

Министерство науки и высшего образования РФ  
Федеральное государственное образовательное учреждение  
высшего образования  
«УФИМСКИЙ ГОСУДАРСТВЕННЫЙ АВИАЦИОННЫЙ ТЕХНИЧЕСКИЙ  
УНИВЕРСИТЕТ»

Кафедра автоматизированных систем управления

Направление подготовки  
09.03.03 Прикладная информатика

**ПОЯСНИТЕЛЬНАЯ ЗАПИСКА**

к курсовой работе по дисциплине «Информационные системы»

«Разработка кроссплатформенного программного продукта на языке JAVA с  
использованием системы контроля версий»

Выполнили:  
Ст. гр. ПИ-223  
Грищенко И.С.  
Мочалина А.С.  
Маломожнова В.М.

Проверил:  
Преподаватель  
Казанцев А.В.

Уфа 2021

# АННОТАЦИЯ

В курсовой работе рассматривается разработка кроссплатформенного программного продукта на языке JAVA.

В первом разделе приведено описание предметной области – ипотечное кредитование.

Во втором разделе представлено “Техническое задание”.

В третьем разделе описан процесс настройки семейств систем windows и linux для работы.

В четвертом разделе описан процесс настройки среды разработки для подключения к системе контроля версии.

В пятом разделе описано какой разработчик ответственен за зону реализации программного кода.

В шестом разделе представлено описание UNIT-тестов, некоторые файлы и процесс сборки.

В седьмом разделе описана настройка программной среды для развёртывания и запуска программного продукта.

В восьмом разделе описано руководство оператора.

Ссылка на репозиторий: [https://github.com/GrishchenkoIS/Kurs\\_project25](https://github.com/GrishchenkoIS/Kurs_project25)

## СОДЕРЖАНИЕ

АННОТАЦИЯ .....	2
Раздел 1. Описание предметной области.....	4
Раздел 2. Техническое задание .....	10
Раздел 3. Настройка среды разработки для операционных систем семейств windows и linux .....	11
Раздел 4. Настройка среды разработки для подключения к системе контроля версий .....	17
Раздел 5. Реализация исходного кода по зонам ответственности .....	22
Раздел 6. Сборка и тестирование программного продукта .....	23
Раздел 7. Настройка программной среды для развертывания и запуска программного продукта.....	26
Раздел 8. Руководство пользователя программного продукта .....	29

## Раздел 1. Описание предметной области

Для данной курсовой работы, которая включает в себя создание ипотечного калькулятора для юридических лиц, рассмотрим ряд следующих понятий. Ипотека—это одна из форм залога, при которой закладываемое недвижимое имущество остается в собственности должника, а кредитор в случае невыполнения последним своего обязательства приобретает право получить удовлетворение за счет реализации данного имущества.

Ипотекой является также залог существующего недвижимого имущества собственника для получения им кредита или займа, которые будут направлены либо на ремонт или строительство, либо на иные нужды по усмотрению заемщика-залогодателя.

Ипотечное кредитование регулирует:

- 1) Федеральный закон РФ "Об ипотеке (залоге недвижимости)" и отдельные законодательные акты Российской Федерации от 16.07.1998 N 102-ФЗ;
- 2) Жилищный кодекс Российской Федерации;
- 3) Федеральный закон РФ "О кредитных историях".

Для точного расчета платежей ипотеки очень удобно использовать специальную программу, доступную каждому, – ипотечный калькулятор. Эта программа, которая содержит набор математических формул, используется для вычисления всех значимых показателей кредита. Важнейшей функцией программы является расчет ипотеки онлайн. С помощью калькулятора заемщик без труда сможет рассчитать все ключевые условия ипотеки: платежи, сумму ипотеки, переплату, сроки и другие.

Для того, чтобы результат расчета ипотеки, совершаемого на калькуляторе, получился точным, обязательно нужно учитывать такие параметры как процентная кредитная ставка, различные платы и комиссии, которые могут иметь место, а также доступная для заемщика сумма первоначального взноса.

Процентная ставка - очень важный параметр при расчёте ипотеки. Измеряется в процентах годовых. Этот параметр показывает сколько процентов начисляется на ваш долг в год.

Юридическое лицо – это зарегистрированная в установленном законом порядке организация, фирма, компания, которая имеет в собственности, хозяйственном ведении или оперативном управлении обособленное имущество и отвечает по своим обязательствам этим имуществом, может от своего имени приобретать и осуществлять имущественные и личные неимущественные права, нести обязанности, быть истцом и ответчиком в суде.

Каждый предприниматель, прежде чем принять решение о покупке нежилой недвижимости по программе коммерческой ипотеки, решает для себя ряд вопросов, касающихся условий кредитования. Одним из основных является ставка рефинансирования.

Ставка рефинансирования - размер процентов в годовом исчислении, подлежащий уплате Банку России со стороны кредитных организаций за предоставленные кредиты. С помощью ставки рефинансирования Центробанк воздействует на ставки межбанковского рынка, ставки по депозитам и кредитам, в числе прочих методов влияет на инфляцию и т.д.

Чем более низкий процент, под который центральный банк дает деньги коммерческим банкам, тем ниже процент, под который эти банки выдают кредит потребителям.

Чем раньше банк узнает о снижении ставки рефинансирования, тем раньше объявит о снижении ставки на ипотечный кредит и тем больше клиентов сможет привлечь.

Цепочка выглядит так:

- 1) ЦБ объявляет о снижении ставки рефинансирования;
- 2) Банки получают деньги под более низкий процент;
- 3) Банки выдают кредиты под более низкий процент;
- 4) Инфляция падает;
- 5) Заемщики чаще берут кредиты;

- 6) Покупательская способность растет, как и производство;
- 7) Наступает момент, когда покупательская способность превышает производство;
- 8) Растет инфляция;
- 9) Ставка рефинансирования повышается.

На 09.12.2020 рефинансированная ставка Центрального банка равняется 4,25 %.

Аннуитетный платеж – платеж, предполагающий выплату как основного долга, так и сумму начисленных процентов по кредиту. В первые месяцы основную часть составляют проценты.

Дифференцированный платеж – платеж, предполагающий выплату процентов по займу и часть основного долга. Общая сумма делится на весь срок равными частями, в то время как проценты каждый месяц начисляются на остаток.

В первое время выплаты наиболее ощутимы, но становятся минимальными к концу срока.

При взятии кредита необходима выбрать вид платежа. Аннуитетный, при котором переплата оказывается более существенней, или дифференцированный, при котором первые платежи гораздо ощутимее последующих.

В пакет необходимых документов для получения коммерческой ипотеки входят:

- 1) Свидетельство о государственной регистрации;  
Свидетельство о постановке на учет в налоговом органе;
- 2) Карточка с образцами подписей и оттиска печати;
- 3) Документы, подтверждающие полномочия лиц, указанных в карточке с образцами подписей и оттиска печати, на распоряжение денежными средствами, находящимися на счете;

4) Документы, подтверждающие полномочия лиц, указанных в карточке с образцами подписей и оттиска печати, на распоряжение денежными средствами, находящимися на счете;

5) Лицензии, патенты на осуществление деятельности, требующиеся в соответствии с законодательством РФ соответствующего разрешения;

6) Информационная карта Клиента – юридического лица, созданного в соответствии с законодательством Российской Федерации;

7) Выписка из ЕГРЮЛ, выданная налоговым органом не ранее, чем за 30 рабочих дней до заключения договоров с Банком;

8) Кредитная история/

В некоторых случаях банки могут требовать и ряд дополнительных документов, на свое усмотрение.

Схема получения выглядит так: клиент заходит в устройство и вводит данные по кредиту, которые предоставляет банк. Если результат удовлетворяет клиента, то он обращается в банк и оформляет кредит(рис.1).

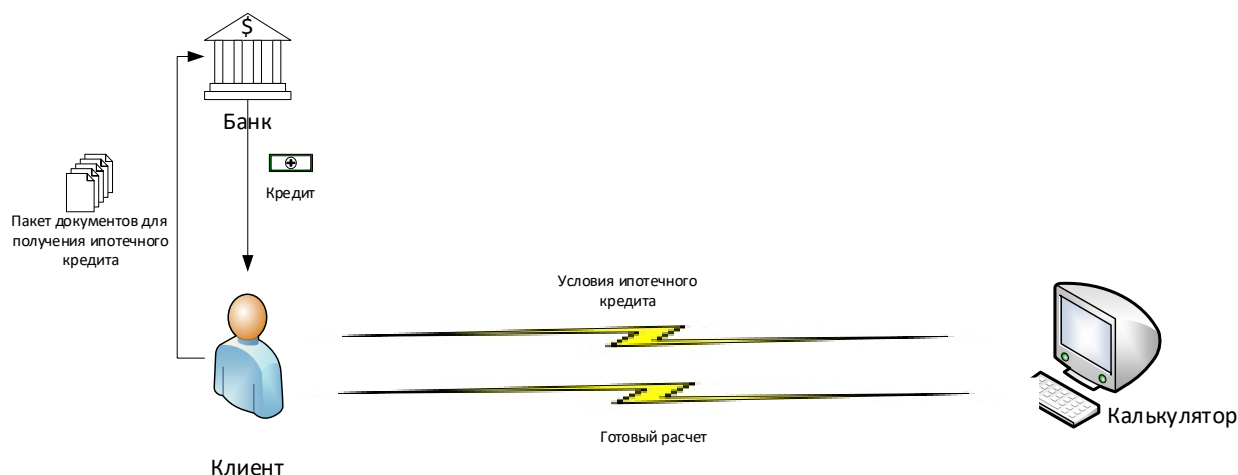


Рис.1. мнемосхема принятия решения об оформлении кредита.

Определение графического изображения для каждого типа информационного взаимодействия мнемосхемы (рис.1). (табл.1)

Таблица 1.

Условное обозначение	Типы взаимодействия
----------------------	---------------------

	электронный
	документальный
	денежный

Взаимодействие пользователя с программой изображено на рисунке 2.

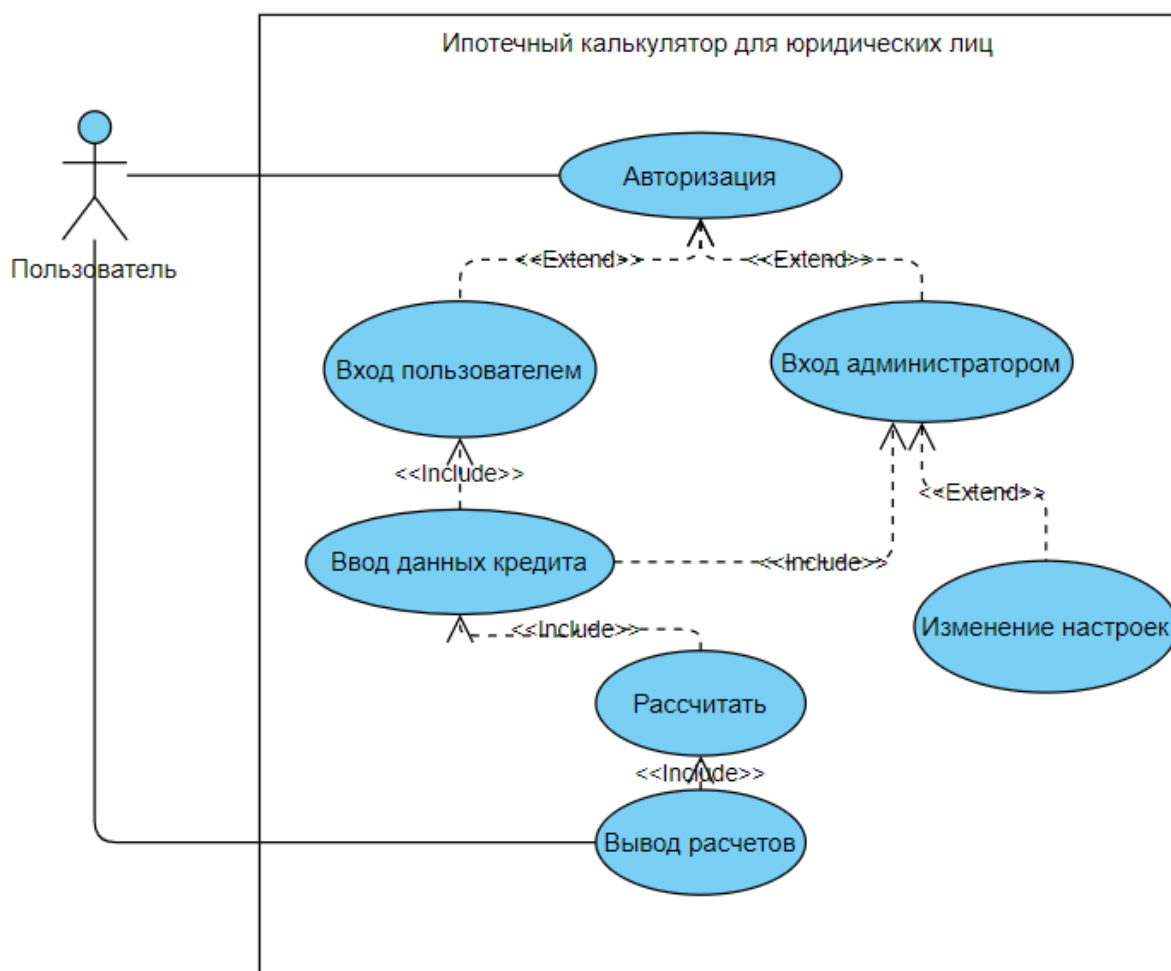


Рис.2. Диаграмма взаимодействия программы ипотечного калькулятора для юридических лиц

Классы, из которых состоит программа изображена на рисунке 3.



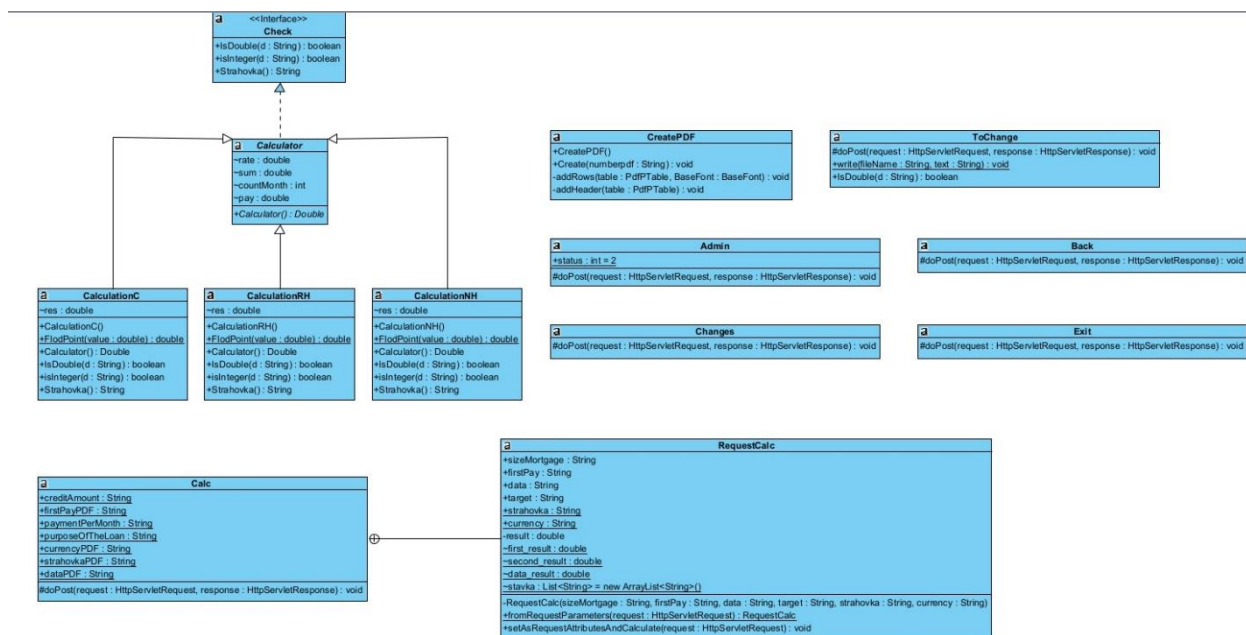


Рис.3. Диаграмма классов программы ипотечного калькулятора для юридических лиц.

## **Раздел 2. Техническое задание**

Техническое задание представлено в приложение 2.

### Раздел 3. Настройка среды разработки для операционных систем семейств windows и linux

В данном разделе описана инструкция по установке и настройке среды разработки для Windows 10, дистрибутивов Linux: Ubuntu 20.04 и OpenSUSE Leap 15.2.

Рассмотрим настройку для windows 10. Необходимо установить следующие инструменты

- 1) Eclipse IDE (Integrated Development Environment) – свободная среда разработки модульных кроссплатформенных приложений.
- 2) JRE (Java Runtime Environment) – минимальная реализация виртуальной машины, необходимая для исполнения Java-приложений, без компилятора и других сред разработки.
- 3) JDK 16 (Java Development Kit) – комплект разработчика на языке Java

Для установки Eclipse IDE необходимо зайти на официальный сайт <https://www.eclipse.org/downloads/> (рис.4).

