

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-221
Самсонов А.А
Степанян С.А
Фахретдинова К.О
Шейфель Ж.В

Проверил:
Преподаватель
Казанцев А.В.

Уфа – 2021

Министерство науки и высшего образования РФ

Федеральное государственное бюджетное образовательное учреждение высшего образования
«УФИМСКИЙ ГОСУДАРСТВЕННЫЙ АВИАЦИОННЫЙ ТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ»

Кафедра автоматизированных систем управления

ЗАДАНИЕ

на курсовую работу по дисциплине «Информационные системы»

Студент	<u>Самсонов А.А.</u>	Группа	<u>ПИ-221</u>	Консультант	<u>Казанцев А.В.</u>
	Фамилия И.О.		номер группы		Фамилия И.О.
Студент	<u>Степанян С.А.</u>	Группа	<u>ПИ-221</u>	Консультант	<u>Казанцев А.В.</u>
	Фамилия И.О.		номер группы		Фамилия И.О.
Студент	<u>Фахретдинова К.О.</u>	Группа	<u>ПИ-221</u>	Консультант	<u>Казанцев А.В.</u>
	Фамилия И.О.		номер группы		Фамилия И.О.
Студент	<u>Шейфель Ж.В.</u>	Группа	<u>ПИ-221</u>	Консультант	<u>Казанцев А.В.</u>
	Фамилия И.О.		номер группы		Фамилия И.О.

1. Тема курсового проекта: Разработка кроссплатформенного программного продукта на языке JAVA с использованием системы контроля версий.
наименование темы

2. Основное содержание:

1. Пояснительная записка с необходимыми материалами.
2. Репозиторий системы контроля версий содержащий программный код с комментариями и необходимую документацию.
3. Требования к оформлению:

3.1. Пояснительная записка должна быть оформлена в текстовом процессоре LibreOffice Writer в соответствии с требованиями СТО УГАТУ. Минимальные требования к оформлению: размер шрифта 14 пунктов; отступы от края листа: отступ слева 2 см. и остальные отступы 0.5 см. В бумажном виде оформляются: титульный лист, задание, календарный план и аннотация, которая содержит ссылку на репозиторий с программным кодом и документацией.

3.2. В пояснительной записке должны содержаться следующие разделы:

Раздел 1. Описание предметной области.

Раздел 2. Техническое задание на создание программного продукта.

Раздел 3. Настройка среды разработки для операционных систем семейств Windows и Linux.

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

Раздел 5. Реализация исходного кода по зонам ответственности.

Раздел 6. Сборка и тестирование программного продукта.

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

Раздел 8. Руководство пользователя программного продукта.

3.3. В приложение выносятся программный код и код тестов.

4. Графическая часть должна включать:

- мнемосхема рассматриваемого процесса;
- диаграммы UML;
- экранные формы инструментальных средств;
- экранные формы, разрабатываемого программного продукта.

Дата выдачи 6 марта 2021 г.

Дата окончания 29 мая 2021 г.

Руководитель Казанцев А.В. ФИО _____

План-график выполнения курсовой работы по дисциплине «Информационные системы»

Наименование этапа работ	Трудоемкость выполнения, час	Процент к общей трудоемкости выполнения	Срок предъявления консультанту	Отметка о выполнении
Получение и согласование задания	1,7	1,7%	27 неделя	
Раздел 1. Описание предметной области	20	20%	29 неделя	
Раздел 2. Техническое задание на создание программного продукта	10	10%	30 неделя	
Раздел 3. Настройка среды разработки для операционных систем семейств Windows и Linux	10	10%	31 неделя	
Раздел 4. Настройка среды разработки для подключения к системе контроля версий	7	7%	32 неделя	
Раздел 5. Реализация исходного кода по зонам ответственности	23	23%	34 неделя	
Раздел 6. Сборка и тестирование программного продукта	8	8%	35 неделя	
Раздел 7. Настройка программной среды для развертывания и запуска программного продукта	10	10%	36 неделя	
Раздел 8. Руководство пользователя программного продукта	10	10%	37 неделя	
Защита	0,3	0,3%	38 неделя	

Содержание

Раздел 1. Описание предметной области.....	5
Раздел 2. Техническое задание на создание программного продукта.....	14
Раздел 3. Настройка среды разработки для операционных систем семейств Windows и Linux.....	15
3.1 Установка и настройка среды разработки и Eclipse IDE Enterprise на Windows 10.....	15
3.2 Установка и настройка среды разработки Eclipse IDE Enterprise на Ubuntu 20.04.....	24
3.3 Установка и настройка среды разработки Eclipse IDE Enterprise на OpenSUSE Leap 15.2.....	33
Раздел 4. Настройка среды разработки для подключения к системе контроля версий.....	43
Раздел 5. Реализация исходного кода по зонам ответственности.....	48
Раздел 6. Сборка и тестирование программного продукта.....	49
Раздел 7. Настройка программной среды для развёртывания и запуска программного продукта.....	53
Раздел 8. Руководство пользователя программного продукта.....	56
Заключение.....	57
Приложения.....	58
Список использованной литературы.....	98

Раздел 1. Описание предметной области.

Кредитование – это финансовые взаимоотношения, в которых одна сторона – кредитор, предоставляет во временное пользование ссуду в денежной или натуральной форме, а вторая сторона – заемщик, пользуется предоставленной ссудой на условиях возвратности и возмещения.

Роль кредита важна в регулировании ликвидности банковской системы, а также в создании эффективного механизма финансирования государственных расходов.

Потребительский кредит - кредит, предоставляемый непосредственно гражданам для приобретения предметов потребления. Такой кредит берут не только для покупки товаров длительного пользования (квартиры, мебель, автомобили и т. п.), но и для прочих покупок (мобильные телефоны, бытовая техника, продукты питания). Он выступает или в форме продажи товаров с отсрочкой платежа, или в форме предоставления банковской ссуды на потребительские цели.

Принято выделять три функции кредита:

— Распределительная функция - заключается в том, что благодаря кредитным отношениям временно свободные денежные средства перераспределяются в пользу тех, кто в них нуждается. Таким образом осуществляется переход активов в более эффективные отрасли экономики.

Возможно внутриотраслевое и межотраслевое перераспределение. В современном мире, когда кредитование происходит преимущественно через банковскую систему, основным является межотраслевое перераспределение средств.

— Эмиссионная функция. Вторая функция кредита заключается в создании кредитных средств обращения и замещения наличных денег.

Кредитные средства в обращении создаются не отдельным банком, а финансовой системой в целом. Этот эффект известен также под названием «банковский мультипликатор».

— Ещё одной функцией кредита является осуществление контроля за эффективностью деятельности экономических субъектов, или контрольная функция. В настоящее время существует несколько видов потребительского кредита, которые могут быть классифицированы по их различным параметрам:

а) Нецелевой. Подобные займы выдаются наличными или на кредитную карту. Важными особенностями кредитов выступает более высокая, по сравнению с целевыми, процентная ставка, а также меньший размер ссуды. Несмотря на это, по мнению многих финансовых аналитиков и специалистов рынка банковских услуг именно данный вид потребительских займов является самым распространенным;

б) Целевой. Выдается банком для приобретения конкретного товара или услуги. Нередко оформление происходит непосредственно в магазине, где заемщик уже подобрал необходимую ему покупку. Зачастую данный вид кредита применяется для оплаты различных услуг, например лечения или туристической поездки;

в) Экспресс-займ. Сравнительно новый вид кредитования, который предоставляется в большинстве случаев в режиме онлайн, выступая, по сути, усовершенствованной и более выгодной для клиента версией микрозаймов. Основными особенностями подобных ссуд выступают высокая процентная ставка и маленький размер кредита;

г) Кредитная банковская карта. Данный вариант оформления ссуды, когда заемные средства поступают на пластик клиента, стремительно набирает популярность. Причинами этого выступает простота, безопасность и удобство процедуры зачисления средств в сочетании с постоянно увеличивающимся количеством карточек, находящихся на руках физических лиц.

В данной курсовой работе будет рассматриваться классический способ кредитования — нецелевой.

