

Пермский филиал федерального государственного автономного  
образовательного учреждения высшего образования  
«Национальный исследовательский университет  
«Высшая школа экономики»

*Факультет экономики, менеджмента и бизнес-информатики*

Вальченко Антон Артемович

**РАЗРАБОТКА ПРИЛОЖЕНИЯ «ДОМАШНИЙ БУХГАЛТЕР»**

*Курсовая работа*

по направлению подготовки 09.03.04 Программная инженерия  
образовательная программа «Программная инженерия»

Руководитель  
старший преподаватель  
кафедры высшей  
математики

---

М. В. Крючков

Пермь, 2018 год

## **Аннотация**

В данной работе было разработано приложение «Домашний Бухгалтер».

Автор: студент НИУ-ВШЭ Пермь Вальченко Антон.

Кафедра информационных технологий в бизнесе.

Работа содержит 46 страниц формата А4 основного текста, включая 3 главы: которые будут тут перечислены.

В основной части содержится 15 картинок и одна таблица с примерами работы программы, описанием алгоритма, моделями интерфейса.

В работе содержится шесть приложений: блок схемы алгоритмов работы с файлами, блок-схемы алгоритмов работы с объектами, блок-схема алгоритма встроенной справки, исходные коды программы на внешнем носителе, тестирование, руководство пользователя.

Библиографический список состоит из семи источников.

В первой главе осуществляется анализ предметной области – проводится обзор аналогов. Как результат, оформлен список критериев оценки конечного продукта.

Вторая глава содержит описание алгоритмов и техник для удовлетворения выдвинутых к программе требований. Результатом данной главы является описание проектирования программы с помощью блок-схем и смоделированный интерфейс.

Третья глава решает задачу реализации и тестирования конечного продукта. Здесь рассмотрены принципы работы программы и тестирование с проверкой полноты по критериям черного ящика. Результатом является конечный продукт с графическим интерфейсом.

## Оглавление

Введение .....	4
1. Анализ предметной области .....	5
1.1. Обзор аналогов .....	5
1.2. Подведение итогов обзора аналогов.....	11
1.3. Требования к конечному продукту.....	12
1.4. Выводы по главе .....	12
2. Проектирование приложения .....	13
2.1. Разработка алгоритмов.....	13
2.2. Разработка интерфейса .....	16
2.3. Выводы по главе .....	17
3. Реализация и тестирование приложения .....	18
3.1. Описание классов, их полей, методов, функций.....	18
3.2. Реализация пользовательского интерфейса.....	19
3.3. Анализ и тестирование разработанного приложения.....	21
3.4. Выводы по главе .....	22
Заключение .....	23
Список сокращений и условных обозначений.....	24
Библиографический список .....	25
Приложение А. Блок-схемы алгоритмов работы с файлами.....	26
Приложение В. Блок-схемы алгоритмов работы с объектами.....	30
Приложение С. Блок-схема алгоритма встроенной справки.....	34
Приложение D. Исходные коды программы.....	35
Приложение Е. Тестирование .....	36
Приложение F. Руководство пользователя.....	40

## **Введение**

Денежные средства заняли прочную позицию в жизни современного человека, а их учёт и контроль – это то, что помогает не только жить на определённую сумму, но и с каждым разом как-то улучшать свою жизнь. Бухгалтерские программы призваны помочь человеку в этом вопросе, так как имеют достаточный функционал для обеспечения его точными расчётами и микропрогнозами по будущим тратам.

Актуальность теме диктует повышение спроса на простые и надёжные программы, предназначенные для людей, которые не имеют достаточного опыта работы с персональным компьютером, чтобы организовать их для себя из стандартных средств.

Объектом исследования проекта являются бухгалтерские программы для домашнего использования. В качестве предмета исследования выступает разработка и модификация программы данного типа.

Цель работы: создать собственную программу, адаптированную для русскоязычных пользователей, имеющих минимальные навыки пользования персональным компьютером. Для достижения поставленной цели требуется выполнить следующие задачи, согласно рекомендациям из [1, 2]:

1. Проанализировать предметную область.
2. Спроектировать приложение.
3. Реализовать приложение.
4. Протестировать приложение.

Научная новизна обусловлена новым применением известных решений и методов, так как бухгалтерские программы широко распространены и, как следствие, хорошо изучены. Для решения различных задач существует большой выбор алгоритмов, доступных для изучения и использования разработчиками программ данного типа.

В работе будут использованы такие методы исследования, как сравнение и теоретический анализ. Метод реализации – ООП.

Практическая значимость работы состоит в расширении знаний в области разработки прикладного программного обеспечения и сопутствующих алгоритмов, в создании программы данного типа.

# **1. Анализ предметной области**

Данная глава посвящена обзору аналогов, анализом особенностей бухгалтерских программ для домашнего использования, определение сильных и слабых сторон существующих решений. Как итог, в данной главе сформированы требования к конечному продукту.

## **1.1. Обзор аналогов**

В данном разделе рассмотрены и проанализированы следующие программы бухгалтерского учёта для домашнего пользования: HomeBank (v. 5.1.6), Домашняя бухгалтерия Lite 6 (v. 6.3), Портмоне (v. 1.0), Учёт расходов (v. 2.32).

### **1.1.1. HomeBank (v. 5.1.6)**

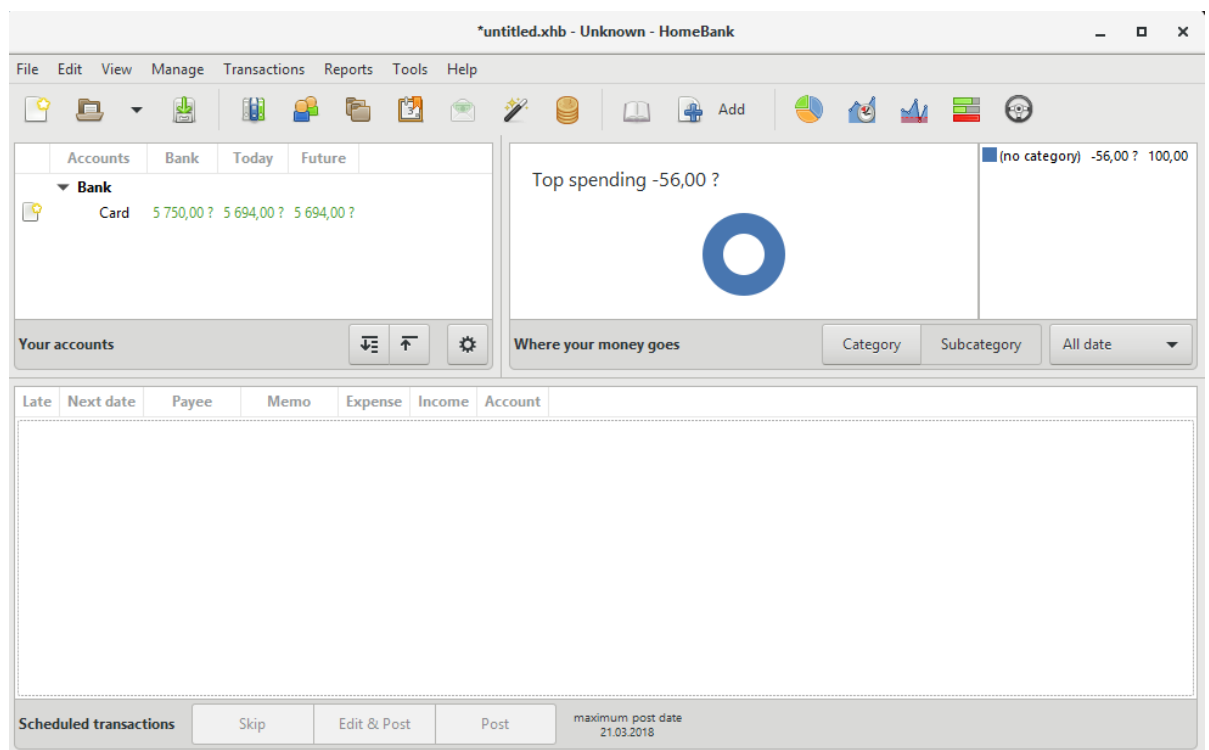
Программа имеет открытый исходный код, написана на языке программирования С, для взаимодействия с пользователем используется графический интерфейс, созданный с помощью библиотеки GTK. Рассмотрен интерфейс главного окна и окна добавления транзакции (см. рис. 1.1, 1.2).

Достоинства программы, выявленные в ходе анализа:

1. Приятный, интуитивно понятный интерфейс.
2. Красивые, современные иконки.
3. Малое количество настроек.
4. Обширная статистика.

Недостатки программы, выявленные в ходе анализа:

1. Неудобно реализовано изменение размеров интерфейса, при сжатии или растяжении главного окна внутренние элементы остаются на своих местах.
2. Чрезмерное количество обязательных полей для создания транзакции.
3. Иконки кнопок плохо отражают действие, совершаемое при нажатии на них.
4. Отсутствует перевод на русский язык.
5. Транзакции без категории не отображаются в общем списке.



**Рисунок 1.1. Главное окно программы**

The 'Add transaction' dialog box contains the following fields and controls:

- Date:** A date picker showing '21.03.2018'.
- Amount:** A text field with a currency symbol '¥/₹', a value of '0,00', and buttons for minus, plus, and a green Euro symbol.
- Payment:** A dropdown menu currently set to '(none)'.
- Info:** A text input field.
- Account:** A dropdown menu.
- Payee:** A dropdown menu.
- Category:** A dropdown menu.
- Status:** A set of buttons: 'None', 'Cleared', 'Reconciled', and 'Remind'.
- Memo:** A text input field.
- Tags:** A text input field.
- Buttons:** 'Close', 'Add & keep', and 'Add' at the bottom.

**Рисунок 1.2 Окно создания транзакции**

### 1.1.2. Домашняя Бухгалтерия Lite 6 (v. 6.3)

Программа имеет закрытый исходный код, распространяется платно, но имеет бесплатную урезанную версию. Рассмотрен интерфейс главного окна и окна добавления транзакции (см. рис. 1.3, 1.4).

Достоинства программы, выявленные в ходе анализа:

1. Реализована функция планирования.
2. Встроенный калькулятор.
3. Выбор между расширенным и компактным внешним видом.
4. Одновременная работа с несколькими валютами.

Недостатки программы, выявленные в ходе анализа:

1. Подавляющее большинство действий находится в меню «Сервис».
2. Часто показываются рекламные баннеры.