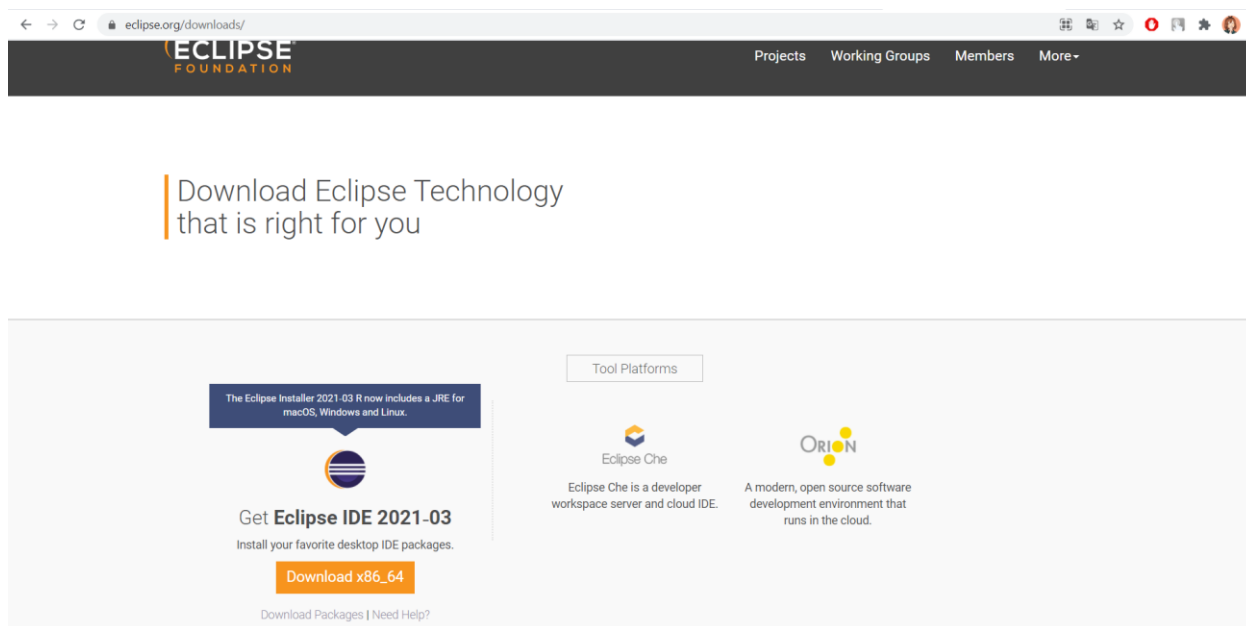


Рис.4. Официальный сайт с установщиком Eclipse IDE

Необходимо нажать на “Download x86\_64” или “Download Packages”. После скачивания установщика запускаем его и выбираем “Eclipse IDE for Java Developers” выбираем путь и нажимаем “INSTALL”.

В установленной Eclipse IDE уже встроен git, JUnit, Maven, поэтому не требуется дополнительная их установка.

Для перехода в проекцию git необходимо нажать на кнопку “open perspective” и выбрать git (рис.5).

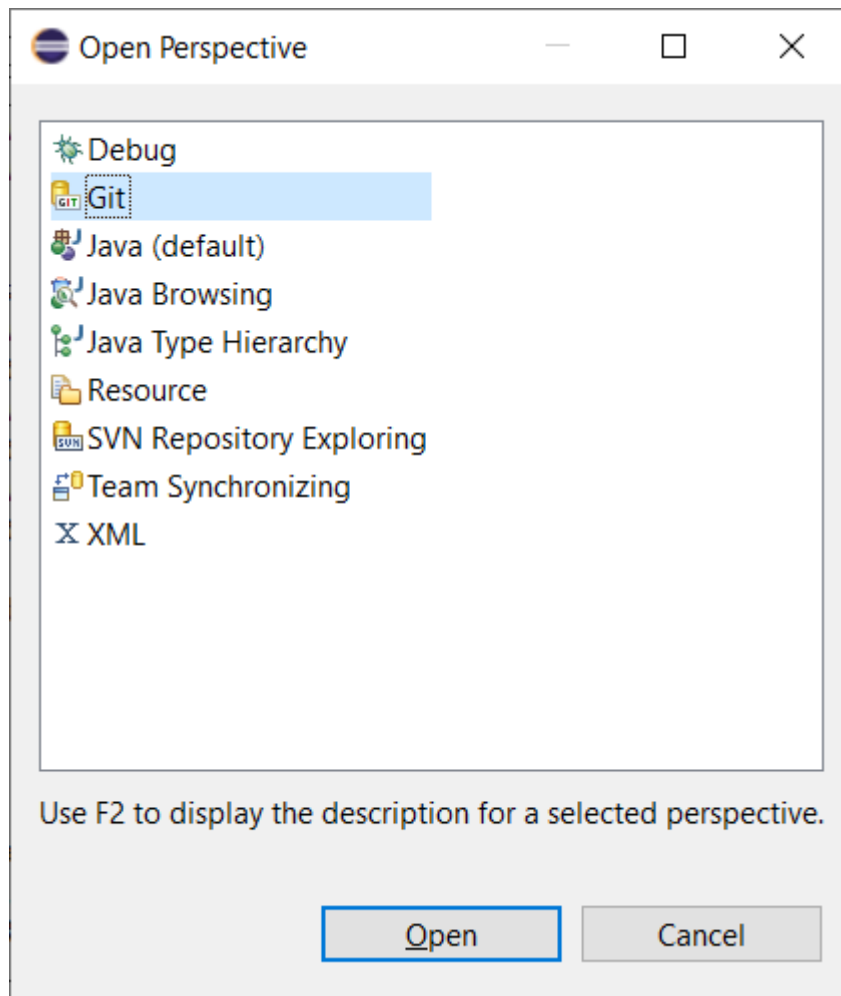


Рис.5. Выбор проекции git

Для создания Maven проекта необходимо выбрать File – New – Project... . В новом окне выбираем Maven – Maven Project. Далее выбираем архитип или simple project. Вводим Group Id и Artifact Id (рис.6). нажимаем Finish

New Maven Project

New Maven project

Configure project

Artifact

Group Id: Project

Artifact Id: Project

Version: 0.0.1-SNAPSHOT

Packaging: jar

Name:

Description:

Parent Project

Group Id:

Artifact Id:

Version: Browse... Clear

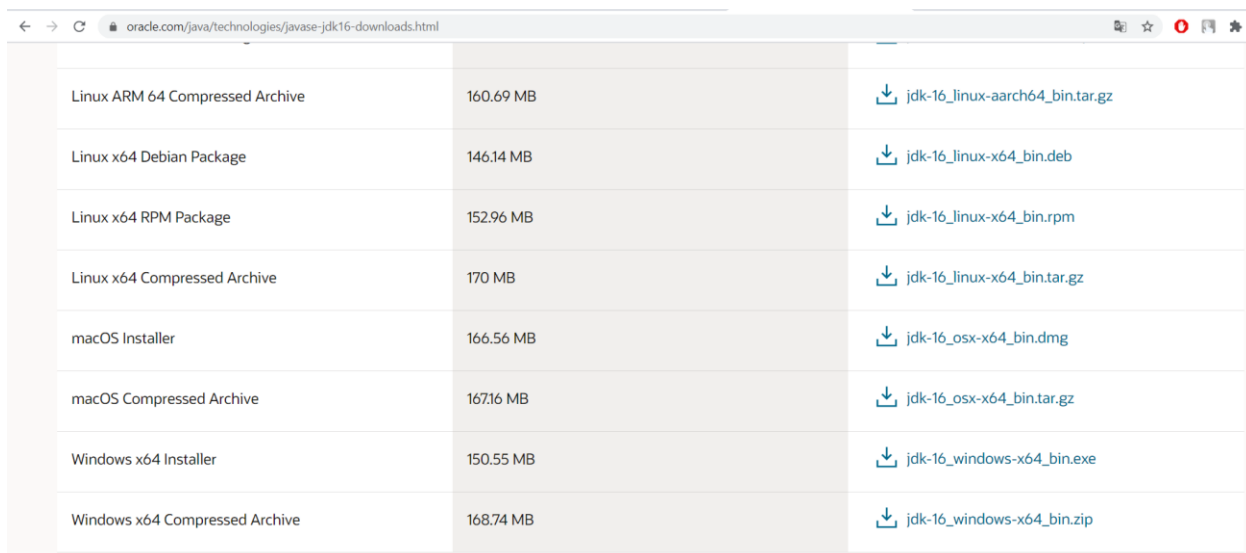
Advanced

? < Back Next > Finish Cancel

Рис.6. Создание Maven project

JRE устанавливается вместе с JDK.

Для установки JDK 16 необходимо перейти на официальный сайт <https://www.oracle.com/java/technologies/javase-jdk16-downloads.html> и выбрать интересующий установщик (рис.7).

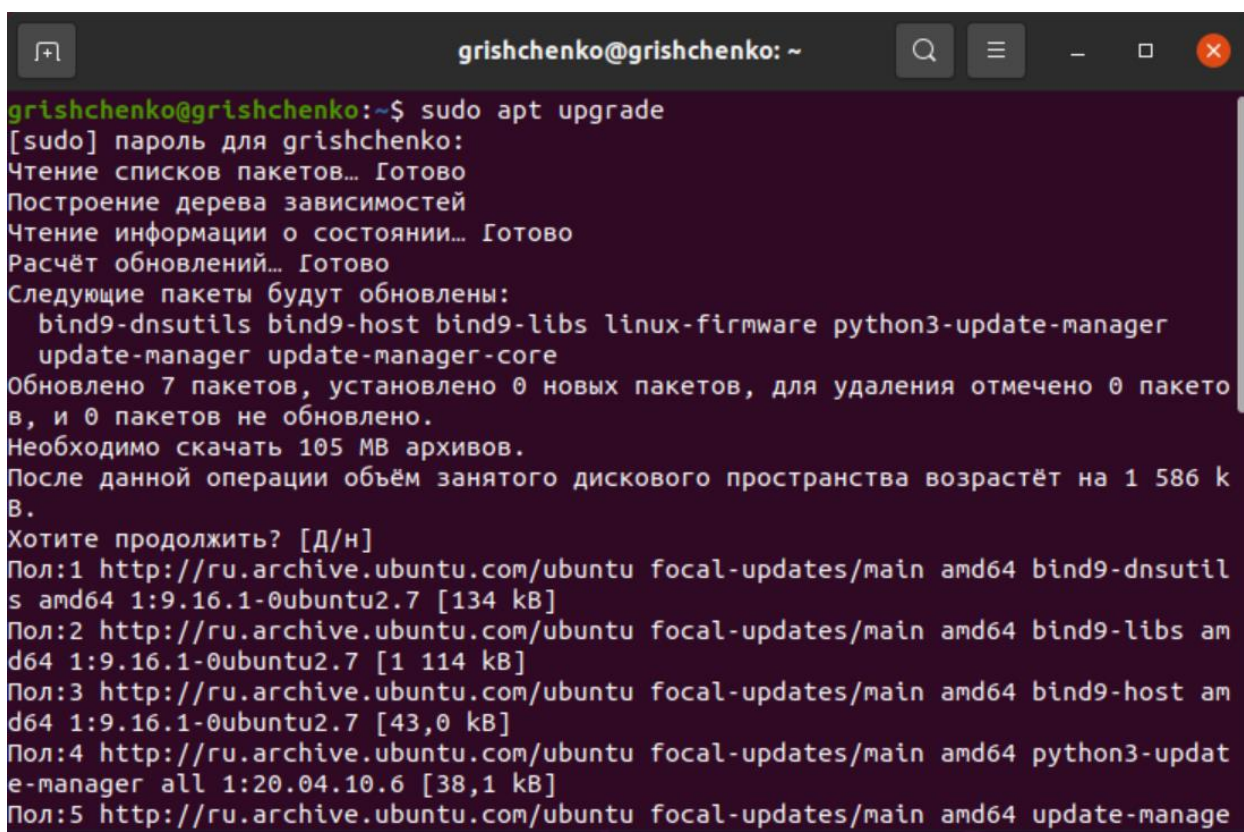


Linux ARM 64 Compressed Archive	160.69 MB	<a href="#">jdk-16_linux-aarch64_bin.tar.gz</a>
Linux x64 Debian Package	146.14 MB	<a href="#">jdk-16_linux-x64_bin.deb</a>
Linux x64 RPM Package	152.96 MB	<a href="#">jdk-16_linux-x64_bin.rpm</a>
Linux x64 Compressed Archive	170 MB	<a href="#">jdk-16_linux-x64_bin.tar.gz</a>
macOS Installer	166.56 MB	<a href="#">jdk-16_osx-x64_bin.dmg</a>
macOS Compressed Archive	167.16 MB	<a href="#">jdk-16_osx-x64_bin.tar.gz</a>
Windows x64 Installer	150.55 MB	<a href="#">jdk-16_windows-x64_bin.exe</a>
Windows x64 Compressed Archive	168.74 MB	<a href="#">jdk-16_windows-x64_bin.zip</a>

Рис.7. официальный сайт с установщиком JDK 16

После скачивания установщика запускаем его, выбираем путь и устанавливаем.

Для работы в дистрибутиве Linux Ubuntu 20.04 с Eclipse IDE необходимо предварительно обновить систему при помощи командой строки и команды *sudo apt upgrade* (рис.8).



```

grishchenko@grishchenko: ~$ sudo apt upgrade
[sudo] пароль для grishchenko:
Чтение списков пакетов... Готово
Построение дерева зависимостей
Чтение информации о состоянии... Готово
Расчёт обновлений... Готово
Следующие пакеты будут обновлены:
  bind9-dnsutils bind9-host bind9-libs linux-firmware python3-update-manager
  update-manager update-manager-core
Обновлено 7 пакетов, установлено 0 новых пакетов, для удаления отмечено 0 пакетов,
и 0 пакетов не обновлено.
Необходимо скачать 105 MB архивов.
После данной операции объем занятого дискового пространства возрастёт на 1 586 k
B.
Хотите продолжить? [Д/н]
Пол:1 http://ru.archive.ubuntu.com/ubuntu focal-updates/main amd64 bind9-dnsutil
s amd64 1:9.16.1-0ubuntu2.7 [134 kB]
Пол:2 http://ru.archive.ubuntu.com/ubuntu focal-updates/main amd64 bind9-libs am
d64 1:9.16.1-0ubuntu2.7 [1 114 kB]
Пол:3 http://ru.archive.ubuntu.com/ubuntu focal-updates/main amd64 bind9-host am
d64 1:9.16.1-0ubuntu2.7 [43,0 kB]
Пол:4 http://ru.archive.ubuntu.com/ubuntu focal-updates/main amd64 python3-updat
e-manager all 1:20.04.10.6 [38,1 kB]
Пол:5 http://ru.archive.ubuntu.com/ubuntu focal-updates/main amd64 update-manage

```

Рис.8. Обновление системы.

Для установки JDK необходимо ввести команду *sudo apt install openjdk-8-jdk* (Рис.9.)

```
Обрабатываются триггеры для mail-ov (2.9.1-1) ...
grishchenko@grishchenko:~$ sudo apt install openjdk-8-jdk
Чтение списков пакетов... Готово
Построение дерева зависимостей
Чтение информации о состоянии... Готово
Уже установлен пакет openjdk-8-jdk самой новой версии (8u282-b08-0ubuntu1~20.04)
.
Обновлено 0 пакетов, установлено 0 новых пакетов, для удаления отмечено 0 пакетов, и 0 пакетов не обновлено.
grishchenko@grishchenko:~$
```

Рис.9. Установка JDK для Ubuntu 20.04

Для установки Eclipse IDE воспользуемся браузером и с официального сайта <https://www.eclipse.org/downloads/> скачаем установщик и повторим процедуру для Eclipse IDE из Windows, запустив eclipse-inst (рис.10).

