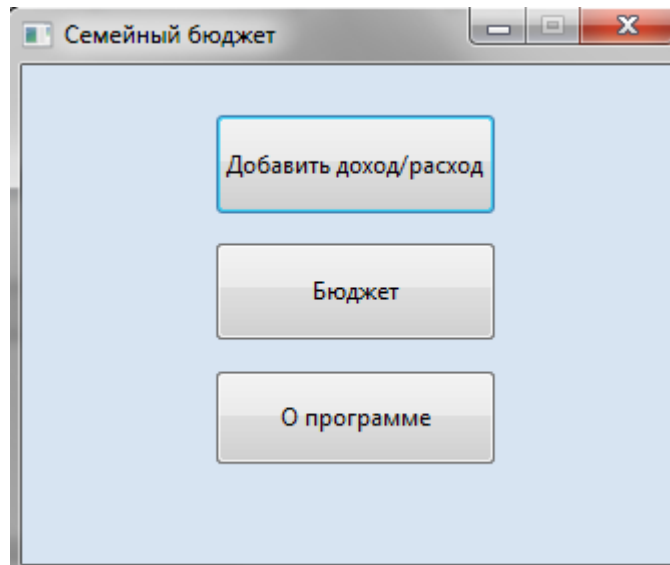


Рис.53 Главное меню

#### 4. Сообщение оператору

Если не заполнить новую запись в таблице, то при попытке ее сохранения программа выдаст ошибку:

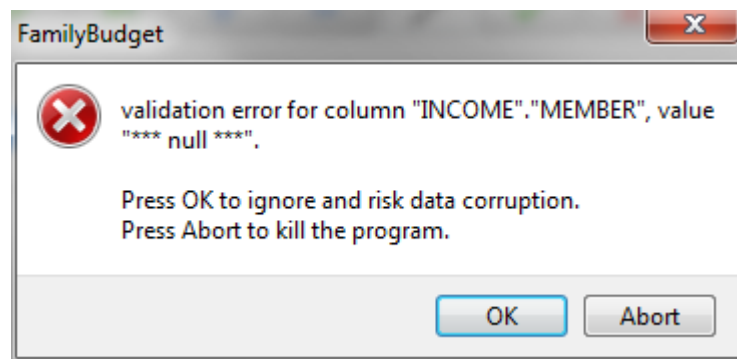


Рис.54 Сохранение незаполненной записи

Нажмите клавишу «OK» и заполните запись.

## **Заключение**

В результате выполнения курсового проекта было разработано приложение «Семейный бюджет» для учета ведения бюджета семьи.

В ходе работы были проанализированы существующие на данный момент разработки, посвященные данному направлению и изучены следующие темы: «Работа с базами данных FireBird», «Разработка приложения с базами данных» и «Программирование в среде Lazarus».

В дальнейшем планируется поменять интерфейс программы, добавить просмотр доходов/расходов по датам, добавить обновление полей итоговых доходов/расходов по мере добавления новых записей без завершения программы, а также сделать обработчик ошибок при заполнении таблиц.

## Список литературы и интернет источников

- 4.1. Официальный сайт 1С: Деньги 8:  
<https://v8.1c.ru/money/>
5. Официальный сайт Alzex Finance:  
<https://www.personalfinances.ru/>
6. Официальный сайт среды разработки Lazarus:  
<https://www.lazarus-ide.org/>
7. Официальный сайт FireBird:  
<https://www.firebirdsql.org/en/server-packages/>
8. Официальный сайт IBExpert:  
<https://www.ibexpert.net/ibe/>  
Бесплатная полнофункциональная версия:  
<https://ru.wikipedia.org/wiki/IBExpert>
9. Статья «Семейный бюджет»:  
<https://infobank.by/prosmotr-konkursa/belorusskie-blogery-au-kto-xochet-zarabotatj-700-rublej/chto-takoe-semejnyj-byudzheth-i-zachem-on-nuzhen/>