Дата	Счет	Категория расхода	Подкатегория расхода	Кол.	Ед. изм.	Сумма	Примечание
20.03.2018	Карта	Комиссия	Обмен валюты	1		5 000,00 p	
Строк: 1							5 000,00 p

	Рубли
<input checked="" type="checkbox"/> Сегодня	0,00
<input checked="" type="checkbox"/> Неделя	5 000,00
<input checked="" type="checkbox"/> Месяц	5 000,00
<input checked="" type="checkbox"/> Всего	5 000,00

Рисунок 1.3. Главное окно программы

***Рисунок 1.4. Окно добавления транзакции***

### **1.1.3. Портмоне (v. 1.0)**

Программа имеет закрытый исходный код, распространяется платно, но имеет бесплатный шестидесятидневный срок, предоставляемый по ключу на e-mail. Рассмотрен интерфейс главного окна и окна добавления транзакции (см. рис. 1.5, 1.6).

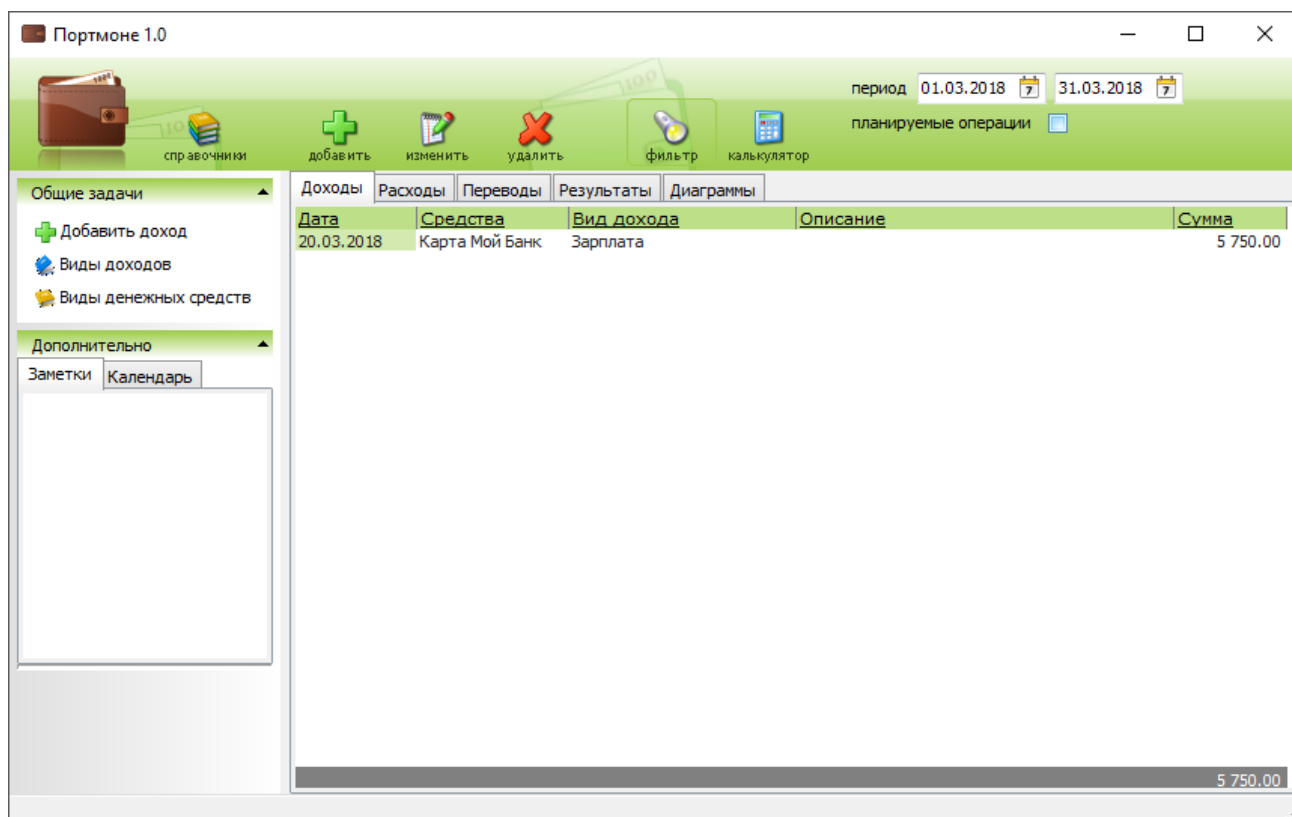
Достоинства программы, выявленные в ходе анализа:

1. Имеет встроенный вызов стандартного калькулятора.
2. «Умный фильтр».
3. Функция планирования.

Недостатки программы, выявленные в ходе анализа:

1. Не имеет функции работы со разными счетами.
2. Нечитаемый шрифт.
3. Высокая цена лицензионного продукта.
4. Недружелюбный интерфейс.





*Рисунок 1.5. Главное окно программы*

Редактор

Доход

Дата: 21.03.2018 | Планируемая операция

Вид средств: [dropdown]

Вид дохода: [dropdown]

Теги: [text]

Сумма: 0

Описание: [text area]

Сохранить | Сохранить и закрыть

*Рисунок 1.6. Окно добавления транзакции*

### 1.1.4. Учёт расходов (v. 2.32)

Программа имеет закрытый исходный код, свободно распространяется. Хранение и обработка данных происходит в базе данных Microsoft Access. Рассмотрено главное окно программы и окно создания расхода (см. рис. 1.7, 1.8).

Достоинства программы, выявленные в ходе анализа:

1. Простой и понятный интерфейс, который не отталкивает пользователя.
2. Минимальный полезный функционал.

Недостатки программы, выявленные в ходе анализа:

1. Не имеет функции работы с доходами.
2. Нет возможности изменения стандартной категории.
3. Не работает без предустановленного Microsoft Access.

Дата ^	Товар	Сумма	Категория	Подкатегория	Место	Примечание
07.09.2017	телевидение	204,00	Коммунальные	Телевидение	КИТС	август - октябрь
08.09.2017	отопление	300,00	Коммунальные	Отопление	ЦРБ	сентябрь
08.09.2017	Вывоз мусора	25,00	Коммунальные	вывоз мусора	ЦРБ	сентябрь
08.09.2017	квартплата	130,00	Коммунальные	квартплата	ЦРБ	сентябрь
08.09.2017	Электроэнергия	93,00	Коммунальные	Электроэнергия	ЦРБ	25235-25085=150 август
08.09.2017	ГАЗ	24,00	Коммунальные	Газ	ЦРБ	2486-2476=10 август
08.09.2017	услуги водоканала	100,00	Коммунальные	Водоканал	ЦРБ	сентябрь
30.09.2017	ГАЗ	40,19	Коммунальные	Газ	ЦРБ	
30.09.2017	Электроэнергия	93,00	Коммунальные	Электроэнергия	ЦРБ	
30.09.2017	квартплата	130,00	Коммунальные	квартплата	ЦРБ	
30.09.2017	отопление	300,00	Коммунальные	Отопление	ЦРБ	
30.09.2017	услуги водоканала	100,00	Коммунальные	Водоканал	ЦРБ	
30.09.2017	Вывоз мусора	25,00	Коммунальные	вывоз мусора	ЦРБ	

Версия: 2.32 от 01.11.2017 г.    Записей: 13, Сумма: 1564,19

Рисунок 1.7. Главное окно программы

Добавление записи о расходах

Дата: 21.03.2018 Категория: Подкатегория:

Сумма: Товар / Услуга: Место:

Примечание:

Сохранить Отменить

Рисунок 1.8. Окно добавления расхода

## 1.2. Подведение итогов обзора аналогов

Как итог обзора, собраны все достоинства и недостатки рассмотренных аналогов, составлен их общий список, а также сравнительная таблица (см. таблицу 1.1).

### 1. Достоинства программ.

- Приятный, интуитивно понятный интерфейс.
- Красивые, современные иконки.
- Обширная статистика.
- Реализована функция планирования.
- Встроенный калькулятор.
- «Умный фильтр».
- Минимальный полезный функционал.

### 2. Недостатки программ.

- Неудобно реализовано изменение размеров интерфейса, при сжатии или растяжении главного окна внутренние элементы остаются на своих местах.
- Чрезмерное количество обязательных полей для создания транзакции.
- Иконки кнопок плохо отражают действие, совершаемое при нажатии на них.
- Не имеет перевода на русский язык.
- Подавляющее большинство действий находится в меню «Сервис».
- Не имеет функции работы с доходами.
- Нет возможности изменения стандартной категории.

**Таблица 1.1. Сравнительная таблица аналогов**

	HomeBank	Домашняя Бухгалтерия Lite 6	Портмоне	Учёт расходов
Качественный интерфейс	+	+	-	+
Минимальный функционал	-	-	+	+
Статистика	+	+	-	+
Русский язык	-	+	+	+
Работа с доходами и расходами	+	+	+	-
Работа с несколькими счетами	+	+	-	-
Работа с категориями	+	+	+	+

### **1.3. Требования к конечному продукту**

Основываясь на проведенном обзоре аналогов и его анализе, выделим наиболее важные функции, которыми должна обладать программа, и существенные недостатки, которых она не должна иметь.

1. Простой, интуитивно понятный интерфейс на русском языке с настройкой размеров окна, связанного с внутренними частями.
2. Минимальный полезный функционал, который не имеет ничего лишнего.
3. Работа как с доходами, так и с расходами.
4. Работа с категориями.
5. Собственная система организации данных, которая не зависит от сторонних компонентов.
6. Код программы должен предусматривать обработку исключительных ситуаций.

### **1.4. Выводы по главе**

В данной главе были рассмотрены четыре аналога разрабатываемой программы, выделены их сильные и слабые стороны, определены наиболее важные достоинства и самые часто встречаемые недостатки, основываясь на которых, были разработаны требования к конечному продукту. В следующей главе будут рассмотрены проблемы, которые могут возникнуть в ходе решения поставленной задачи, описаны пути их решения и предложены алгоритмы, реализующие основные и вспомогательные функции.

## **2. Проектирование приложения**

Вторая глава содержит подробное изложение разрабатываемых алгоритмов, описание используемых средств реализации графического интерфейса, обоснование их выбора, для удовлетворения сформированным требованиям. Результат данной главы – описание проектирования конечного продукта с помощью блок-схем разработанных алгоритмов и эскизами графического интерфейса.