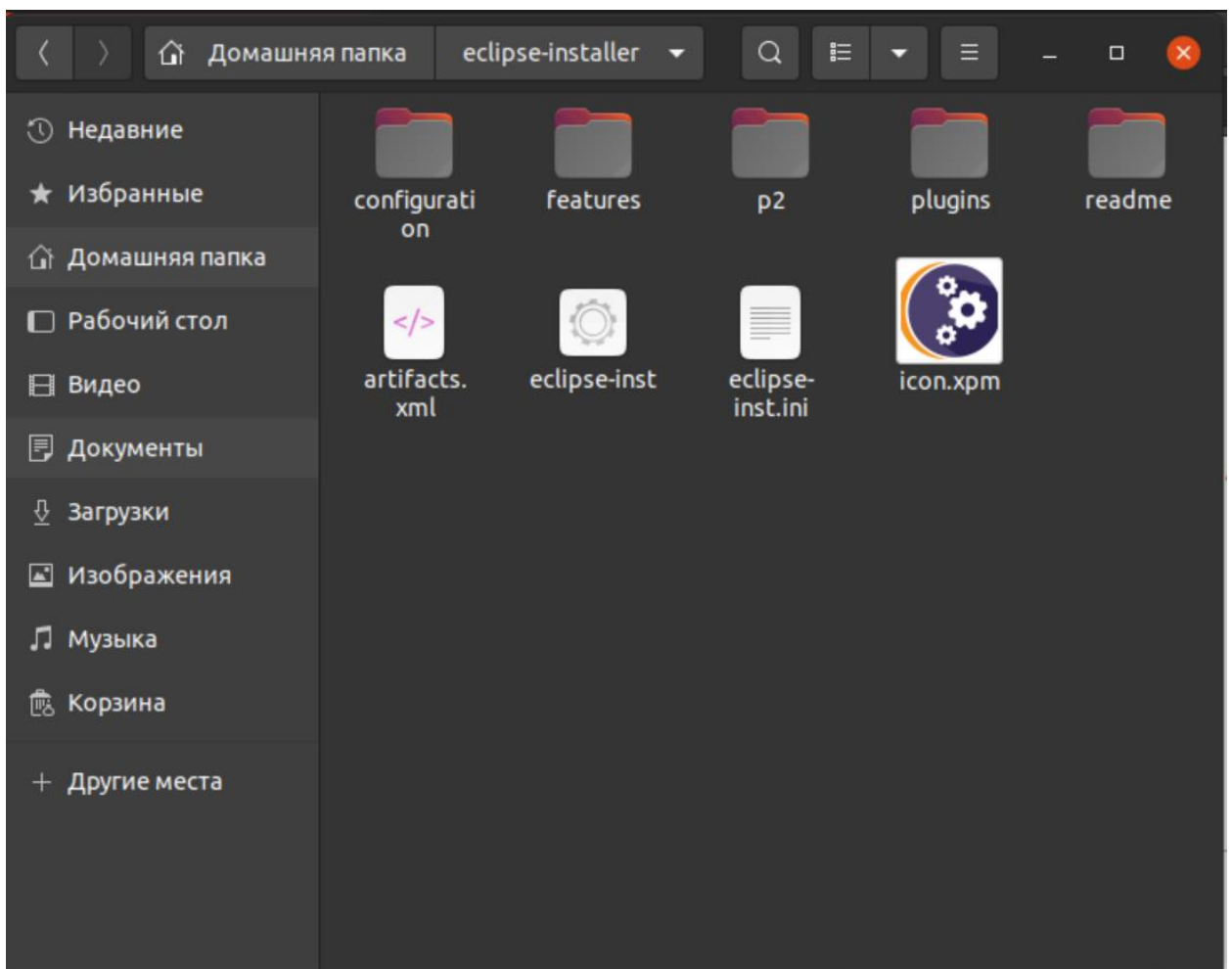
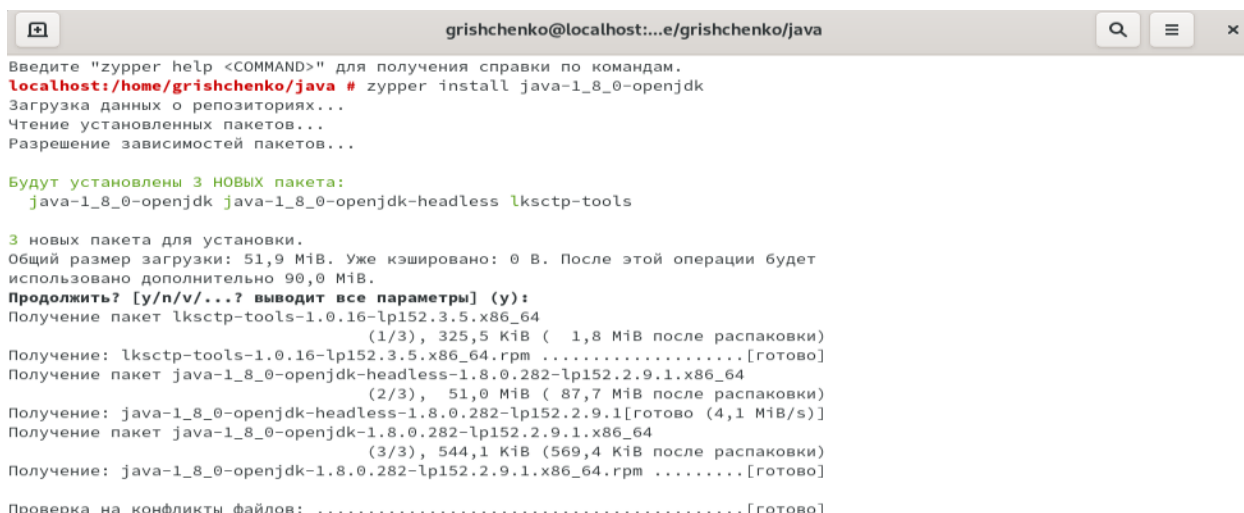


Рис.10. Установщик Eclipse IDE для Ubuntu 20.04

Для работы в дистрибутиве Linux OpenSuse Leap 15.2 с Eclipse IDE необходимо обновить систему с помощью команды *sudo zypper update*.

Для установки JDK необходимо войти в режим администратора при помощи введения команды *su* и дальнейшего заполнения пароля. Ввести команду *zypper install java-1\_8\_0-openjdk* (рис.11)



```
grishchenko@localhost:~$ grishchenko/java
Введите "zypper help <COMMAND>" для получения справки по командам.
localhost:/home/grishchenko/java # zypper install java-1_8_0-openjdk
Загрузка данных о репозиториях...
Чтение установленных пакетов...
Разрешение зависимостей пакетов...

Будут установлены 3 НОВЫХ пакета:
  java-1_8_0-openjdk  java-1_8_0-openjdk-headless  lksctp-tools

3 новых пакета для установки.
Общий размер загрузки: 51,9 MiB. Уже кэшировано: 0 B. После этой операции будет
использовано дополнительно 90,0 MiB.
Продолжить? [y/n/v/...? выводит все параметры] (y):
Получение пакет lksctp-tools-1.0.16-lp152.3.5.x86_64
(1/3), 325,5 KiB ( 1,8 MiB после распаковки)
Получение: lksctp-tools-1.0.16-lp152.3.5.x86_64.rpm .....[готово]
Получение пакет java-1_8_0-openjdk-headless-1.8.0.282-lp152.2.9.1.x86_64
(2/3), 51,0 MiB ( 87,7 MiB после распаковки)
Получение: java-1_8_0-openjdk-headless-1.8.0.282-lp152.2.9.1[готово (4,1 MiB/s)]
Получение пакет java-1_8_0-openjdk-1.8.0.282-lp152.2.9.1.x86_64
(3/3), 544,1 KiB (569,4 KiB после распаковки)
Получение: java-1_8_0-openjdk-1.8.0.282-lp152.2.9.1.x86_64.rpm .....[готово]
Проверка на конфликты файлов: .....[готово]
```

Рис.11. Установка JDK в OpenSUSE Leap 15.2

Установка Eclipse на OpenSUSE 15.2 схожа с установкой Eclipse на Ubuntu 20.04. Воспользуемся браузером и с официального сайта <https://www.eclipse.org/downloads/> скачаем установщик и повторим процедуру для Eclipse IDE из Windows, запустив eclipse-inst (рис.12).

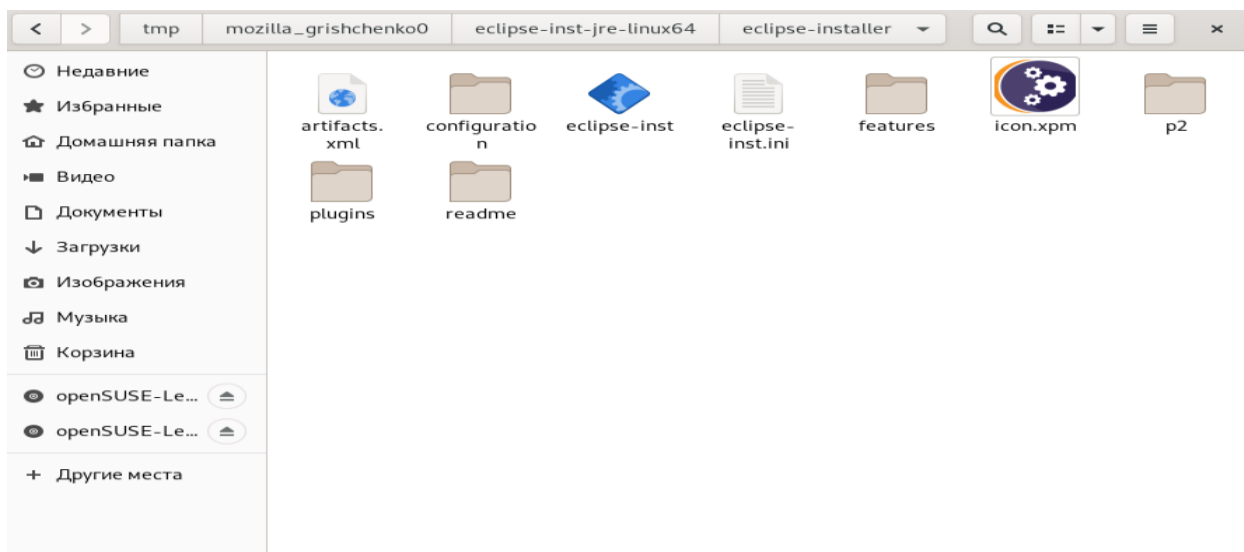


Рис.12. Установщик Eclipse IDE для OpenSuse Leap 15.2



## Раздел 4. Настройка среды разработки для подключения к системе контроля версий

В данном разделе описана инструкция по настройке Eclipse IDE для работы с системой контроля версии Git, где в качестве хостинга IT-проекта выступает веб-сервис GitHub.

Ссылка на репозиторий: [https://github.com/GrishchenkoIS/Kurs\\_project25](https://github.com/GrishchenkoIS/Kurs_project25)

При установке Eclipse IDE в новых версиях уже установлен Git, но также рассмотрим обычную установку. Так как процесс одинаков для всех используемых операционных сред, опишем только процесс для Windows.

Для начала установки в Eclipse необходимо нажать на Help – install new software, в поле “work with:” вводим “Latest Eclipse Release - <https://download.eclipse.org/releases/juno>”, в “Collaboration” выбираем “Eclipse EGit” и “Eclipse EGit Mylyn GitHub Feature” далее происходит установка (рис.13).

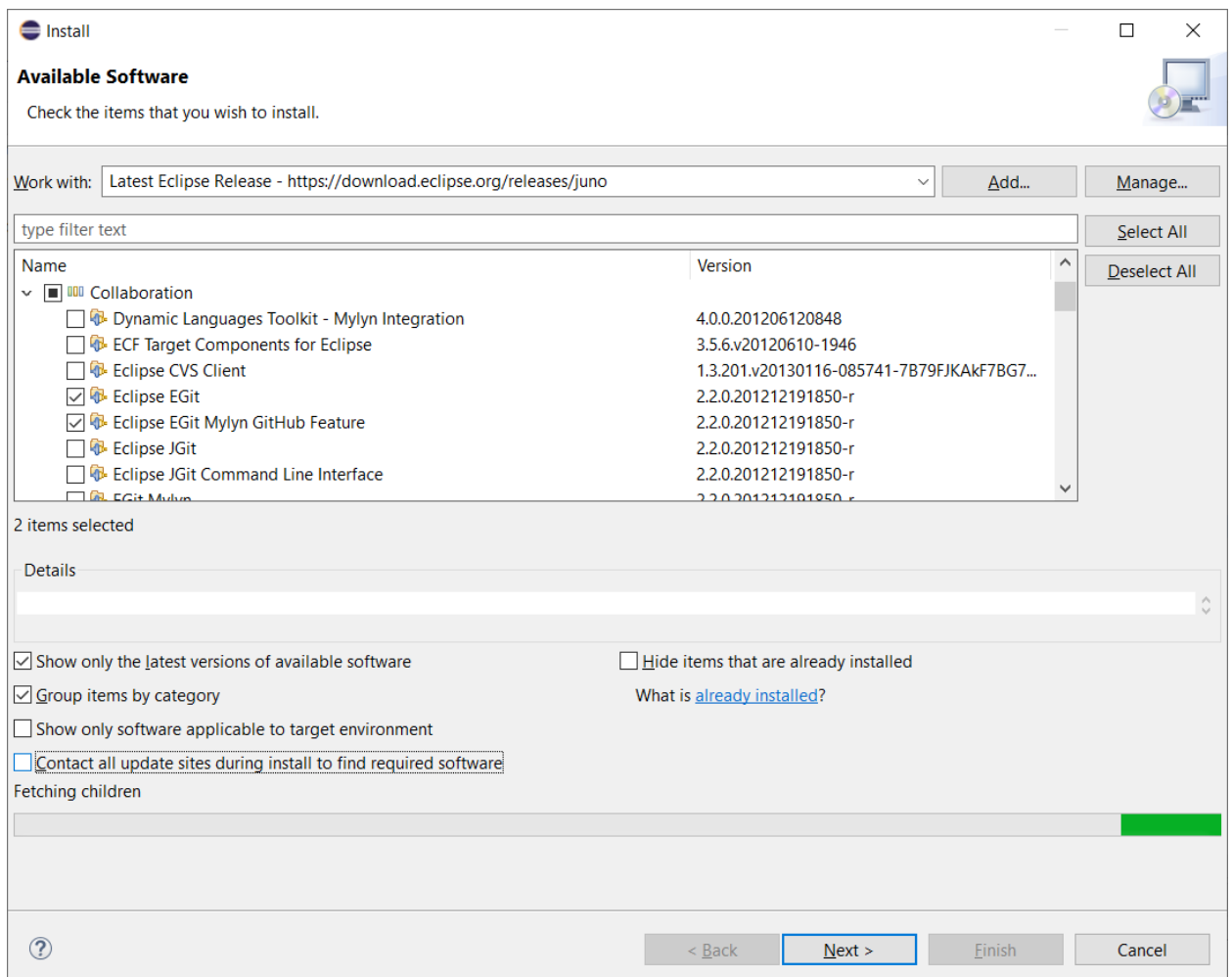


Рис.13. Установка Git.

После установки нужно перейти в проекцию Git. Нажимаем на “Open perspective” и выбираем Git (рис.14.).

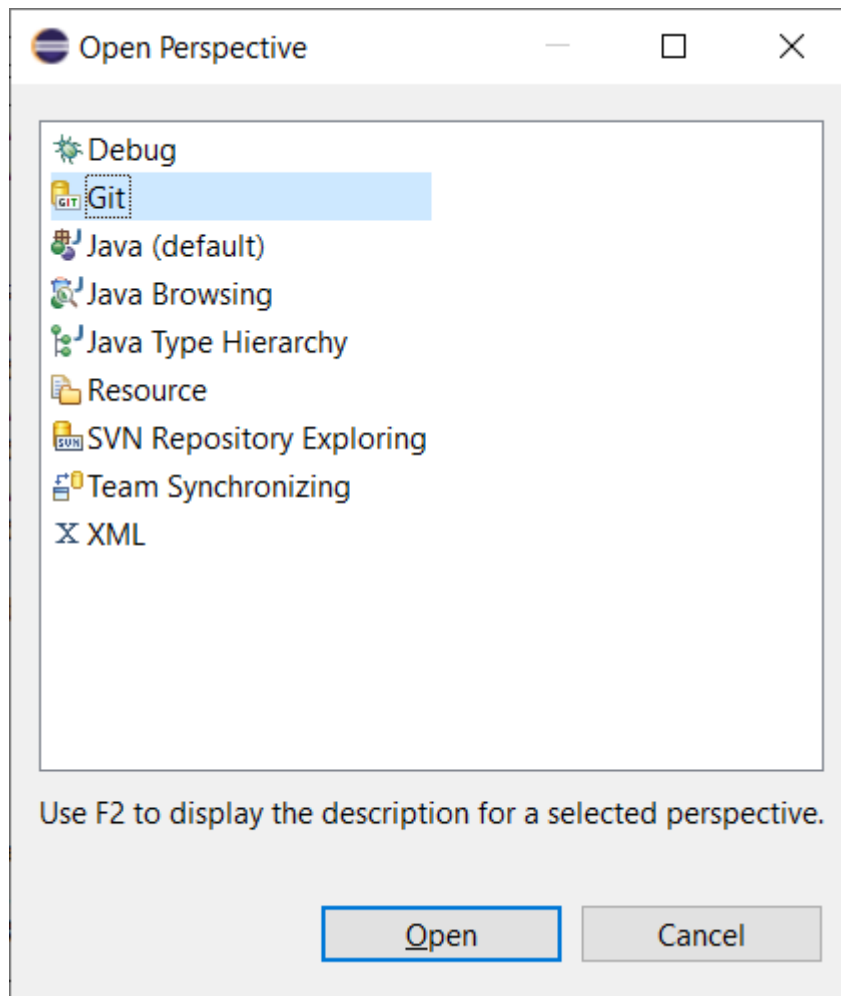


Рис.14. Переход в проекцию Git.

В пустой проекции будет предложено 3 действия. Выбираем клонирование – “Clone a Git repository”. В выпавшем окне в поле “URI” вводим ссылку на репозиторий GitHub - [https://github.com/GrishchenkoIS/Kurs\\_project25](https://github.com/GrishchenkoIS/Kurs_project25). При необходимости вводим логин и пароль (рис.15).

**Clone Git Repository**

**Source Git Repository**

Enter the location of the source repository.

**Location**

URI: thub.com/GrishchenkoIS/Kurs\_project25 Local Folder... Local Bundle File...

