Министерство науки и высшего образования Российской Федерации Федеральное государственное бюджетное образовательное учреждение высшего образования

ТОМСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ СИСТЕМ УПРАВЛЕНИЯ И РАДИОЭЛЕКТРОНИКИ (ТУСУР)

Кафедра комплексной информационной безопасности электронно-вычислительных систем (КИБЭВС)

## КАЛЬКУЛЯТОР РАСХОДОВ

Курсовая работа по дисциплине «Основы программирования» Пояснительная записка

|  |  |
| --- | --- |
|  | Обучающийся гр. 722-1  \_ С.Н. Балачкова  (подпись) (И.О. Фамилия)  (дата) |
| (оценка) | Руководитель практики от Университета:  Доцент каф. БИС, канд. техн. наук  (должность, ученая степень, звание)  \_ С.С. Харченко  (подпись) (И.О. Фамилия)  (дата) |

Томск 2023

Министерство науки и высшего образования Российской Федерации Федеральное государственное бюджетное образовательное учреждение высшего образования

ТОМСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ СИСТЕМ УПРАВЛЕНИЯ

И РАДИОЭЛЕКТРОНИКИ (ТУСУР)

Кафедра комплексной информационной безопасности электронно- вычислительных систем (КИБЭВС)

КАЛЬКУЛЯТОР РАСХОДОВ НА ЯЗЫКЕ C#

ТЕХНИЧЕСКОЕ ЗАДАНИЕ

На 13 листах

|  |  |
| --- | --- |
| СОГЛАСОВАНО  Доц. кафедры БИС  С.С. Харченко  « » 2023 г. | РАЗРАБОТЧИК  Студентом гр. 722-1  С.Н. Балачкова  « » 2023 г. |

Томск 2023

1 Общие сведения

* 1. Полное наименование системы и её условное обозначение Полное наименование системы: «Десктопное приложение «Калькулятор

расходов на языке C#».

* 1. Заказчик

Заказчиком является Томский государственный университет систем управления и радиоэлектроники, кафедра комплексной информационной безопасности электронно-вычислительных систем (КИБЭВС).

* 1. Исполнитель

Исполнителем является студент группы 722-1 Балачкова София Николаевна

* 1. Основания разработки

Основанием для разработки является задание на выполнение курсовой работы по дисциплине “Основы программирования” для студентов направления подготовки 10.05.03 – Информационная безопасность автоматизированных систем, профиля «Информационная безопасность в кредитно-финансовой сфере».

1. Назначение и цель создания системы
   1. Назначение системы

Десктопное приложение «Калькулятор расходов» было выбрано для упрощения жизни студентов и обывателей города. Таким образом, данная система должна быть полезна для пользователя, а разработка этой системы должна углубить знания и умения в разработке программ и приложений на языке программирования C#.

* 1. Цели создания системы

Целью разработки является получение опыта в разработке приложений с интерактивным графическим интерфейсом и создание программы

«Калькулятор расходов» для нужд студентов.

1. Характеристика объектов автоматизации
   1. Объект автоматизации

Объектом автоматизации является возможность для пользователей – студентов видеть свои расходы в течение определенного периода. Также это позволит студентам грамотно рассчитывать и корректировать свои привычки, чтобы расходовать меньшее количество средств

1. Требования к системе
   1. Требования к структуре и функционированию Приложение должно выполнять следующие функции:

* возможность для автоматизации и регистрации;
* возможность выбора периода сбора статистики расходов;
* возможность вывода суммы расходов за выбранный период.
  1. Перечень подсистемы, их назначение и основные характеристики

В системе предлагается выделить следующие функциональные подсистемы:

* подсистема графического интерфейса, для более удобного взаимодействия с приложением;
* подсистема авторизации;
* подсистема регистрации;
* подсистема взаимодействия с базой данных;
  1. Требования к надёжности

При возникновении сбоев в аппаратном обеспечении, включая разряд аккумулятора устройства, информационная система восстанавливает свою работоспособность после устранения сбоев и корректного перезапуска аппаратного обеспечения (за исключением случаев повреждения рабочих носителей информации с исполняемым программным кодом).

* 1. Требования по безопасности

Все технические решения, использованные при создании системы, а также при определении требований к аппаратному обеспечению, соответствуют действующим нормам и правилам техники безопасности, пожарной безопасности, а также охраны окружающей среды при эксплуатации.