### **2.1. Разработка алгоритмов**

Данная программа изначально должна запустить графический интерфейс, а после дожидаться команд от пользователя и выполнять их, обрабатывая исключительные ситуации, согласно методам, описанным в [3]. База данных содержится в отдельном файле и представляет собой класс, описывающий поля, содержащие, коллекции созданных объектов. Первоначально доступны лишь функции работы с файлами базы данных, а именно открытие файла или его создание по следующим алгоритмам (блок-схемы представлены в приложении А, см. рис. А.1, А.2), если уже идёт работа с файлом, то его можно сохранить, либо отменить последнее изменение (блок-схемы представлены в приложении А, см. рис. А.3, А.4).

Алгоритм открытия файла:

1. Запросить полный адрес к файлу.
2. Если файловый поток открыт, то закрыть его, иначе продолжить.
3. Создать новый файловый поток с параметром чтения.
4. Записать данные из файла во внутренний объект базы данных.
5. Закрыть файловый поток.

Алгоритм создания файла:

1. Запросить полный адрес к новому файлу.
2. Создать файловый поток с параметром записи.
3. Создать пустой внутренний объект базы данных.
4. Записать данные из внутреннего объекта базы данных в файл.
5. Закрыть файловый поток.

Алгоритм сохранения файла:

1. Запросить полный адрес файла, в который происходит сохранение.
2. Если адрес не совпадает с адресом исходного файла, то изменить адрес на новый и создать файл с новым адресом, иначе продолжить.
3. Создать файловый поток с параметром чтения и записи.
4. Записать данные из внутреннего объекта базы данных в файл.
5. Закрыть файловый поток.

Алгоритм отмены последнего изменения:

1. Если переключатель изменения не в положении «изменено», то прервать выполнение алгоритма, иначе продолжить.
2. Записать данные из буфера во внутренний объект базы данных.
3. Очистить буфер.
4. Перевести переключатель изменения в положение «не изменено».

Основное взаимодействие программы и пользователя будет производиться на основе отправленных пользователем команд: добавление объекта, изменение объекта, удаление объекта, анализ имеющихся данных (блок-схемы представлены в приложении В, см. рис. В.1-В.4).

Алгоритм добавления нового объекта в базу данных:

1. Сохранить в буфер исходное состояние базы данных.
2. Запросить у пользователя параметры создаваемого объекта.
3. Если объект с заданными параметрами уже существует в базе данных, то вывести ошибку и прервать выполнение алгоритма, иначе продолжить.
4. Создать новый экземпляр объекта с заданными параметрами.
5. Записать новый объект в базу данных.
6. Зафиксировать изменения в базе данных.
7. Перевести переключатель изменения в положение «изменено».

Алгоритм изменения объекта в базе данных:

1. Сохранить в буфер исходное состояние базы данных.
2. Запросить у пользователя параметры изменяемого объекта.

3. Если объекта с заданными параметрами не существует в базе данных, то вывести ошибку и прервать выполнение алгоритма, иначе продолжить.
4. Запросить у пользователя новые параметры объекта.
5. Если объект с новыми параметрами уже существует в базе данных, то вывести ошибку и прервать выполнение алгоритма, иначе продолжить.
6. Создать новый объект с заданными новыми параметрами.
7. Записать новый экземпляр объект на место исходного в базе данных.
8. Зафиксировать изменения в базе данных.
9. Перевести переключатель изменения в положение «изменено».

Алгоритм удаления объекта из базы данных:

1. Сохранить в буфер исходное состояние базы данных.
2. Запросить у пользователя параметры удаляемого объекта.
3. Если объекта с заданными параметрами не существует в базе данных, то вывести ошибку о прервать выполнение алгоритма, иначе продолжить.
4. Удалить из базы данных зависимости, связанные с удаляемым объектом.
5. Удалить объект из базы данных.
6. Зафиксировать изменения в базе данных.
7. Перевести переключатель изменения в положение «изменено».

Алгоритм анализа имеющихся данных:

1. Получить из базы данных все объекты.
2. Сопоставить объекты и сгруппировать их по одинаковым признакам.
3. Провести аналитические вычисления, необходимые для составления диаграммы.
4. Вывести составленную диаграмму пользователю.
5. Удалить из памяти временные данные.

Чтобы облегчить и ускорить обучение пользователя взаимодействию с программой, будет реализована встроенная справка, включающая в себя полный обзор основных функций программы и наглядную демонстрацию работы с ними. Вызываться справка будет из

главного меню и состоять из выполнения действий по заранее разработанному алгоритму (блок-схема представлена в приложении С, см. рис С.1).

## **2.2. Разработка интерфейса**

Графически интерфейс программы будет реализован с помощью API Windows Forms, входящего в состав Microsoft .NET Framework для языка программирования C#. В ходе программы будет использоваться три разные вида форм: главная форма, форма работы с объектами, справочная форма.

Меню в главной форме (см. рис. 2.1) находится на верхней панели и представляет собой ниспадающий список действий, логически сгруппированных по схожести действий и объектам, на которые они направлены. В центральную область формы выводится информация из базы данных, разделённая по разным вкладкам в зависимости от объектов, из каждой вкладки можно перейти в любую другую.

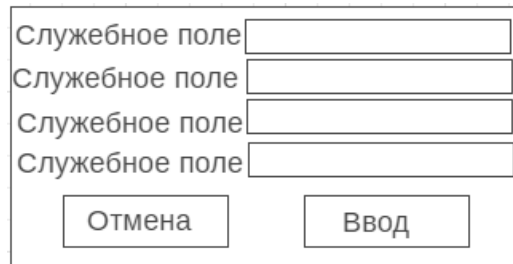
Для удобства пользователя, при запуске программы будет выводиться сообщение, предлагающее открыть файл, создать файл или пройти краткий обзор функций программы и просмотреть наглядную демонстрацию работы с программой.



***Рисунок 2.1. Окно главной формы***



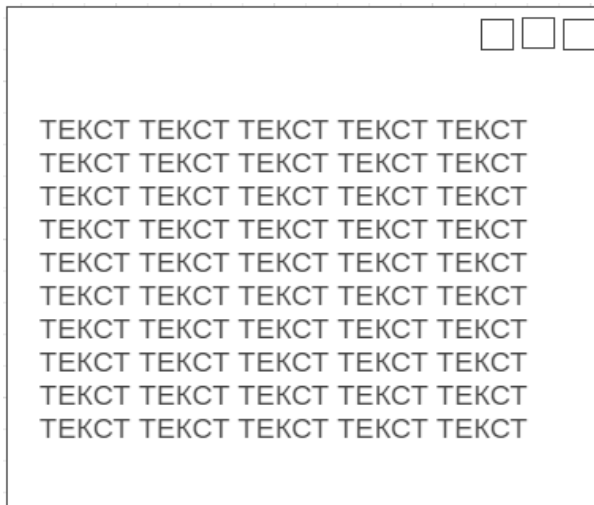
Для организации работы с объектами будет реализована форма (см. рис. 2.2), имеющая служебные поясняющие подписи, поля для введения пользователем параметров объекта, а также возможность подтвердить выполнение действия или его отменить.



The form consists of a rectangular frame. Inside, there are four horizontal input fields stacked vertically. To the left of each input field is the text 'Служебное поле' (Service field). Below the input fields, there are two buttons: 'Отмена' (Cancel) on the left and 'Ввод' (Input) on the right.

**Рисунок 2.2. Форма работы с объектом**

Служебная форма (см. рис. 2.3) будет иметь вид простого окна, наполненного служебной информацией без дополнительных функций.



The form is a window with a standard title bar (minimize, maximize, close buttons). The main area of the window is filled with ten lines of placeholder text, each line containing the word 'ТЕКСТ' repeated five times. The text is centered within the window.

**Рисунок 2.3. Служебная форма**

## **2.3. Выводы по главе**

В данной главе были определены основные алгоритмы, которые будут использоваться в программе для обработки информации, взаимодействия с файлами и пользователем. Описан и спроектирован графический интерфейс, выбрана система и средства его разработки, составлен макет основных окон. В следующей главе будет описана разработка и отладка продукта, согласно проектированию, составление тестовых сценариев, согласно полноте тестирования по критериям черного ящика, описанным в [4].

### **3. Реализация и тестирование приложения**

В данной главе выполнена разработка и тестирование продукта, приведено подробное описание классов, интерфейсов, используемых методов и реализованных функций, рассмотрены исключительные ситуации и методы их обработки. Результатом является программа, реализованная на языке программирования C# и комплект тестов.

#### **3.1. Описание классов, их полей, методов, функций**

В этом разделе описываются основные классы, реализованные для приложения, их поля, документируются методы и функции, разработанные по алгоритмам, приведенным в разделе 2.1, оптимизированным в соответствии с рекомендациями, изложенными в [5].

##### **3.1.1. Класс PersonalAccount**

Конечный, внутренний, сериализуемый класс (исходный код находится на внешнем носителе в приложении D, папка Wallet, файл PersonalAccount.cs), описывающий встроенную базу данных. Имеет защищённые, согласно инкапсуляции – одному из основных понятий ООП [6], поля, содержащие коллекции трёх основных видов объектов: объект счёт, на котором хранятся средства пользователя, объект категория, который определяет вид дохода или расхода, объект транзакция, описывающий параметры доходной или расходной операции. Класс имеет единственный конструктор без входных данных, который создаёт пустые коллекции, готовые для записи информации, реализует встроенный интерфейс ICloneable для того, чтобы в буфере хранилась не ссылка на предыдущее состояние, а полная копия, а в случае отказа она может быть восстановлена.

##### **3.1.2. Класс Account**

Конечный, внутренний, сериализуемый класс (исходный код находится на внешнем носителе в приложении D, папка Wallet, файл Account.cs), описывающий счёт. Имеет два защищенных поля, одно из которых содержит название счёта, а второе состояние баланса. Для доступа к полю баланса реализовано два свойства: первое возвращает и задаёт сумму на балансе объектом класса Money, описанного в 3.1.3, второе возвращает сумму на балансе в виде строки.

### **3.1.3. Класс Money**

Конечный, внутренний, сериализуемый класс (исходный код находится на внешнем носителе в приложении D, папка Service, файл Money.cs), представляющий деньги. Имеет два защищенных поля, которые описывают сумму денег, разделённую на часть до запятой и часть после запятой для более удобной работы с бинарными операциями и сравнениями, пустой конструктор и конструктор с параметрами. Класс реализует интерфейс ICorrect (исходный код находится на внешнем носителе в приложении D, папка Service, файл ICorrect.cs) и содержит служебные методы: CheckDecimal – рекурсивный метод, последовательно выделяющий из целые из дробной, ToDecimal – переводит целую часть в дробную, IsValid – выполняет проверку входящей строки на корректность, IsOnlyDigits – проверяет, содержит ли входная строка только цифры. Так же, для класса были переопределены операции бинарного сложения и вычитания.