Host: github.com

Repository path: /GrishchenkoIS/Kurs\_project25

**Connection**

Protocol: https

Port:

**Authentication**

User: GrishchenkoIS

Password: .....

☐ Store in Secure Store

? < Back Next > Finish Cancel

Рис.15. Клонирование репозиторий Git.

Для получения рабочей копии необходимо перейти к “Working Tree”, вызвать контекстное меню и выбрать “Import projects...” (рис.16).

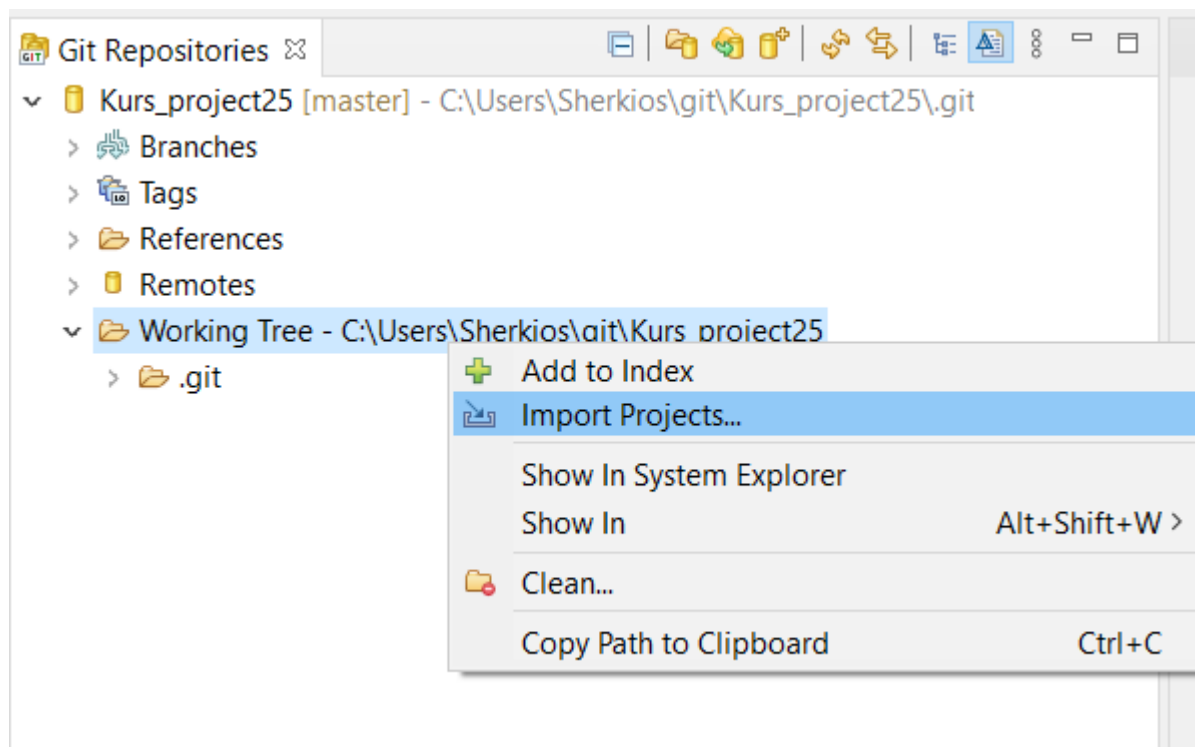


Рис.16. Создание рабочей копии.

Для работы с git необходимо вызвать контекстное меню у рабочей копии и выбрать “team” (рис.17).

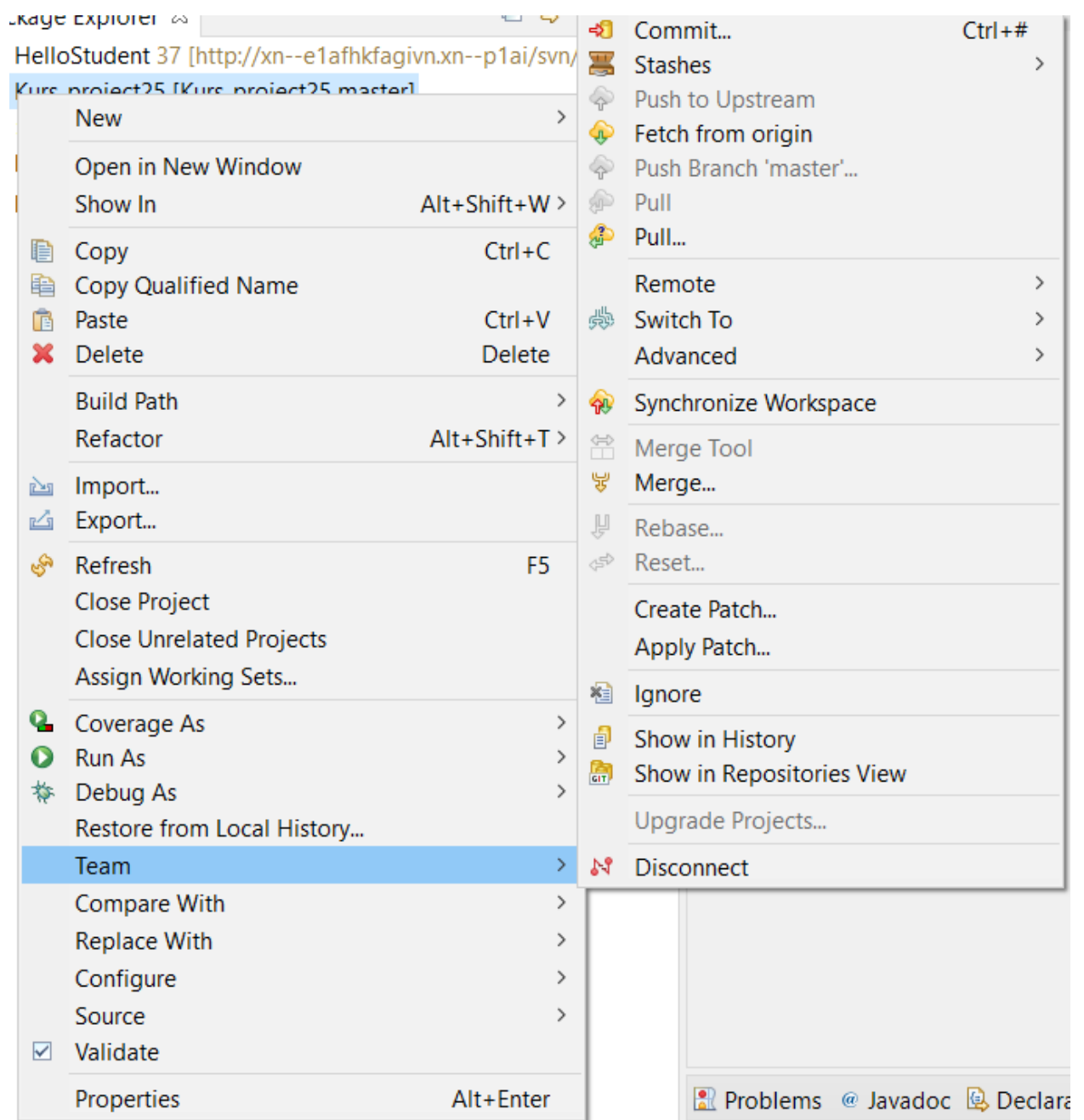


Рис.17. Работа с Git.

## **Раздел 5. Реализация исходного кода по зонам ответственности**

Зоны ответственности указаны в таблице 1

Таблица 1. Зоны ответственности

<b>№</b>	<b>ФИО разработчика/модератора</b>	<b>Зона ответственности</b>
1	Грищенко Игорь Сергеевич	Описание по зоне ответственности: index.jsp,author.jsp,changes.jsp,output.jsp. Стили: style.css, reset.css.  Класс: CreatePDF.java
2	Мочалина Алина Сергеевна	Описание по зоне ответственности:  Класс:Admin.java ,Calc.java, ToChange.java,Changes.java, Exit.java,Back.java.
3	Маломожнова Валентина Михайловна	Описание по зоне ответственности: Интерфейс:Check.java  Абстрактный класс:Calculator.java  Класс: CalculationC.java, CalculationNH.java, CalculationRH.java

Ссылка на репозиторий : [https://github.com/GrishchenkoIS/Kurs\\_project25](https://github.com/GrishchenkoIS/Kurs_project25)

## Раздел 6. Сборка и тестирование программного продукта

Зона ответственности Junit тестов описана в таблице 1

Таблица 1 – Описание UNIT – тестов

№	ФИО разработчика/модератора	Описание UNIT-теста	№ приложения
1	Грищенко Игорь Сергеевич	Проверяет строку на то что она является числом типа double	См. Приложение П-1
2	Мочалина Алина Сергеевна	Проверяет метод write. Проверяет файл на содержащую строку	См. Приложение П-2
3	Маломожнова Валентина Михайловна	Проверяет метод округления до сотых	См. Приложение П-3

Сборка проекта можно осуществить как вручную, так и автоматически.

Инструкция для ручной сборки:

Для ручной сборки необходимо перейти на сайт <https://dashboard.heroku.com/apps/kurs25/deploy/github>.

Далее во вкладке deploy(рис.18) в “Connect to Github” ввести название репозитория(рис.19) и нажать “Connect”

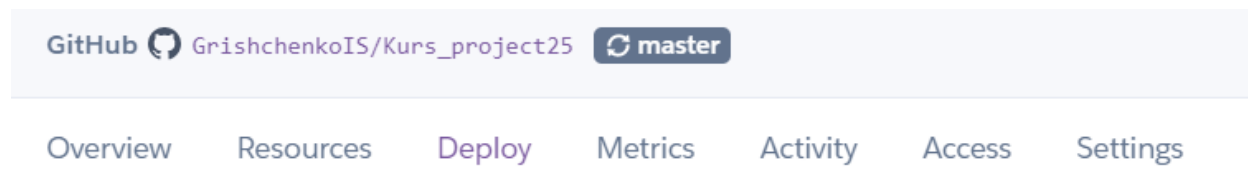


Рис.18. Вкладка deploy

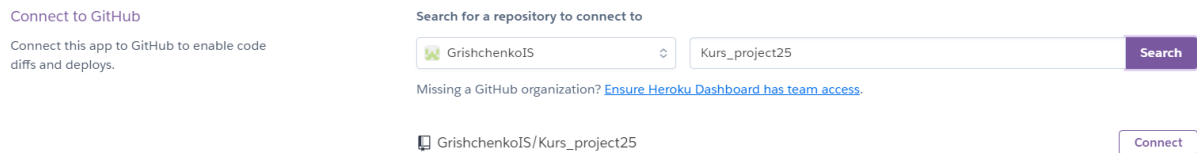


Рис.19. Ввод названия репозитория

После подключения репозитория там же найти “Manual deploy”, выбрать ветку репозитория и нажать “deploy branch”

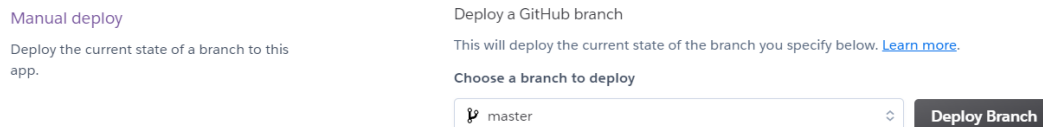


Рис.20. Сборка проекта

Каталог src/main/java содержит java классы..

Admin.java – сервлет для авторизации пользователя.

Author.java –сервлет для перехода на страницу авторизации.

Back.java – сервлет для выхода из аккаунта.

Calc.java –сервлет для принятия вводим данных и расчета с последующим переходом к странице результатов.

CalculationC.java, CalculationRH.java, CalculationNH.java – классы вычисления в зависимости от выбранной цели кредита.

Calculator.java – абстрактный класс для создания CalculationC.java, CalculationRH.java, CalculationNH.java.

Check.java – интерфейс для абстрактного класса Calculator.java.

Exit.java – сервлет для перехода на начальную страницу.

ToChange.java – сервлет для изменения ставки.

Каталог src/test/java содержит UNIT-тесты.

TestCheck.java - Проверяет строку на то что она является числом типа double.

TestToChange.java - Проверяет метод write. Проверяет файл на содержащую строку.



TestFlodPoint.java - Проверяет метод округления до сотых.

Каталог artifacts – содержит файлы Calculate.war, web-ranner.jar.

Каталог WEBINF – содержит web.xml.

Проект содержит каталог css в которой хранятся файлы reset.css, style.css, которые необходимы для размещения элементов jsp файла.

Файлы author.jsp, changes.jsp, index.jsp, output.jsp нужны для отображения страницы на веб-сервисе, в них описано как будет выглядеть страница.

Файл pom.xml содержит информацию о проекте и различных деталях конфигурации, которые используются Maven для создания проекта

## Раздел 7. Настройка программной среды для развертывания и запуска программного продукта

Для сборки и тестирования приложения используется веб сервис Travis CI. Для работы необходимо настроить Travis CI:

- 1) на сайте <https://travis-ci.org/> авторизоваться через github.
  - 2) нажать на кнопку «Activate all repositories using github apps».
  - 3) создать в корневом каталоге и написать содержание
- (рис.1) `before_script` – выполнение команд перед скриптом  
`script` – запуск скрипта сборки  
`after_script` – выполнение команд после этапа запуска  
`after_success` – выполнение команд при успешной сборке  
`after_failure` – выполнение команд при ошибке сборки

```
language: java
before_script:
- echo "Starting building"
script:
- mvn clean package
- mvn package
after_script:
- echo "Script finished"
after_success:
- echo "Build ready"
- ls -l $TRAVIS_BUILD_DIR/laba4/target
after_failure:
- echo "Build failure"
- env
```

Рис.21. Содержание .travis.yml

Для развёртывания проекта существует готовая среда Heroku. Чтобы развернуть приложения нам необходимо авторизоваться на сайте. В созданном приложении подключиться к репозиторию с проектом на github, во вкладке “deploy” найти “Automatic deploys” и нажать “Enable automatic deploys” (рис.2)

#### Automatic deploys

Enables a chosen branch to be automatically deployed to this app.



You can now change your main deploy branch from "master" to "main" for both manual and automatic deploys, please follow the instructions [here](#).

#### Enable automatic deploys from GitHub

Every push to the branch you specify here will deploy a new version of this app. **Deploys happen automatically:** be sure that this branch is always in a deployable state and any tests have passed before you push. [Learn more](#).

#### Choose a branch to deploy

master

☐ Wait for CI to pass before deploy

Only enable this option if you have a Continuous Integration service configured on your repo.

Enable Automatic Deploys

Рис.22. Автоматическое развертывания

Данный калькулятор работает в ОС Windows и Linux (рис.23,24,25)