Основные документы, регламентирующие потребительское кредитование:

Федеральный закон "О потребительском кредите (займе)" от 21.12.2013 N 353-ФЗ [9, с. 33]

- Статья 5. «Условия договора потребительского кредита (займа)»;
- Статья 6. «Полная стоимость потребительского кредита (займа)»;
- Статья 7. «Заключение договора потребительского кредита»;
- Статья 9. «Проценты по договору потребительского кредита (займа)».

Гражданский кодекс РФ (часть вторая) от 26.01.1996 N 14-ФЗ (ред. от 27.12.2019 с изм. от 28.04.2020):

- Статья 809. «Проценты по договору займа»;
- Статья 810. «Обязанность заемщика возвратить сумму займа»;
- Статья 811. «Последствия нарушения заемщиком договора займа»;
- Статья 819. «Кредитный договор»;
- Статья 820. «Форма кредитного договора»;
- Статья 821. «Отказ от предоставления или получения кредита».

Программный продукт «Калькулятор потребительского кредита», разрабатываемый в данной курсовой работе предназначен для самостоятельного расчёта процентной ставки и ежемесячного платежа. Также программный продукт позволяет узнать общую сумму выплат по кредиту с переплатой.

Для подробного представления исследуемого бизнес-процесса рассмотрим мнемосхему, представленную на рисунке 1.

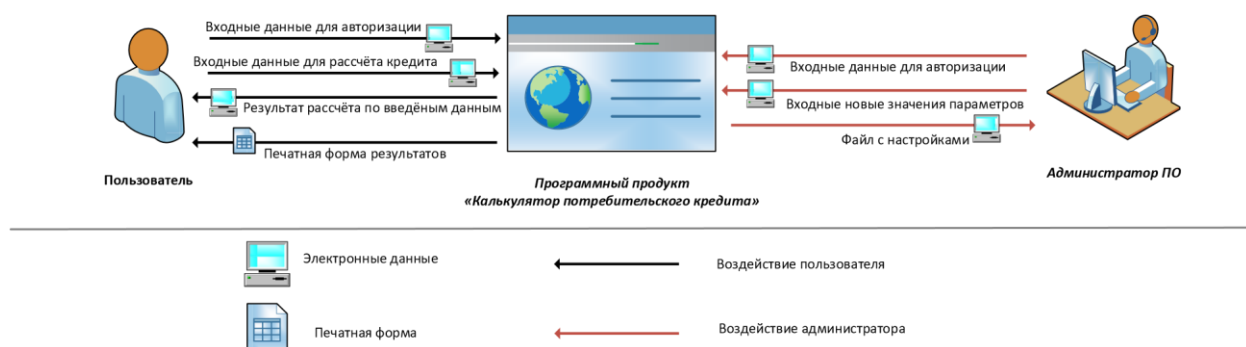


Рисунок 1 – Мнемосхема рассматриваемого бизнес-процесса.

На мнемосхеме изображён процесс расчёта потребительского кредита с помощью программного продукта. Пользователь вводит данные для авторизации: логин и пароль, выбирает условие кредита и указывает будет ли он досрочно погашать его. Далее вводит данные для расчёта: сумма кредита и срок кредитования.

После того, как программа совершит расчёты по входным данным, пользователь получит результат, который также можно получить в печатной форме.

Администратор может ввести новые значения параметров: изменить процентную ставку по кредиту, скорректировать минимальные и максимальные значения суммы и срока для расчёта кредита, предварительно пройдя авторизацию.

Функциональные возможности программного продукта отражает поведенческая диаграмма (диаграмма вариантов использования), представленная на рисунке 2.

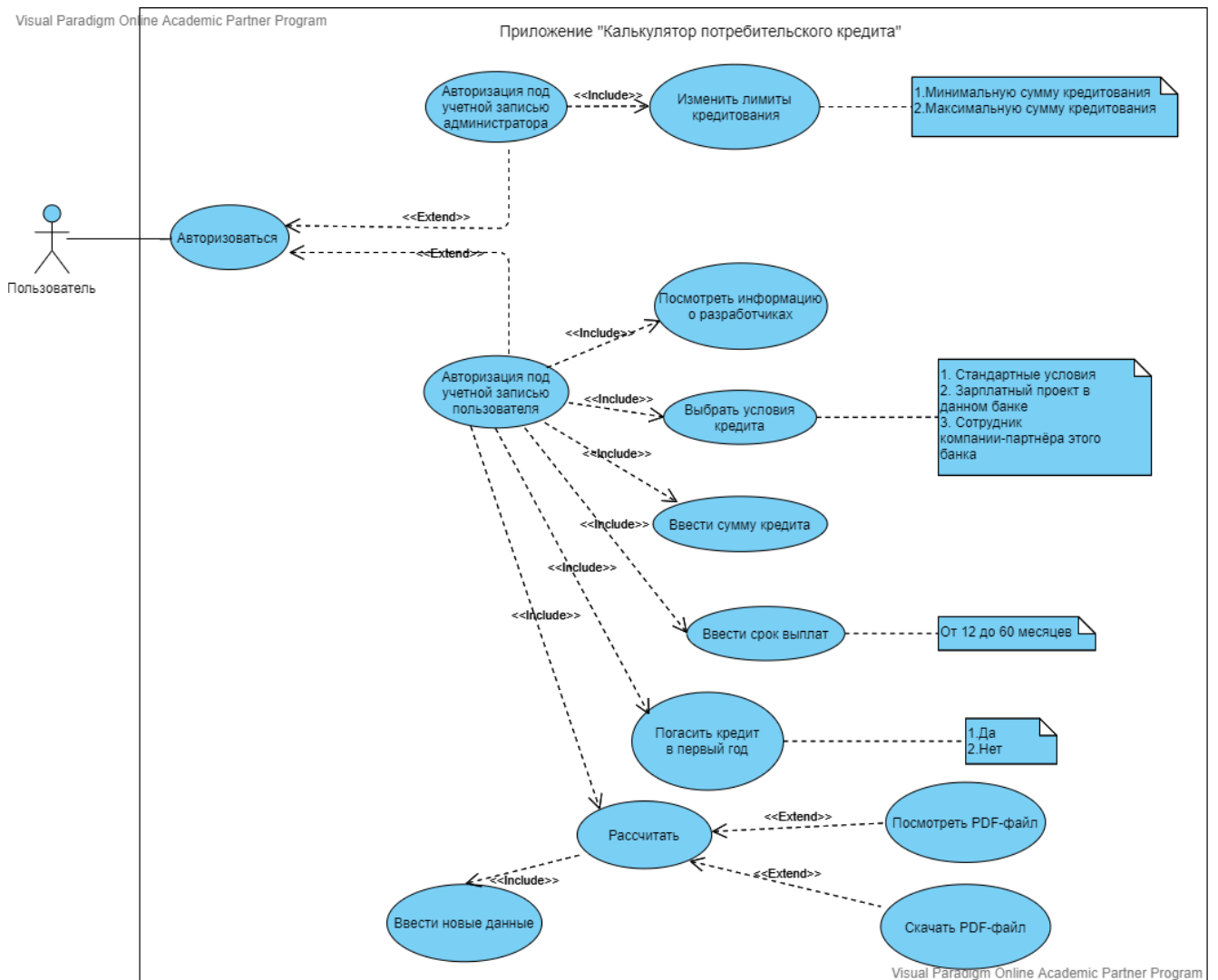


Рисунок 2 – Диаграмма вариантов использования.

Общую структуру классов системы демонстрирует структурная диаграмма (диаграмма классов), представленная на рисунке 3.