### **3.1.4. Класс Category**

Конечный, внутренний, сериализуемый класс (исходный код находится на внешнем носителе в приложении D, папка Finances, файл Category.cs), представляющий категории доходов и расходов, имеет следующие защищённые поля: название категории, суммарный доход по категории, суммарный расход по категории.

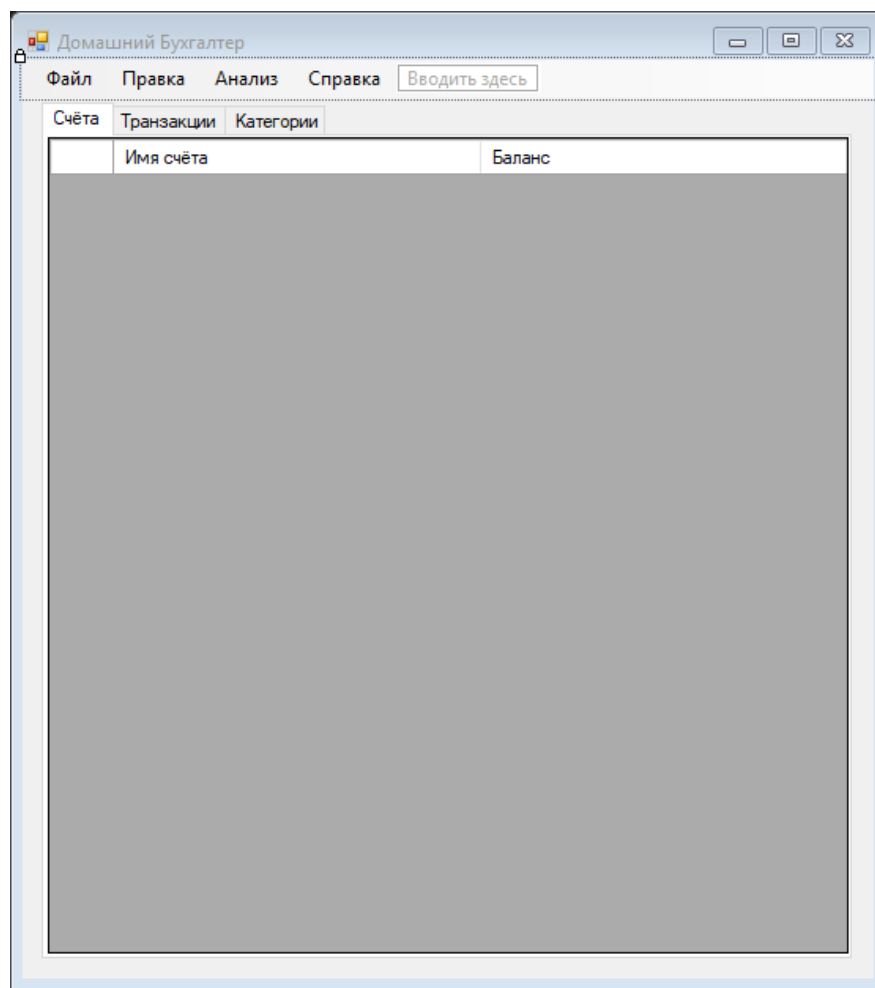
### **3.1.5. Класс Transaction**

Конечный, внутренний, сериализуемый класс (исходный код находится на внешнем носителе в приложении D, папка Finances, файл Transaction.cs), описывающий доходы и расходы, имеет следующие поля: сумма, категория, к которой относится, статья, дата, место списания или зачисления суммы, определение относится транзакция к расходам или к доходам.

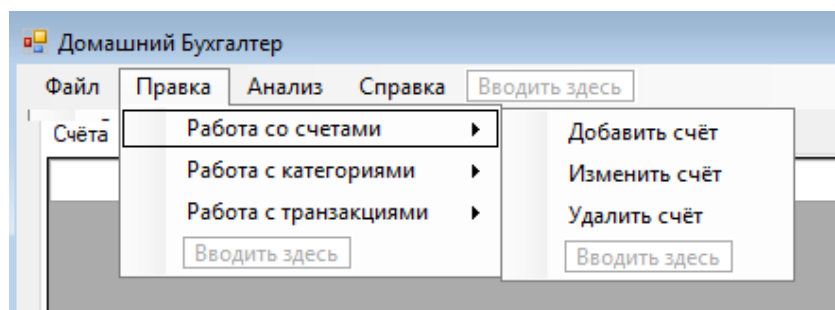
## **3.2. Реализация пользовательского интерфейса**

Для взаимодействия с пользователем были определены три типа форм: главная форма, форма работы с объектами, справочная форма. Интерфейс реализован с помощью API Windows Forms, входящего в состав .NET Framework, при разработке использовали макеты, представленные в разделе 2.2.

Главная форма (см. рис. 3.1) содержит основное меню – верхняя панель, на которой представлены и сгруппированы по категориям все действия (см. рис. 3.2), доступные пользователю. Для вывода информации из базы данных используется встроенный класс DataGridView, использовавшийся согласно документации, представленной в [7], а для экономии места используется несколько вкладок.

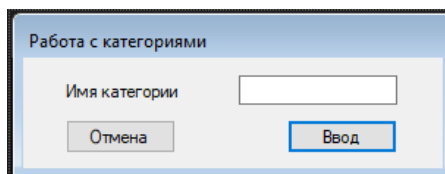


**Рисунок 3.1. Главное окно программы**

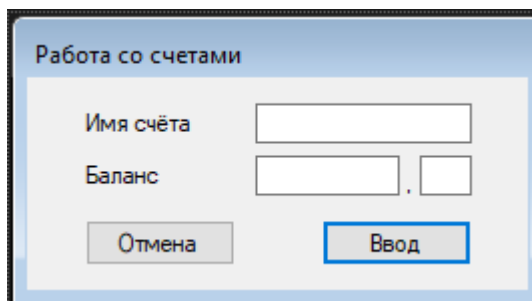


**Рисунок 3.2. Сгруппированные действия в главном меню**

Для взаимодействия с объектами, используется более компактная форма (см. рис. 3.3, 3.4), открывающаяся поверх главной и блокирующая её работу до конца выполнения действия.



*Рисунок 3.3. Пример окна для работы с объектами №1*



*Рисунок 3.4 Пример окна для работы с объектами №2*

### **3.3. Анализ и тестирование разработанного приложения**

Тестирование производилось на персональном компьютере с операционной системой Windows 10 Pro (1709).

В ходе тестирования функций работы с файлами были обнаружены следующие исключительные ситуации, приводящие к аварийному завершению программы: открытие занятого файла, сохранение в занятый файл, сохранение пустой базы данных, открытие пустой базы данных, открытие файлов, которые не предусмотрены для работы с программой. Для каждой ситуации было обнаружено место возникновения и вставлен обработчик.

Были обнаружены проблемы с корректным отображением данных в таблицах DataGridView и обновлением данных в ходе использования программы.

Так же были обнаружены ошибки, связанные с отображением сравнительной диаграммы и прикреплением подписей.

Все найденные ошибки были исправлены, а во все места, где они встречались, встроены обработчики.

Полученный продукт был проверен по критериям, разработанным в пункте 1.3. Так, интерфейс программы не требует специальных настроек, логическая часть реализует минимальный полезный набор функций таких, как работа с файлами, работа с объектами, анализ данных; весь продукт написан на русском языке, собственная, внутренняя логика организации данных, реализована работа с доходами, расходами, категориями, предусмотрена обработки исключительных ситуаций. Таким образом, все критерии, которые были выдвинуты в пункте 1.3, выполнены.

По итогу тестирования была составлена полная таблица тестов (приложение Е, таблица Е.1).

### **3.4. Выводы по главе**

В данной главе реализованы основные алгоритмы, представленные и описанные с помощью блок-схем в пункте 2.1, на языке программирования С#, описаны все основные классы и интерфейсы, которые реализованы в программе, описана разработка интерфейса, выделены основные моменты. После реализации было проведено тестирование и отладка, составлена таблица тестов, по которой было проведено повторное тестирование, завершившееся успешно.

## **Заключение**

По итогу работы было спроектировано и реализовано приложение на языке программирования С# для платформы .NET.

Для достижения поставленной цели были выполнены следующие задачи: проанализирована предметная область, в ходе обзора и анализа аналогов были выделены те части функционала приложения, на которые нужно обратить внимание, сформированы требования к конечному продукту, разработаны основные алгоритмы, смоделирован интерфейс, описаны основные классы программы и реализован интерфейс по созданному макету, проведены тестирование и отладка программы.

Так как программа имеет открытый исходный код и содержит в себе минимальный полезный функционал, она может быть использована в будущем, как фундамент для разработки более крупного проекта.

В ходе работы были получены и развиты навыки объектно-ориентированного программирования, работы в среде разработки Microsoft Visual Studio, работа с системой контроля версий GitHub; получен опыт анализа и оценки работы и поведения других программ. Были приобретены знания в области контроля финансов, практические навыки тестирования, разработки тестов средствами Microsoft Visual Studio и создании пользовательской документации (приложение F).

## **Список сокращений и условных обозначений**

ООП – объектно-ориентированное программирование.

GTK – GIMP (GNU Image Manipulation Program) ToolKit. Кроссплатформенная библиотека элементов интерфейса.

API (рус. интерфейс прикладного программирования) – application programming interface.

