МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ

Федеральное государственное бюджетное образовательное учреждение

высшего образования

**«Вятский государственный университет»**

Факультет автоматики и вычислительной техники

Кафедра электронных вычислительных машин

Допущено к защите

Руководитель проекта

\_\_\_\_\_\_\_\_\_\_/Клюкин В.Л./

(подпись) (Ф.И.О)

«\_\_\_»\_\_\_\_\_\_\_\_\_\_2023 г.

Разработка приложения для расчета полной стоимости потребительского кредита

Пояснительная записка курсового проекта по дисциплине

«Комплекс знаний бакалавра в области программного и аппаратного обеспечения вычислительной техники»

ТПЖА.09.03.01.512982 ПЗ

Разработал студент группы ИВТ-21 \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ /Категов А.Д./

Руководитель /Клюкин В.Л./

Консультант \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_/Долженкова М.Л./

Работа защищена с оценкой «\_\_\_\_\_\_\_\_\_\_\_\_» \_\_\_\_\_\_\_\_\_\_\_\_\_\_

(оценка) (дата)

Члены комиссии \_\_\_\_\_\_\_\_ \_\_\_\_\_/ /

(подпись)

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

(подпись)

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

(подпись)

Киров 2023

Реферат

Категов А. Д. Разработка приложения для расчета полной стоимости потребительского кредита. ТПЖА.09.03.01.512982 ПЗ: Курс. проект / ВятГУ, каф. ЭВМ; рук. Клюкин В.Л. - Киров, 2023. – ПЗ 47с, 3 табл., 15 рис.

РАЗРАБОТКА ПРИЛОЖЕНИЯ, ТЕОРИЯ ПРОЦЕНТА, НЕМЕДЛЕННЫЙ АННУИТЕТ, АННУИТЕТ С ПЕРЕМЕННЫМ ПЛАТЁЖНЫМ ПЕРИОДОМ, ПОСТОЯННЫЙ АНАЛИТИЧЕСКИЙ КАЛЕНДАРЬ, ПОЛНАЯ СТОИМОСТЬ ПОТРЕБИТЕЛЬСКОГО КРЕДИТА.

Объект исследования и разработки – математический аппарат необходимый для разработки алгоритма расчета полной стоимости потребительского, аннуитетного кредита с переменным платежным периодом.

Цель курсового проекта – разработка алгоритма расчета полной стоимости потребительского, аннуитетного кредита с переменным платежным периодом, соответствующего указаниям центрального банка РФ и требованиям заказчика.

Результатом выполнения курсового проекта является исходный код приложения, рассчитывающего полную стоимость потребительского, аннуитетного кредита с переменным платежным периодом, в соответствии с указаниями центрального банка РФ и требованиями заказчика.

Оглавление

Введение 5

1. Анализ предметной области 6
2. Постановка задачи 13
3. Разработка алгоритмов 17
4. Пример расчета 29
5. Разработка структуры программы 31
6. Разработка интерфейса пользователя 35
7. Программная реализация 37
8. Тестирование 39

Заключение 41

Перечень сокращений 42

Приложение A. Список литературы 43

Приложение Б. Листинг кода 44

Введение

В современной экономике кредит является одним из основных условий и предпосылок экономического развития государства, а также важной и неотъемлемой частью экономического роста. Объем этих кредитов увеличивается с каждым годом, а жесткие условия межбанковской конкуренции на рынке заставляют банки искать пути повышения привлекательности для клиентов.

Этот процесс требует не только значительных инвестиций, обширной и разветвленной сети банковских филиалов, современных банковских технологий, но и глубокого понимания системы индивидуального кредитования.

Причиной разработки проекта стало желание изучить структуру современных кредитных отношений.

Актуальность данного проекта заключается в высокой востребованности услуг потребительского кредитования, и, следовательно, высокой необходимости контроля, оценки и сравнения кредитных предложений.

В ходе курсового проекта необходимо ознакомиться с структурой современных кредитных отношений. Разработать алгоритм расчета полной стоимости потребительского кредита с переменным платежным периодом, в соответствии с указаниями центрального банка РФ и требованиями заказчика.

Целью курсового проекта является создание исходного кода рабочего прототипа программы для расчёта полной стоимости потребительского, аннуитетного кредита с переменным платежным периодом.

1. Анализ предметной области

В данном разделе производится обзор предметной области, даются определения необходимым понятиям, связанным с банковской системой и теорией процента. Также проводится обзор, аналилз и сравнение сущесствующих аналогов.

* 1. Обзор предметной области

Банковское кредитование предприятий и населения относится к традиционным видам банковских услуг. Наибольшая часть активов КБ по-прежнему помещена в кредитные операции [1]. Кредиты играют важную роль в экономике, обеспечивая доступ к финансовым ресурсам для реализации различных потребительских и бизнес-проектов [1]. С течением времени, с развитием технологий и изменением потребительских запросов, структура и условия кредитования становятся все более разнообразными и сложными.

Из-за разнообразия условий и изменений в формулах осуществлять контроль и оценку кредитных предложений стало крайне сложно. Даже внимательного изучения договора и конкретных условий кредитования стало недостаточно, чтобы точно определить выгоду от кредита и его конкретную стоимость для заемщика. Для сравнения кредитных предложений и определения того, какой кредит будет более выгодным для заемщика была разработана специальная величина под названием полная стоимость кредита.

Изначально в России не было практики того, что банки афишируют свои ссуды. Она появилась на фоне многочисленных конфликтов между займодателями и заемщиками.

Исторически, ПСК использовалась для более точного сравнения различных кредитных предложений, и она должна была помочь заемщикам оценить, насколько выгодно кредитное предложение на самом деле, учитывая все платежи и условия. Однако, законодатели и индустрия кредитования усложняли в формулу расчета ПСК, для адаптации ее к различным видам кредитов.

В итоге, полная стоимость кредита (ПСК) стала показателем, который используется для оценки затрат, связанных с получением кредита, и для сравнения различных кредитных предложений. Она представляет собой процентную ставку, выраженную в процентах годовых, которые являются годовой доходностью банка, она учитывает не только номинальную процентную ставку, но и все дополнительные затраты, комиссии и платежи, связанные с кредитом. Важно понимать, что ПСК включает в себя все эти дополнительные расходы, делая оценку реальной стоимости кредита более прозрачной и информативной.

ПСК не может расширяться или сокращаться, по решению какой-либо из сторон сделки или же третьих лиц. Есть четкий список допустимых расходов, которые могут быть включены в ПСК.

В настоящее время все банки обязаны указывать на первой странице договора ПСК, благодаря федеральному закону от 21.12.2013 N 353-ФЗ (последняя редакция от 24.07.2023) "О потребительском кредите (займе)" [2]. Если банк игнорирует это обязательство и не указывает на первой странице ПСК, то договор подлежит расторжению, а все средства, которые внес заемщик по переплате, возвращаются ему [2].

* 1. Анализ приложения «Кредитный калькулятор 2.0»

Произведем разбор приложения для расчета аннуитетного кредита под названием «Кредитный калькулятор 2.0». На рисунке 1 представлена экранная форма интерфейса приложения.

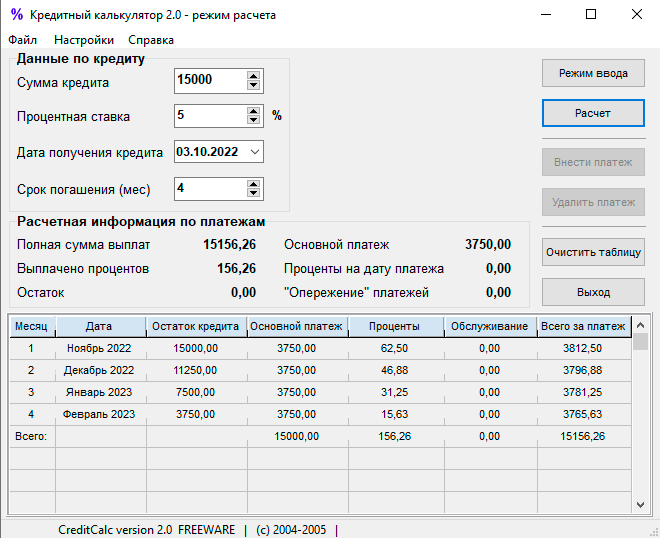


Рисунок 1 – Экранная форма интерфейса приложения «Кредитный калькулятор 2.0»

Приложение «Кредитный калькулятор 2.0» производит расчет ежемесячного платежа аннуитетного кредита и выводит на экран график его погашения, так же пользователь может увидеть полную сумму выплат и сумму процентов по кредиту.

Однако данное приложение обладает рядом недостатков. В графике погашения кредита не просчитываются даты платежей, что делает его неполным и не информативным. При просчетах реальных кредитов учитываются выходные и праздничные дни. Отсутствие учета выходных и праздничных дней при просчете кредита в данном приложении делает результат расчетов неверным. Отсутсвует расчет переплаты в процентах от суммы платежей, что тоже влияет на информативность. Отсутсвует расчет ПСК и, следовательно, при использовании данного приложения невозможно сравнить и оценить кредитные приложения.

Таким образом, приложение «Кредитный калькулятор 2.0» может дать только приблизительную информацию о кредите. Использовать его для точного расчета, оценки и сравнения аннуитетных кредитов не представляется возможным.