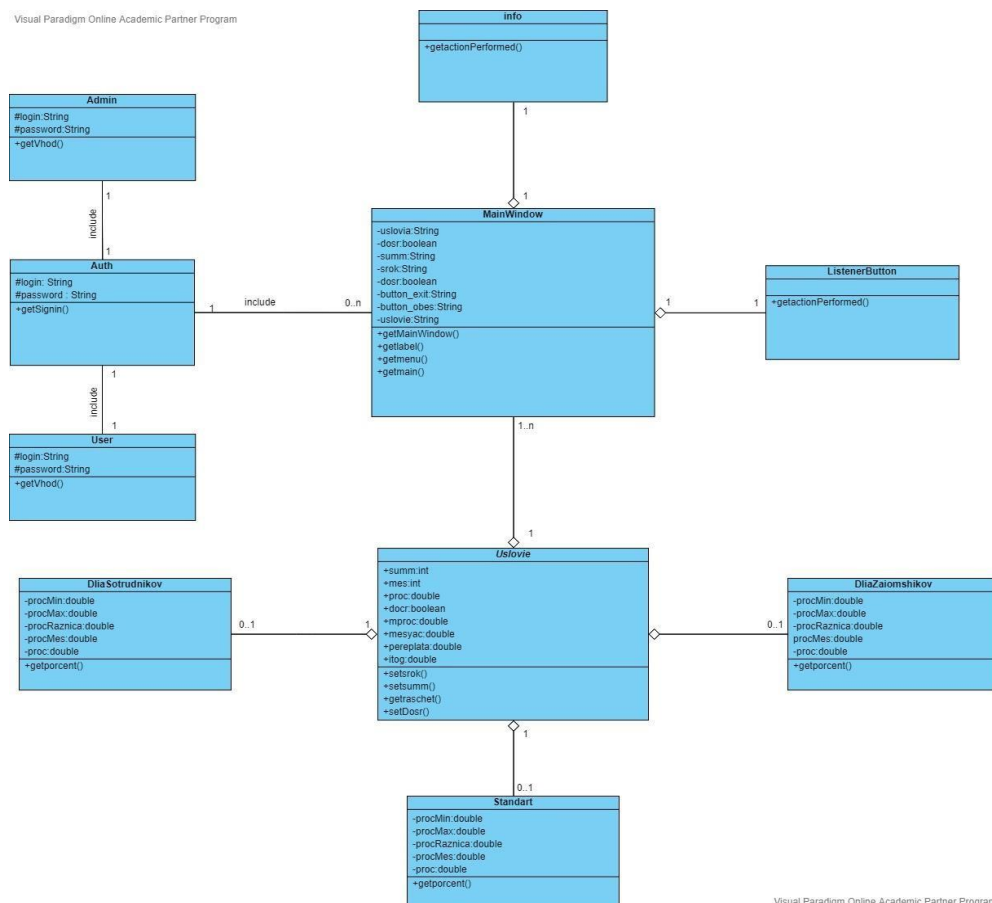


Рисунок 3 – Структурная диаграмма

Описание входной и выходной информации.

Входная информация относительно всех пользователей:

- Данные авторизации: логин и пароль.

Входная информация относительно администратора:

- Значения новых параметров (для корректировки коэффициентов процентной ставки, для изменения минимальных и максимальных даты суммы на кредитование).

Выходная информация относительно администратора:

- Файл с настройками.

Входная информация относительно простого пользователя:

- Данные для расчёта: условие для кредитования (стандартные/ зарплатный проект/ условия для партнёров банка), сумма кредита, срок кредитования.

Выходная информация относительно простого пользователя:

- Результаты расчётов: процентная ставка, ежемесячный платёж, переплата по кредиту, общая сумма с переплатой;
- Результаты расчётов в печатной форме.

Математическая модель программного продукта.

В данной курсовой работе ежемесячная сумма выплат будет рассчитываться с помощью аннуитетного платежа.

Для начала необходимо назначить ежегодный процент с учётом различных факторов. Ставка нашего банка определяется исходя из ключевой ставки ЦБ РФ 4,25%. Процент высчитывается относительно выбора условия кредита: стандартные условия, держу зарплатный проект в данном банке, являюсь сотрудником компании-партнёров этого банка.

1. Стандартные условия:

Если сумма кредита составит от 50000 до 250000 рублей, то минимальный процент составит 16,99%, максимальный 23,49%.

Если сумма кредита составит от 250000 до 700000 рублей, то минимальный процент составит 14,99%, максимальный 18,99%.

Если сумма кредита составит от 700000 до 3000000 рублей, то минимальный процент составит 11,99%, максимальный 18,99%.

2. Держу зарплатный проект в данном банке:

Если сумма кредита составит от 50000 до 250000 рублей, то минимальный процент составит 13,49%, максимальный 23,49%.

Если сумма кредита составит от 250000 до 700000 рублей, то минимальный процент составит 12,99%, максимальный 18,99%.

Если сумма кредита составит от 700000 до 3000000 рублей, то минимальный процент составит 11,99%, максимальный 18,99%.

3. Являюсь сотрудником компании-партнёров этого банка:

Если сумма кредита составит от 50000 до 250000 рублей, то минимальный процент составит 15,99%, максимальный 23,49%.

Если сумма кредита составит от 250000 до 700000 рублей, то минимальный процент составит 13,99%, максимальный 18,99%.

Если сумма кредита составит от 700000 до 3000000 рублей, то минимальный процент составит 11,99%, максимальный 18,99%.

Также заёмщик может выбрать условие «Не планирую досрочно погашать кредит в первый год». Данное условие позволит уменьшить процентную ставку на две единицы.

После выбора условия идут вычисления итоговой процентной ставки по формуле (1) и (2).

Вычисления месячного процента для вычисления итоговой процентной ставки по формуле (1).

$$\text{ПРОЦ}_M = \frac{\text{ПРОЦ}_{\max} - \text{ПРОЦ}_{\min}}{48} \quad (1)$$

где ПРОЦ_M — срок кредитования в месяцах;

ПРОЦ_{\max} — максимальная процентная ставка;

ПРОЦ_{\min} — минимальная процентная ставка.

После высчитывается итоговый процент по формуле (2).

$$\text{ПРОЦ} = \text{ПРОЦ}_{\max} - \text{ПРОЦ}_{\min} \cdot (M - 12) \quad (2)$$

где, ПРОЦ_{\max} — максимальная процентная ставка;

ПРОЦ — итоговая процентная ставка;

M — срок кредитования в месяцах.

Чтобы воспользоваться аннуитетным платежом, помимо процентной ставки, необходимо рассчитать также ежемесячную процентную ставку. Формула ежемесячной процентной ставки рассчитывается по формуле (3).

$$\text{ПРОЦ}_{\text{мес}} = \frac{\text{ПРОЦ}}{1200} \quad (3)$$

где $\text{ПРОЦ}_{\text{мес}}$ — ежемесячный процент;

ПРОЦ — итоговая процентная ставка.

После нахождения ежемесячного процента, можно рассчитать ежемесячный аннуитетный платёж с помощью формулы (4).

$$\text{ПЛ} = \text{СУМ}_3 \cdot \left(\text{ПРОЦ}_{\text{мес}} + \frac{\text{ПРОЦ}_{\text{мес}}}{(1 + \text{ПРОЦ}_{\text{мес}})^M} - 1 \right) \quad (4)$$

где ПЛ — ежемесячный аннуитетный платёж, руб.;

СУМ_3 — сумма займа, руб.;

$\text{ПРОЦ}_{\text{мес}}$ — ежемесячный процент;

M — срок кредитования в месяцах.

Затем необходимо найти переплату по кредиту, используя ежемесячный аннуитетный платёж с помощью формулы (5).

$$\text{ПЕР} = M \cdot \text{ПЛ} - \text{СУМ}_3 \quad (5)$$

где ПЕР — переплата по кредиту, руб.;

M — срок кредитования в месяцах;

ПЛ — ежемесячный аннуитетный платёж, руб.;

СУМ_3 — сумму займа.

В итоге для нахождения общей суммы долга необходимо сложить переплату по кредиту и сумму займа используя формулу (6).

$$\text{СУМ}_{\text{общ}} = \text{СУМ}_3 + \text{ПЕР} \quad (6)$$

$\text{СУМ}_{\text{общ}}$ — сумма кредита, руб.;

СУМ_3 — сумма займа, руб.;

ПЕР — переплата по кредиту, руб.

Раздел 2. Техническое задание на создание программного продукта

Техническое задание для рассмотрения и изучения представлено в приложении А-1.

Раздел 3. Настройка среды разработки для операционных систем семейств Windows и Linux.