## Приложение 1. Модуль fmain

```

unit fmain;

{$mode objfpc}{$H+}

interface

uses
  Classes, SysUtils, Forms, Controls, Graphics, Dialogs, StdCtrls,
  money, budget, aboutProgram;

type

  { Tmain }

  Tmain = class(TForm)
    budget: TButton; //бюджет
    about: TButton; // о программе
    tables: TButton; // таблицы
    procedure aboutClick(Sender: TObject); //обработчик О программе
    procedure budgetClick(Sender: TObject); // обработчик бюджет
    procedure tablesClick(Sender: TObject);
  // обработчик ввод дохода/расхода
  private

  public

  end;

var
  ument: Tmain;

implementation
  uses dm;

  {$R *.lfm}

  { Tmain }

  procedure Tmain.tablesClick(Sender: TObject);
  begin
    ument2.Show();// форма ввода дохода/расхода
  end;

  procedure Tmain.budgetClick(Sender: TObject);
  begin
    ument3.show(); // форма бюджета
  end;

  procedure Tmain.aboutClick(Sender: TObject);
  begin
    ument4.show();//форма о программе
  end;
end.

```



## Приложение 2. Модуль budget

```

unit budget;

{$mode objfpc}{$H+}

interface

uses
    Classes, SysUtils, Forms, Controls, Graphics, Dialogs, StdCtrls,
    DBGrids,
    ComCtrls, Types;

type

    { Tmain3 }

    Tmain3 = class(TForm)
        cmb1: TComboBox; //комбобокс для категорий доходов
        cmb2: TComboBox; //комбобокс для категорий расходов
        DBGrid1: TDBGrid;
        //компонент отображения результата запроса фильтрации доходов по
        категориям
        DBGrid2: TDBGrid;
        //компонент отображения результата запроса фильтрации расходов по
        категориям
        DBGrid3: TDBGrid;
        //компонент отображения результата запроса суммы доходов по
        категориям
        DBGrid4: TDBGrid;
        //компонент отображения результата запроса суммы расходов по
        категориям
        Label1: TLabel;
        Label2: TLabel;
        IncomeSum: TLabel; //отображение результата запроса суммы доходов
        Label3: TLabel;
        Label4: TLabel;
        outgoSum: TLabel; //отображение результата запроса суммы расходов
        PageControl1: TPageControl; //окно с вкладками
        TabSheet1: TTabSheet; //вкладка Доходы
        TabSheet2: TTabSheet; //вкладка Расходы
        procedure cmb1Change(Sender: TObject);
        //вывод доходов по категориям
        procedure cmb2Change(Sender: TObject);
        //вывод расходов по категориям
        procedure FormCreate(Sender: TObject);
        //обработчик события OnCreate
    private
    public

```

```

        procedure loadtable(tablename : string; cmb : TCombobox);
//создание выпадающего списка в комбобоксе
        end;

var
    umein3: Tumein3;

implementation
uses dm;

{$R *.lfm}

{ Tumein3 }

procedure Tumein3.FormCreate(Sender: TObject);
begin
    loadtable('income', cmb1); //комбобокс для таблицы доходов
    loadtable('outgo', cmb2);  //комбобокс для таблицы расходов
    outgoSum.Caption := 'Итого расходов : ' +
IntToStr(dm.dmb.ibqOutgoSum.FieldValues['d']);
// выводим в текст label значение запроса суммы расходов
    IncomeSum.Caption := 'Итого доходов : ' +
IntToStr(dm.dmb.ibqIncomeSum.FieldValues['d']);
// выводим в текст label значение запроса суммы доходов
end;

procedure Tumein3.cmb1Change(Sender: TObject);
//вывод доходов по категориям
var
    ind:string;
begin
    dm.dmb.ibqIncome.close;
    if cmb1.ItemIndex <> 0 then //если выбран пункт "всего"
    begin
        ind := cmb1.text;
        dm.dmb.ibqIncome.sql[1] := 'where category='+chr(39)+ind+chr(39);
    end
    else dm.dmb.ibqIncome.SQL[1]:='';
//иначе оставляем запрос как есть и выводим все категории
    dm.dmb.ibqIncome.open;
end;

procedure Tumein3.cmb2Change(Sender: TObject);
//вывод расходов по категориям
var
    ind:string;
begin
    dm.dmb.ibqOutgo.close;
    if cmb2.ItemIndex <> 0 then //если выбран пункт "всего"