* 1. Анализ калькулятора от сайта «www.sravni.ru»

Произведем разбор калькулятора от сайта «www.sravni.ru». На рисунке 2 представлена экранная форма интерфейса приложения.

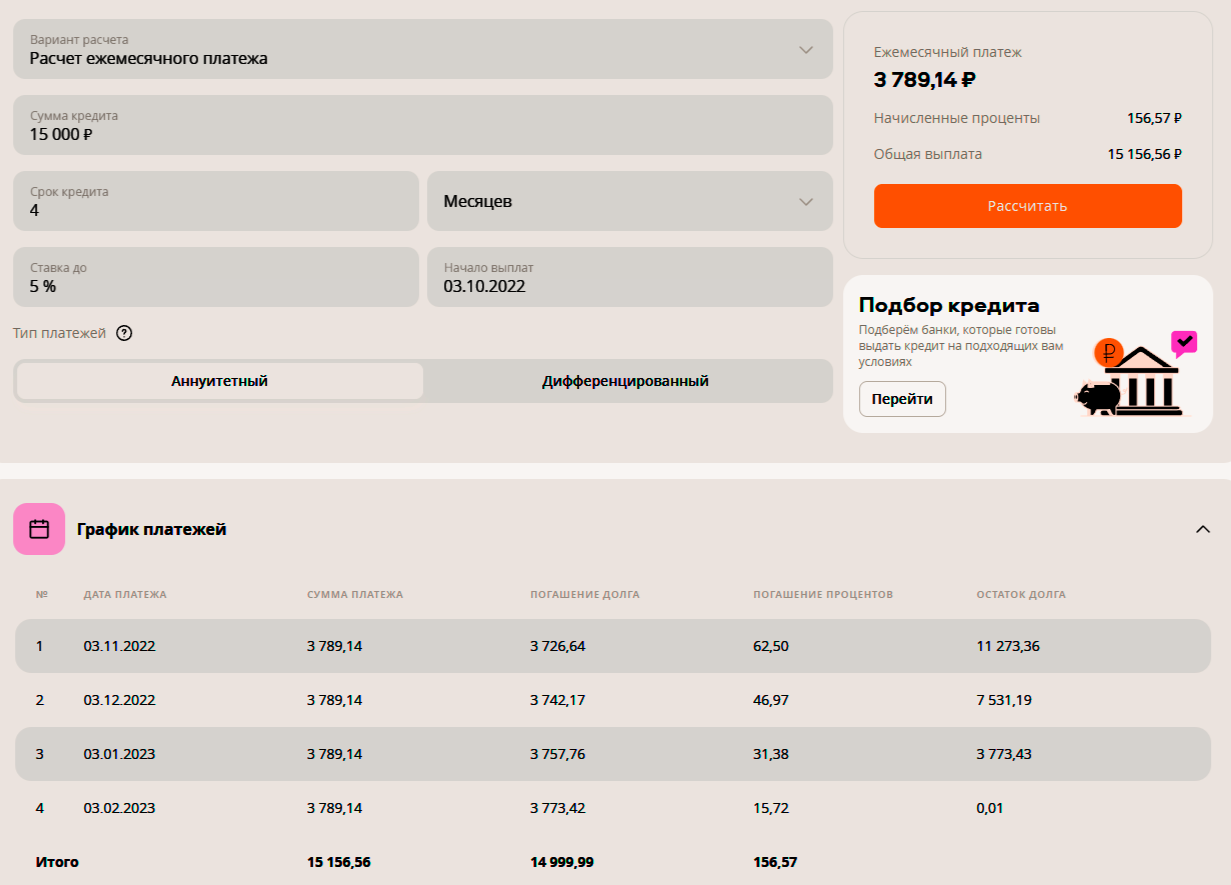


Рисунок 2 – Экранная форма калькулятора на сайте «www.sravni.ru»

Данный калькулятор производит расчет ежемесячного платежа аннуитетного кредита и выводит на экран график его погашения, так же пользователь может увидеть полную сумму выплат и сумму процентов по кредиту.

Однако данный калькулятор также обладает рядом недостатков. В графике погашения кредита не учитываются выходные и праздничные дни, что делает его неверным. При просчетах реальных кредитов учитываются выходные и праздничные дни. Отсутствие учета выходных и праздничных дней при просчете кредита делает результат расчетов неверным. Отсутсвует расчет переплаты в процентах от суммы платежей, что тоже влияет на информативность. Отсутсвует расчет ПСК и, следовательно, при использовании данного приложения невозможно сравнить и оценить кредитные приложения.

Таким образом, калькулятор от сайта «www.sravni.ru» может дать только приблизительную информацию о кредите. Использовать его для точного расчета, оценки и сравнения аннуитетных кредитов не представляется возможным.

* 1. Анализ калькулятора от сайта «calculator-credit.ru»

Произведем разбор калькулятора от сайта «calculator-credit.ru». На рисунке 3 представлена экранная форма интерфейса приложения.

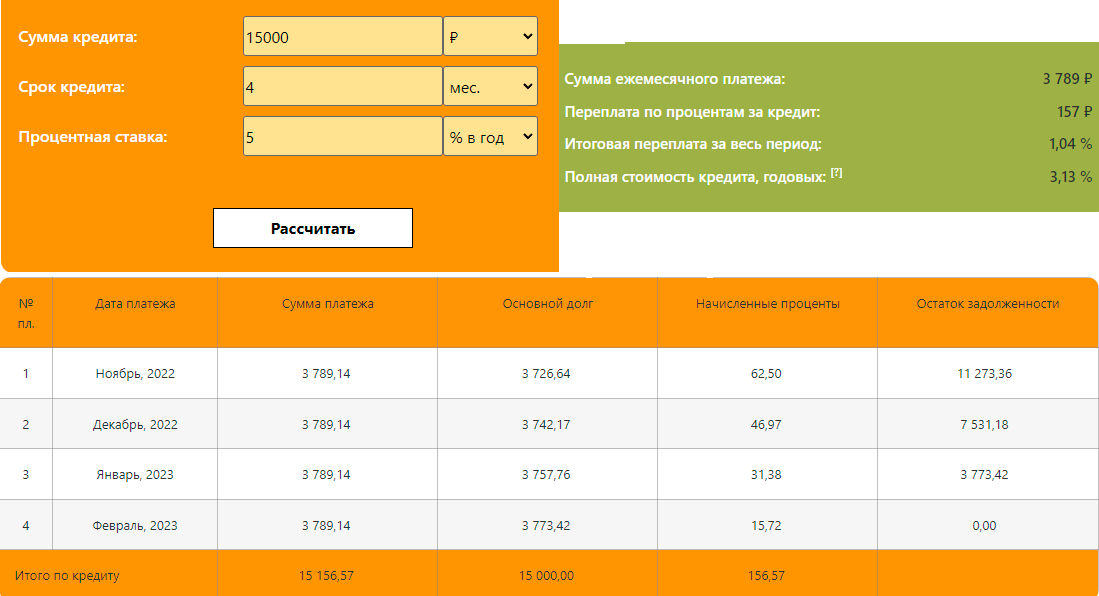


Рисунок 3 – Экранная форма калькулятора на сайте «calculator-credit.ru»

Данный калькулятор производит расчет ежемесячного платежа аннуитетного кредита и выводит на экран график его погашения, так же пользователь может увидеть сумму процентов по кредиту, расчет переплаты в процентах от суммы платежей и ПСК.

Данный калькулятор обладает рядом недостатков. В графике погашения кредита не просчитываются даты платежей, что делает его неполным и не информативным. Отсутствие учета выходных и праздничных дней при просчете кредита делает результат расчетов неверным. Сайт не предоставляет формулу расчета ПСК. При заданных параметрах ПСК не может быть меньше 5%, следовательно, расчет выполнен неверно и, при использовании данного приложения невозможно верно сравнить и оценить кредитные приложения.

Таким образом, калькулятор от сайта «calculator-credit.ru» может дать только приблизительную информацию о кредите. Использовать его для точного расчета, оценки и сравнения аннуитетных кредитов не представляется возможным.

* 1. Общие выводы из анализа существующий аналогов

Для более удобного представления, информация о рассмотренных аналогах кредитного калькулятора представлена в таблице 1.

Таблица 1 – Аналоги кредитного калькулятора

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Аналог | Расчет дат платежей | Расчет ежемесячного платежа | График погашения кредита | Расчет переплаты | Расчет ПСК |
| «Кредитный калькулятор 2.0» | отсутствует | неверен | неполон | неверен | отсутствует |
| сайт «www.sravni.ru» | неверен | неверен | неверен | неверен | отсутствует |
| cайт «calculator-credit.ru» | отсутствует | неверен | неполон | неверен | неверен |

Из таблицы 1 видно, что, среди рассмотренных кредитных калькуляторов ни один не производит верный расчет аннуитетного кредита и его ПСК. Таким образом, получить полную и достоверную информацию о кредите возможно только в отделении банка, при его непосредственном оформлении. Отсутствие достоверной информации о кредитном предложении и ограничения во времени могут приводить к ошибкам и недопониманиям. Следовательно, существует потребность в приложении, которое будет осущесвлять расчет аннуитетных кредитов, их ПСК и обеспечивать пользователя точной информацией.