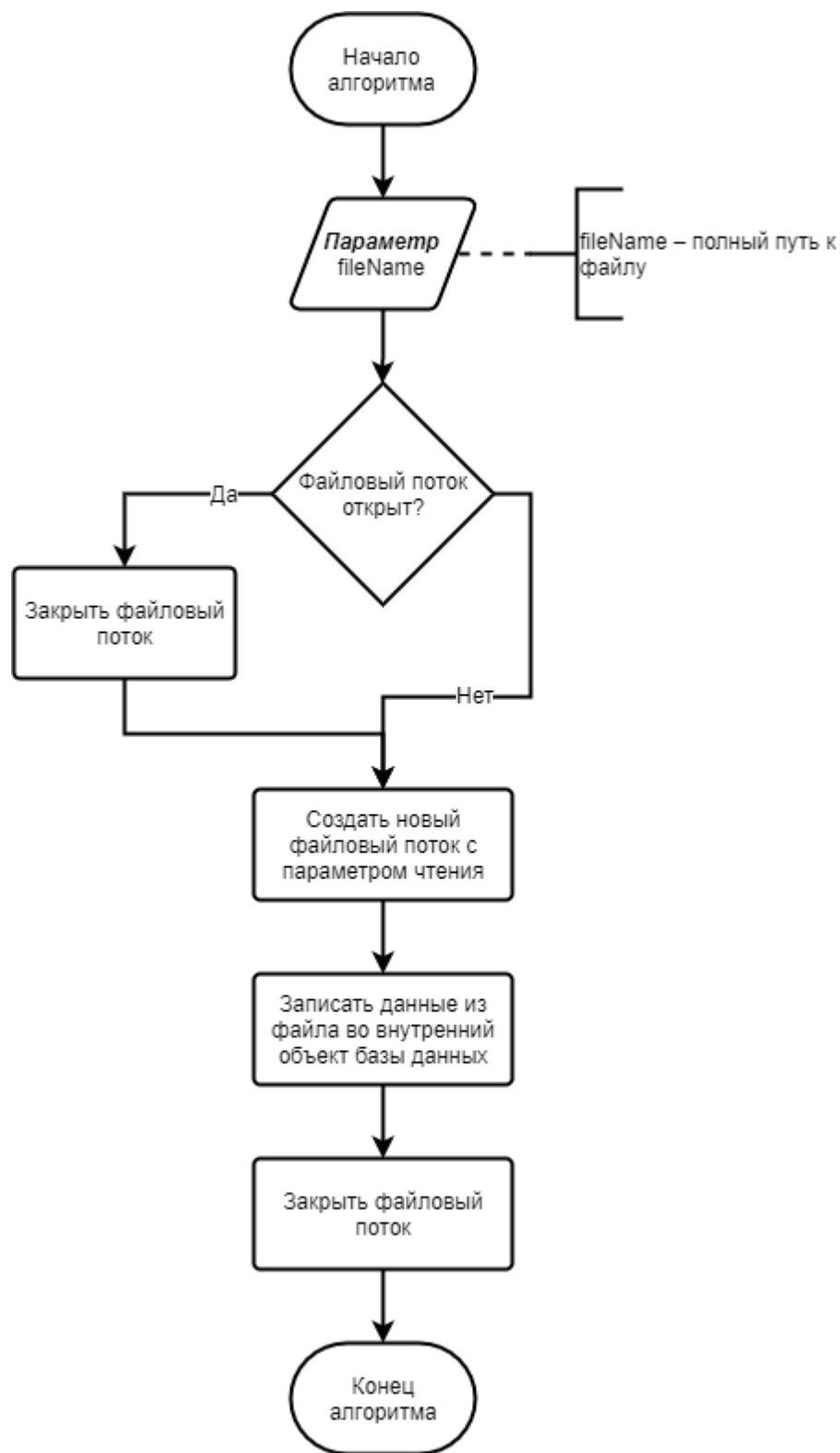
.NET – .NET Framework, программная платформа, основанная на общезыковой системе среде исполнения.



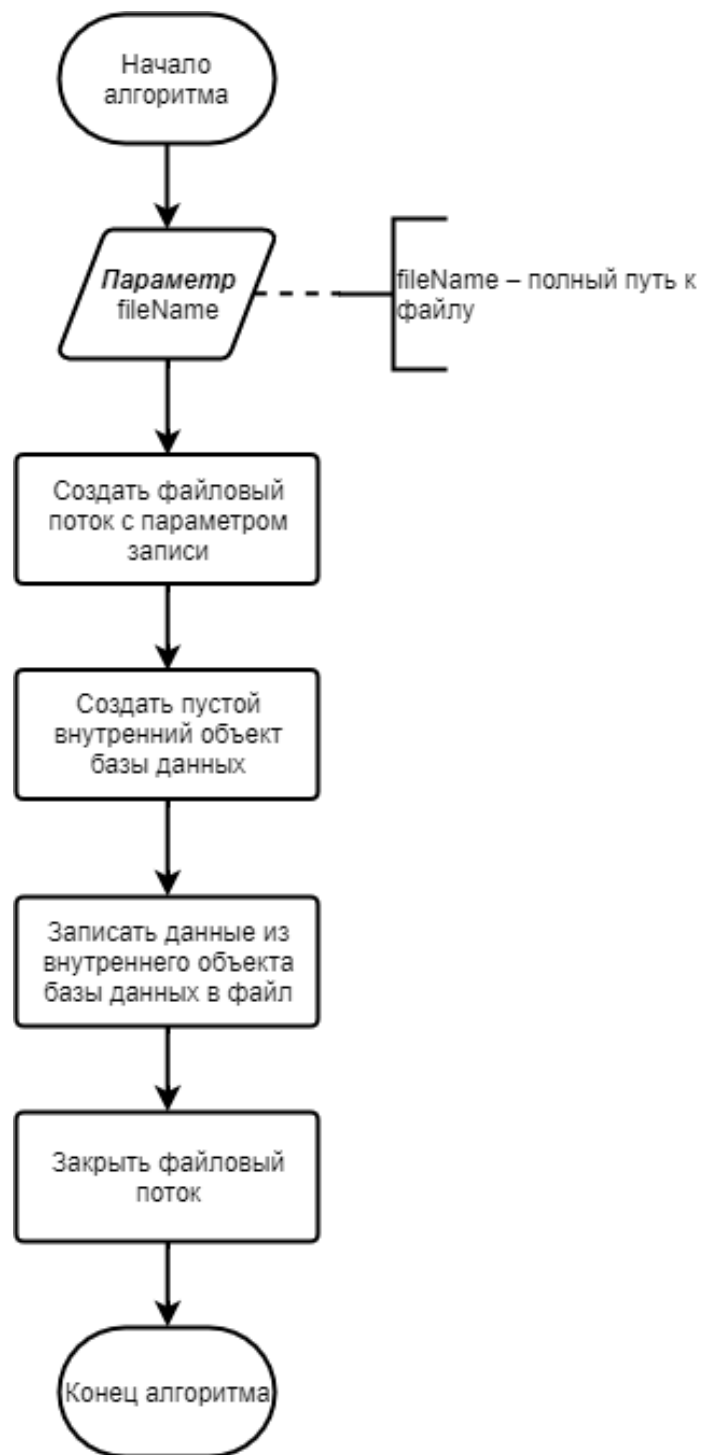
## **Библиографический список**

1. Терехов А. Н. Технология программирования. - М.: Интернет-Ун-т Информ. Технологий, 2006.
2. Технология программирования / Ю. Ю Громов, О. Г. Иванова, М. П. Беляев, Ю. В. Минин, - Тамбов: ФГБОУ ВПО «ТГТУ», 2013.
3. Подбельский В. В. Язык С# Базовый курс. - М.: Финансы и статистика, 2011.
4. Плаксин М. А. Тестирование и отладка программ для профессионалов будущих и настоящих. - 2-е изд. - М.: БИНОМ. Лаборатория знаний, 2013.
5. Правила по оптимизации кода (для начинающих) [Электронный ресурс] // Хабрахабр URL: <https://habrahabr.ru/sandbox/97393/> (дата обращения: 09.03.2018).
6. Почему ООП не инкапсуляция, наследование и полиморфизм, или как я научился не волноваться и полюбил разметку [Электронный ресурс] // Хабрахабр URL: <https://habrahabr.ru/post/307040/> (дата обращения: 12.02.2018).
7. Класс DataGridView (System.Windows.Forms) // Microsoft Developer Network [Электронный ресурс] URL: [https://msdn.microsoft.com/ru-ru/library/system.windows.forms.datagridview\(v=vs.110\).aspx](https://msdn.microsoft.com/ru-ru/library/system.windows.forms.datagridview(v=vs.110).aspx) (дата обращения: 11.03.2018).

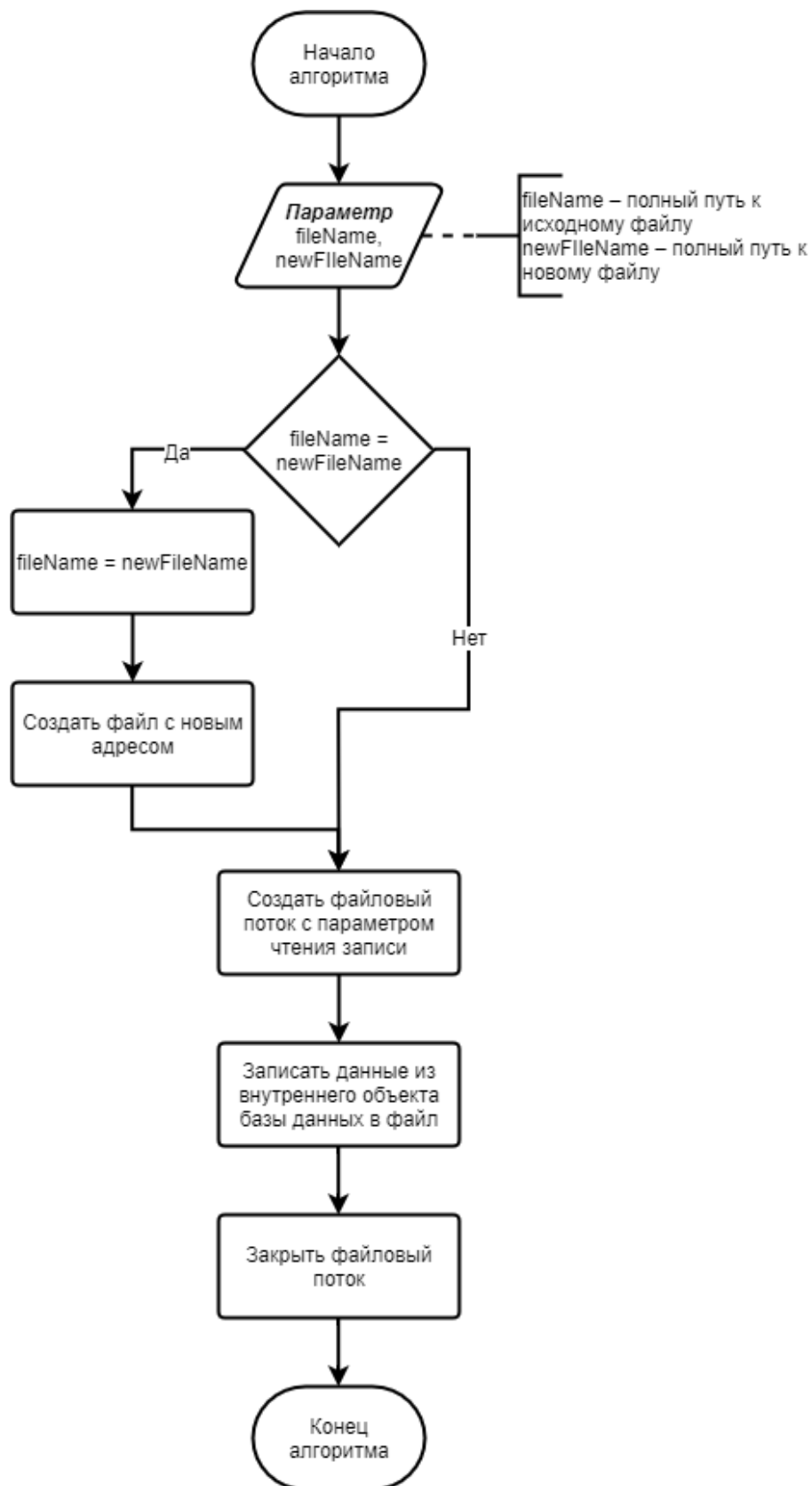
## Приложение А. Блок-схемы алгоритмов работы с файлами