В данном разделе мы будем рассматривать установку и настройку среды разработки Eclipse на следующих операционных системах: Windows 10, Linux Ubuntu 20.04, Linux OpenSUSE Leap 15.2.

3.1 Установка и настройка среды разработки и Eclipse IDE Enterprise на Windows 10

Для работы со средой разработки Eclipse IDE Enterprise необходимо для начала установить JDK (Java Development Kit) на сайте <https://www.oracle.com/ru/java/technologies/javase-jdk15-downloads.html> (смотреть рисунок 4).

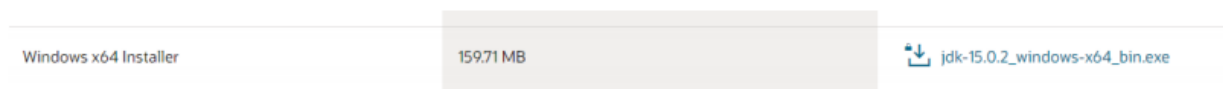


Рисунок 4 – Скачивание JDK

После установки JDK необходимо скачать и настроить Maven. Для начала необходимо скачать архив с официального сайта <https://maven.apache.org/download.cgi> (смотреть рисунок 5).

Files

Maven is distributed in several formats for your convenience. Simply pick a ready-made binary distribution archive and follow the [installation instructions](#). Use a source archive if you intend to build Maven yourself.

In order to guard against corrupted downloads/installations, it is highly recommended to [verify the signature](#) of the release bundles against the public [KEYS](#) used by the Apache Maven developers.

	Link	Checksums	Signature
Binary tar.gz archive	apache-maven-3.6.3-bin.tar.gz	apache-maven-3.6.3-bin.tar.gz.sha512	apache-maven-3.6.3-bin.tar.gz.asc
Binary zip archive	apache-maven-3.6.3-bin.zip	apache-maven-3.6.3-bin.zip.sha512	apache-maven-3.6.3-bin.zip.asc
Source tar.gz archive	apache-maven-3.6.3-src.tar.gz	apache-maven-3.6.3-src.tar.gz.sha512	apache-maven-3.6.3-src.tar.gz.asc
Source zip archive	apache-maven-3.6.3-src.zip	apache-maven-3.6.3-src.zip.sha512	apache-maven-3.6.3-src.zip.asc

• [Release Notes](#)

Рисунок 5 – Скачивание Maven

После скачивания необходимо распаковать скачанный архив в любую удобную пользователю папку (смотреть рисунок 6).

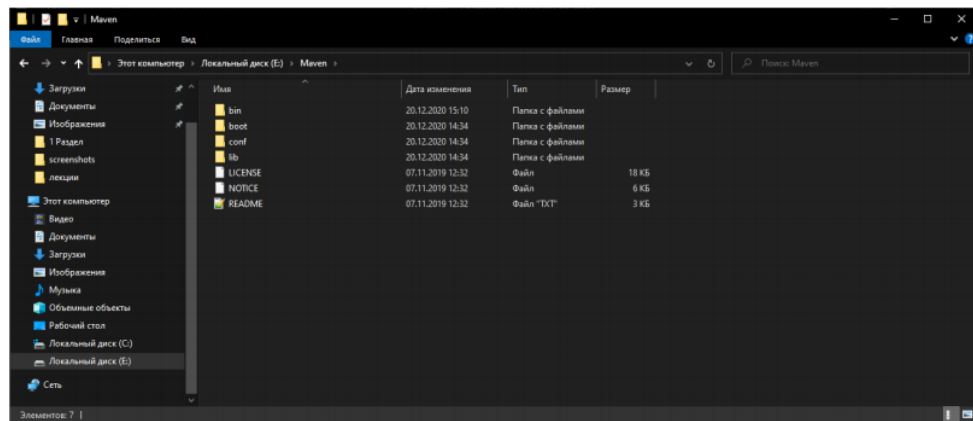


Рисунок 6 – Содержимое архива Maven

После проделанных операций необходима настройка установленных программ. Для этого нужно правой клавишей нажать на «Мой компьютер», нажать на «Свойства», после на «Дополнительные параметры системы», затем выбрать «Переменные среды». После в «Переменные среды для пользователя» создать или изменить следующие переменные (смотреть рисунок 7):

1. Создать JAVA_HOME, путь необходимо указать на папку с JDK.
2. Создать M2_HOME, путь к распакованному архиву Maven.
3. Создать M2, путь %M2%/bin.
4. Изменить переменную Path, добавив к концу %M2%.

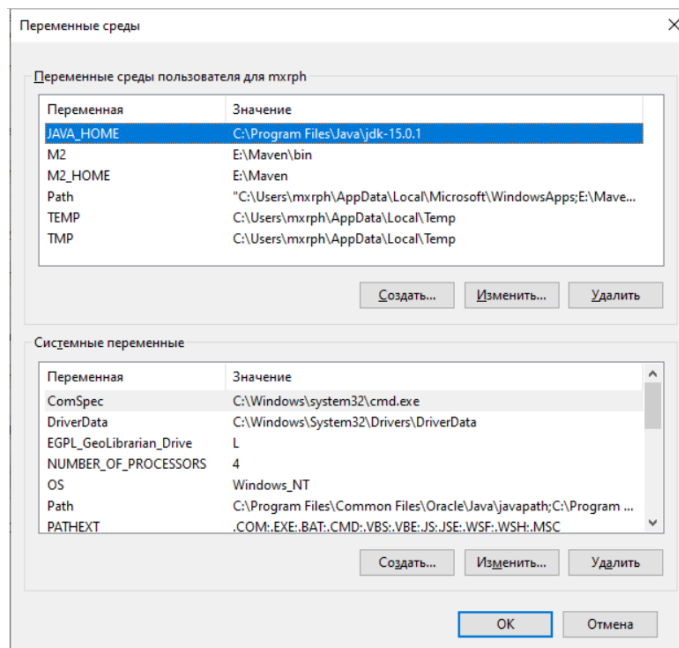


Рисунок 7 – Переменные среды пользователя

Также в «Системные переменные» в переменную Path добавить переменную с путём переменной M2 (смотреть рисунок 8).

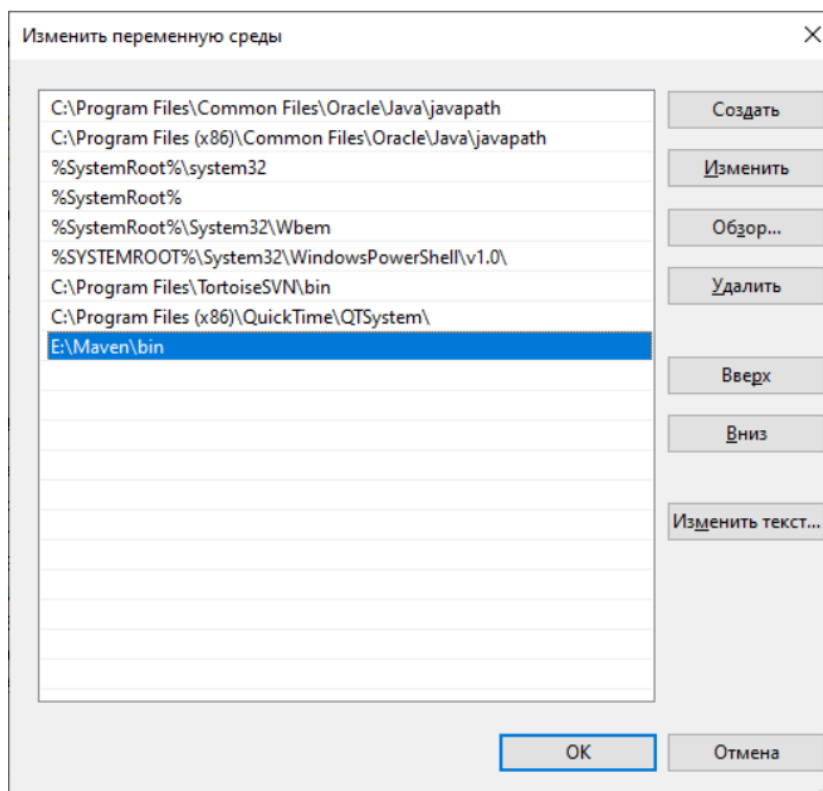


Рисунок 8 – Системные переменные

Проверить установку Java и Maven можно с помощью командной строки.