Для того чтобы облегчить работу с индивидуальными кредитами для работников банка, а для клиентов сделать процедуру получения кредита более прозрачной, было принято решение разработать приложение для расчета полной стоимости потребительского кредита.

1. Постановка задачи
   1. Полное наименование системы

Полное наименование системы: Приложение для расчета полной стоимости потребительского кредита "Расчет ПСК".

Краткое наименование системы: Расчет ПСК.

* 1. Назначение системы

Приложение "Расчет ПСК" предназначено для информационного обеспечения процессов АО КБ "Хлынов", в части исполнения следующих процессов:

– произведение расчета полной стоимости кредита;

– произведение расчета ежемесячного платежа;

– построение графика погашения кредита;

– произведение расчета переплаты.

"Расчет ПСК" предполагается использовать в отделениях банка "Хлынов" при работе с клиентами, для выполнения вышеперечисленных процессов.

* 1. Цели создания системы

Основными целями создания приложения "Расчет ПСК" являются:

– Повышение эффективности исполнения процессов, перечисленных выше, путем сокращения операций, выполняемых «вручную»;

– Повышение качества оказания услуг за счет полноты, достоверности и удобства форматов отображения информации;

– Повышение информационной открытости и прозрачности оказания услуг АО КБ "Хлынов", повышение удобства и комфорта клиентов банка.

Для реализации поставленных целей система должна решать следующие задачи:

– Ввод данных о кредите;

– Произведение расчетов;

– Вывод информации на экран;

* 1. Перечень подсистем, их назначение и основные характеристики

В состав приложения Расчет ПСК должны входить следующие подсистемы:

* Подсистема вычисления дат платежей
* Подсистема расчета ежемесячного платежа
* Подсистема построения графика погашения кредита
* Подсистема расчета дополнительной информации о кредите

Подсистема вычисления дат платежей предназначена для расчета дат платежей по дате получения и сроку выдачи кредита. Подсистема должна учитывать, что платеж должен выпадать только на рабочий день.

Подсистема расчета ежемесячного платежа предназначена для расчета ежемесячного платежа аннуитетного графика погашения по сумме кредита и процентной ставке.

Подсистема построения графика по гашению кредита предназначена для расчета аннуитетного графика погашения кредита и вывода его на экран.

Подсистема расчета дополнительной информации о кредите предназначена для расчета полной стоимости кредита и переплаты

* 1. Требования к численности и квалификации персонала системы

Для эксплуатации приложения Расчет ПСК определена роль пользователя. Пользователи системы должны иметь опыт работы с персональным компьютером на базе операционных систем Microsoft Windows на уровне квалифицированного пользователя и свободно осуществлять базовые операции в стандартных Windows.

* 1. Требования по стандартизации и унификации

Взаимодействие пользователей с прикладным программным обеспечением, входящим в состав системы должно осуществляться посредством визуального графического интерфейса. Интерфейс системы должен быть понятным и удобным, не должен быть перегружен графическими элементами. Навигационные элементы должны быть выполнены в удобной для пользователя форме. Ввод-вывод данных системы, прием управляющих команд и отображение результатов их исполнения должны выполняться в интерактивном режиме. Интерфейс должен соответствовать современным эргономическим требованиям и обеспечивать удобный доступ к основным функциям и операциям системы.

Система должна обеспечивать корректную обработку аварийных ситуаций, вызванных неверными действиями пользователей, неверным форматом или недопустимыми значениями входных данных.

* 1. Требования к функциям (задачам), выполняемым системой

Подсистема вычисления дат платежей должна по дате получения кредита и сроку выплат вычислять календарные даты платежей с учетом того, что каждый платеж должен выпадать на рабочий день.

Подсистема расчета ежемесячного платежа должна по сумме кредита, процентной ставке и количеству платежей вычислять ежемесячный аннуитетный платеж.

Подсистема построения графика погашения кредита должна по сумме ежемесячного платежа и датам платежей рассчитывать аннуитетный график погашения кредита и выводить его на экран.

Подсистема расчета дополнительной информации о кредите должна по графику погашения кредита рассчитывать полную стоимость кредита и переплату.

1. Разработка алгоритмов

В данном разделе будут определены функциональные диаграммы. Они создаются на ранних этапах проектирования систем, для выявления основных функций и алгоритмов проектируемой системы. Для этой цели используется методология IDEF0 (Integrated Definition Function Modeling), а для документирования каждой функции IDEF3 (workflow diagramming).

Контекстная диаграмма IDEF0, определяющая границы моделируемой системы, представлена на рисунке 4.

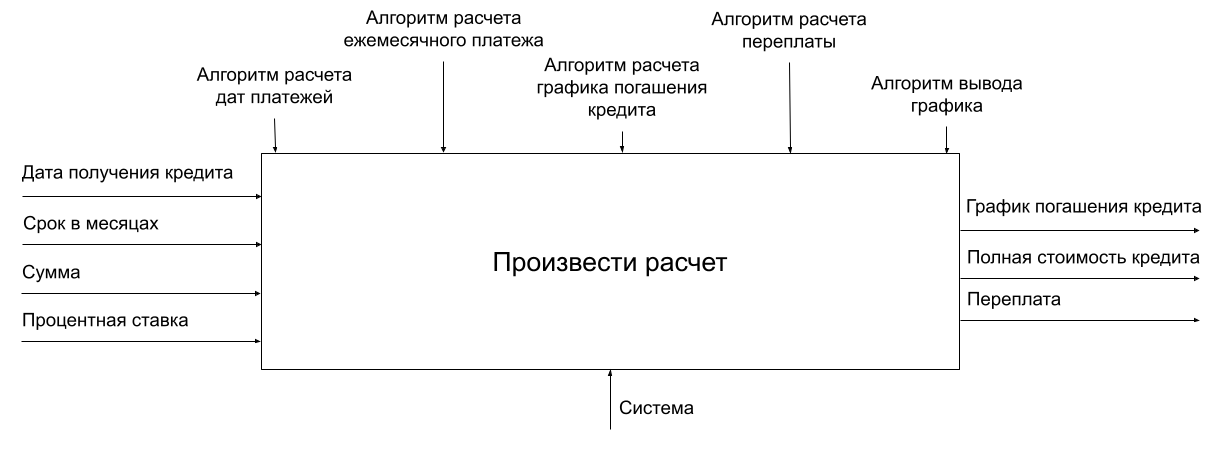


Рисунок 4 – Контекстная диаграмма IDEF0

Дополняющая рисунок 4, детализирующая диаграмма IDEF0 представлена на рисунке 5.

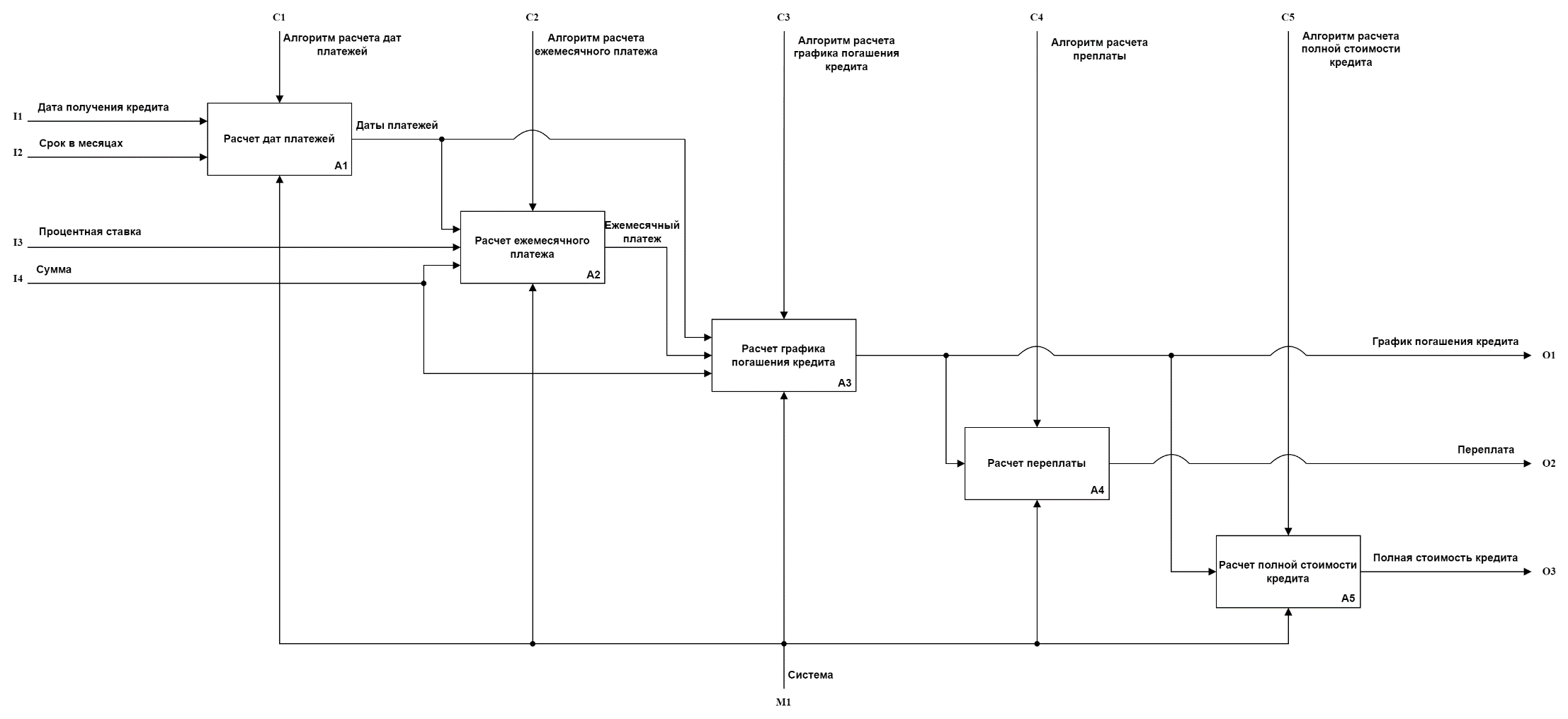


Рисунок 4 – Детализирующая диаграмма IDEF0

Диаграмма IDEF3, описывающая сценарий процесса А1 (Расчет дат платежей) представлена на рисунке 5.