*Рисунок А.1. Открытие файла*



***Рисунок А.2. Создание файла***

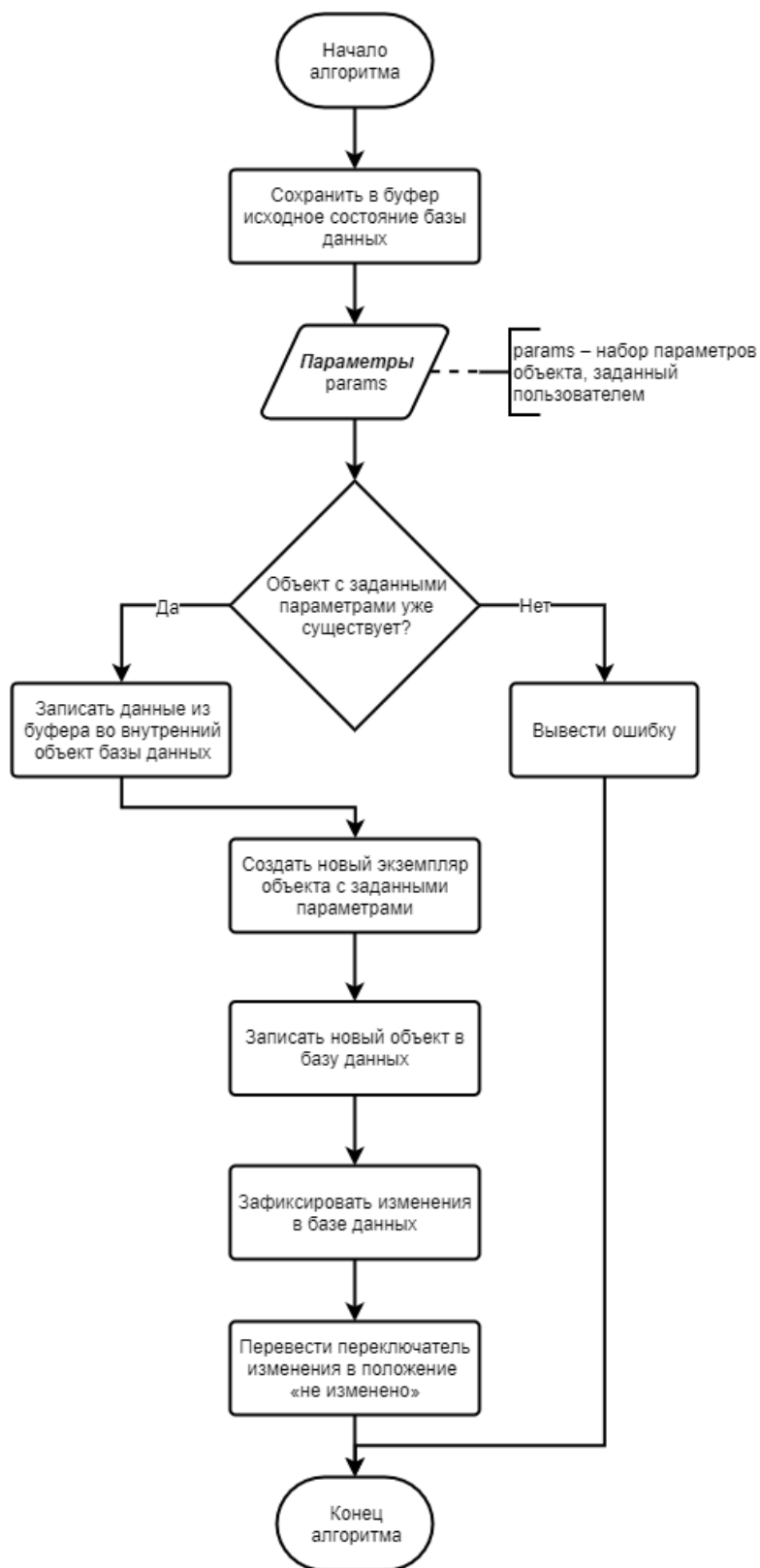


**Рисунок А.3. Сохранение файла**

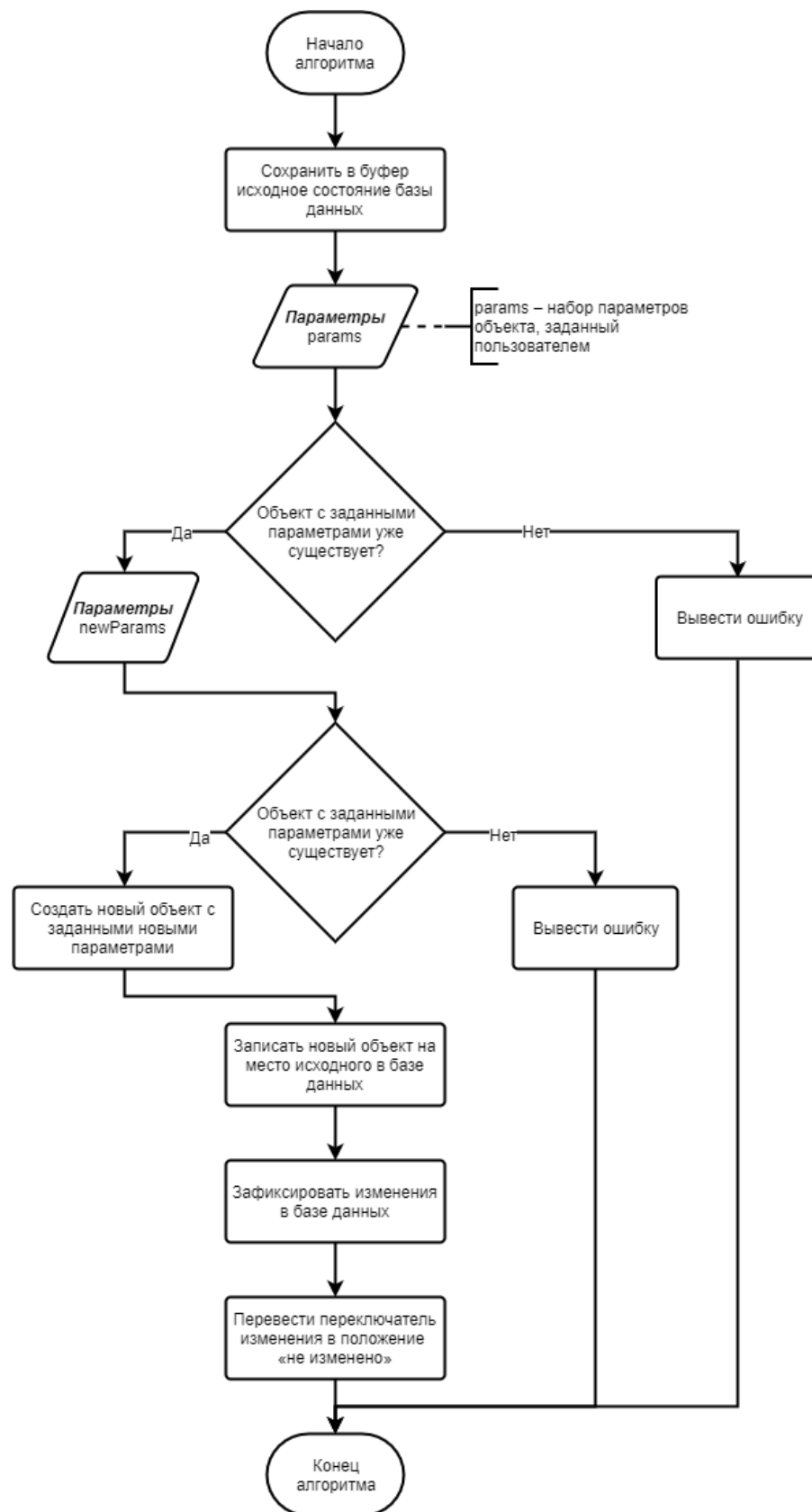


*Рисунок А.4. Отмена последнего изменения*

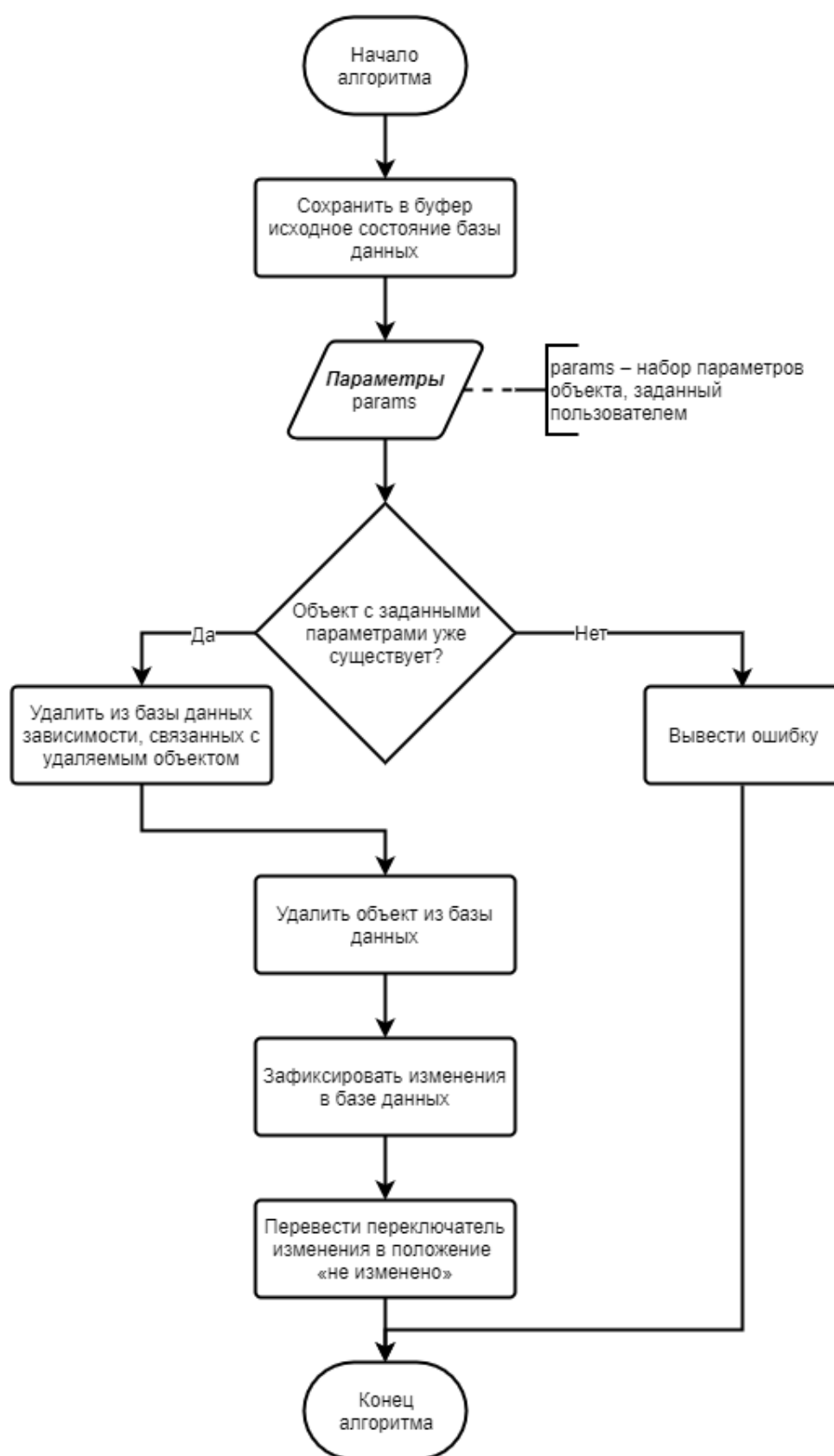
## Приложение В. Блок-схемы алгоритмов работы с объектами



*Рисунок В.1. Добавление нового объекта в базу данных*



**Рисунок В.2. Изменение объекта в базе данных**



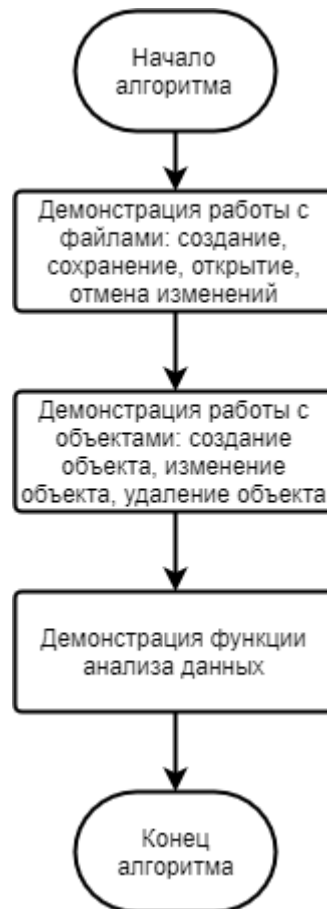
**Рисунок В.3. Удаление объекта из базы данных**





*Рисунок В.4. Анализ имеющихся данных*

## Приложение С. Блок-схема алгоритма встроенной справки



*Рисунок С.1. Алгоритм встроенной справки*

## **Приложение D. Исходные коды программы**

## Приложение Е. Тестирование

*Таблица Е.1. Комплект тестов*

№	Начальное состояние программы	Действие пользователя	Отклик программы
1	Главная форма, меню «Файл»	Клик по кнопке «Открыть файл»	Вызов диалога выбора файла
2	Главная форма, меню «Файл»	Клик по кнопке «Создать файл»	Вызов диалога выбора файла
3	Главная форма, меню «Файл», в программе не открыт файл	Клик по кнопке «Сохранить»	Вывод ошибки о том, что нет файла для сохранения
4	Главная форма, меню «Файл», в программе открыт файл	Клик по кнопке «Сохранить»	Вывод сообщения о том, что файл сохранён
5	Главная форма, меню «Файл», в программе не открыт файл	Клик по кнопке «Сохранить как...»	Вывод диалога выбора файла, после вывод ошибки о том, что нет файла для сохранения
6	Главная форма, меню «Файл», в программе не открыт файл	Клик по кнопке «Отменить»	Вывод ошибки о том, что нет действий для отмены
7	Главная форма, меню «Файл», в программе не открыт файл, нет действий для отмены	Клик по кнопке «Отменить»	Вывод ошибки о том, что нет действий для отмены
8	Главная форма, меню «Файл», в программе открыт файл, присутствует действие для его отмены	Клик по кнопке «Отменить»	Вывод сообщения о том, что последнее действие отменено
9	Главная форма, меню «Файл», все изменения зафиксированы	Клик по кнопке «Выход»	Завершение работы программы
10	Главная форма, меню «Файл», изменения не зафиксированы	Клик по кнопке «Выход»	Вывод предложения зафиксировать изменения в открытом файле, после завершение программы
11	Главная форма	Клик по вкладке «Счета»	Открыта вкладка «Счета»
12	Главная форма	Клик по вкладке «Транзакции»	Открыта вкладка «Транзакции»
13	Главная форма	Клик по вкладке «Категории»	Открыта вкладка «Категории»

№	Начальное состояние программы	Действие пользователя	Отклик программы
14	Главная форма, меню «Правка», работа со счетами	Клик по кнопке «Добавить счёт»	Вывод формы работы со счетами и передача управления ей
15	Главная форма, меню «Правка», работа со счетами	Клик по кнопке «Изменить счёт»	Вывод формы работы со счетами и передача управления ей
16	Главная форма, меню «Правка», работа со счетами	Клик по кнопке «Удалить счёт»	Вывод формы работы со счетами и передача управления ей
17	Главная форма, меню «Правка», работа с категориями	Клик по кнопке «Добавить категорию»	Вывод формы работы с категориями и передача управления ей
18	Главная форма, меню «Правка», работа с категориями	Клик по кнопке «Изменить категорию»	Вывод формы работы с категориями и передача управления ей
19	Главная форма, меню «Правка», работа с категориями	Клик по кнопке «Удалить категорию»	Вывод формы работы с категориями и передача управления ей
20	Главная форма, меню «Правка», работа с транзакциями	Клик по кнопке «Добавить транзакцию»	Вывод формы работы с транзакциями и передача управления ей