* 1. Требования к эксплуатации, техническому обслуживанию, ремонту и хранению

Для эксплуатации разрабатываемой информационной системы необходимы следующие условия:

* компьютер под управлением операционной системы Windows 11;
* предустановленный .Net Framework v 6.0;
* питание компьютера от сети или батареи;
* предустановленная СУБД MS SQL 2019;
* наличие таких периферийных устройств, как мышь и клавиатура, для взаимодействия.
  1. Требования к защите информации от несанкционированного доступа

Доступ к работе с интерфейсом системы имеют только авторизованные пользователи.

* 1. Требования к функциям разработчика

Роль разработчика заключается в обновлении и пополнении системы новыми функциями, а также исправление возможных ошибок в функционировании системы.

* 1. Требования к функциям разработчика

Пользователь может использовать все функции, которыми обладает система.

* 1. Описание процессов и функций работы с системой

Процессы и функции, выполняемые при эксплуатации системы, приведены в разбивке по подсистемам: подсистема графического интерфейса, для более удобного взаимодействия с приложением, подсистема авторизации, подсистема регистрации, подсистема взаимодействия с базой данных. Процессы, реализованные под управлением различных подсистем, реализуются на основе системных процедур, которые являются составной частью функции системы. Системные процедуры группируются в соответствии с их назначением:

* графический интерфейс пользователя;
* авторизация/регистрация пользователей.
  1. Требования к информационному обеспечению системы Компоненты системы должны активно взаимодействовать с системой

управления базой данных (СУБД). Обмен информацией с СУБД должен происходить автоматически. Уровень хранения данных в системе должен быть

построен на основе современных реляционных или объектно-реляционных СУБД. Доступ к данным должен быть предоставлен только авторизованным пользователем.

* 1. Требования к программному обеспечению
* ОС Windows 11;
* СУБД MS SQL 2019;
* язык программирования C#;
* .Net Framework 6.0;
* установленное ПО.

1. Состав и содержание работ по созданию системы

Состав и содержание работ по созданию системы приведены в таблице

5.1.

Таблица 5.1 – Этапы разработки

|  |  |  |  |
| --- | --- | --- | --- |
| № | Этап | Результат | Срок выполнения |
| 1 | Проектирование  системы | UML диаграмма  классов | 1.05.23 – 20.06.23 |
| 2 | Разработка ПО | Приложение,  репозиторий на github | 20.06.23 – 5.09.23 |
| 3 | Создание БД и подключение к СУБД | SQL запросы  инициализации и миграции БД | 5.09.23 – 1.11.23 |
| 4 | Тестирование и отладка  приложения | Набор unit-test, отчёт о  тестировании | 1.11.23 – 1.12.23 |
| 5 | Защита курсовой работы | Пояснительная записка | 1.12.23 – 14.01.24 |

1. Порядок контроля и приёмки системы
   1. Перечень этапов испытаний и проверок

Этапы испытаний подразделяются на предварительные и приемочные. Предварительные испытания проводятся во время сдачи проекта разработчиком совместно с заказчиком. Все подсистемы испытываются одновременно на корректность взаимодействия подсистем, влияние подсистем друг на друга, то есть испытания проводятся комплексно. Во время приемочных испытаний оценивается:

* полнота и качество реализации функций, указанных в настоящем техническом задании;
* демонстрация объективно-ориентированного подхода при реализации функций, указанных в настоящем техническом задании;
* выполнение каждого требования, относящиеся к интерфейсу системы;
* полнота действий, доступных пользователю:

1. регистрация и авторизация пользователей;
2. взаимодействие с графическим интерфейсом;
3. работоспособность основных функций аналогичных приложений для навигации;
4. выход авторизованного пользователя из приложения.

При проверке регистрации производится несколько попыток регистрации пользователя.

При авторизации пробуется авторизация зарегистрированного пользователя и не зарегистрированного пользователя.

При проверке корректного выхода авторизированного пользователя из приложения при нажатии на выход из приложения должна открыться главная страница приложения, и, чтобы продолжить работу с приложением, нужно

заново авторизоваться. Также просматриваются коды графического интерфейса, коды выполнения запросов и просматривается база данных.