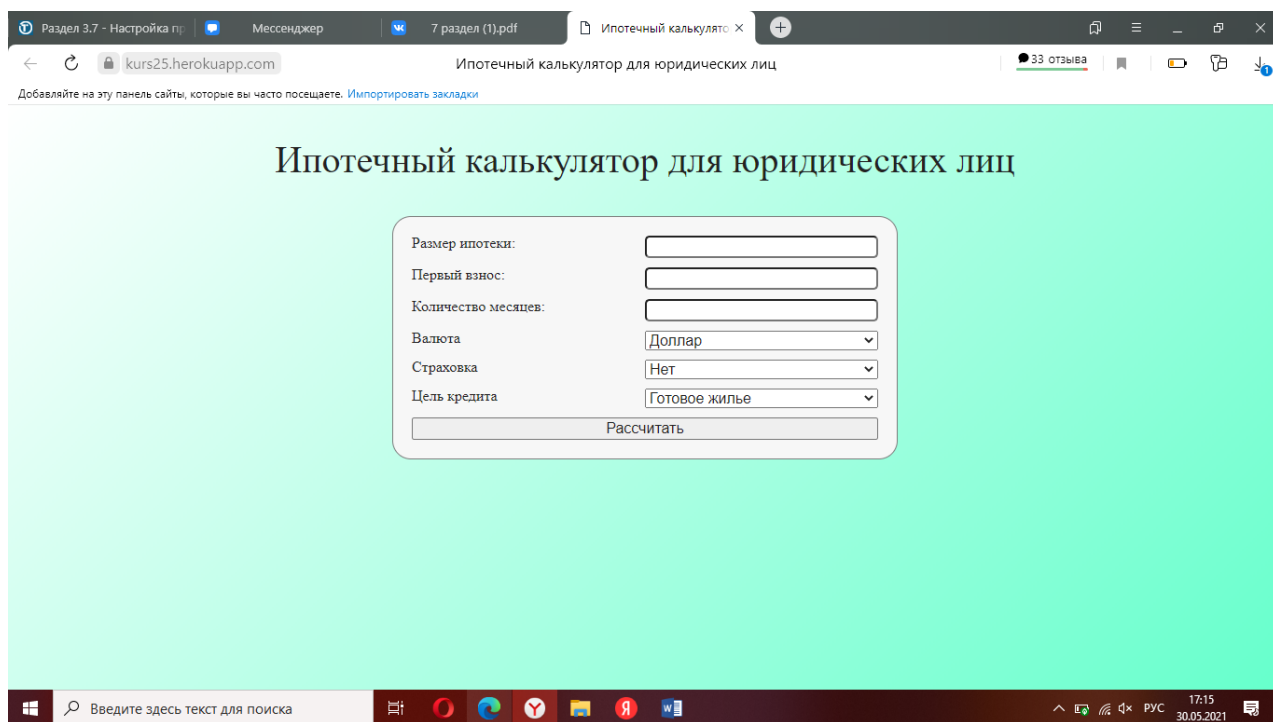


Рис.23. - Windows

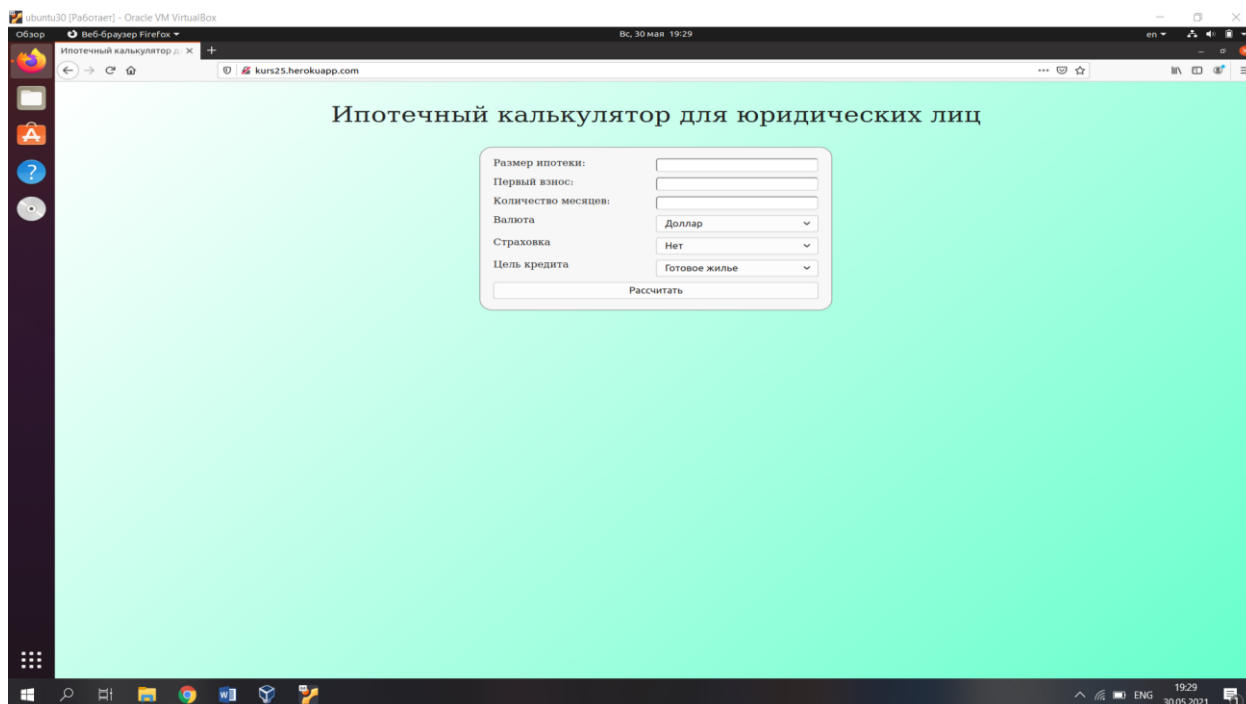


Рис.24. - Linux Ubuntu

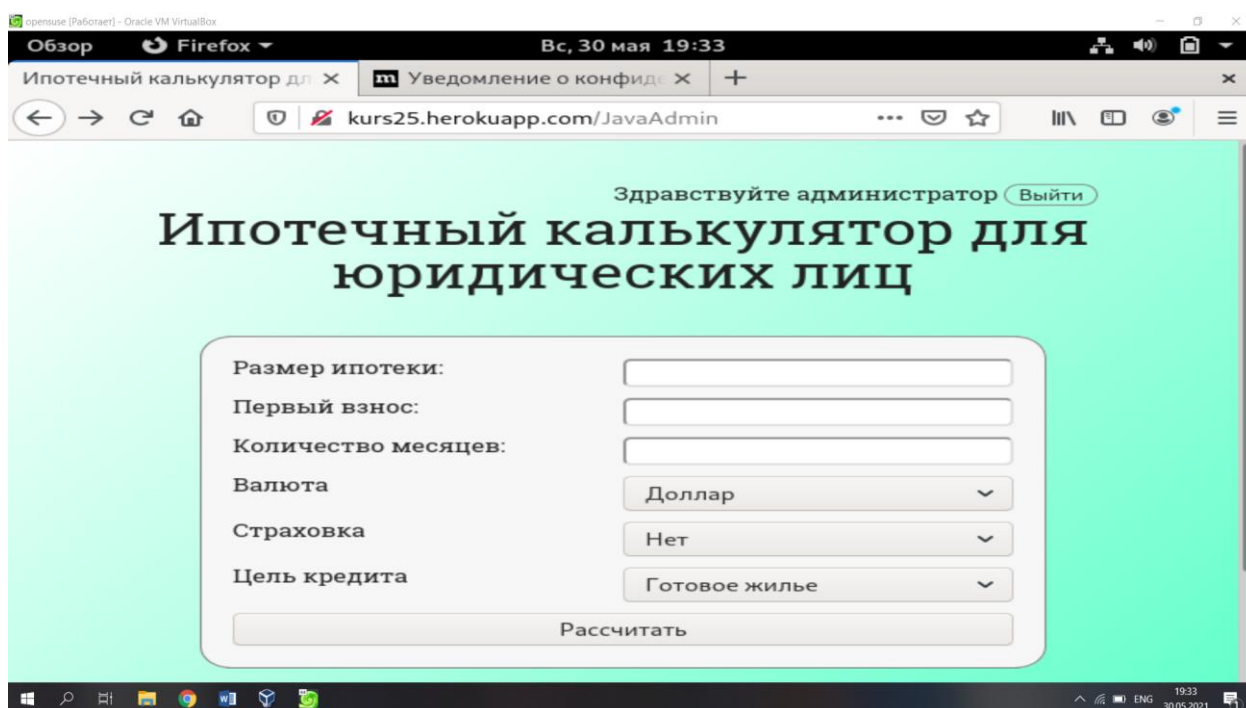


Рис.25. – Linux OpenSUSE

## **Раздел 8. Руководство пользователя программного продукта**

Руководство пользователя представлено в приложение 1

**ПРИЛОЖЕНИЕ 1**

**ФБГОУ ВО УФИМСКИЙ ГОСУДАРСТВЕННЫЙ АВИАЦИОННЫЙ  
ТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ**

**СОГЛАСОВАНО**

Доцент кафедры АСУ  
ФГОУ УГАТУ  
Личная      Расшифровка  
подпись      подписи  
27.03.2021

**УТВЕРЖДАЮ**

Студент группы ПИ-223  
ФИРТ ФГОУ УГАТУ,  
модератор  
Личная      Расшифровка  
подпись      подписи  
26.03.2021

**ИПОТЕЧНЫЙ КАЛЬКУЛЯТОР ДЛЯ ЮРИДИЧЕСКИХ ЛИЦ**

**Руководство оператора**

**ЛИСТ УТВЕРЖДЕНИЯ**

**АСУ-1304.300025.000 ТУ**

**(Электронный)**

**Листов 2**

**СОГЛАСОВАНО**

Доцент кафедры АСУ  
ФГОУ УГАТУ  
Личная      Расшифровка  
подпись      подписи  
29.03.2021

Представитель команды  
разработчиков

Студент группы ПИ-223  
ФИРТ ФГОУ УГАТУ,  
модератор  
Личная      Расшифровка  
подпись      подписи  
29.03.2021

Подп. и дата	
Инв. № инв.	
Взам. инв. №	
Подп. и дата	
Инв. № инв.	

Утвержден  
АСУ-1304.300025.000 ТУ

## ИПОТЕЧНЫЙ КАЛЬКУЛЯТОР ДЛЯ ЮРИДИЧЕСКИХ ЛИЦ

Техническое задание  
АСУ-1304.300025.000 ТУ  
(Электронный)  
Листов 7

Инд N подл	Подп и дата	Взам инд N	Инд N дубл	Подп и дата

## СОДЕРЖАНИЕ

1. Назначение программы .....	4
2. Условия выполнения .....	4
3. Выполнение программы .....	4
4. Сообщение оператору .....	7



## **1. Назначение программы**

“Ипотечный калькулятор для юридических лиц” – программа, помогающая сформулировать представление о будущих платежах по аннуитетным платежам по кредиту. Программа позволяет выбрать цель кредита и изменить ставку для этого кредита, выбрать валюту, сумму, первый платеж, количество месяцев и страховку.

Программа может быть запущена на любом устройстве поддерживающие браузеры и имеющие доступ к интернету. Так же программой могут пользоваться как простой пользователь, так и администратор.

Данный калькулятор предназначен для банка «Сбережения»

## **2. Условия выполнения программы**

Компьютер пользователя должен иметь следующие минимальные технические характеристики:

- Процессор – Intel Pentium 1.5 ГГц;
- Объем оперативной памяти – 256 Мб;
- Дисковая подсистема – 40 Гб;
- Устройство чтения компакт-дисков (DVD-ROM); – Сетевой адаптер
- 100 Мбит.

Для работы с Калькулятором у пользователя должен быть установлен браузер.

Пользователь работает с программой через веб-браузер, где у него есть доступ к веб-приложению, расположенному по определённом url-адресу

## **3. Выполнение программы**

Для доступа к функционалу программы необходимо зайти по адресу <https://kurs25.herokuapp.com/>.

После пользователю будет доступна авторизация для сохранения данных. Для авторизации нужно нажать на кнопку “Авторизоваться” и на новой странице ввести логин и пароль и нажать кнопку “Ввод” (рис.1)

Авторизация

Введите логин:

Введите пароль:

Рис.1. Ввод логина и пароля

После успешной регистрации/авторизации пользователя, произойдет переход на основную страницу программы (рис.2)

1

2

3

4

5

6

Рис.2. Основная страница программы

На основной странице находятся:

- 1) Поле ввода размера ипотеки
- 2) Поле ввода первого взноса
- 3) Поле ввода кол-ва месяцев
- 4) Выпадающее меню с выбором валюты
- 5) Выпадающее меню с выбором наличие ставки
- 6) Выпадающее меню с выбором цели кредита

Для расчетов необходимо ввести данные и нажать кнопку “Расчитать” (рис.3)

Размер ипотеки: 2000000

Первый взнос: 10000

Количество месяцев: 12

Валюта: Доллар

Страховка: Нет

Цель кредита: Готовое жилье

Рассчитать

Рис.3. Ввод данных

Далее будет произведен расчет вашего кредита (рис.4)

## Результаты вычислений

Размер ипотеки: 2000000	Первый взнос: 10000
Количество месяцев: 12	Валюта: Доллар
Страховка: Нет	Цель кредита: Готовое жилье
Результат: 168128.21 Долларов ежемесячно	

[Открыть PDF-файл](#) [Скачать PDF-файл](#) [Назад](#)

Рис.4. Результаты

Для расчётов других данных нужно нажать на кнопку “Назад”.

Для изменения ставок необходимо авторизоваться как администратор и нажать на кнопку “изменить” (рис.5).

Настройка ставки : [Изменить](#)

Рис.5. Изменение ставок.

После в поля ввода “Новая ставка готового жилья”, “Новая ставка новостройки”, “Новая ставка коммерческая” вводим новую ставку и жмем кнопку “Ввод” (рис.6).

Новая ставка готового жилья (8.3) :

Новая ставка новостройки (8.6) :

Новая ставка коммерческая (7) :

Рис.6. Ввод изменений для ставок

Когда расчеты готовы, можно вывести их в отдельный файл, который впоследствии можно будет скачать (рис.7).

Размер ипотеки: 200000      Первый взнос: 10000  
 Количество месяцев: 12      Валюта: Доллар  
 Страховка: Нет      Цель кредита: Готовое жилье

Результат: 17565.63 Долларов ежемесячно

[Открыть PDF-файл](#)      [Скачать PDF-файл](#)     

Рис.7. Открытие/скачивание файла

#### 4. Сообщения оператору

При авторизации, если введенные данные не существуют, то выводится сообщение об ошибке авторизации (рис. 8).

## Авторизация

Введите логин:

Введите пароль:

Не верный логин или пароль

Рис.8. Ошибка при авторизации

При вводе не корректных данных в форму изменения ставки , то выводится сообщение об ошибке изменений (рис.9.)

The image shows a mortgage calculator interface with a light green background. A white rounded rectangle contains the input fields. The fields are: 'Размер ипотеки:' (empty), 'Первый взнос:' (empty), 'Количество месяцев:' (empty), 'Валюта' (dropdown with 'Доллар'), 'Страховка' (dropdown with 'Нет'), and 'Цель кредита' (dropdown with 'Готовое жилье'). Below these is a 'Рассчитать' button. Below the white rectangle, a red error message 'Не верно введены данные' is displayed, followed by the text 'Изменить ставку : ' and a button labeled 'Изменить'.

Размер ипотеки:	<input type="text"/>
Первый взнос:	<input type="text"/>
Количество месяцев:	<input type="text"/>
Валюта	Доллар ▾
Страховка	Нет ▾
Цель кредита	Готовое жилье ▾
<input type="button" value="Рассчитать"/>	

Не верно введены данные

Изменить ставку :

Рис.9. Ошибка введения не корректной ставки

**ПРИЛОЖЕНИЕ 2**

**ФБГОУ ВО УФИМСКИЙ ГОСУДАРСТВЕННЫЙ АВИАЦИОННЫЙ  
ТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ**

**СОГЛАСОВАНО**

Доцент кафедры АСУ  
ФГОУ УГАТУ  
Личная      Расшифровка  
подпись      подписи  
27.03.2021

**УТВЕРЖДАЮ**

Студент группы ПИ-223  
ФИРТ ФГОУ УГАТУ,  
модератор  
Личная      Расшифровка  
подпись      подписи  
26.03.2021

**ИПОТЕЧНЫЙ КАЛЬКУЛЯТОР ДЛЯ ЮРИДИЧЕСКИХ ЛИЦ**

**Техническое задание**

**ЛИСТ УТВЕРЖДЕНИЯ**

**АСУ-1304.300025.000 ТУ**

**(Электронный)**

**Листов 2**

**СОГЛАСОВАНО**

Доцент кафедры АСУ  
ФГОУ УГАТУ  
Личная      Расшифровка  
подпись      подписи  
29.03.2021

Представитель команды  
разработчиков

Студент группы ПИ-223  
ФИРТ ФГОУ УГАТУ,  
модератор  
Личная      Расшифровка  
подпись      подписи  
29.03.2021

Подп. и дата	
Инв. №	
Взам. инв. №	
Подп. и дата	
Инв. №	

Утвержден  
АСУ-1304.300025.000 ТУ

## **ИПОТЕЧНЫЙ КАЛЬКУЛЯТОР ДЛЯ ЮРИДИЧЕСКИХ ЛИЦ**

**Техническое задание**  
**АСУ-1304.300025.000 ТУ**  
**(Электронный)**  
**Листов 11**

Инд N подл	Подп и дата	Взам инд N	Инд N учбр	Подп и дата

## СОДЕРЖАНИЕ

1. ВЕДЕНИЕ .....	4
2. ОСНОВАНИЕ ДЛЯ РАЗРАБОТКИ .....	4
3. НАЗНАЧЕНИЕ РАЗРАБОТКИ .....	4
3.1. Функциональное назначение программы .....	4
3.2. Эксплуатационное назначение программы .....	4
4. ТРЕБОВАНИЯ К ПРОГРАММЕ ИЛИ ПРОГРАММНОМУ ИЗДЕЛИЮ ....	4
4.1. Требования к функциональным характеристикам .....	4
4.2. Требования к надежности .....	5
4.3. Условия эксплуатации .....	5
4.4. Требование к составу и параметрам технических средств .....	5
4.5. Требования к информационной и программной совместимости .....	6
4.6. Требования к маркировке и упаковке .....	6
4.7. Требования к транспортированию и хранению .....	6
4.8. Конфигурация сервера .....	6
5. ТРЕБОВАНИЕ К ПРОГРАММНОЙ ДОКУМЕНТАЦИИ .....	6
6. ТЕХНИЧЕСКО-ЭКОНОМИЧЕСКИЕ ПОКАЗАТЕЛИ .....	6
7. СТАДИИ И ЭТАПЫ РАЗРАБОТКИ .....	6
8. ПОРЯДОК КОНТРОЛЯ И ПРИЕМКИ .....	7



## **1. ВЕДЕНИЕ**

Наименования программы “ипотечный калькулятор для юридических лиц”. Данная АС является бесплатным веб-сервисом для юридических лиц и предназначена для расчетов ежемесячного платежа по аннуитетному кредиту

Юридическое лицо – это зарегистрированная в установленном законом порядке организация, фирма, компания, которая имеет в собственности, хозяйственном ведении или оперативном управлении обособленное имущество и отвечает по своим обязательствам этим имуществом, может от своего имени приобретать и осуществлять имущественные и личные неимущественные права, нести обязанности, быть истцом и ответчиком в суде.

Аннуитетный платеж – платеж, предполагающий выплату как основного долга, так и сумму начисленных процентов по кредиту. В первые месяцы основную часть составляют проценты.