```

```

begin
    ind := cmb2.text;
    dm.dmb.ibqOutgo.sql[1] := 'where category='+chr(39)+ind+chr(39);
end
else dm.dmb.ibqOutgo.SQL[1]:='';
//иначе оставляем запрос как есть и выводим все категории
dm.dmb.ibqOutgo.open;
end;

procedure Tmain3.loadtable(tablename : string; cmb : TCombobox);
// создаем комбобокс с выпадающим списком
begin
    dm.dmb.ibq.close(); //закрыть запрос
    if not (dm.dmb.ibq.Transaction.Active) then
dm.dmb.ibq.Transaction.StartTransaction;
//запустить транзакцию, если неактивна
    dm.dmb.ibq.SQL.Text := 'select distinct category from ' +
tablename; //запрос на список неповторяющихся категорий
    dm.dmb.ibq.open; //открыть запрос
    while not (dm.dmb.ibq.eof) do //пока не конец запроса
        begin
            cmb.items.add(dm.dmb.ibq.FieldName('CATEGORY').AsString);
//берем название поля и добавляем в свойство items
            dm.dmb.ibq.next; //сдвиг
        end;
    end;
end;
end.

```

### Приложение 3. Модуль AboutProgram

```
unit aboutProgram;

{$mode objfpc}{$H+}

interface

uses
  Classes, SysUtils, Forms, Controls, Graphics, Dialogs, StdCtrls;

type

  { Tmain4 }

  Tmain4 = class(TForm)
    Memo1: TMemo; //текстовый редактор

  private

  public

  end;

var
  umain4: Tmain4;

implementation

{$R *.lfm}

{ Tmain4 }

end.
```

## Приложение 4. Модуль dm

```

unit dm;

{$mode objfpc}{$H+}

interface

uses
  Classes, SysUtils, db, IBDatabase, IBTable, IBCustomDataSet,
  IBQuery,
  IBXServices, StdCtrls, Dialogs;

type

  { Tdmb }

Tdmb = class(TDataModule)
  //ресурсы запросов
  dsqOutgoCat: TDataSource;
  dsqIncomeCat: TDataSource;
  dsqOutgoSum: TDataSource;
  dsqIncomeSum: TDataSource;
  dsqOutgo: TDataSource;
  dsqIncome: TDataSource;
  dsOutgo: TDataSource;
  dsIncome: TDataSource;

  ibdb: TIBDatabase; //база данных
  ibtr: TIBTransaction; //транзакция

  ibIncome: TIBTable; //таблица доходов бд

  //поля таблицы доходов
  ibIncomeCATEGORY: TIBStringField;
  ibIncomeDATA: TDateField;
  ibIncomeID: TIBIntegerField;
  ibIncomeMEMBER: TIBStringField;
  ibIncomePROFIT: TIBBCDField;

  ibOutgo: TIBTable; //таблица расходов

  //поля таблицы расходов
  ibOutgoCATEGORY: TIBStringField;
  ibOutgoCOST: TIBBCDField;
  ibOutgoDATA: TDateField;
  ibOutgoID: TIBIntegerField;
  ibOutgoMEMBER: TIBStringField;