21	Главная форма, меню «Правка», работа с транзакциями	Клик по кнопке «Изменить транзакцию»	Вывод формы работы с транзакциями и передача управления ей
22	Главная форма, меню «Правка», работа с транзакциями	Клик по кнопке «Удалить транзакцию»	Вывод формы работы с транзакциями и передача управления ей
23	Главная форма	Клик по кнопке «Анализ»	Вывод справочной формы с построенной диаграммой доходов и расходов
24	Главная форма, меню «Справка»	Клик по кнопке «Помощь»	Запуск сценария знакомства пользователя с программой

№	Начальное состояние программы	Действие пользователя	Отклик программы
25	Главная форма, меню «Справка»	Клик по кнопке «О программе»	Вывод окна с краткой информацией о программе
26	Форма добавления объекта	Введена корректная информация	Данные приняты, объект создан, таблицы обновлены, форма закрыта, управление передано главной форме
27	Форма добавления объекта	Введена информация о уже существующем объекте	Вывод ошибки, что файл введёнными данными уже существует, заново вызвана форма добавления объекта
28	Форма добавления объекта	Введена некорректная информация	Вывод ошибки о некорректности введённых данных, заново вызвана форма добавления объекта
29	Форма изменения объекта	Выбран существующий объект	Вызов формы добавления объекта, передача управления ей
30	Форма изменения объекта	Выбран несуществующий объект	Вывод ошибки о том, что такой объект не существует, заново вызвана форма изменения объекта
31	Форма удаления объекта	Введена информация о существующем объекте	Вывод предупреждения о последствиях удаления объекта, после удаление
32	Форма удаления объекта	Введена информация о несуществующем объекте	Вывод ошибки о том, что такой объект не существует, заново вызвана форма удаления объекта
33	Главная форма, в программе существует информация о счетах, категориях и транзакциях	Клик по кнопке «Анализ»	Вывод окна с построенной круговой диаграммой о расходах и доходах по категориям

№	Начальное состояние программы	Действие пользователя	Отклик программы
34	Главная форма, в программе не существует информация о счетах, или категориях, или транзакциях	Клик по кнопке «Анализ»	Вывод ошибки о том, что недостаточно данных для анализа
35	Главная форма, все изменения сохранены	Клик по стандартной кнопке «Закрыть программу» в правом верхнем углу формы	Закрытие программы
36	Главная форма, есть несохраненные изменения	Клик по стандартной кнопке «Закрыть программу» в правом верхнем углу формы	Предложения сохранить изменения, после закрытие программы

# **Приложение F. Руководство пользователя**

## **ПРИЛОЖЕНИЕ «ДОМАШНИЙ БУХГАЛТЕР»**

### **Руководство пользователя**

6 листов

Пермь, 2018 год



## **Г.1. Введение**

Программа «Домашний бухгалтер» – приложение для учёта и анализа данных о доходах и расходах, может быть использована в бытовой сфере, где предусматривается недокументируемый расход или приход средств.

Программа «Домашний Бухгалтер» работает с неограниченным количеством счетов, транзакций, категорий, имеет функцию анализа.

Для взаимодействия с программой необходимо иметь базовые минимальные навыки работы с персональным компьютером, понимать термин «файл».

Перед работой с программой пользователю необходимо ознакомиться с этим руководством.

## **Г.2. Назначение и условия использования**

Программа «Домашний Бухгалтер» предназначена для автоматизации учёта доходов и расходов, анализа и сохранения этих данных. При ручном выполнении данных действий возможны арифметические ошибки, а при обработке больших объёмов информации занимает продолжительное время.