Область применения программного продукта является сфера недвижимости.

## **2. ОСНОВАНИЕ ДЛЯ РАЗРАБОТКИ**

Основанием для разработки АС ипотечного калькулятора для юридических лиц являются необходимость написания курсовой работы по дисциплине “Информационные системы”:

Организация, утвердившая документ: ФГБОУ ВО Уфимский

Государственный Авиационный Технический Университет.

Дата утверждения документа: 27.03.2021

## **3. НАЗНАЧЕНИЕ РАЗРАБОТКИ**

### **3.1. Функциональное назначение программы**

Функциональным назначением программы является расчет ежемесячного платежа по аннуитетным кредитам.

### **3.2. Эксплуатационное назначение программы**

Программа должна быть использована клиентом, желающим использовать ее или администратором, желающим изменить коэффициенты.

## **4. ТРЕБОВАНИЕ К ПРОГРАММЕ ИЛИ ПРОГРАММНОГО ИЗДЕЛИЮ**

### **4.1 Требования к функциональным характеристикам**

Программа должна иметь авторизацию обычного пользователя и администратора. После авторизации пользователю будет предложена возможность ввести данные для расчета ипотечного кредита, администратору также будет доступно изменение коэффициентов. После

расчетов на экран будут выводиться итоги расчетов. Описанная логика программы также изображена на диаграмме взаимодействия программы ипотечного калькулятора для юридических лиц

Программное обеспечение будет является кроссплатформенным вебприложением.

Кроссплатформенность - способность программного обеспечения работать с несколькими аппаратными платформами или операционными системами.

Функциональные возможности:

- 1) авторизация пользователя и администратора;
- 2) просмотр информации;
- 3) изменение информации;
- 4) просмотр результата.

Программа будет запускаться на семействе Linux. На дистрибутивах таких как Ubuntu 20.04 и OpenSUSE.

#### **4.2. Требования к надежности**

Для надежного функционирования программы необходимо обеспечить бесперебойное питание технических средств (хостинг, компьютер), персонал обслуживающий хостинг должен иметь необходимый уровень квалификации.

Для защиты информации должна быть авторизация пользователя, которая позволит либо только использовать программу, либо использовать программу и изменять коэффициенты.

#### **4.3. Условия эксплуатации**

Требование к эксплуатации:

Система электропитания должна обеспечивать защитное отключение при перегрузках и коротких замыканиях в цепях нагрузки, а также аварийно-ручное отключение.

Все внешние элементы технических средств системы, находящиеся под напряжением, должны иметь защиту от случайного прикосновения, а сами технические средства иметь зануление или защитное заземление.

#### **4.4. Требование к составу и параметрам технических средств**

Для корректной работы АС в состав комплекса должны входить следующие средства:

- Серверы БД;
- Веб сервер;
- ПК пользователей.

Для защиты аппаратуры от бросков напряжения и коммутационных помех должны применяться сетевые фильтры.

Требования к техническим характеристикам серверов БД:

- Процессор – 2 x Intel Xeon 3 ГГц;
- Объем оперативной памяти – 16 Гб;
- Дисковая подсистема – 4 x 146 Гб;

- Устройство чтения компакт-дисков (DVD-ROM);
- Сетевой адаптер – 100 Мбит.

Требования к техническим характеристикам веб сервера:

- Процессор – 2 x Intel Xeon 3 ГГц;
- Объем оперативной памяти – 16 Гб;
- Дисковая подсистема – 4 x 146 Гб;
- Устройство чтения компакт-дисков (DVD-ROM);
- Сетевой адаптер – 100 Мбит.

Требования к техническим характеристикам ПК пользователя:

Процессор – Intel Pentium 1.5 ГГц;

- Объем оперативной памяти – 256 Мб;
- Дисковая подсистема – 40 Гб;
- Устройство чтения компакт-дисков (DVD-ROM);
- Сетевой адаптер – 100 Мбит.

#### **4.5. Требования к информационной и программной совместимости**

Данное программное изделие будет работать на языке Java.

На системе должна быть установлена виртуальная машина Java

#### **4.6. Требования к маркировке и упаковке**

Требования к маркировкам отсутствуют.

#### **4.7. Требования к транспортированию и хранению**

Требования к транспортировке и хранению отсутствуют.

#### **4.8. Конфигурация сервера**

- CPU – 1 ядро 2ГГц
- RAM – 1,5 ГБ
- Диск – 10 ГБ
- Образ – Debian 9

### **5. ТРЕБОВАНИЕ К ПРОГРАММНОЙ ДОКУМЕНТАЦИИ**

Предварительный состав программной документации включает в себя:

- техническое задание;
- руководство оператора (пользователя);
- руководство администратора;
- текст программы.

### **6. ТЕХНИЧЕСКО-ЭКОНОМИЧЕСКИЕ ПОКАЗАТЕЛИ**

Не рассчитывается.

### **7. СТАДИИ И ЭТАПЫ РАЗРАБОТКИ**

План график работы приведен ниже.

Таблица 1 - План-график выполнения курсовой работы

Наименование этапа работ	Трудоемкость выполнения, час	Процент к общей трудоемкости выполнения
Получение и согласование задания	1,7	1,7%
Раздел 1. Описание предметной области	20	20%
Раздел 2. Техническое задание на создание программного продукта	10	10%
Раздел 3. Настройка среды разработки для операционных систем семейств Windows и Linux	10	10%
Раздел 4. Настройка среды разработки для подключения к системе контроля версий	7	7%
Раздел 5. Реализация исходного кода по зонам ответственности	23	23%
Раздел 6. Сборка и тестирование программного продукта	8	8%
Раздел 7. Настройка программной среды для развертывания и запуска программного продукта	10	10%
Раздел 8. Руководство пользователя программного продукта	10	10%
Защита	0,3	0,3%
Итого	100	100

## 8. ПОРЯДОК КОНТРОЛЯ И ПРИЕМКИ

Виды испытаний будут проводиться путем использования JUnit тестов. В тестах будет проверяться конечный результат на ожидаемые данные.

Общие требования к приемке работы:

- 1) Настройка среды Eclipse в нескольких операционных системах разных семейств;
- 2) Клонирование репозитория GitHub, извлечение рабочей копии и выполнение основных команд;

- 3) Работа с сервисом Travis CI;
- 4) Выполнить развертывание и запуск программного продукта;
- 5) Знание своей зоны ответственности.

## Программный код. Сервлет Admin.java

```
package kurs25;

import java.io.BufferedReader;
import java.io.DataOutput;
import java.io.DataOutputStream;
import java.io.File;
import java.io.FileNotFoundException;
import java.io.FileReader;
import java.io.FileWriter;
import java.io.IOException;
import java.io.InputStream;
import java.io.InputStreamReader;
import java.nio.file.Path;
import java.util.ArrayList;
import java.util.List;
import java.util.Properties;
import java.util.Scanner;
import javax.servlet.ServletException;
import javax.servlet.annotation.WebServlet;
import javax.servlet.http.HttpServlet;
import javax.servlet.http.HttpServletRequest;
import javax.servlet.http.HttpServletResponse;

@WebServlet(name="Admin", urlPatterns="/JavaAdmin")
public class Admin extends HttpServlet {

    public static int status = 2 ;

    protected void doPost(HttpServletRequest request, HttpServletResponse response) throws
ServletException, IOException {

        try {
            InputStream ins = getServletContext().getResourceAsStream("password");
            if (ins == null) {
                System.out.println("Failed in reading file");
            } else {
                BufferedReader br = new BufferedReader((new InputStreamReader(ins));
                String word = "";

                boolean statusWord = false;

                while ((word = br.readLine()) != null) {

                    String[] check = word.split(" ");
                    if (check[0].equals(request.getParameter("login"))) {
                        String[] words = word.split(" ");

                        if (words[0].equals(request.getParameter("login")) &&
words[1].equals(request.getParameter("password")) && words[2].equals("1")) {
                            request.setAttribute("incorrect", "");
                            request.setAttribute("changes", "<label for=\"changes\"
class=\"changes__text\"> Изменить ставку :</label>\n"
+ "
<input type=\"submit\" name=\"sign\"
value=\"Изменить\" class=\"changes__submit input\">");
                            request.setAttribute("display", "none");
                            request.setAttribute("admin", "<div class=\"header__form\">\n"
+ "
<label for=\"\" class=\"header__text\">
Здравствуйте администратор</label>\n"
```

```

value="Выйти" class="header__input input"></div>");
        + "        <input type=\"submit\" name=\"sign\"
        status = 1;
        request.getRequestDispatcher("/index.jsp").forward(request, response);
        break;
    } else if
(words[0].equals(request.getParameter("login")) && words[1].equals(request.getParameter("password")) &&
words[2].equals("0")) {
        request.setAttribute("incorrect", "");
        request.setAttribute("display", "none");
        request.setAttribute("admin", "<div
class=\"header__form\">\n"
        + "        <label for=\""
class="header__text"> Здравствуйте пользователь</label>\n"
        + "        <input type=\"submit\"
name="sign" value="Выйти" class="header__input input"></div>");
        status = 0;
        request.getRequestDispatcher("/index.jsp").forward(request,
response);
        break;
    } else {

        request.setAttribute("incorrect", "Не верный логин или пароль");

        request.setAttribute("changes", "");

        request.getRequestDispatcher("/author.jsp").forward(request, response);
        break;

    }

    } else {
        statusWord = true;
    }
}
if (statusWord) {
        request.setAttribute("incorrect", "Не верный логин или
пароль");
        request.setAttribute("changes", "");
        request.getRequestDispatcher("/author.jsp").forward(request,
response);
    }
}
} catch (FileNotFoundException e) {
    e.printStackTrace();
}
}
}

```

## Программный код. Сервлет Calc.java

```

package kurs25;

import java.io.BufferedReader;
import java.io.File;
import java.io.FileInputStream;
import java.io.FileReader;
import java.io.FileWriter;
import java.io.IOException;
import java.io.InputStream;
import java.io.InputStreamReader;
import java.net.URLDecoder;
import java.nio.file.Path;
import java.nio.file.Paths;
import java.util.ArrayList;
import java.util.List;

import javax.servlet.ServletException;
import javax.servlet.annotation.WebServlet;
import javax.servlet.http.HttpServlet;
import javax.servlet.http.HttpServletRequest;
import javax.servlet.http.HttpServletResponse;

import kurs25.CreatePDF;

@WebServlet(name="Calc", urlPatterns="/JavaCalc")
public class Calc extends HttpServlet {

    public static String creditAmount;
    public static String firstPayPDF;
    public static String paymentPerMonth;
    public static String purposeOfTheLoan;
    public static String currencyPDF;
    public static String strahovkaPDF;
    public static String dataPDF;

    protected void doPost(HttpServletRequest request, HttpServletResponse response) throws
ServletException, IOException {

        RequestCalc Calc = RequestCalc.fromRequestParameters(request);

        File currentClass = new File(URLDecoder.decode(Calc.class
.getProtectionDomain()
.getCodeSource()
.getLocation()
.getPath(), "UTF-8"));

        String filepath = currentClass.getParent();
        File currentClass2 = new File(URLDecoder.decode(filepath, "UTF-8"));
        filepath = currentClass2.getParent();

        InputStream ins = new FileInputStream(filepath + "/Stavka");
        if (ins == null) {
            System.out.println("Failed in reading file");
        } else {
            BufferedReader br = new BufferedReader(new FileReader(filepath + "/Stavka"));
            String word;
            while ((word = br.readLine()) != null) {
                Calc.stavka.add(word);
            }
        }
    }
}

```



```
}
```

```
Calc.setAsRequestAttributesAndCalculate(request);  
if ((Admin.status == 0) || (Admin.status == 1)) {  
    request.getRequestDispatcher("/output.jsp").forward(request, response);  
    Calc.stavka.clear();  
  
    CreatePDF PDF = new CreatePDF();  
    String goals = "Hello";  
    PDF.Create(goals);  
} else {  
    request.getRequestDispatcher("/author.jsp").forward(request, response);  
}  
}
```

```
}
```

```
public static class RequestCalc {  
    public final String sizeMortgage;  
    public final String firstPay;  
    public final String data;  
    public final String target;  
    public static String strahovka;  
    public static String currency;  
    private double result;  
    static double first_result;  
    static double second_result;  
    static double data_result;  
    static List<String> stavka = new ArrayList<String>();  
}
```

```
    private RequestCalc (String sizeMortgage, String firstPay, String data, String target, String  
    strahovka, String currency) {  
        this.sizeMortgage = sizeMortgage;  
        this.firstPay = firstPay;  
        this.data = data;  
        this.target = target;  
        RequestCalc.strahovka = strahovka;  
        RequestCalc.currency = currency;  
    }  
}
```

```
public static RequestCalc fromRequestParameters(HttpServletRequest request) {  
    return new RequestCalc(  
        request.getParameter("sizeMortgage"),  
        request.getParameter("firstPay"),  
        request.getParameter("data"),  
        request.getParameter("target"),  
        request.getParameter("strahovka"),  
        request.getParameter("currency")  
    );  
}
```

```
public void setAsRequestAttributesAndCalculate(HttpServletRequest request) {  
    request.setAttribute("first_result", sizeMortgage);  
    request.setAttribute("second_result", firstPay);  
    request.setAttribute("data_result", data);  
    if (target.equals("ready house")) {  
        request.setAttribute("target_result", "Готовое жилье");  
    } else if (target.equals("new house")) {  
        request.setAttribute("target_result", "Новостройка");  
    } else {  
        request.setAttribute("target_result", "Коммерческое");  
    }  
    if (strahovka.equals("No")) {  
    }  
}
```

```

        request.setAttribute("strahovka_result", "Нет");
    } else {
        request.setAttribute("strahovka_result", "Есть");
    }
    if (currency.equals("dollar")) {
        request.setAttribute("currency_result", "Доллар");
        request.setAttribute("currency_for_result", "Долларов ежемесячно");
    } else if (currency.equals("rub")) {
        request.setAttribute("currency_result", "Рубль");
        request.setAttribute("currency_for_result", "Рублей ежемесячно");
    } else {
        request.setAttribute("currency_result", "Евро");
        request.setAttribute("currency_for_result", "Евро ежемесячно");
    }

    try {
        first_result=Double.parseDouble(sizeMortgage);
    }
    catch (NumberFormatException e) {
        first_result=0;
        request.setAttribute("first_result", 0);
    }
    try {
        second_result=Double.parseDouble(firstPay);
    } catch (NumberFormatException e) {
        second_result=0;
        request.setAttribute("second_result", 0);
    }
    try {
        data_result=Double.parseDouble(data);
    } catch (NumberFormatException e) {
        data_result=0;
        request.setAttribute("data_result", 0);
    }
    if (second_result < 0 || first_result <= 0) {
        result = 0;
        request.setAttribute("result", result);
    } else {
        if (target.equals("ready house")) {
            CalculationRH calculation = new CalculationRH();
            request.setAttribute("result", calculation.res);
            result = calculation.res;
        }
        if (target.equals("new house")) {
            CalculationNH calculation = new CalculationNH();
            request.setAttribute("result", calculation.res);
            result = calculation.res;
        }
        if (target.equals("commercial")) {
            CalculationC calculation = new CalculationC();
            request.setAttribute("result", calculation.res);
            result = calculation.res;
        }
    }

    creditAmount = Double.toString(first_result);
    firstPayPDF = Double.toString(second_result);
    paymentPerMonth = Double.toString(result);
    purposeOfTheLoan = target;
    if (currency.equals("dollar")) {
        currencyPDF = "Dollar";
    }

```

```
    } else if (currency.equals("rub")) {  
        currencyPDF = "Rub";  
    } else {  
        currencyPDF = "Euro";  
    }  
    dataPDF = Double.toString(data_result);  
    if (strahovka.equals("No")) {  
        strahovkaPDF = "No";  
    } else {  
        strahovkaPDF = "Yes";  
    }  
}  
  
}  
  
}
```

## Программный код. Сервлет Back.java

```
package kurs25;

import java.io.IOException;

import javax.servlet.ServletException;
import javax.servlet.annotation.WebServlet;
import javax.servlet.http.HttpServlet;
import javax.servlet.http.HttpServletRequest;
import javax.servlet.http.HttpServletResponse;
@WebServlet(name="Back", urlPatterns={"/Back"})
public class Back extends HttpServlet {

    protected void doPost(HttpServletRequest request, HttpServletResponse response) throws
ServletException, IOException {
        if ( Admin.status == 1 ) {
            request.setAttribute("incorrect", "");
            request.setAttribute("changes", "<label for=\"changes\" class=\"changes__text\">
Настройка ставки :</label><n\"
                                + \"          <input type=\"submit\" name=\"sign\" value=\"Изменить\"
class=\"changes__submit input\">");
            request.setAttribute("display", "none");
            request.setAttribute("admin", "<div class=\"header__form\"><n\"
                                + \"          <label for=\"\" class=\"header__text\"> Здравствуйте
администратор</label><n\"
                                + \"          <input type=\"submit\" name=\"sign\" value=\"Выйти\"
class=\"header__input input\"></div>");

        } else if (Admin.status == 0) {
            request.setAttribute("incorrect", "");
            request.setAttribute("display", "none");
            request.setAttribute("admin", "<div class=\"header__form\"><n\"
                                + \"          <label for=\"\" class=\"header__text\"> Здравствуйте
пользователь</label><n\"
                                + \"          <input type=\"submit\" name=\"sign\" value=\"Выйти\"
class=\"header__input input\"></div>");

        }
        request.getRequestDispatcher("/index.jsp").forward(request, response);
    }
}
```