```

```

ibqIncome: TIBQuery; //запрос к таблице Доходов

//поля запроса
ibqIncomeCATEGORY: TIBStringField;
ibqIncomeDATA: TDateField;
ibqIncomeMEMBER: TIBStringField;
ibqIncomePROFIT: TIBBCDField;

ibqIncomeSum: TIBQuery; //запрос на сумму доходов
ibqIncomeSumD: TIBBCDField; //поле запроса

ibq: TIBQuery; //запрос для комбобокса

ibqOutgo: TIBQuery; //запрос к таблице Расходов

//поля запроса
ibqOutgoCATEGORY: TIBStringField;
ibqOutgoCOST: TIBBCDField;
ibqOutgoDATA: TDateField;
ibqOutgoMEMBER: TIBStringField;

ibqOutgoSum: TIBQuery; //запрос на сумму расходов
ibqOutgoSumD: TIBBCDField; //поле запроса

ibqIncomeCat: TIBQuery; //запрос по категориям доходов

//поля запроса
ibqIncomeCatCATEGORY: TIBStringField;
ibqIncomeCatTOTAL: TIBBCDField;

ibqOutgoCat: TIBQuery; //запрос по категориям расходов

//поля запроса
ibqOutgoCatCATEGORY: TIBStringField;
ibqOutgoCatTOTAL: TIBBCDField;
OpenDialog1: TOpenDialog;

procedure DataModuleCreate(Sender: TObject); //дата-модуль

private

public
end;

var
    dmb: Tdmb;

implementation
uses budget ;

```

```

{$R *.lfm}

{ Tdmb }

procedure Tdmb.DataModuleCreate(Sender: TObject);    //дата-модуль
begin
    if(OpenDialog1.Execute) then
    begin
        ibdb.DatabaseName :=
ExtractFileDir(OpenDialog1.FileName)+'\BQ.FDB'; //выбор пути бд
        ibdb.Connected:=true;
        end;
        ibtr.StartTransaction; //транзакция
        ibIncome.Open; // таблица доходов
        ibOutgo.Open; // таблица расходов
        ibqIncome.open; // запрос к таблице доходов для комбобокса
        ibqOutgo.open; // запрос к таблице расходов для комбобокса
        ibqOutgoSum.Open; // запрос для общей суммы расходов
        ibqIncomeSum.Open; // запрос для общей суммы расходов
        ibqIncomeCat.Open;
        // запрос для суммы доходов по каждой категории дохода
        ibqOutgoCat.Open; // запрос для суммы расходов по каждой категории
расхода
        end;
    end.

```

## Приложение 5. Модуль money

```

unit money;

{$mode objfpc}{$H+}

interface

uses
  Classes, SysUtils, Forms, Controls, Graphics, Dialogs, ComCtrls,
  DBCtrls,
  DBGrids, dm;

type

  { Tmain2 }

  Tmain2 = class(TForm)
    DBGrid1: TDBGrid; //компонент отображения таблицы доходов
    DBGrid2: TDBGrid; //компонент отображения таблицы расходов
    DBNavigator1: TDBNavigator;
    //компонент для работы с полями т. доходов
    DBNavigator2: TDBNavigator;
    //компонент для работы с полями т. расходов
    PageControl1: TPageControl; //окно с вкладками
    TabSheet1: TTabSheet; //вкладка "Доходы"
    TabSheet2: TTabSheet; //вкладка "Расходы"
  private

  public

  end;

var
  ument2: Tmain2;

implementation

{$R *.lfm}

end.

```



## Приложение 6. Модуль FamilyBudget

```

program FamilyBudget;

{$mode objfpc}{$H+}

uses
    {$IFDEF UNIX}{$IFDEF UseCThreads}
    cthreads,
    {$ENDIF}{$ENDIF}
    Interfaces, // this includes the LCL widgetset
    Forms, fmain, money, dm, budget, aboutProgram
    { you can add units after this };

{$R *.res}

begin
    RequireDerivedFormResource:=True;
    Application.Scaled:=True;
    Application.Initialize;
    Application.CreateForm(Tumain, umain); //экземпляр формы umain
    Application.CreateForm(Tumain2, umain2); //экземпляр формы umain2
    Application.CreateForm(Tdmb, dmb); //экземпляр формы dmb
    Application.CreateForm(Tumain3, umain3); //экземпляр формы umain3
    Application.CreateForm(Tumain4, umain4); //экземпляр формы umain4
    Application.Run;
end.

```