Приемка результатов должна осуществляться в сроки, установленные таблицей 5.1. Результаты проектирования системы предоставляются в электронном виде с помощью ЭИОС sdo.tusur.ru. Результаты разработки программного обеспечения будут располагаться на репозитории GitHub. Результаты тестирования будут представлены в электронном виде с помощью ЭИОС sdo.tusur.ru.

* 1. Общие требования к приемке работы

Приемка осуществляется представителями Заказчика и Исполнителя. Все создаваемые в рамках настоящей работы программные изделия передаются Заказчику, как в виде готовых модулей, так и в виде исходных кодов, представляемых в электронной форме на репозитории GitHub.

1. Требования к составу и содержанию работ по подготовке объекта автоматизации к вводу системы в действие

Для обеспечения готовности объекта к вводу системы в действие провести комплекс мероприятий:

* загрузка файлов приложения;
* проведение предварительных испытаний;
* проверка приемочных испытаний.

1. Требования к документированию

Состав программной документации:

* задание на курсовую работу;
* техническое задание (ТЗ);
* пояснительная записка (ПЗ);
* документация к системе в электронном виде. Документация должна быть оформлена с использованием:

 ГОСТ 34.602-2020;

* ОС ТУСУР 01-2021 для технического задания;
* ОС ТУСУР 01-2021 для пояснительной записки.

Реферат

Отчет содержит 42 страниц, 2 таблицы, 28 рисунков, 4

источника, 4 приложения.

Десктопное приложение «Калькулятор расходов».

Цель работы – разработать программу, которая позволяет пользователям авторизироваться и отслеживать свои расходы.

В результате работы были разработана программа, которая после авторизации пользователя выводит сумму расходов за определенный период.

Разработка программы проводилась на языке программирования С# в

программе Visual Studio 2022.

Исходные коды программы расположены в репозитории по адресу: .

Отчет составлен в текстовом редакторе Microsoft Office Word 2021. Пояснительная записка была оформлена согласовано ОС ТУСУР 01-

2022[2].

Abstract

The report contains 42 pages, 2 tables, 28 figures, 4 sources, 4 appendices. Expense calculator desktop application.

The goal of the work is to develop a program that allows users to log in and track their expenses.

As a result of the work, a program was developed that, after user authorization, displays the amount of expenses for a certain period.

The program was developed in the C# programming language in Visual Studio 2022 program.

The source codes of the program are located in the repository at: .

The report was compiled in the text editor Microsoft Office Word 2021. The explanatory note was drawn up and agreed upon by OS TUSUR 01-

2022[2].

Министерство науки и высшего образования Российской Федерации Федерального государственное бюджетное образовательное учреждение высшего образования

«ТОМСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ СИСТЕМ УПРАВЛЕНИЯ И РАДИОЭЛЕКТРОНИКИ» (ТУСУР)

Кафедра комплексной информационной безопасности электронно- вычислительных систем (КИБЭВС)

УТВЕРЖДАЮ

Зав. кафедрой БИС

А.А. Шелупанов

« » 20 г.

Задание

на курсовую работу по дисциплине «Основы программирования» студенту гр.722-1 факультета безопасности

Балачковой Софии Николаевне Тема работы: Калькулятор расходов.

Цель работы: Разработка десктопного приложения: «Калькулятор расходов». Получение навыков программирования на языках высоко уровня.

Срок сдачи студентом законченной работы « » 20 г. Исходные данные к работе: Программа вычисляет сумму расходов

пользователя за определенный период времени и показывает динамику расходов: увеличились или уменьшились.

Задание: разработать калькулятор расходов, который вычисляет сумму расходов пользователя за определенный период времени и показывает динамику расходов.

Требования к используемым технологиям: язык программирования C#,

.Net Framework 6.

## Оглавление

[ВВЕДЕНИЕ 19](#_bookmark0)

1. [ОБЗОР 20](#_bookmark1)
   1. [Описание алгоритма работы игры “Угадай число” 20](#_bookmark2)
   2. [Описание программы 20](#_bookmark3)
   3. [Преимущества и недостатки 20](#_bookmark4)
   4. [Аналоги программы 21](#_bookmark5)
   5. [Обоснование выбранных технологий 21](#_bookmark6)
2. [Проектирование 23](#_TOC_250003)
   1. Словесное описание алгоритма 23
   2. [Блок схема алгоритма 24](#_TOC_250002)
   3. [Проектирование структуры программы 25](#_TOC_250001)
   4. [Проектирование графического интерфейса 26](#_TOC_250000)
      1. Форма «Игра угадай число» 27
3. Реализация приложения 29
4. Тестирование 33
   1. UNIT тесты 33
   2. Программа и методика испытаний 36
   3. Ручное тестирование 38
5. Руководство пользователя 45

Заключение 49

Список использованных источников 50

Приложение А 51

Приложение Б 53

Приложение В 55

Приложение Г 56

Приложение Д 57

# ВВЕДЕНИЕ

Цель работы: получение и улучшение навыков программирования на языке высокого уровня C#.