## Программный код. Сервлет Change.java

```
package kurs25;

import java.io.BufferedReader;
import java.io.File;
import java.io.FileInputStream;
import java.io.FileReader;
import java.io.FileWriter;
import java.io.IOException;
import java.io.InputStream;
import java.net.URLDecoder;
import java.util.ArrayList;
import java.util.List;

import javax.servlet.ServletException;
import javax.servlet.annotation.WebServlet;
import javax.servlet.http.HttpServlet;
import javax.servlet.http.HttpServletRequest;
import javax.servlet.http.HttpServletResponse;
@WebServlet(name="Change", urlPatterns="/JavaToChange")
public class Changes extends HttpServlet {

    protected void doPost(HttpServletRequest request, HttpServletResponse response) throws
ServletException, IOException {
        File currentClass = new File(URLDecoder.decode(Changes.class
        .getProtectionDomain()
        .getCodeSource()
        .getLocation()
        .getPath(), "UTF-8"));

        String filepath = currentClass.getParent();
        File currentClass2 = new File(URLDecoder.decode(filepath, "UTF-8"));
        filepath = currentClass2.getParent();

        List<String> stavka = new ArrayList<String>();

        InputStream ins = new FileInputStream(filepath + "/Stavka");
        if (ins == null) {
            System.out.println("Failed in reading file");
        } else {
            BufferedReader br = new BufferedReader(new FileReader(filepath + "/Stavka"));
            String word;
            while ((word = br.readLine()) != null) {
                stavka.add(word);
            }

            request.setAttribute("stavkaRH", stavka.get(0));
            request.setAttribute("stavkaNH", stavka.get(1));
            request.setAttribute("stavkaC", stavka.get(2));
            request.getRequestDispatcher("/changes.jsp").forward(request, response);
        }
    }
}
```

## Программный код. Сервлет Exit.java

```
package kurs25;

import java.io.FileWriter;
import java.io.IOException;

import javax.servlet.ServletException;
import javax.servlet.annotation.WebServlet;
import javax.servlet.http.HttpServlet;
import javax.servlet.http.HttpServletRequest;
import javax.servlet.http.HttpServletResponse;

@WebServlet(name="Exit", urlPatterns="/JavaExit")
public class Exit extends HttpServlet {
    protected void doPost(HttpServletRequest request, HttpServletResponse response) throws
    ServletException, IOException {
        Admin.status = 2;
        request.getRequestDispatcher("/author.jsp").forward(request, response);
    }
}
```

## Приложение Б-6. Программный код. Сервлет ToChange.java

```

package kurs25;

import java.io.BufferedReader;
import java.io.File;
import java.io.FileNotFoundException;
import java.io.FileReader;
import java.io.FileWriter;
import java.io.IOException;
import java.io.InputStream;
import java.io.InputStreamReader;
import java.io.PrintWriter;
import java.net.URLDecoder;

import javax.servlet.ServletException;
import javax.servlet.annotation.WebServlet;
import javax.servlet.http.HttpServlet;
import javax.servlet.http.HttpServletRequest;
import javax.servlet.http.HttpServletResponse;
@WebServlet(name="ToChange", urlPatterns="/JavaChange")
public class ToChange extends HttpServlet {

    protected void doPost(HttpServletRequest request, HttpServletResponse response) throws
ServletException, IOException {
        File currentClass = new File(URLDecoder.decode(ToChange.class
        .getProtectionDomain()
        .getCodeSource()
        .getLocation()
        .getPath(), "UTF-8"));

        String filepath = currentClass.getParent();
        File currentClass2 = new File(URLDecoder.decode(filepath, "UTF-8"));
        filepath = currentClass2.getParent();

        if ((IsDouble(request.getParameter("StReadyHouse"))) &&
(IsDouble(request.getParameter("StComercial"))) && (IsDouble(request.getParameter("StNewHouse")))) {
            write(filepath + "/Stavka", request.getParameter("StReadyHouse") + "\n" +
request.getParameter("StNewHouse") + "\n" + request.getParameter("StComercial"));
        } else {
            request.setAttribute("incChanges", "Введены не верные данные");
        }

        if (Admin.status == 1) {
            request.setAttribute("incorrect", "");
            request.setAttribute("changes", "<label for=\"changes\" class=\"changes__text\"> Изменить ставку
:</label>\n"
+ "
<input type=\"submit\" name=\"sign\" value=\"Изменить\"
class=\"changes__submit input\">");
            request.setAttribute("display", "none");
            request.setAttribute("admin", "<div class=\"header__form\">\n"
+ "
<label for=\"\" class=\"header__text\"> Здравствуйте
администратор</label>\n"
+ "
<input type=\"submit\" name=\"sign\" value=\"Выйти\"
class=\"header__input input\"></div>");

        } else if (Admin.status == 0) {
            request.setAttribute("incorrect", "");
            request.setAttribute("display", "none");
            request.setAttribute("admin", "<div class=\"header__form\">\n"
+ "
<label for=\"\" class=\"header__text\"> Здравствуйте
пользователь</label>\n"

```

```

        + "
class=\"header__input input\"></div>");
        }
        request.getRequestDispatcher("/index.jsp").forward(request, response);
    }
    public static void write(String fileName, String text) {
        File file = new File(fileName);

        try {
            if(!file.exists()){
                file.createNewFile();
            }
            FileWriter out = new FileWriter(file.getAbsolutePath(), false);

            try {
                out.write(text);
            } finally {
                out.close();
            }
        } catch(IOException e) {
            throw new RuntimeException(e);
        }
    }
    public boolean IsDouble(String d) {
        try {
            Double.parseDouble(d);
            return true;
        } catch (NumberFormatException e) {
            return false;
        }
    }
}

```



## Программный код. Класс CalculationC.java

```

package kurs25;
import javax.swing.JLabel;
import javax.swing.JOptionPane;
public class CalculationC extends Calculator {
    double res;
    public CalculationC() {

        res = FlodPoint(Calculator());
    }
    public static double FlodPoint(double value) {
        value = Math.round(value * 100);
        value = value/100;
        return value;
    }

    @Override
    public Double Calculator() {
        double sum = Calc.RequestCalc.first_result - Calc.RequestCalc.second_result;
        rate = 7.3;
        if (Strahovka().equals("Exist")) {
            rate = rate - 1;
        }
        rate = rate/12/100/12;
        if (Calc.RequestCalc.data_result == 0 || sum == 0) {
            return (double) 0;
        } else {
            pay = rate * Math.pow(1+rate, Calc.RequestCalc.data_result)/(Math.pow(1+rate,
Calc.RequestCalc.data_result)-1)*(Calc.RequestCalc.first_result+Calc.RequestCalc.second_result);
        }
        return pay;
    }

    @Override
    public boolean IsDouble(String d) {
        // TODO Auto-generated method stub
        return false;
    }

    @Override
    public boolean isInteger(String d) {
        // TODO Auto-generated method stub
        return false;
    }

    public String Strahovka() {
        String info = (String) Calc.RequestCalc.strahovka;
        return info;
    };
}

```

## Программный код. Класс CalculationNH.java

```

package kurs25;
import javax.swing.JLabel;
import javax.swing.JOptionPane;
public class CalculationNH extends Calculator {
    double res;
    public CalculationNH() {

        res = FlodPoint(Calculator());
    }
    public static double FlodPoint(double value) {
        value = Math.round(value * 100);
        value = value/100;
        return value;
    }

    @Override
    public Double Calculator() {
        double sum = Calc.RequestCalc.first_result - Calc.RequestCalc.second_result;
        rate = 8.6;
        if (Strahovka().equals("Exist")) {
            rate = rate - 1;
        }
        rate = rate/12/100/12;
        if (Calc.RequestCalc.data_result == 0 || sum == 0) {
            return (double) 0;
        } else {
            pay = rate * Math.pow(1+rate, Calc.RequestCalc.data_result)/(Math.pow(1+rate,
Calc.RequestCalc.data_result)-1)*(Calc.RequestCalc.first_result+Calc.RequestCalc.second_result);
        }
        return pay;
    }

    @Override
    public boolean IsDouble(String d) {
        // TODO Auto-generated method stub
        return false;
    }

    @Override
    public boolean isInteger(String d) {
        // TODO Auto-generated method stub
        return false;
    }

    public String Strahovka() {
        String info = (String) Calc.RequestCalc.strahovka;
        return info;
    };
}

```

## Программный код. Класс CalculationRH.java

```

package kurs25;
import javax.swing.JLabel;
import javax.swing.JOptionPane;
public class CalculationRH extends Calculator {
    double res;
    public CalculationRH() {

        res = FlodPoint(Calculator());
    }
    public static double FlodPoint(double value) {
        value = Math.round(value * 100);
        value = value/100;
        return value;
    }

    @Override
    public Double Calculator() {
        double sum = Calc.RequestCalc.first_result - Calc.RequestCalc.second_result;
        rate = 8.3;
        if (Strahovka().equals("Exist")) {
            rate = rate - 1;
        }
        rate = rate/12/100/12;
        if (Calc.RequestCalc.data_result == 0 || sum == 0) {
            return (double) 0;
        } else {
            pay = rate * Math.pow(1+rate, Calc.RequestCalc.data_result)/(Math.pow(1+rate,
Calc.RequestCalc.data_result)-1)*(Calc.RequestCalc.first_result+Calc.RequestCalc.second_result);
        }
        return pay;
    }
    @Override
    public boolean IsDouble(String d) {
        // TODO Auto-generated method stub
        return false;
    }

    @Override
    public boolean isInteger(String d) {
        // TODO Auto-generated method stub
        return false;
    }

    public String Strahovka() {
        String info = (String) Calc.RequestCalc.strahovka;
        return info;
    };
}

```

## Программный код. Класс CreatePDF.java

```

package kurs25;

import java.awt.Desktop;
import java.io.File;
import java.io.FileNotFoundException;
import java.io.FileOutputStream;
import java.io.IOException;
import java.net.MalformedURLException;
import java.net.URL;
import java.net.URLDecoder;
import java.util.stream.Stream;

import javax.servlet.jsp.PageContext;

import com.itextpdf.text.BadElementException;
import com.itextpdf.text.BaseColor;
import com.itextpdf.text.Document;
import com.itextpdf.text.DocumentException;
import com.itextpdf.text.Font;
import com.itextpdf.text.FontFactory;
import com.itextpdf.text.Image;
import com.itextpdf.text.Paragraph;
import com.itextpdf.text.Phrase;
import com.itextpdf.text.pdf.BaseFont;
import com.itextpdf.text.pdf.PdfPCell;
import com.itextpdf.text.pdf.PdfPTable;
import com.itextpdf.text.pdf.PdfWriter;

public class CreatePDF {

    public CreatePDF() {

    }

    public void Create(String numberpdf) throws IOException {

        Document document = new Document(); //создание класса Document

        String filepath =
CreatePDF.class.getProtectionDomain().getCodeSource().getLocation().getPath();

        File currentClass = new File(URLDecoder.decode(CreatePDF.class
        .getProtectionDomain()
        .getCodeSource()
        .getLocation()
        .getPath(), "UTF-8"));

        filepath = currentClass.getParent();
        File currentClass2 = new File(URLDecoder.decode(filepath, "UTF-8"));
        filepath = currentClass2.getParent();

        filepath=filepath + "/Check.pdf";
        try {
            PdfWriter.getInstance(document, new FileOutputStream(filepath));
        } catch (FileNotFoundException | DocumentException e) {
            e.printStackTrace();
        }
    }
}

```

```

Font fontpath =FontFactory.getFont(filepath + "/times.ttf");
document.open();

BaseFont times =fontpath.getBaseFont() ;

String string_pdf = "Result for Calculate";
Paragraph paragraph = new Paragraph();
paragraph.add(new Paragraph(string_pdf, new Font(times,14)));

String string_pdf2 = "";
paragraph.add(new Paragraph(string_pdf2, new Font(times,14)));

try {
    document.add(paragraph);
} catch (DocumentException e1) {
    e1.printStackTrace();
}

//организация перехода на следующую строку
paragraph.clear();
String string_pdf3 = " ";
paragraph.add(new Paragraph(string_pdf3, new Font(times,14)));

try {
    document.add(paragraph);
} catch (DocumentException e1) {
    e1.printStackTrace();
}

//организация перехода на следующую строку
paragraph.clear();
paragraph.add(new Paragraph(string_pdf3, new Font(times,14)));

try {
    document.add(paragraph);
} catch (DocumentException e1) {
    e1.printStackTrace();
}

//добавление таблицы
PdfPTable table = new PdfPTable(7); //создание таблицы с 4 столбцами
addHeader(table);
addRows(table, times);

try {
    document.add(table);
} catch (DocumentException e) {
    e.printStackTrace();
}

document.close(); //закрытие и сохранение документа PDF
}

private void addRows(PdfPTable table, BaseFont BaseFont) {

    //заполнение таблицы вводимыми значения в текстовые поля на главной форме
    Phrase cell1 = new Phrase(Calc.creditAmount,new Font(BaseFont,14));
    Phrase cell2 = new Phrase(Calc.firstPayPDF,new Font(BaseFont,14));
    Phrase cell3 = new Phrase(Calc.paymentPerMonth,new Font(BaseFont,14));

```

```

Phrase cell4 = new Phrase(Calc.purposeOfTheLoan,new Font(BaseFont,14));
Phrase cell5 = new Phrase(Calc.currencyPDF,new Font(BaseFont,14));
Phrase cell6 = new Phrase(Calc.dataPDF,new Font(BaseFont,14));
Phrase cell7 = new Phrase(Calc.strahovkaPDF,new Font(BaseFont,14));

```

```

        table.addCell(cell1);
        table.addCell(cell2);
        table.addCell(cell3);
        table.addCell(cell4);
        table.addCell(cell5);
        table.addCell(cell6);
        table.addCell(cell7);

```

```

    }

```

```

private void addHeader(PdfPTable table) {

```

```

    Stream.of("credit amount", "first pay", "payment per month", "purpose of the loan", "currency", "data",
"insurance")

```

```

    .forEach(columnTitle -> {
        PdfPCell header = new PdfPCell();
        header.setBackgroundColor(BaseColor.LIGHT_GRAY);
        header.setBorderWidth(2);
        header.setPhrase(new Phrase(columnTitle));
        table.addCell(header);
    });

```

```

}
}

```

Программный код. Абстрактный класс Calculator.java

```
package kurs25;

import javax.swing.JButton;
import javax.swing.JFrame;
import javax.swing.JPanel;

public abstract class Calculator implements Check {
    double rate;
    double sum;
    int countMonth;
    double pay;

    public abstract Double Calculator();
}
```

Программный код. Интерфейс Check.java

```
package kurs25;  
  
public interface Check{  
    public boolean IsDouble(String d);  
    public boolean isInteger(String d);  
    public String Strahovka();  
}
```



## UNIT-тесты. TestCheck.java

```
package kurs25;

import static org.junit.Assert.*;

import org.junit.BeforeClass;
import org.junit.Test;

public class TestCheck implements Check{

    @BeforeClass
    public static void setUpBeforeClass() throws Exception {
    }

    @Test
    public void test() {
        assertTrue(IsDouble("5"));
    }

    //класс проверяет строку на наличие только цифр, на цифровой тип Double
    @Override
    public boolean IsDouble(String d) {
        try {
            Double.parseDouble(d);
            return true;
        } catch (NumberFormatException e) {
            return false;
        }
    }

    @Override
    public boolean isInteger(String d) {
        // TODO Auto-generated method stub
        return false;
    }

    @Override
    public String Strahovka() {
        // TODO Auto-generated method stub
        return null;
    }

}
```

## UNIT-тесты. TestFlodPoint.java

```
package kurs25;

import static org.junit.Assert.*;

import org.junit.BeforeClass;
import org.junit.Test;

public class TestFlodPoint {
    static boolean correct = false;
    @BeforeClass
    public static void setUpBeforeClass() throws Exception {
        double value = 74.56767876456;
        double out = CalculationC.FlodPoint(value);
        if (out == 74.57) {
            correct = true;
        }
    }

    @Test
    public void test() {
        assertTrue(correct);
    }
}
```

## UNIT-тесты. TestToChange.java

```

package kurs25;

import static org.junit.Assert.*;

import java.io.BufferedReader;
import java.io.File;
import java.io.FileReader;
import java.io.FileWriter;
import java.io.IOException;

import org.junit.BeforeClass;
import org.junit.Test;

public class TestToChange {
    static boolean status = false;

    @BeforeClass
    public static void setUpBeforeClass() throws Exception {
    }

    @Test
    public void test() throws IOException {

        write("password", "123");

    }

    public static void write(String fileName, String text) {
        //Определяем файл
        File file = new File(fileName);

        try {
            //проверяем, что если файл не существует то создаем его
            if(!file.exists()){
                file.createNewFile();
            }

            //PrintWriter обеспечит возможности записи в файл
            FileWriter out = new FileWriter(file.getAbsolutePath(), false);

            try {
                //Записываем текст у файл
                out.write(text);
            } finally {
                //После чего мы должны закрыть файл
                //Иначе файл не запишется
                out.close();
            }
        }
        BufferedReader br = new BufferedReader(new FileReader(fileName));
        String word;
        word = br.readLine();
        if (word.equals("123")) {
            status = true;
            assertTrue(status);
        }
        catch(IOException e) {
            throw new RuntimeException(e);}
    }
}