Так как кредит выдается на срок в месяцах, а платежи осуществляются один раз в месяц, то работа А1.1.1 (Расчет даты следующего платежа) выполняется смещением текущей даты на один месяц вперед.

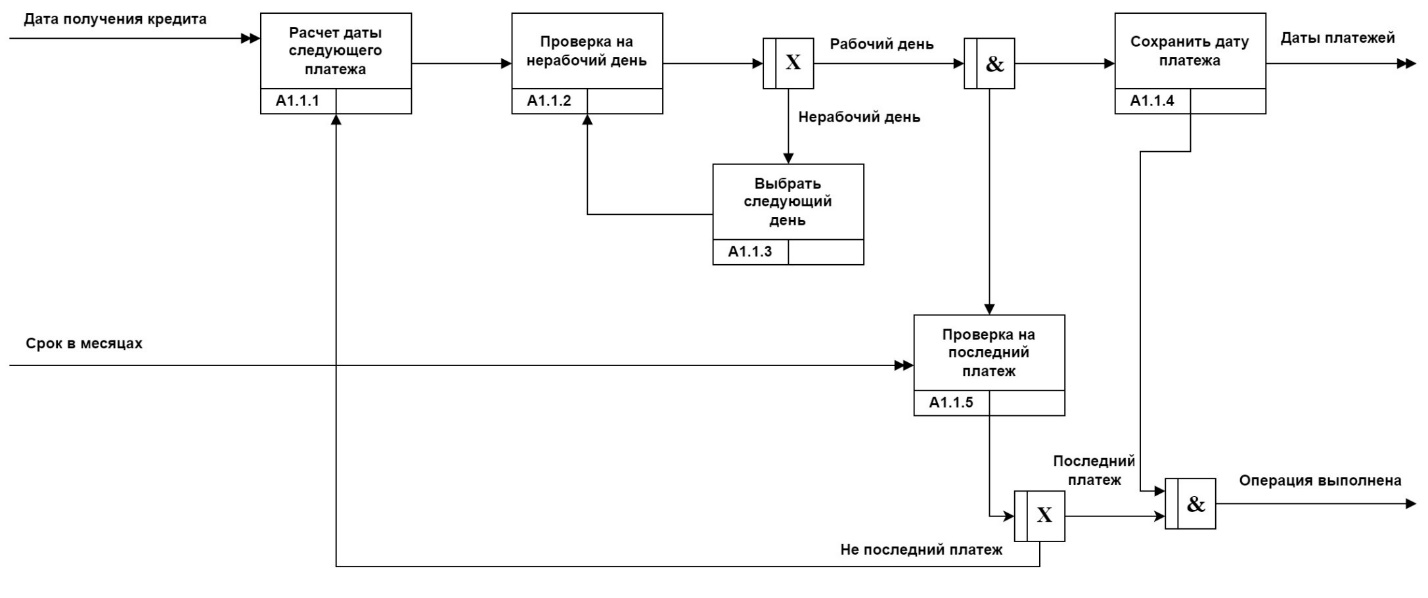


Рисунок 5 – Расчет дат платежей

Диаграмма IDEF3, описывающая сценарий процесса А2 (Расчет ежемесячного платежа) представлена на рисунке 6.

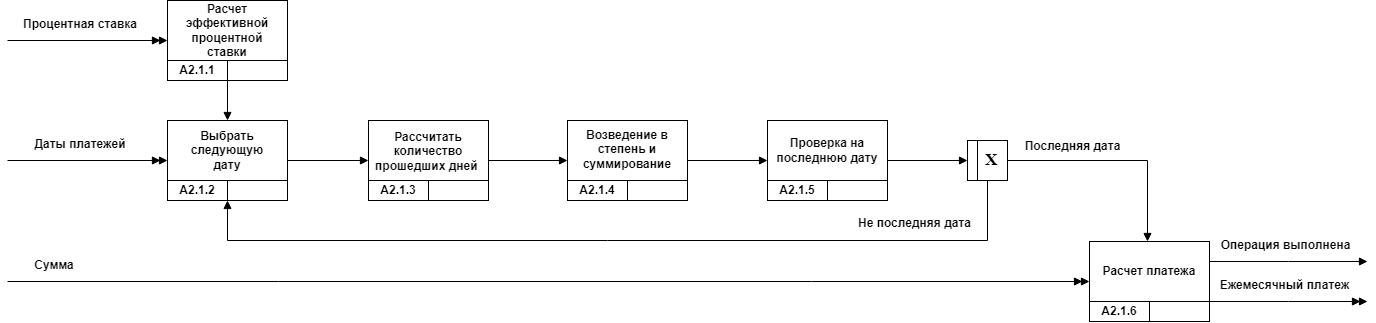


Рисунок 6 – Расчет ежемесячного платежа

Расчет эффективной процентной ставки I, в процессе А2.1.1, производится по формуле: I = i \ 365, где i – номинальная процентная ставка. Так как согласно ЦБ РФ период конвертации процентов равен 1 дню, а продолжительность календарного года признается равной 365 дням [2].

Для расчета ежемесячного платежа аннуитета, выплачиваемого с меньшей частотой, чем начисляются проценты, в теории процента существует формула. Она имеет следующий вид:

где k – количество периодов конвертации процентов за 1 платежный период, n – срок аннуитета измеряемый в периодах конвертации процента, i – эффективная процентная ставка за период конвертации процентов, S – сумма кредита. Однако она применима только для аннуитетов с постоянным платежным периодом k, для переменного платежного периода необходимо модифицировать формулу расчета параметра А.

Для этого заметим, что формула параметра А является частичной суммой геометрического числового ряда, тогда, при его развертывании, формула будет иметь вид:

теперь, зная что параметр k – переменный, формулу следует привести к виду:

где k – количество периодов конвертации процентов прошедших с последнего платежа, n – срок аннуитета измеряемый в периодах конвертации процента.

Таким образом, мы получаем формулу параметра А для аннуитетов с переменным платежным периодом. При этом не стоит забывать, что первым платежом считается само получение крадита, тогда k1 – это количество периодов конвертации процентов прошедших с даты взятия кредита.

Диаграмма IDEF3, описывающая сценарий процесса А3 (Расчет графика погашения кредита) представлена на рисунке 7.

В работе А3.1.3 для определения количества дней между двумя датами используется их перевод в юлианский день с последующим вычитанием. Формула перевода даты григорианского календаря в юлианский день имеет следующий вид:

где Y – григорианский год, M – григорианский месяц, D – григорианский день.

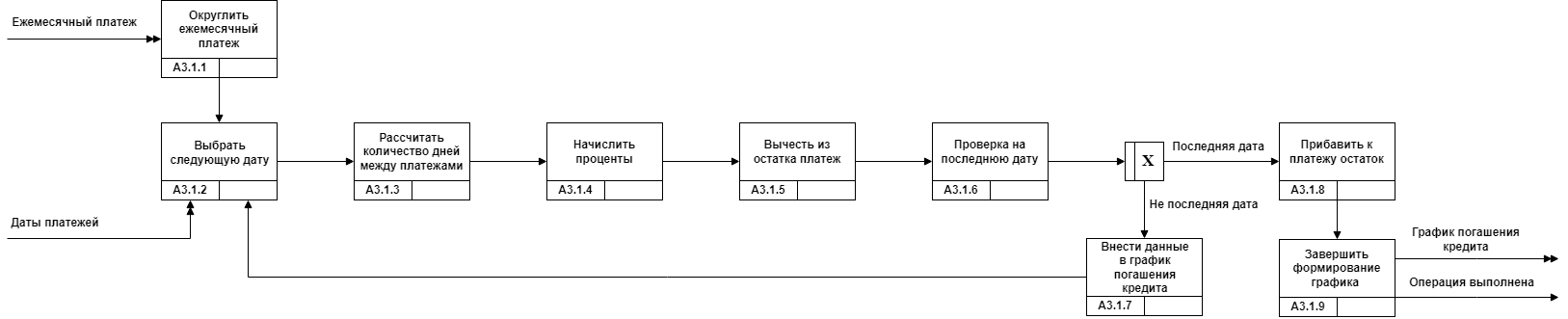


Рисунок 7 – Расчет графика погашения кредита

Диаграмма IDEF3, описывающая сценарий процесса А4 (Расчет переплаты) представлена на рисунке 8.