Её можно вызвать с помощью клавишной комбинации Win + R, в выпадающем окне ввести команду cmd (смотреть рисунок 9).

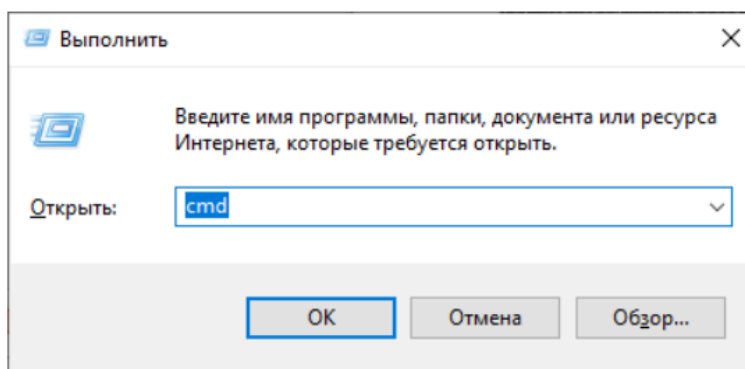


Рисунок 9 – Вызов командной строки

Для проверки Java необходимо написать `java -version`, для Maven `-mvn version` (смотреть рисунок 10).

```
C:\Windows\system32\cmd.exe
Microsoft Windows [Version 10.0.19041.388]
(c) Корпорация Майкрософт (Microsoft Corporation), 2020. Все права защищены.

C:\Users\mxrph>java -version
java version "15.0.1" 2020-10-20
Java(TM) SE Runtime Environment (build 15.0.1+9-18)
Java HotSpot(TM) 64-Bit Server VM (build 15.0.1+9-18, mixed mode, sharing)

C:\Users\mxrph>mvn -version
Apache Maven 3.6.3 (cecedd343002696d0abb50b32b541b8a6ba2883f)
Maven home: E:\Maven\bin\..
Java version: 15.0.1, vendor: Oracle Corporation, runtime: C:\Program Files\Java\jdk-15.0.1
Default locale: ru_RU, platform encoding: Cp1251
OS name: "windows 10", version: "10.0", arch: "amd64", family: "windows"

C:\Users\mxrph>
```

Рисунок 10 – Проверка установленных программ

После успешной настройки программ необходимо скачать и установить среду разработки Eclipse IDE Enterprise. Для её установки нужно перейти на сайт <https://www.eclipse.org/downloads/> и начать загрузку для разрядности вашей операционной системы. При запуске выбрать пункт “Eclipse IDE for Enterprise Java and Web Developers”, затем произвести конфигурацию установки (смотреть рисунок 11–14).



Рисунок 11 – Кнопка скачивания Eclipse IDE



Рисунок 12 – Выбор версии Eclipse IDE



Рисунок 13 – Окно установки с выбором пути



Рисунок 14 – Итоговое окно установки Eclipse IDE

Изначально в Eclipse IDE предварительно установлен Git. Для того, чтобы пользователь смог удостовериться в его наличии, необходимо выполнить следующие действия: нажать на Window > Perspective > Open Perspective > Other. В выпадающем окне выбрать Git (смотреть рисунок 15).

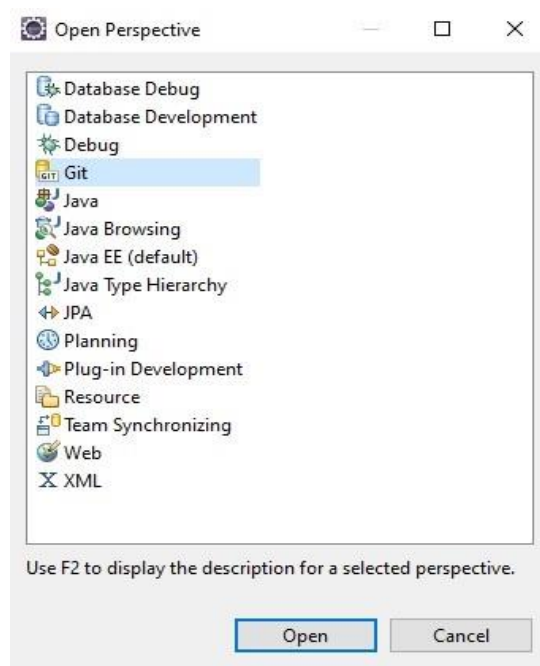


Рисунок 15 — Демонстрация наличия Git

Для создания Maven проекта в Windows 10 нужно нажать на File > New Project > Project > Maven > Maven Project > Поставить галочку Create a simple project (skip from archetype) после произвести конфигурацию проекта (смотреть рисунок 16–19).