```

## Веб-страница author.jsp

```

<% @ page language="java" contentType="text/html; charset=UTF-8"
    pageEncoding="UTF-8"% >
<html>
<head>
    <meta charset="Cp1251">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Авторизация</title>
    <link rel="stylesheet" type="text/css" href="css/reset.css">
    <link rel="stylesheet" type="text/css" href="css/style.css">
</head>
<body>
    <header class="header container">
        <h1 class="header__name">Авторизация</h1>
    </header>
    <div class="main">
        <form action="{pageContext.request.contextPath}/JavaAdmin" method="post" class="main__forms">
            <div class="login__form">
                <label for="login" class="main__text">Введите логин: </label>
                <input type="text" name="login" id="login" value="{login}" class="main__input">
                <label for="password" class="main__text">Введите пароль: </label>
                <input type="text" name="password" id="password" value="{password}" class="main__input">
                <input type="submit" name="sign" value="Ввод" class="main__submit">
            </div>
        </form>
        <h2 class="header__incorrect">{incorrect}</h2>
    </div>
</body>
</html>

```

## Веб-страница changes.jsp

```

<% @ page language="java" contentType="text/html; charset=UTF-8"
    pageEncoding="UTF-8"% >
<html>
<head>
    <meta charset="Cp1251">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Изменение ставки</title>
    <link rel="stylesheet" type="text/css" href="css/reset.css">
    <link rel="stylesheet" type="text/css" href="css/style.css">
</head>
<body>
    <header class="header container">
        <h1 class="header__name">Измена ставки по кредиту</h1>
    </header>
    <div class="main">
        <form action="{pageContext.request.contextPath}/JavaChange" method="post" class="main__forms">
            <div class="main__form">
                <label for="size" class="main__text">Новая ставка готового жилья ({stavkaRH}) : </label>
                <input type="text" name="StReadyHouse" id="StReadyHouse" value="{StReadyHouse}"
class="main__input">
                <label for="size" class="main__text">Новая ставка новостройки ({stavkaNH}) : </label>
                <input type="text" name="StNewHouse" id="StNewHouse" value="{StNewHouse}"
class="main__input">
                <label for="size" class="main__text">Новая ставка коммерческая ({stavkaC}) : </label>
                <input type="text" name="StComercial" id="StComercial" value="{StComercial}"
class="main__input">
                <input type="submit" name="sign" value="Ввод" class="main__submit">
            </div>
        </form>
    </div>
</body>
</html>

```

## Веб-страница index.jsp

```

<% @ page language="java" contentType="text/html; charset=UTF-8"
    pageEncoding="UTF-8"% >
<!DOCTYPE html>
<html>
<head>
<meta charset="Cp1251">
<title>Ипотечный калькулятор для юридических лиц</title>
<link rel="stylesheet" type="text/css" href="css/reset.css">
<link rel="stylesheet" type="text/css" href="css/style.css">
</head>
<body>

    <header class="header container" >
    <form action="{pageContext.request.contextPath}/JavaExit" method="post" class="header__admin">
        ${admin}
    </form>

    <h1 class="header__name">Ипотечный калькулятор для юридических лиц</h1>
</header>
<div class="main">
    <form action="{pageContext.request.contextPath}/JavaCalc" method="post" class="main__forms">
        <div class="main__form">
            <label for="size" class="main__text">Размер ипотеки: </label>
            <input type="text" name="sizeMortgage" id="sizeMortgage" value="{sizeMortgage}"
class="main__input">
            <label for="firstPay" class="main__text">Первый взнос: </label>
            <input type="text" name="firstPay" id="firstPay" value="{firstPay}" class="main__input">
            <label for="data" class="main__text">Количество месяцев: </label>
            <input type="text" name="data" id="data" value="{data}" class="main__input">
            <label for="" class="main__text">Валюта </label>
            <select name="currency" id="currency" value="{currency}">
                <option value="dollar">Доллар</option>
                <option value="rub">Рубль</option>
                <option value="euro">Евро</option>
            </select>
            <label for="" class="main__text">Страховка </label>
            <select name="strahovka" id="strahovka" value="{strahovka}">
                <option value="No">Нет</option>
                <option value="Exist">Есть</option>
            </select>
            <label for="" class="main__text">Цель кредита </label>
            <select name="target" id="target" value="{target}">
                <option value="ready house">Готовое жилье</option>
                <option value="new house">Новостройка</option>
                <option value="commercial">Коммерческая</option>
            </select>
            <input type="submit" name="sign" value="Рассчитать" class="main__submit">
        </div>
    </form>
</div>
<div class="changes container" >
    <div class="changes__changes">
        <p style="color:red">${incChanges}</p>
        <form action="{pageContext.request.contextPath}/JavaToChange" method="post"
class="changes__form">

```

```
        ${changes}
    </form>
</div>
</div>

</body>
</html>
```

## Веб-страница output.jsp

```

<% @ page language="java" contentType="text/html; charset=UTF-8"
    pageEncoding="UTF-8"% >
<!DOCTYPE html>
<html>
<head>
    <meta charset="Cp1251">
    <title>Ипотечный калькулятор для юридических лиц</title>
    <link rel="stylesheet" type="text/css" href="css/reset.css">
    <link rel="stylesheet" type="text/css" href="css/style.css">
</head>
<body>
<header class="header container">
    <h1 class="header__name">Результаты вычислений</h1>
</header>
<div class="result container">
    <p class="result__text text"><strong>Размер ипотеки: </strong>${first_result}</p>
    <p class="result__text text"><strong>Первый взнос: </strong>${second_result}</p>
    <p class="result__text text"><strong>Количество месяцев: </strong>${data_result}</p>
    <p class="result__text text"><strong>Валюта: </strong>${currency_result}</p>
    <p class="result__text text"><strong>Страховка: </strong>${strahovka_result}</p>
    <p class="result__text text"><strong>Цель кредита: </strong>${target_result}</p>
    <p class="result__result text">Результат: ${result} ${currency_for_result}</p>
    <form action="${pageContext.request.contextPath}/Back" method="post" class="result__form">
        <input type="submit" name="sign" value="Назад">
    </form>
    <a href="Check.pdf">Открыть PDF-файл</a>
    <a href="Check.pdf" download>Скачать PDF-файл</a>
</div>
</body>
</html>

```



## Стиль reset.css

```

/* http://meyerweb.com/eric/tools/css/reset/
   v2.0-modified | 20110126
   License: none (public domain)
*/

html, body, div, span, applet, object, iframe,
h1, h2, h3, h4, h5, h6, p, blockquote, pre,
a, abbr, acronym, address, big, cite, code,
del, dfn, em, img, ins, kbd, q, s, samp,
small, strike, strong, sub, sup, tt, var,
b, u, i, center,
dl, dt, dd, ol, ul, li,
fieldset, form, label, legend,
table, caption, tbody, tfoot, thead, tr, th, td,
article, aside, canvas, details, embed,
figure, figcaption, footer, header, hgroup,
menu, nav, output, ruby, section, summary,
time, mark, audio, video {
    margin: 0;
    padding: 0;
    border: 0;
    font-size: 100%;
    font: inherit;
    vertical-align: baseline;
}

/* make sure to set some focus styles for accessibility */
:focus {
    outline: 0;
}

/* HTML5 display-role reset for older browsers */
article, aside, details, figcaption, figure,
footer, header, hgroup, menu, nav, section {
    display: block;
}

body {
    line-height: 1;
}

ol, ul {
    list-style: none;
}

blockquote, q {
    quotes: none;
}

blockquote:before, blockquote:after,
q:before, q:after {
    content: "";
    content: none;
}

table {
    border-collapse: collapse;
}

```

```

        border-spacing: 0;
    }

    input[type=search]::-webkit-search-cancel-button,
    input[type=search]::-webkit-search-decoration,
    input[type=search]::-webkit-search-results-button,
    input[type=search]::-webkit-search-results-decoration {
        -webkit-appearance: none;
        -moz-appearance: none;
    }

    input[type=search] {
        -webkit-appearance: none;
        -moz-appearance: none;
        -webkit-box-sizing: content-box;
        -moz-box-sizing: content-box;
        box-sizing: content-box;
    }

    textarea {
        overflow: auto;
        vertical-align: top;
        resize: vertical;
    }

/**
 * Correct `inline-block` display not defined in IE 6/7/8/9 and Firefox 3.
 */

audio,
canvas,
video {
    display: inline-block;
    *display: inline;
    *zoom: 1;
    max-width: 100%;
}

/**
 * Prevent modern browsers from displaying `audio` without controls.
 * Remove excess height in iOS 5 devices.
 */

audio:not([controls]) {
    display: none;
    height: 0;
}

/**
 * Address styling not present in IE 7/8/9, Firefox 3, and Safari 4.
 * Known issue: no IE 6 support.
 */

[hidden] {
    display: none;
}

/**
 * 1. Correct text resizing oddly in IE 6/7 when body `font-size` is set using
 *    `em` units.
 * 2. Prevent iOS text size adjust after orientation change, without disabling
 *    user zoom.
 */

```

```

html {
    font-size: 100%; /* 1 */
    -webkit-text-size-adjust: 100%; /* 2 */
    -ms-text-size-adjust: 100%; /* 2 */
}

/**
 * Address `outline` inconsistency between Chrome and other browsers.
 */

a:focus {
    outline: thin dotted;
}

/**
 * Improve readability when focused and also mouse hovered in all browsers.
 */

a:active,
a:hover {
    outline: 0;
}

/**
 * 1. Remove border when inside `a` element in IE 6/7/8/9 and Firefox 3.
 * 2. Improve image quality when scaled in IE 7.
 */

img {
    border: 0; /* 1 */
    -ms-interpolation-mode: bicubic; /* 2 */
}

/**
 * Address margin not present in IE 6/7/8/9, Safari 5, and Opera 11.
 */

figure {
    margin: 0;
}

/**
 * Correct margin displayed oddly in IE 6/7.
 */

form {
    margin: 0;
}

/**
 * Define consistent border, margin, and padding.
 */

fieldset {
    border: 1px solid #c0c0c0;
    margin: 0 2px;
    padding: 0.35em 0.625em 0.75em;
}

/**
 * 1. Correct color not being inherited in IE 6/7/8/9.
 * 2. Correct text not wrapping in Firefox 3.

```

```
* 3. Correct alignment displayed oddly in IE 6/7.
*/
```

```
legend {
    border: 0; /* 1 */
    padding: 0;
    white-space: normal; /* 2 */
    *margin-left: -7px; /* 3 */
}
```

```
/**
 * 1. Correct font size not being inherited in all browsers.
 * 2. Address margins set differently in IE 6/7, Firefox 3+, Safari 5,
 *    and Chrome.
 * 3. Improve appearance and consistency in all browsers.
 */
```

```
button,
input,
select,
textarea {
    font-size: 100%; /* 1 */
    margin: 0; /* 2 */
    vertical-align: baseline; /* 3 */
    *vertical-align: middle; /* 3 */
}
```

```
/**
 * Address Firefox 3+ setting `line-height` on `input` using `!important` in
 * the UA stylesheet.
 */
```

```
button,
input {
    line-height: normal;
}
```

```
/**
 * Address inconsistent `text-transform` inheritance for `button` and `select`.
 * All other form control elements do not inherit `text-transform` values.
 * Correct `button` style inheritance in Chrome, Safari 5+, and IE 6+.
 * Correct `select` style inheritance in Firefox 4+ and Opera.
 */
```

```
button,
select {
    text-transform: none;
}
```

```
/**
 * 1. Avoid the WebKit bug in Android 4.0.* where (2) destroys native `audio`
 *    and `video` controls.
 * 2. Correct inability to style clickable `input` types in iOS.
 * 3. Improve usability and consistency of cursor style between image-type
 *    `input` and others.
 * 4. Remove inner spacing in IE 7 without affecting normal text inputs.
 *    Known issue: inner spacing remains in IE 6.
 */
```

```
button,
html input[type="button"], /* 1 */
input[type="reset"],
input[type="submit"] {
```

```

    -webkit-appearance: button; /* 2 */
    cursor: pointer; /* 3 */
    *overflow: visible; /* 4 */
}

/**
 * Re-set default cursor for disabled elements.
 */

button[disabled],
html input[disabled] {
    cursor: default;
}

/**
 * 1. Address box sizing set to content-box in IE 8/9.
 * 2. Remove excess padding in IE 8/9.
 * 3. Remove excess padding in IE 7.
 *    Known issue: excess padding remains in IE 6.
 */

input[type="checkbox"],
input[type="radio"] {
    box-sizing: border-box; /* 1 */
    padding: 0; /* 2 */
    *height: 13px; /* 3 */
    *width: 13px; /* 3 */
}

/**
 * 1. Address `appearance` set to `searchfield` in Safari 5 and Chrome.
 * 2. Address `box-sizing` set to `border-box` in Safari 5 and Chrome
 *    (include `-moz` to future-proof).
 */

input[type="search"] {
    -webkit-appearance: textfield; /* 1 */
    -moz-box-sizing: content-box;
    -webkit-box-sizing: content-box; /* 2 */
    box-sizing: content-box;
}

/**
 * Remove inner padding and search cancel button in Safari 5 and Chrome
 * on OS X.
 */

input[type="search"]::-webkit-search-cancel-button,
input[type="search"]::-webkit-search-decoration {
    -webkit-appearance: none;
}

/**
 * Remove inner padding and border in Firefox 3+.
 */

button::-moz-focus-inner,
input::-moz-focus-inner {
    border: 0;
    padding: 0;
}

/**

```

```
* 1. Remove default vertical scrollbar in IE 6/7/8/9.  
* 2. Improve readability and alignment in all browsers.  
*/
```

```
textarea {  
    overflow: auto; /* 1 */  
    vertical-align: top; /* 2 */  
}
```

```
/**  
 * Remove most spacing between table cells.  
 */
```

```
table {  
    border-collapse: collapse;  
    border-spacing: 0;  
}
```

```
html,  
button,  
input,  
select,  
textarea {  
    color: #222;  
}
```

```
::-moz-selection {  
    background: #b3d4fc;  
    text-shadow: none;  
}
```

```
::selection {  
    background: #b3d4fc;  
    text-shadow: none;  
}
```

```
img {  
    vertical-align: middle;  
}
```

```
fieldset {  
    border: 0;  
    margin: 0;  
    padding: 0;  
}
```

```
textarea {  
    resize: vertical;  
}
```

```
.chromeframe {  
    margin: 0.2em 0;  
    background: #ccc;  
    color: #000;  
    padding: 0.2em 0;  
}
```

## Стиль style.css

```
html {  
    height: 100%;  
}  
body {  
    background: linear-gradient(-45deg,#66ffcc,#ffffff);  
}  
.container {  
    max-width: 1020px;  
    margin: 0 auto;  
}  
.header {  
    max-width: 1020px;  
    margin: 40px auto;  
}  
    .header__name {  
        text-align: center;  
        font-size: 40px;  
    }  
    .header__admin {}  
    .header__form {  
        width: 400px;  
        margin-left: 100%;  
        transform: translate(-100%);  
    }  
.input {  
    background-color: transparent;  
    border: solid 1px gray;  
    border-radius: 20px;  
    font-size: 14px;  
}  
.input:hover {  
    background-color: gray;  
}  
.main {  
    margin: 0 auto;  
}  
.text {  
    font-size: 20px;  
}  
.main__forms {  
    max-width: 500px;  
    background-color: #f7f7f7;  
    border: 1px solid #7d7d7d;  
    margin: 0 auto;  
    max-height: 100%;  
    justify-content: center;  
    border-radius: 20px;  
    padding: 20px 20px;  
}  
.main__form {  
    display: grid;  
    grid-template-columns: repeat(2,1fr);  
    grid-row-gap: 10px;
```

```

}
.main__text {
    min-width: 150px;
}
.main__input {
    min-width: 150px;
    border-radius: 5px;
}
.main__input[type=text] {
    padding-left: 10px;
}

.main__submit {
    grid-column-start: 1;
    grid-column-end: 3;
}
.result {
    background-color: #f7f7f7;
    border: 1px solid #7d7d7d;
    display: grid;
    max-width: 800px;
    grid-template-columns: repeat(2, 1fr);
    border-radius: 20px;
    grid-row-gap: 10px;
    padding: 20px 20px;
}
.result__text {}
.result__result {
    text-align: center;
    grid-column-start: 1;
    grid-column-end: 3;
}
.result__form {
    text-align: right;
    grid-column-start: 1;
    grid-column-end: 3;
}
.changes {
    margin-top: 100px;
    align-items: center;
}
.changes__changes {
    max-width: 300px;
    margin: 0 auto;
}
.changes__form {
}
.changes__text {}
.changes__submit {}

.header__incorrect {
    color: red;
    width: 100%;
    text-align: center;
}
.login__form {
    display: grid;
    grid-template-columns: repeat(1, 1fr);
    grid-row-gap: 10px;
}

```