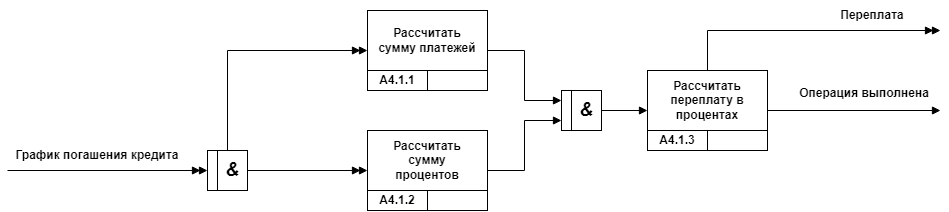


Рисунок 8 – Расчет переплаты

В работе А4.1.3 расчет переплаты с процентах от суммы платежей производится по формуле: G = (P \* 100) \ R, где P – сумма платежей, R – сумма процентов.

Диаграмма IDEF3, описывающая сценарий процесса А5 (Расчет ПСК) представлена на рисунке 9.

Порядок расчета ПСК участниками финансового рынка устанавливается ЦБ РФ [2]. ПСК определяется в процентах годовых по формуле:

где i – процентная ставка базового периода, C – число базовых периодов в календарном году, расчет ПСК необходимо производить с точностью до третьего знака после запятой. Продолжительность календарного года признается ЦБ РФ равной тремстам шестидесяти пяти дням [2].

где В – базовый платежный период (среднее количество периодов конвертации процентов в платежном периоде), Тk - количество периодов конвертации процентов в k-м платежном периоде (денежном потоке), n – количество денежных потоков.

Процентная ставка базового периода i определяется как наименьшее положительное решение уравнения:

где – сумма k-го денежного потока, qk – количество полных базовых периодов с момента выдачи кредита до даты k-го денежного потока, еk – срок, выраженный в долях базового периода, с момента завершения k-го базового периода до даты k-го денежного потока, n – количество денежных потоков, i – процентная ставка базового периода.

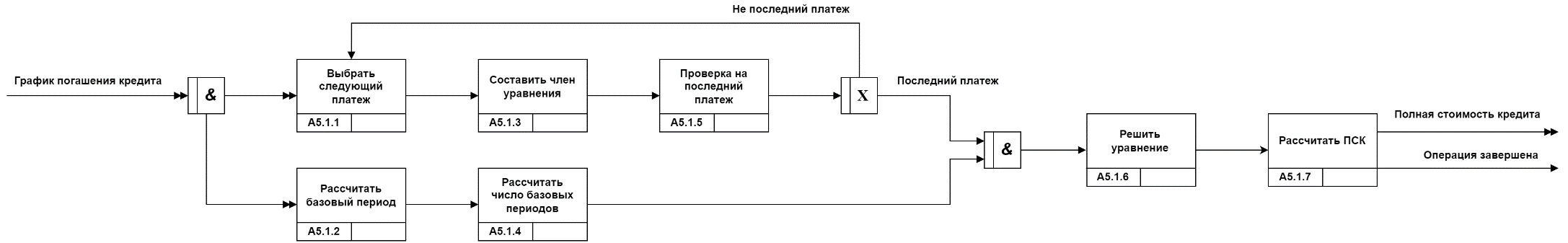


Рисунок 9 – Расчет ПСК

В работе А5.1.6 решение уравнения выполняется методом Regula falsi. Данный метод имеет сверхлинейную сходимость, порядок равен примерно 1,6, и не требует вычисление производной.

Метод Ньютона, хотя и имеет квадратичную сходимость, не используется, так как вычисление производной сложной функции затруднительно и может отнять много времени. Также метод Ньютона, в отличии от Regula falsi, не имеет безусловной сходимости.

При решении одного уравнения с помощью компьютера адекватным выбором является метод деления пополам. Хотя деление пополам выполняется не так быстро, как другие методы, оно имеет гарантированую сходимость с полезной скоростью. Компьютер, используя деление пополам, решит уравнение с желаемой точностью настолько быстро, что нет необходимости экономить время, используя менее надежный метод [3].

Если компьютерной программе приходиться решать уравнения очень много раз. То время, сэкономленное более быстрыми методами, может быть значительным [3]. Исходя из этого использование метода Regula falsi является обоснованным, так как он не уступает в надежности методу деления пополам, но превосходит его в скорости сходимости. Формула метода Regula falsi имеет вид:

где – интервал изоляции, f – функция.

1. Пример расчета

В данном разделе расположен пример расчета полной стоимости кредита.

Исходные данные: сумма кредита S – 15000; срок в месяцах n – 4; процентная ставка I – 0,05; дата взятия кредита – 03.10.2022.

График погашения кредита представлен в таблице 1.

Таблица 2 – График погашения кредита

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| № | Дата | Период | Сумма | Осн. долг | Проценты | Остаток |
| 0 | 03.10.2022 | 0 | 0 | 0 | 0 | 15000 |
| 1 | 03.11.2022 | 31 | 3790,62 | 3726,8 | 63,82 | 11273,2 |
| 2 | 05.12.2022 | 32 | 3790,62 | 3741,1 | 49,52 | 7532,1 |
| 3 | 09.01.2023 | 35 | 3790,62 | 3754,43 | 36,19 | 3777,67 |
| 4 | 03.02.2023 | 25 | 3790,66 | 3777,67 | 12,99 | 0 |
| Q | 123 | 123 | 15162,52 | 15000 | 162,52 | 0 |

Расчет нормативной процентной ставки:

Расчет ежемесячного платежа:

Расчет базового платежного периода:

B = ( / 4 = 30,75

Расчет процентной ставки базового периода:

Решаем уравнение и получаем: i = 0,004324178;

Расчет числа базовых периодов в календарном году:

Расчет ПСК:

Таким образом были разработаны алгоритмы для полного анализа аннуитетного кредита.

1. Разработка структуры программы

В данном разделе описана общая структура приложения, описаны модули и, как они взаимодействуют между собой. Для визуализации структуры приложения будет использован язык UML.

UML — язык графического описания для объектного моделирования в области разработки программного обеспечения. Он позволяет разработчикам программного обеспечения достигнуть соглашения в графических обозначениях для представления общих понятий и больше сконцентрироваться на проектировании.

Диаграмма вариантов использования, описывающая типичное взаимодействие между пользователем и системой, представлена на рисунке 10.



Рисунок 10 – Диаграмма вариантов использования

Диаграмма последовательности, отражающая поток событий, происходящих в рамках варианта использования, представлена на рисунке 11.

На диаграмме кооперации представлена вся та информация, которая есть и на диаграмме последовательности, но кооперативная диаграмма подругому описывает поток событий. Из нее легче понять связи между объектами, однако, труднее уяснить последовательность событий. Диаграмма кооперации представлена на рисунке 12.

Исходя из всего вышесказанного, в приведенных диаграммах была определена структура программы.