Для работы программы нужен персональный компьютер, работающий под операционной системой Windows 7 и выше, на котором установлен программный пакет «.NET Framework» версии 4.6.1 или выше. Для успешного запуска и работы программы компьютеру необходимо обладать минимальными характеристиками, приведёнными ниже:

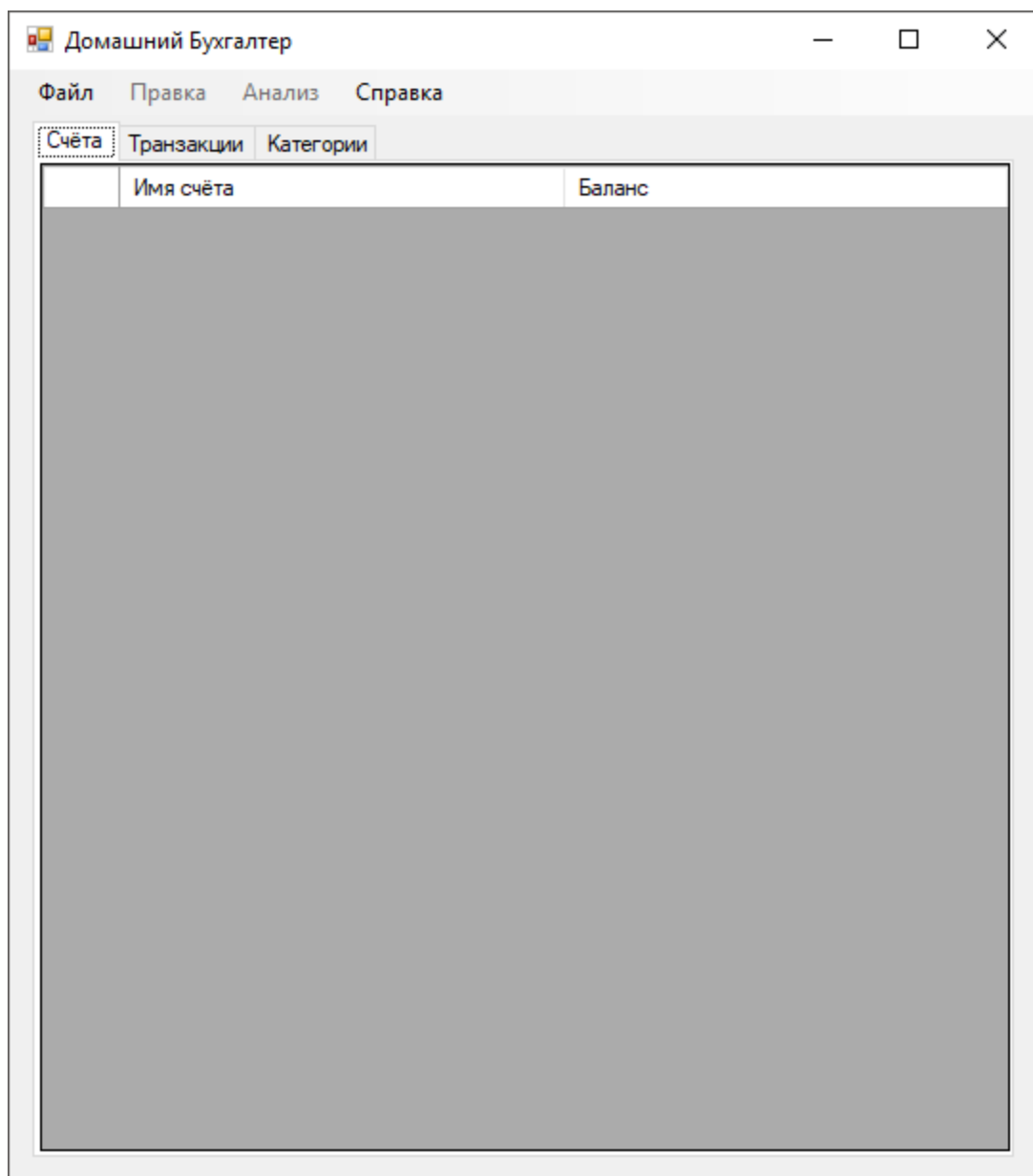
1. Процессор с тактовой частотой 1 ГГц и более.
2. ОЗУ 1 Гб и более.
3. Жесткий диск объемом не менее 1 Гб.
4. Манипулятор типа «мышь».
5. Физическую клавиатуру.

Входная информация для программы – это совокупность заданных пользователем параметров объектов.

Для работы с программой пользователю не требуется дополнительных знаний, кроме базовых навыков работы с компьютером.

### Ф.3. Подготовка к работе

Дистрибутив программы содержит вспомогательный файлы, исполняемый файл «Home Bookkeeper.exe», данное руководство пользователя. Для запуска программы нужно открыть исполняемый файл. Запуск программы и отображение главного окна (см. рис. F.1) означает, что всё готово к работе и конфигурация системы удовлетворяет минимальным требованиям.



*Рисунок F.1. Главное окно программы*

## **Ф.4 Описание операций**

Раздел содержит развернутое описание всех возможностей программы. Операции в программе могут быть выполнены двумя различными способами: через главное меню, либо с помощью нажатия комбинации клавиш на клавиатуре.

### **Ф.4.1. Создание нового файла**

Функция создания нового файла позволяет создать новую базу данных для последующей работы с ней. Функция доступна пользователю в любом состоянии программы и может быть вызвана следующими способами:

1. Подачей команды «Файл → Создать».
2. Комбинацией клавиш на клавиатуре «Ctrl+N».

После вызова функции будет открыто диалоговое окно создания файла, в котором пользователю нужно указать место хранения файла, его имя и нажать кнопку «Сохранить».

### **Ф.4.2. Открытие существующего файла**

Функция открытия существующего файла позволяет открыть уже созданную базу данных для последующей работы с ней. Функция доступна пользователю в любом состоянии программы и может быть вызвана следующими способами:

1. Подачей команды «Файл → Открыть».
2. Комбинацией клавиш на клавиатуре «Ctrl+O».

После вызова функции будет открыто диалоговое окно открытия файла, в котором пользователю нужно указать место хранения файла, его имя и нажать кнопку «Сохранить».

### **Ф.4.3. Сохранение существующего файла в заданное место**

Функция сохранения существующего файла в заданное место позволяет сохранить уже созданную базу данных для последующей работы с ней. Функция доступна пользователю в любом состоянии программы и может быть вызвана следующими способами:

1. Подачей команды «Файл → Сохранить как...».
2. Комбинацией клавиш на клавиатуре «Ctrl+Shift+S».

После вызова функции будет открыто диалоговое окно сохранения файла, в котором пользователю нужно указать место хранения файла, его имя и нажать кнопку «Сохранить».

#### **Ф.4.4. Сохранение существующего файла**

Функция сохранения существующего файла позволяет сохранить уже созданную базу данных для последующей работы с ней. Функция доступна пользователю в любом состоянии программы и может быть вызвана следующими способами:

1. Подачей команды «Файл → Сохранить».
2. Комбинацией клавиш на клавиатуре «Ctrl+S».

После вызова функции пользователю будет сообщено, что операция выполнена успешно.

#### **Ф.4.5. Отмена последнего действия**

Функция отмены последнего действия позволяет отменить последнее действие с базой данных и вернуть её в предыдущее состояние вне зависимости от внесённого изменения, кроме отмены предыдущего. Функция доступна пользователю в любом состоянии программы и может быть вызвана следующими способами:

1. Подачей команды «Файл → Отменить».
2. Комбинацией клавиш на клавиатуре «Ctrl+Z».

После вызова функции пользователю будет сообщено, что операция выполнена успешно.

#### **Ф.4.6. Добавления объекта в базу данных**

Функция добавления объекта в базу данных позволяет пользователю добавить новый счёт, категорию или транзакцию. Функция доступна пользователю, когда в программу загружена база данных из файла или создана новая, и может быть вызвана следующими способами:

1. Подачей команды «Правка → Работа с счетами/категориями/транзакциями → Добавление счёта/категории/транзакции».

2. Комбинацией клавиш на клавиатуре «Ctrl+A» при открытой вкладке, соответствующей создаваемому объекту.

После вызова функции пользователю будет выведено окно с полями для заполнения, в которые он должен ввести данные.

#### **Г.4.7. Изменение объекта в базе данных**

Функция изменения объекта в базе данных позволяет пользователю изменить существующий счёт, категорию или транзакцию. Функция доступна пользователю, когда в программу загружена база данных из файла или создана новая, и может быть вызвана следующими способами:

1. Подачей команды «Правка → Работа с счетами/категориями/транзакциями → Изменение счёта/категории/транзакции».
2. Комбинацией клавиш на клавиатуре «Ctrl+R» при открытой вкладке, соответствующей создаваемому объекту.

После вызова функции пользователю будет выведено окно с полями для заполнения, в которые он должен ввести данные.

#### **Г.4.8. Удаление объекта из базы данных**

Функция удаления объекта из базы данных позволяет пользователю удалить существующий счёт, категорию или транзакцию. Функция доступна пользователю, когда в программу загружена база данных из файла или создана новая, и может быть вызвана следующими способами:

1. Подачей команды «Правка → Работа с счетами/категориями/транзакциями → Удаление счёта/категории/транзакции».
2. Комбинацией клавиш на клавиатуре «Ctrl+D» при открытой вкладке, соответствующей создаваемому объекту.

После вызова функции пользователю будет выведено сообщение с предупреждением о последствиях удаления объектов, а после окно с полями для заполнения, в которые он должен ввести данные.

#### **Ф.4.9. Анализ данных**

Функция анализа данных позволяет пользователю составить круговую диаграмму расходов и доходов, сгруппированных по категориям. Функция доступна пользователю в любом состоянии программы и может быть вызвана подачей команды «Анализ». После вызова функции пользователю будет выведено окно сформированной диаграммой.

#### **Ф.4.10. Вызов справки**

Функция вызова справки доступна пользователю в любом состоянии программы и может быть вызвана подачей команды «Справка → Помощь». После вызова функции пользователю будет на примерах показана работа с программой и выведено окно со всеми горячими клавишами.

#### **Ф.4.11. Вызов окна «О программе»**

Функция вызова окна «О программе» доступна пользователю в любом состоянии программы и может быть вызвана подачей команды «Справка → О программе».

### **Ф.5. Аварийные ситуации**

Во время эксплуатации программы могут возникать исключительные ситуации, для которых не предусмотрена обработка встроенными средствами. В таком случае необходимо перезапустить программу и выполнить заново требуемые действия.