## ОБЗОР

* 1. **Описание алгоритма работы игры «Калькулятор расходов»**

«Калькулятор расходов» - программа, которая собирает от пользователя все его затраты, а затем выводит сумму расходов за определенный период.

## Описание программы

Программа предлагает выбрать границы, в которых будет производиться расчет общей суммы расходов. С клавиатуры вводится начало и конец границы. После чего в этом промежутке вычисляется сумма расходов, введенных заранее.

## Преимущества и недостатки

Преимущества:

* Вычисление суммы затраченных средств: человеку не требуется это делать вручную;

Недостатки:

* Маленький функционал. Помимо суммы расходов на данном этапе ничего выводиться не планируется.

## Аналоги программы

Аналогичный механизм работает в банковских приложениях: Тинькофф, Сбер(СБОЛ), Альфа-банк и так далее.

В банковских приложениях данный механизм не только высчитывает сумму расходов, но и на основе собранных данных дает рекомендации, как тратить меньше.

## Обоснование выбранных технологий

Как платформа пользовательского интерфейса была выбрана Windows Forms. Другие технологии (WPF, UWP, Xamarin) не будут использованы, так как они сложнее для изучения, чем Windows Forms, а также имеют функции, которые не понадобятся для выполнения работы. То есть Windows Forms была выбрана, потому что с помощью неё проще проектировать простой GUI и она имеет достаточно высокую производительность для приложений с простой графикой (без анимации, прозрачности, 3D, и т.д.).

В качестве языка программирования был выбран язык C#. Есть аналоги C#, с помощью которых можно создать приложение Windows Forms: C++, F#, Visual Basic

.NET. Но ни один из этих аналогов не выбран: C++ не будет использован, так как скорость разработки на нём будет невысокой; F# не будет использован, так как он имеет высокий порог входа и совсем небольшое сообщество разработчиков, так как не очень популярен, по сравнению с C#; Visual Basic

.NET не будет использован, так как он имеет более низкую производительность и развивается очень медленно. То есть C# был выбран, потому что приложение будет разрабатываться для Windows, с использованием платформы Windows Forms, на данном языке проще программировать, благодаря автоматической «сборке мусора», большого разнообразия синтаксических конструкций, большого количества библиотек и шаблонов, C#

объединяет лучшие идеи современных языков программирования Java, C++, Visual Basic и т.д., язык понятный и подходящий для новичков, а также имеет строгую типизацию.

Была выбрана технология .NET Framework. Есть более продвинутые аналоги этой технологии. Один из них - .Net 5, и он не будет использован, так как некоторые сторонние библиотеки .NET, пакеты NuGet, технологии и платформы, существующие в .NET Framework, недоступны в .NET 5. Так как приложение разрабатывается для Windows, использование более продвинутых платформ излишне, так как они имеют функции, которые не понадобятся для выполнения работы. То есть .NET Framework был выбран, потому что механизмы работы с классическим приложением Windows Forms реализованы в нем лучше всего.

## Проектирование

* 1. **Словесное описание алгоритма (псевдокод)**

А.1 Начало;

А.2 запросить у пользователя начальную и конечную даты периода (например, "с" и "по");

А.3 инициализировать переменную "сумма\_расходов" как ноль; А.4 инициализировать пустой список "расходы\_за\_период"; А.5 пока текущая дата не превышает "по";

А.5.1 запросить у пользователя дату и сумму расходов за этот день;

А.5.2 если дата входит в указанный период (больше или равна "с" и меньше или равна "по"):

А.5.2.1 добавить сумму расходов к "сумма\_расходов";

А5.2.2 добавить запись (дата, сумма) в "расходы\_за\_период"; А.5.3 перейти к следующей дате;

А.6 вывести "Сумма расходов за указанный период: сумма\_расходов"; А.7 если "расходы\_за\_период" не пустой:

А.7.1 вычислить первую и последнюю записи в "расходы\_за\_период";

А.7.2 определить, увеличились или уменьшились расходы, сравнив первую и последнюю записи;

А.7.3 вывести сообщение о динамике расходов: "Расходы в период с [первая\_дата] по [последняя\_дата] [увеличились/уменьшились] на [разница]";

А.8 конец.