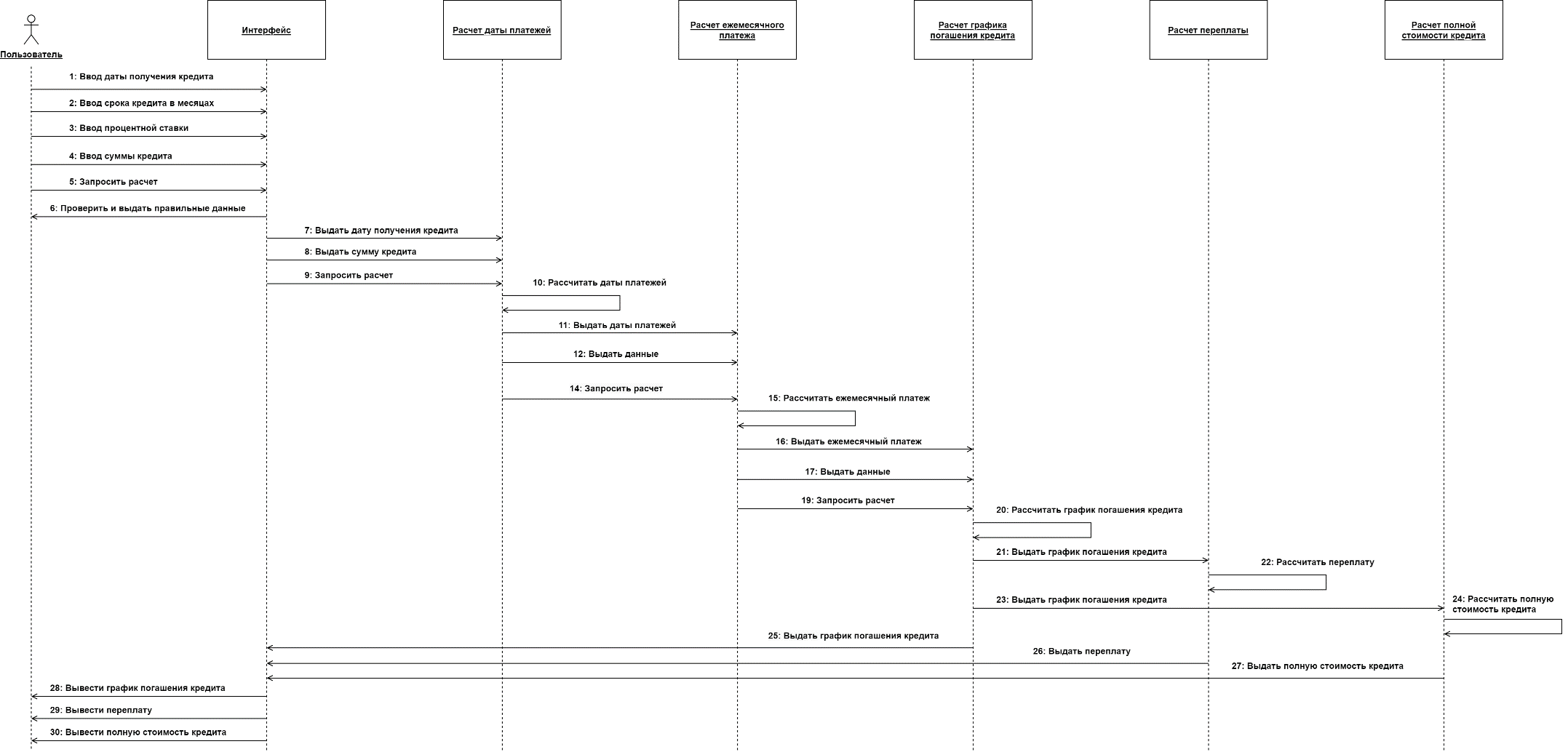


Рисунок 11 – Диаграмма последовательности

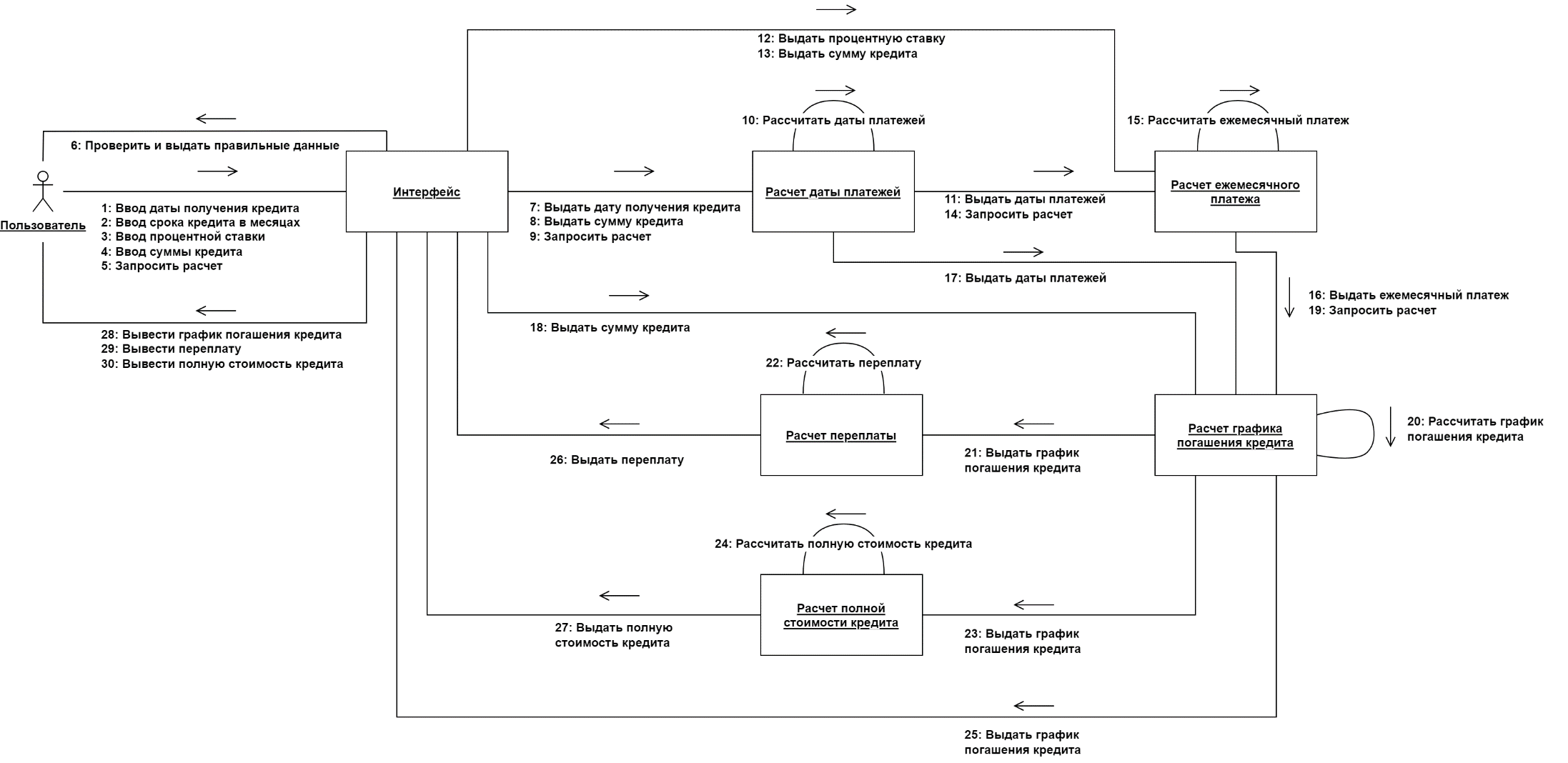


Рисунок 12 – Диаграмма кооперации

1. Разработка интерфейса пользователя

Взаимодействие пользователей с прикладным программным обеспечением, входящим в состав системы должно осуществляться посредством визуального графического интерфейса. Интерфейс системы должен быть понятным и удобным, не должен быть перегружен графическими элементами. Средства редактирования информации должны удовлетворять принятым соглашениям в части использования функциональных клавиш, режимов работы, поиска, использования оконной системы. Ввод-вывод данных системы, прием управляющих команд и отображение результатов их исполнения должны выполняться в интерактивном режиме. Интерфейс должен соответствовать современным эргономическим требованиям и обеспечивать удобный доступ к основным функциям и операциям системы.

Интерфейс должен быть рассчитан на преимущественное использование манипулятора типа «мышь», то есть управление системой должно осуществляется с помощью кнопок. Клавиатурный режим ввода должен используется главным образом при заполнении и/или редактировании текстовых и числовых полей экранных форм.

Экранные формы должны проектироваться с учетом требований унификации:

– все экранные формы пользовательского интерфейса должны быть выполнены в едином графическом дизайне, с одинаковым расположением основных элементов управления и навигации;

– для обозначения сходных операций должны использоваться сходные графические значки, кнопки и другие управляющие (навигационные) элементы. Термины, используемые для обозначения типовых операций (добавление информационной сущности, редактирование поля данных), а также последовательности действий пользователя при их выполнении, должны быть унифицированы;

– внешнее поведение сходных элементов интерфейса (реакция на наведение указателя «мыши», переключение фокуса, нажатие кнопки) должны реализовываться одинаково для однотипных элементов.

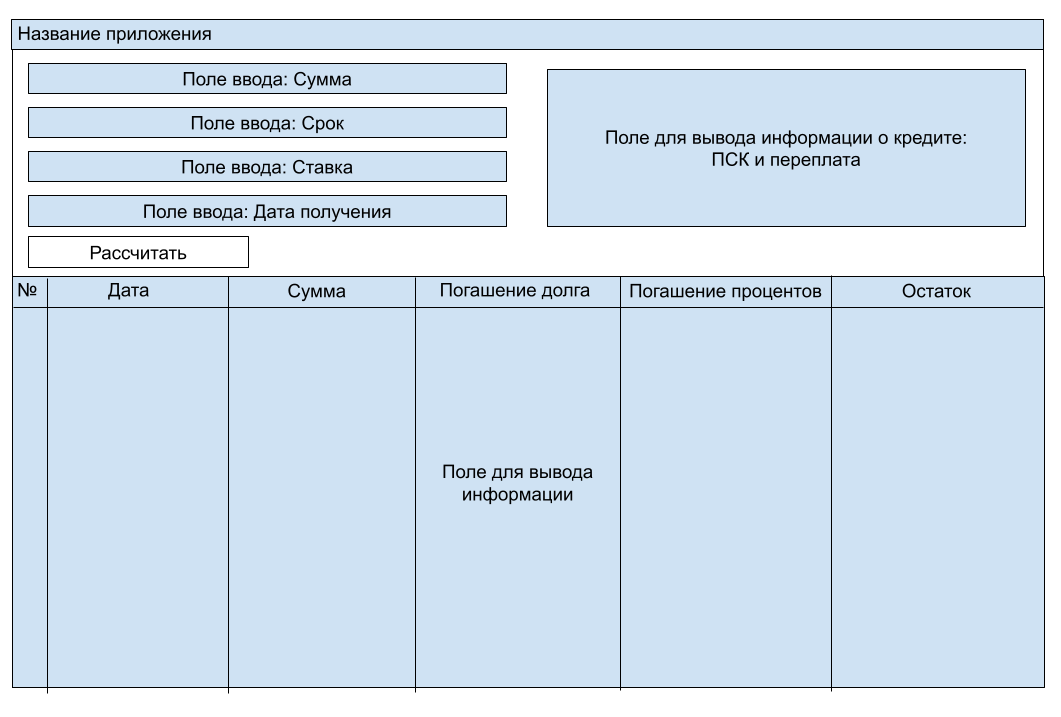


Рисунок 13 – Макет интерфейса приложения

Таким образом, в соответствии со всеми требованиями, был разработан интерфейс пользователя.