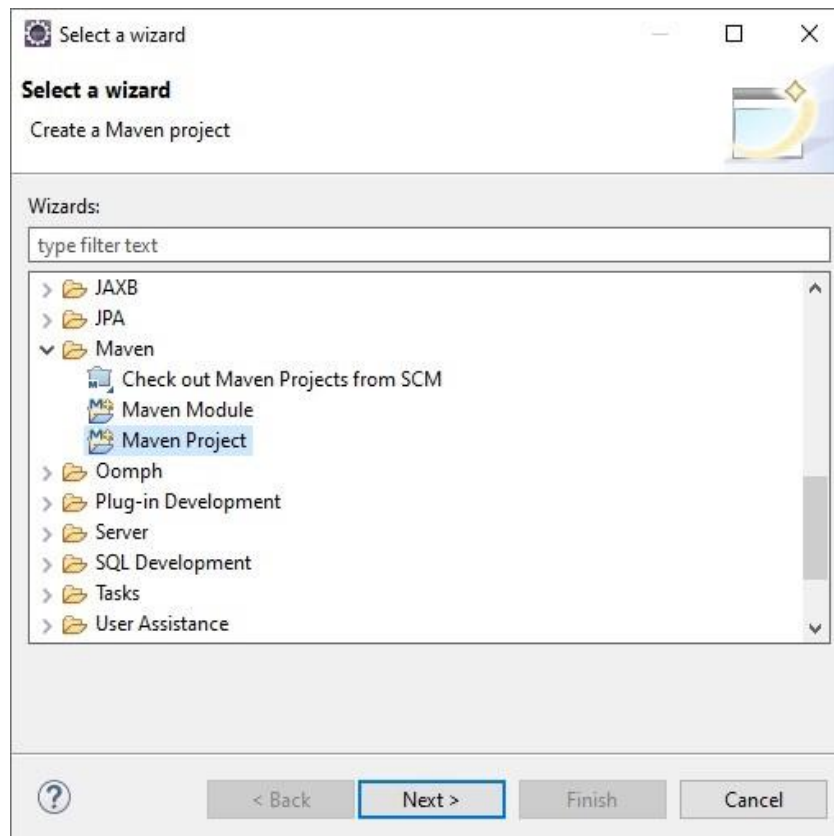


Рисунок 16 — Создание Maven Project

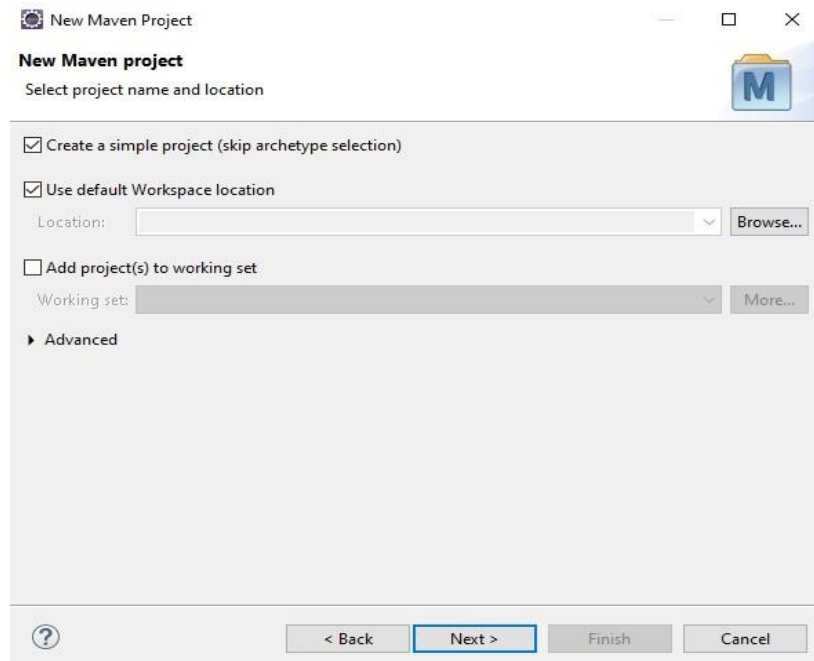


Рисунок 17 — Выбор пути нового Maven Project

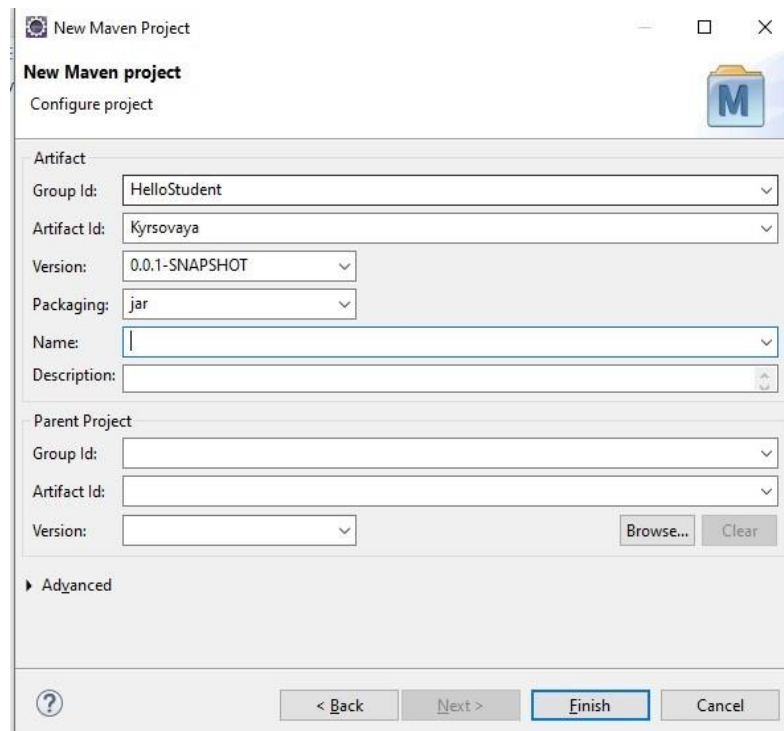


Рисунок 18 — Конфигурация Maven Project

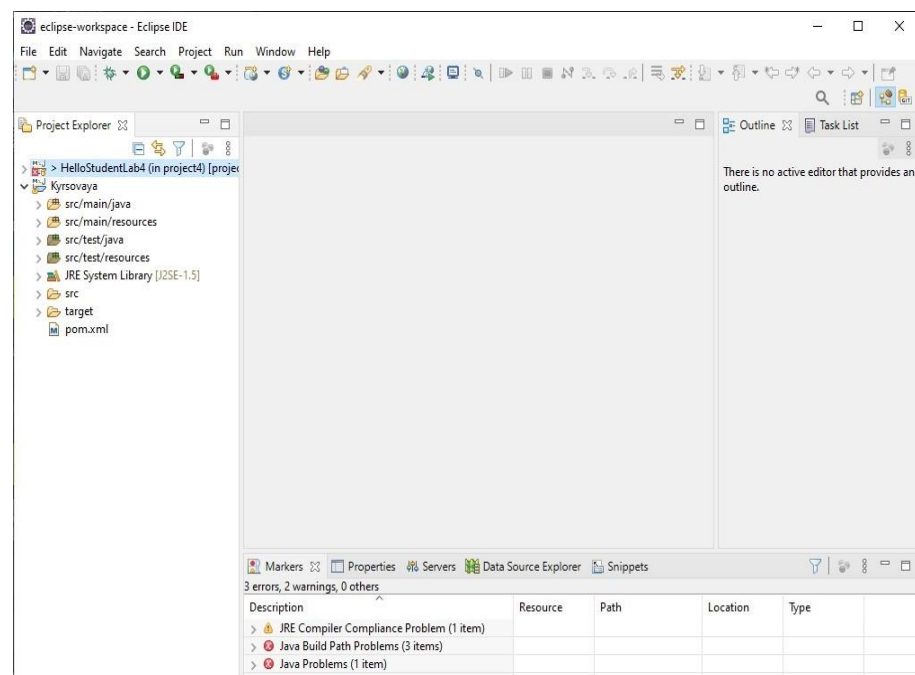


Рисунок 19 — Итог создания Maven Project на Windows 10

3.2 Установка и настройка среды разработки Eclipse IDE Enterprise на Ubuntu 20.04

Для работы с Eclipse IDE на Ubuntu 20.04 необходимо скачать и установить JDK. Для этого мы откроем терминал с помощью нажатия правой клавишей мыши по рабочему столу > Открыть в терминале (смотреть рисунок 20).