## Блок-схема алгоритма

На рисунках 2.1 и 2.2 представлена блок-схема программы.

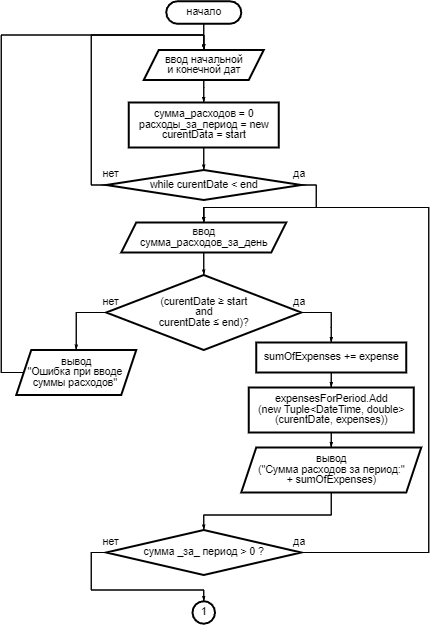


Рисунок 2.1 – Блок-схема работы программы

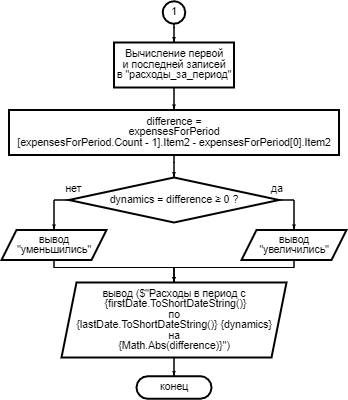


Рисунок 2.2 – Продолжение блок-схемы

## Проектирование структуры программы

Были составлены: UML-диаграмма классов и диаграмма прецедентов.

Диаграмма прецедентов определяет функциональные требования к системе и описывает типичные взаимодействия между пользователями.

UML-диаграмма классов и диаграмма прецедентов представлены на рисунках 2.3 и 2.4 соответственно.