1. Программная реализация

В данном разделе описана программная реализация, описаны классы и их взаимодействие. Для визуализации будет использован язык UML. Диаграмма классов представленна на рисунке 15.

В диаграмме классов класс Form зависит от классов TDate и Rounding. Расчет полной стоимости кредита реализуется классом TPSK, при этом TPSK зависит от полей TGraphic и атрибутов Rounding. Класс TGraphic является составляющей частью класса Form.

Класс TGraphic описывает сущность для хранения всей информации о кредите. Граничный класс Form описывает экранную форму приложения. Класс TDate описывает сущность для хранения дат и операций над ними. Класс Rounding реализует операции округления.

Экранная форма приложения для расчета полной стоимости потребительского кредита представленна на рисунке 14. Исходный код программы представлен в приложении Б.

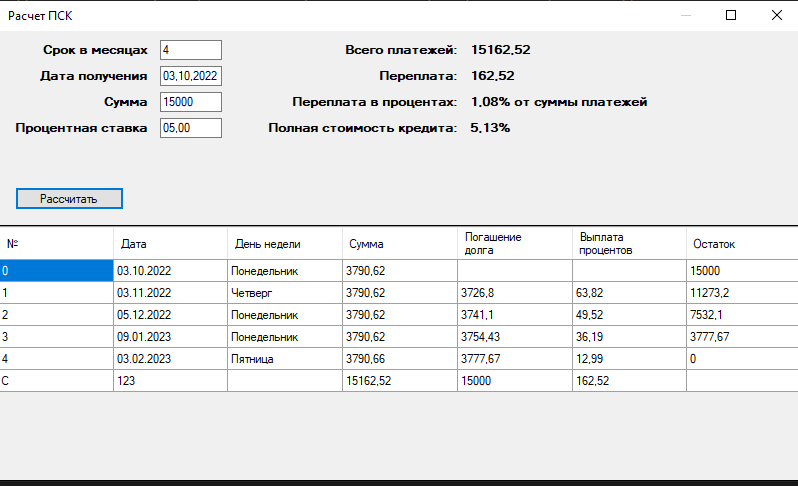


Рисунок 14 – Экранная форма приложения для расчета ПСК

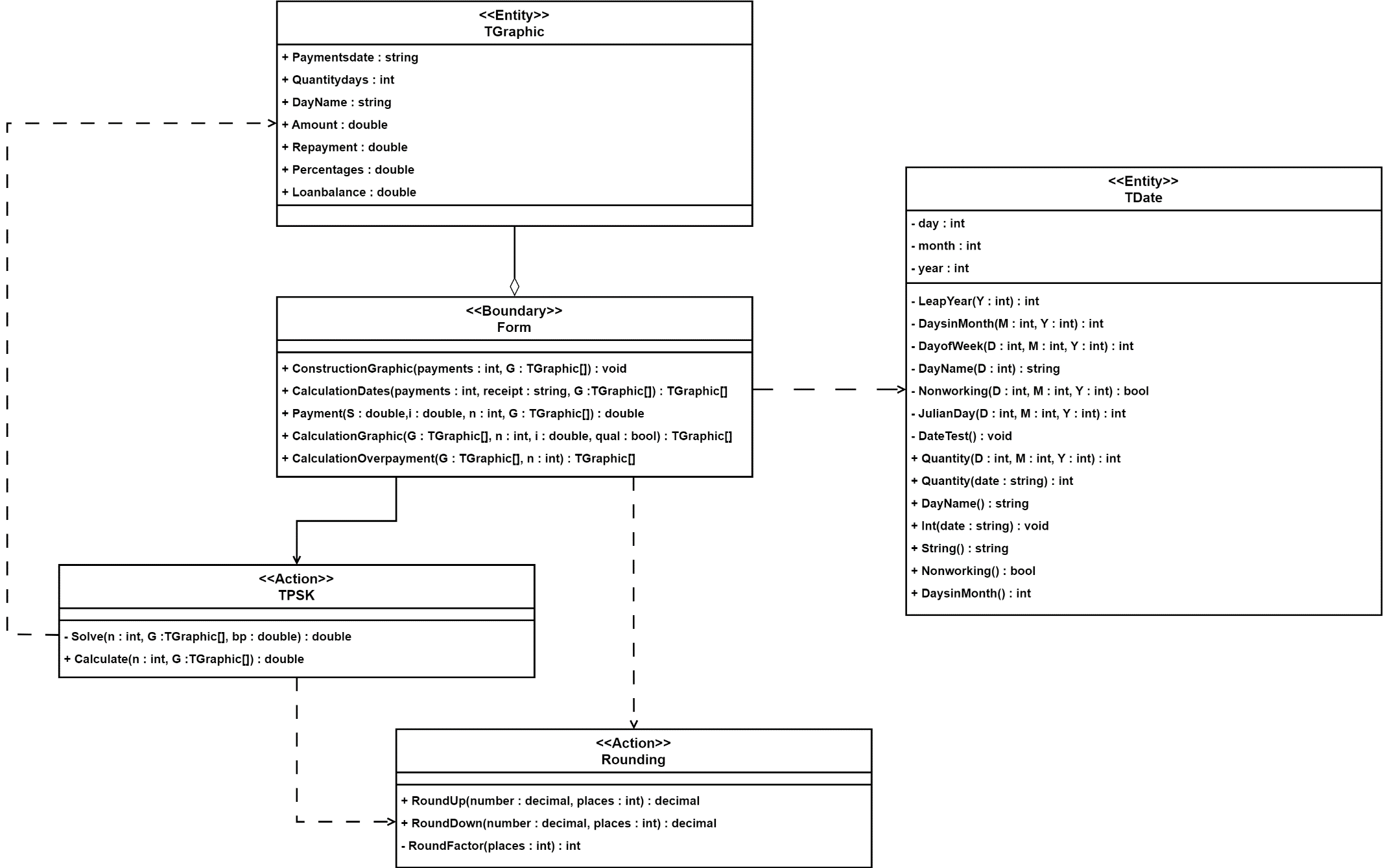


Рисунок 15 – Диаграмма классов

1. Тестирование

Так как алгоритмы, требуемые для расчета ПСК, после создания были согласованы с заказчиком, то осталось установить соответствует ли программа внешним спецификациям. Следовательно, для тестирования была выбрана стратегия «черного ящика» и метод эквивалентного разбиения

На уровне специальных требований возможны следующие эквивалентные разбиения:

1. Срок в месяцах от 1 до 999;
2. Дата получения должна быть существующей;
3. Сумма от 10 до 999999999;
4. Процентная ставка от 0.01 до 0.99

Выделение классов эквивалентности представлено в таблице 3.

Таблица 3 – Выделение классов эквивалентности

|  |  |  |
| --- | --- | --- |
| Входные условия | Правильные классы эквивалентности | Неправильные классы эквивалентности |
| Срок в месяцах | 1-999(1) | >999(2)  <1(3) |
| День | 1-31(4) | <1(5)  >31(6) |
| Месяц | 1-12(7) | <1(8)  >12(9) |
| Год | 0-9999(10) | <1(11)  >9999(12) |
| Сумма | 10-999999999(13) | <10(14)  >999999999(15) |
| Процентная ставка | 0.01-0.99(16) | <0.01(17)  >0.99(18) |

Построение тестов:

1. DIM 1: Срок в месяцах = 4; Дата = 03.10.2022; Сумма = 15000; Процентная ставка = 0.05. Покрывает тесты: 1, 4, 7, 10, 13, 16;
2. DIM 2: Срок в месяцах = 0; Дата = 0.0.0; Сумма = 0; Процентная ставка = 0. Покрывает тесты: 3, 5, 8, 11, 14, 17;
3. DIM 3: Срок в месяцах = 1000; Дата = 33.13.10000; Сумма = 1000000000; Процентная ставка = 1. Покрывает тесты: 2, 6, 9, 12, 15, 18.

Выводы по тестированию: в ходе тестирования ошибок не выявлено.

Заключение

В ходе данного курсового проекта был реализован исходный код приложения, рассчитывающего полную стоимость потребительского, аннуитетного кредита с переменным платежным периодом, в соответствии с указаниями центрального банка РФ и требованиями заказчика.

Был осуществлён анализ предметной области, ознакомление с устройством современной банковской системы и методами индивидуального кредитования. Также был произведен обзор, анализ и сравнение существующих аналогов.

В процессе работы был исследован математический аппарат необходимый для разработки алгоритма расчета полной стоимости потребительского, аннуитетного кредита с переменным платежным периодом. С его помощью был разработан алгоритм расчета полной стоимости потребительского, аннуитетного кредита с переменным платежным периодом, соответствующего указаниям центрального банка РФ и требованиям заказчика.