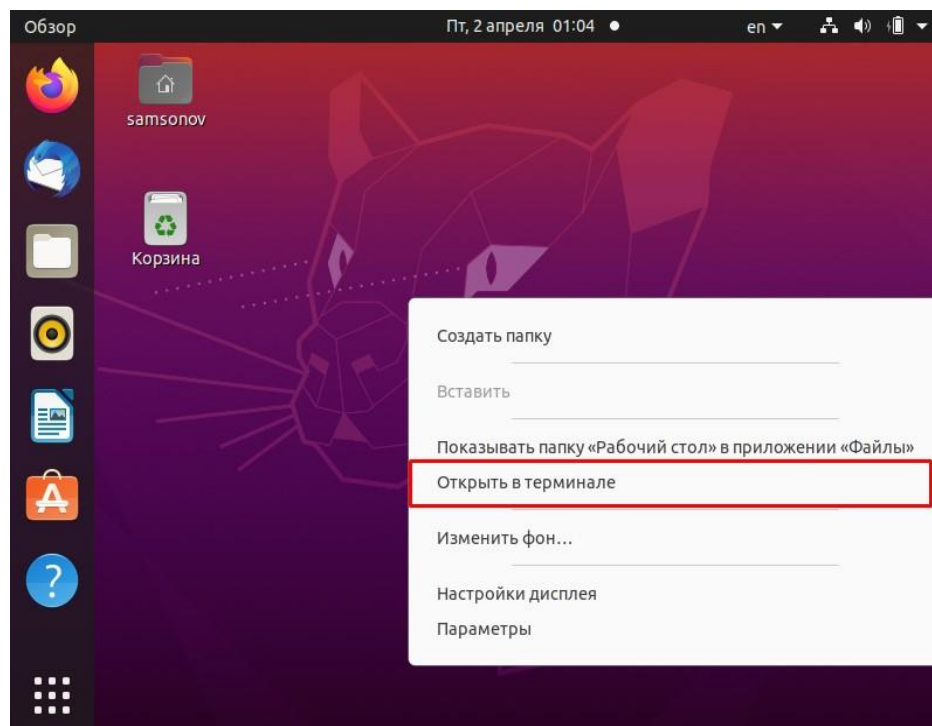


Рисунок 20 — Открытие терминала

После скачиваем JDK. Чтобы это сделать необходимо написать в командную строку `sudo apt install openjdk-8-jdk` (смотреть рисунок 21) и проверяем с помощью `java -version` (смотреть рисунок 22).


```
samsonov@mxrph: ~/Рабочий стол
Обновлено 0 пакетов, установлено 0 новых пакетов, для удаления отмечено 0 пакетов, и 0 пакетов не обновлено.
samsonov@mxrph:~/Рабочий стол$ sudo apt install openjdk-8-jdk
Чтение списков пакетов... Готово
Построение дерева зависимостей
Чтение информации о состоянии... Готово
Будут установлены следующие дополнительные пакеты:
ca-certificates-java fonts-dejavu-extra java-common libatk-wrapper-java
libatk-wrapper-java-jni libice-dev libpthread-stubs0-dev libsm-dev
libx11-dev libxau-dev libxcb1-dev libxdmcp-dev libxt-dev
openjdk-8-jdk-headless openjdk-8-jre openjdk-8-jre-headless
x11proto-core-dev x11proto-dev xorg-sgml-doctools xtrans-dev
Предлагаемые пакеты:
default-jre libice-doc libsm-doc libx11-doc libxcb-doc libxt-doc
openjdk-8-demo openjdk-8-source visualvm icedtea-8-plugin
fonts-ipafont-gothic fonts-ipafont-mincho fonts-wqy-microhei
fonts-wqy-zenhei
Следующие НОВЫЕ пакеты будут установлены:
ca-certificates-java fonts-dejavu-extra java-common libatk-wrapper-java
libatk-wrapper-java-jni libice-dev libpthread-stubs0-dev libsm-dev
libx11-dev libxau-dev libxcb1-dev libxdmcp-dev libxt-dev openjdk-8-jdk
openjdk-8-jdk-headless openjdk-8-jre openjdk-8-jre-headless
x11proto-core-dev x11proto-dev xorg-sgml-doctools xtrans-dev
Обновлено 0 пакетов, установлено 21 новых пакетов, для удаления отмечено 0 пакет
```

Рисунок 21 — Скачивание JDK для Ubuntu 20.04

```
samsonov@mxrph:~/Рабочий стол$ java -version
openjdk version "1.8.0_282"
OpenJDK Runtime Environment (build 1.8.0_282-8u282-b08-0ubuntu1~20.04-b08)
OpenJDK 64-Bit Server VM (build 25.282-b08, mixed mode)
samsonov@mxrph:~/Рабочий стол$
```

Рисунок 22 — Проверка установки JDK.

После успешной установки необходимо скачать и установить среду разработки Eclipse IDE Enterprise. Для её установки нужно перейти на сайт <https://www.eclipse.org/downloads/>, выбрать Download Packages (смотреть рисунок 23) и выбрать Linux_x86_x64 (смотреть рисунок 24). После скачивания в загрузках появится архив, который нужно будет распаковать (смотреть рисунок 25), затем запустить eclipse-inst (смотреть рисунок 26).



Рисунок 23 — Кнопка Download Packages

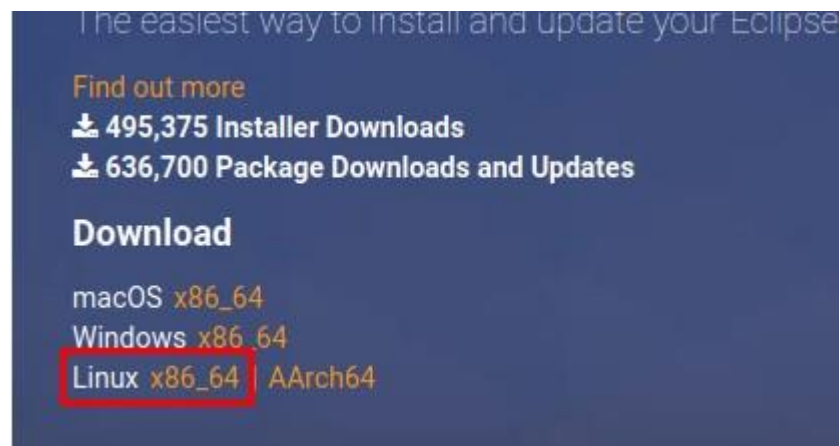


Рисунок 24 — Выбор скачивания Eclipse IDE для Linux

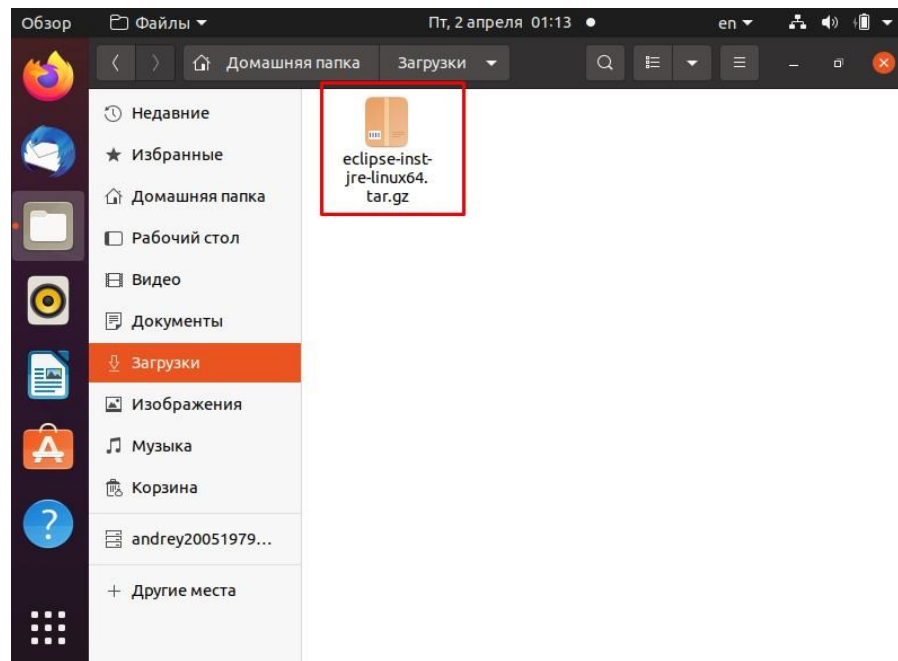


Рисунок 25 — Скачанный архив

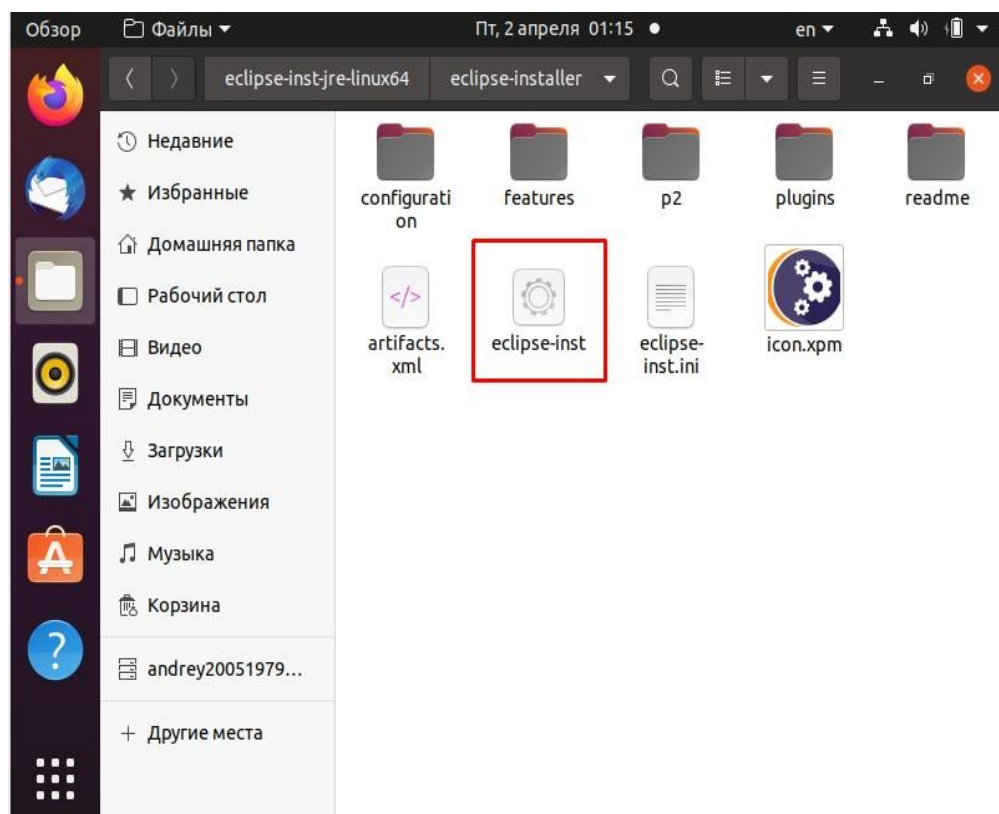


Рисунок 26 — Установочный файл Eclipse IDE

После запуска установочного файла необходимо выбрать пункт “Eclipse IDE for Enterprise Java and Web Developers” (смотреть рисунок 27) и после произвести установку (смотреть рисунок 28–29).