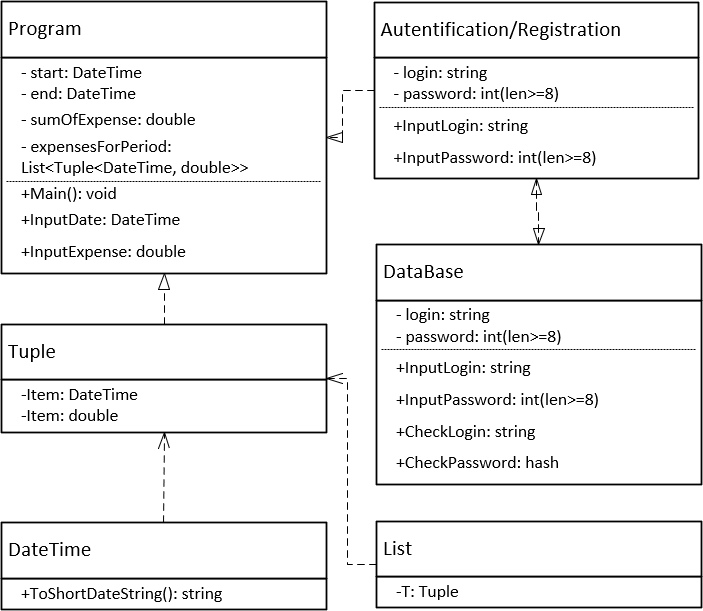


Рисунок 2.3 – UML-диаграмма классов

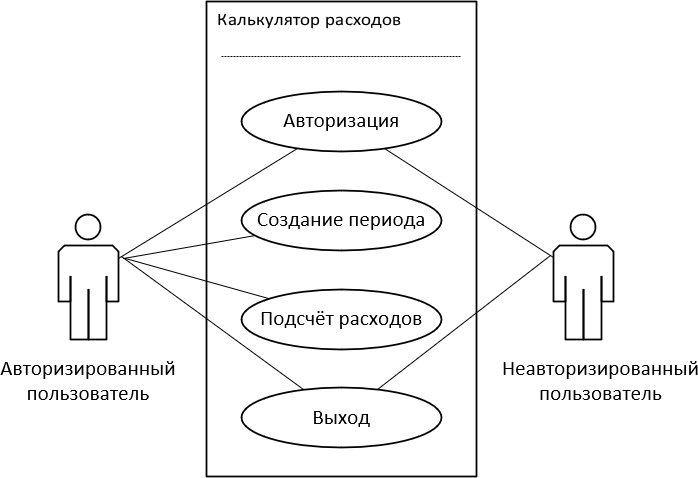


Рисунок 2.3 – Диаграмма прецедентов

## Проектирование графического интерфейса

Была создана форма для авторизации пользователя. Для этого было создано:

* два элемента «TextBox»;
* один элемент «Label»;
* два элемента «Button» («Вход» и «Регистрация»). Форма «Авторизация» представлена на рисунке 2.3.1

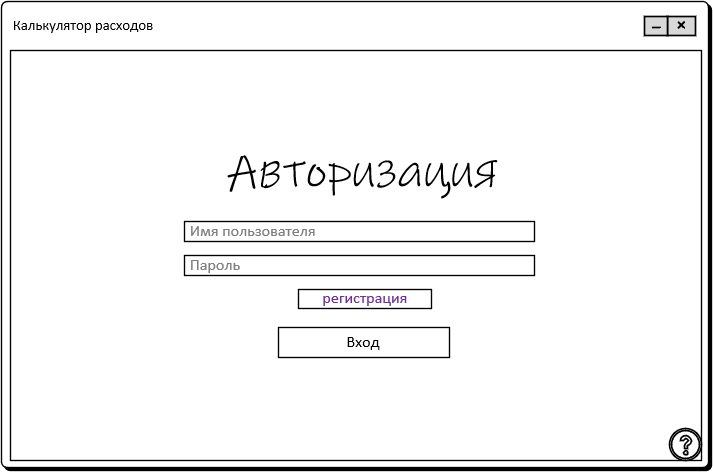


Рисунок 2.3.1 – Форма «Авторизация»

## Форма «Калькулятор расходов»

Была создана форма для реализации калькулятора расходов. Для этого были созданы:

* два элемента «DateTimePicker»;
* один элемент «TextBox»;
* пять элементов «Label»;
* один элемент «Button» («Начать подсчёт»);
* два элемента «RichTextBox».

Форма «Калькулятор расходов» представлена на рисунке 2.3.2.

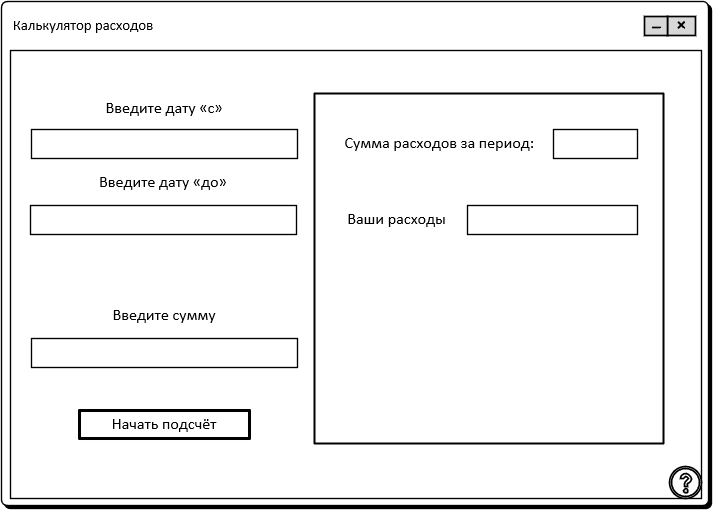


Рисунок 2.3.2 – Форма «Калькулятор расходов»

При нажатии на кнопку «Начать подсчёт» в элементах «RichTextBox» будут отображаться результаты работы программы: сумма расходов и подведение итога: увеличились или уменьшились расходы по сравнению с предыдущим периодом.

При закрытии приложения и его повторном открытии происходит возврат к форме «Авторизация» и сброс ранее введённых логина и пароля.