Перечень сокращений

КБ – коммерческий банк

ПСК – полная стоимость кредита

ЦБ РФ – Центральный банк Российской Федерации

Приложение А

(справочное)

Список используемой литературы

1. Деньги, кредит, банки: кафедра Финансов и кредитования / Никитин В.М., Юдина И.Н.; Барнаул: Издательство «Азубка», 2004.

2. Федеральный закон от 21.12.2013 N 353-ФЗ (последняя редакция от 24.07.2023) "О потребительском кредите (займе)".

3. Elementary Numerical Analysis: an algorithmic approach (3rd ed.) / Conte, S.D., Boor, Carl de; McGraw-Hill Book Company, 1980; <https://epdf.tips/elementary-numerical-analysis-an-algorithmic-approach-5ea6d2e17561c.html>

Приложение Б

(обязательно)

Исходный код программы

using System;

public struct TGraphic

{

public string Paymentsdate; // дата платежа

public int Quantitydays; // количество дней прошедших с даты получения кредита

public string DayName; // название дня

public double Amount; // сумма платежа

public double Repayment; // переплата

public double Percentages; // проценты

public double Loanbalance; // остаток кредита

}

public class TPSK

{

private static double Solve(int n, TGraphic[] G, double bp) // составление и решение уравнения

{

double function(double x) // составление уравнения

{

double y = G[0].Loanbalance;

double Q = 1;

int q = 0;

for (int j = 1; j <= n; j++)

{

int tmp = (int)Rounding.RoundDown((decimal)(G[j].Quantitydays / bp), 0);

if (tmp > q)

{

q = tmp;

Q = Q \* (1 + x);

}

double e = (G[j].Quantitydays - G[j - 1].Quantitydays - bp) / bp;

y = y + (G[j].Amount / ((1 + e \* x) \* Q));

}

return y;

}

bool Sign(double x) // определение знака числа

{

if (x >= 0) { return true; }

else { return false; }

}

bool interval(double x, double y) // определение присутсвия корня

{

double fx = function(x);

double fy = function(y);

if (Sign(fx) != Sign(fy)) { return true; }

else { return false; }

}

bool IIsola(double x, double y) // определение интервала изоляции

{

double f1x = (function(x + 0.1) - function(x - 0.1)) / 0.2;

double f1y = (function(y + 0.1) - function(y - 0.1)) / 0.2;

if (Sign(f1x) != Sign(f1y)) { return false; }

else { return true; }

}

double a = 0;

double b = 1;

if ((IIsola(a, b) == false) || (interval(a, b) == false))

{

while (interval(a, b) == false)

{

b = b + 1;

}

while (IIsola(a, b) == false)

{

b = b - 0.01;

}

}

do {

double c = (a \* function(b) - b \* function(a)) / (function(b) - function(a));

if (interval(a, c) == true) { b = c; }

else { a = c; }

} while (Math.Abs(a - b) > 2 \* 0.0001);

return a;

}

public static double Calculate(int n, TGraphic[] G) // вычисление полной стоимости кредита

{

double bp = (double)G[n].Quantitydays / n;

double cbp = 365 / bp;

double i = Solve(n, G, bp);

return cbp \* i \* 100;

}

}

namespace PSK

{

public partial class Form1 : Form // класс формы

{

public Form1() // конструктор формы

{

InitializeComponent();

}

void СonstructionGraphic(int payments, TGraphic[] G) // вывод графика погашения кредита на экран

{

Grid1.Rows.Clear();

Grid1.Rows.Add(0, G[0].Paymentsdate, G[0].DayName, G[0].Amount, "", "", -G[0].Loanbalance);

for (int i = 1; i <= payments; i++)

{

Grid1.Rows.Add(i, G[i].Paymentsdate, G[i].DayName, G[i].Amount, G[i].Repayment, G[i].Percentages, G[i].Loanbalance);

}

Grid1.Rows.Add("C", G[payments].Quantitydays, "", Rounding.RoundUp((decimal)G[payments + 1].Amount, 2), Rounding.RoundUp((decimal)G[payments + 1].Repayment, 2), G[payments + 1].Percentages, "");

}

TGraphic[] СalculationDates(int payments, ref string receipt, TGraphic[] G) // расчет дат погашения кредита

{

TDate Original = new TDate(receipt);

receipt = Original.String();

TDate Next = new TDate(receipt);

G[0].Paymentsdate = Original.String();

G[0].DayName = Original.DayName();

for (int i = 1; i <= payments; i++)

{

Next.day = Original.day;

Next.month = Next.month + 1;

if (Next.month == 13)

{

Next.year = Next.year + 1;

Next.month = 1;

}

int tmp = Next.DaysinMonth();

if (Next.day > tmp) { Next.day = Next.day - (Next.day - tmp); }

while (Next.Nonworking())

{

if (Next.DaysinMonth() > Next.day) { Next.day++; }

else

{

Next.day = 1;

if (Next.month != 12) { Next.month++; }

else

{

Next.month = 1;

Next.year++;

}

}

}

G[i].Paymentsdate = Next.String();

G[i].DayName = Next.DayName();

G[i].Quantitydays = Original.Quantity(Next.day, Next.month, Next.year);

}

return G;

}

double Payment(double S, double i, int n, TGraphic[] G) // расчет ежемесячного платежа

{

double I = i / 365;

double V = 1 / (1 + I);

double A = 0;

for (int j = 1; j <= n; j++)

{

double tmp = Math.Pow(V, G[j].Quantitydays);

A = A + tmp;

}

return S / A;

}

TGraphic[] СalculationGraphic(TGraphic[] G, int n, double i, out bool qual) // расчет графика погашения кредита

{

TGraphic[] Original = G;

qual = true;

TDate date = new TDate();

double payment = (double)Rounding.RoundDown((decimal)G[0].Amount, 2);

i = i / 365;

for (int j = 1; j <= n - 1; j++)

{

date.Int(G[j - 1].Paymentsdate);

double tmp = G[j - 1].Loanbalance \* Math.Pow(i + 1, date.Quantity(G[j].Paymentsdate));

G[j].Amount = payment;

G[j].Percentages = (double)Rounding.RoundDown((decimal)(tmp - G[j - 1].Loanbalance), 2);

G[j].Repayment = (double)Rounding.RoundDown((decimal)(payment - G[j].Percentages), 2);

if (payment - G[j].Percentages < 0)

{

qual = false;

return Original;

}

G[j].Loanbalance = (double)Rounding.RoundDown((decimal)(tmp - payment), 2);

}

date.Int(G[n - 1].Paymentsdate);

double tmp2 = G[n - 1].Loanbalance \* Math.Pow(i + 1, date.Quantity(G[n].Paymentsdate)) + ((G[0].Amount - payment) \* n);

G[n].Amount = (double)Rounding.RoundUp((decimal)tmp2, 2);

G[n].Percentages = (double)Rounding.RoundUp((decimal)(tmp2 - G[n - 1].Loanbalance), 2);

G[n].Repayment = (double)Rounding.RoundUp((decimal)(tmp2 - G[n].Percentages), 2);

if (tmp2 - G[n].Percentages < 0)

{

qual = false;

return Original;

}

G[n].Loanbalance = 0;

G[0].Amount = payment;

return G;

}

TGraphic[] CalculationOverpayment(TGraphic[] G, int n) // расчет переплаты

{

G[n + 1].Repayment = G[0].Loanbalance;

G[0].Loanbalance = -G[0].Loanbalance;

for (int j = 1; j <= n; j++)

{

G[n + 1].Amount = G[n + 1].Amount + G[j].Amount;

G[n + 1].Percentages = G[n + 1].Percentages + G[j].Percentages;

}

labelPayment.Text = Convert.ToString((double)Rounding.RoundDown((decimal)G[n + 1].Amount, 2));

labelRepayment.Text = Convert.ToString((double)Rounding.RoundDown((decimal)G[n + 1].Percentages, 2));

labelPercent.Text = Convert.ToString((double)Rounding.RoundDown((decimal)((G[n + 1].Percentages \* 100) / G[n + 1].Repayment), 2)) + "% от суммы платежей";

return G;

}

private void btnCalculate\_Click(object sender, EventArgs e) // реакция на нажатие кнопки btn

{

if (int.TryParse(TextBox1.Text, out int n) == false) { n = 0; }

if (double.TryParse(TextBox3.Text, out double S) == false) { S = 0; }

if (double.TryParse(TextBox4.Text, out double i) == false) { i = 0; }

i = i / 100;

TGraphic[] Graphic = { };

bool quality = false;

n = n + 1;

while (quality == false)

{

n = n - 1;

Graphic = new TGraphic[n + 2];

string DateTest = TextBox2.Text;

Graphic = СalculationDates(n, ref DateTest, Graphic);

TextBox2.Text = DateTest;

Graphic[0].Amount = Payment(S, i, n, Graphic);

Graphic[0].Loanbalance = S;

Graphic = СalculationGraphic(Graphic, n, i, out quality);

if (quality == false) { continue; }

Graphic = CalculationOverpayment(Graphic, n);

labelPSK.Text = Convert.ToString((double)Rounding.RoundDown((decimal)TPSK.Calculate(n, Graphic), 2)) + "%";

СonstructionGraphic(n, Graphic);

}

}

}

}