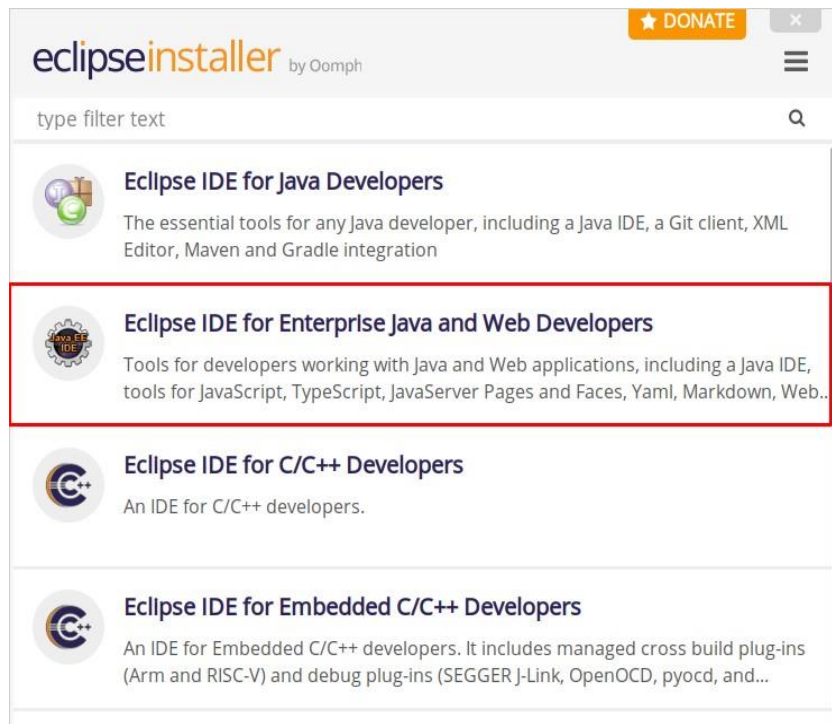


Рисунок 27 — Выбор версии Eclipse IDE

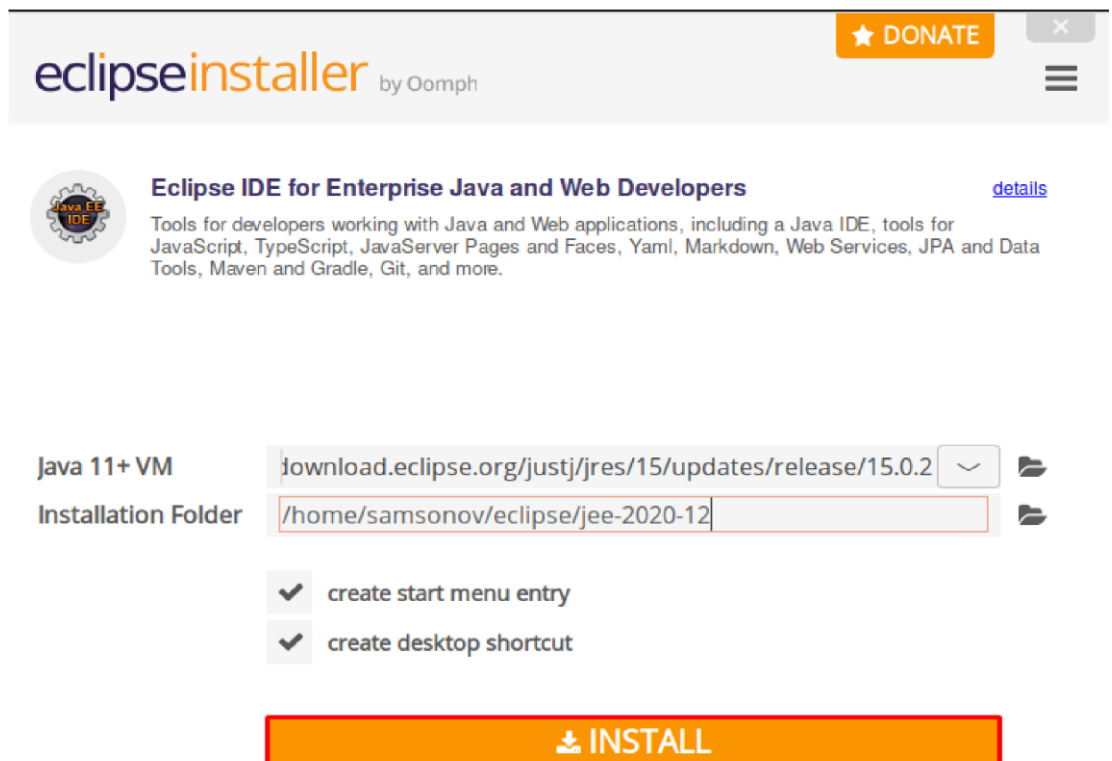


Рисунок 28 — Окно установки с выбором пути

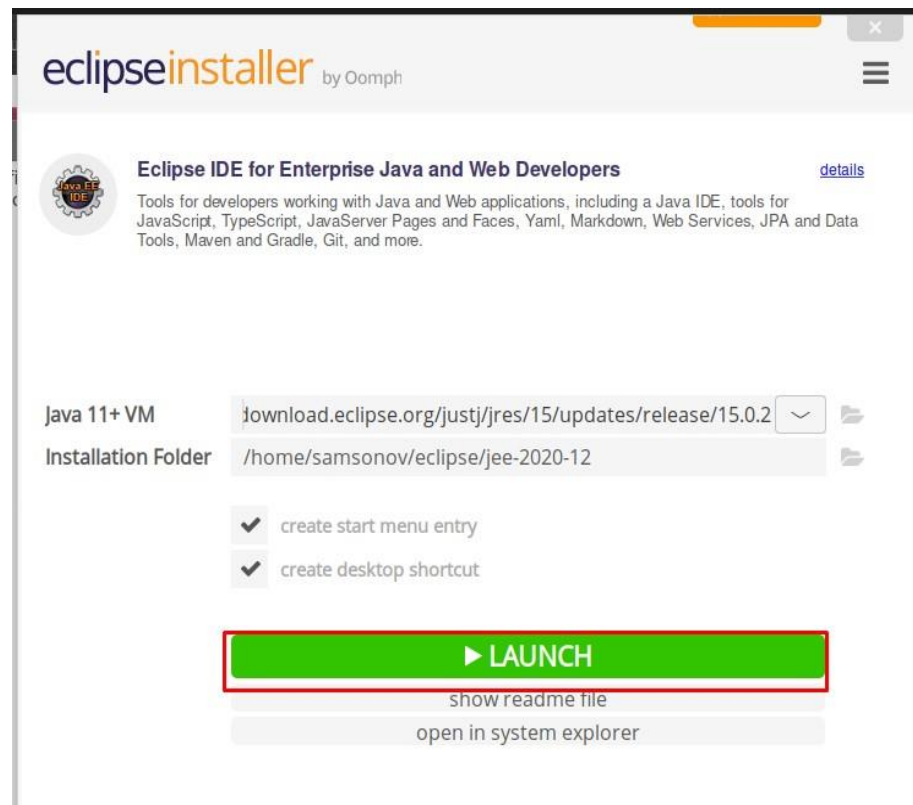


Рисунок 29 – Итоговое окно установки

Изначально в Eclipse IDE предварительно установлен Git. Для того, чтобы пользователь смог удостовериться в его наличии, необходимо выполнить следующие действия: нажать на Window > Perspective > Open Perspective > Other. В выпадающем окне выбрать Git (смотреть рисунок 30).

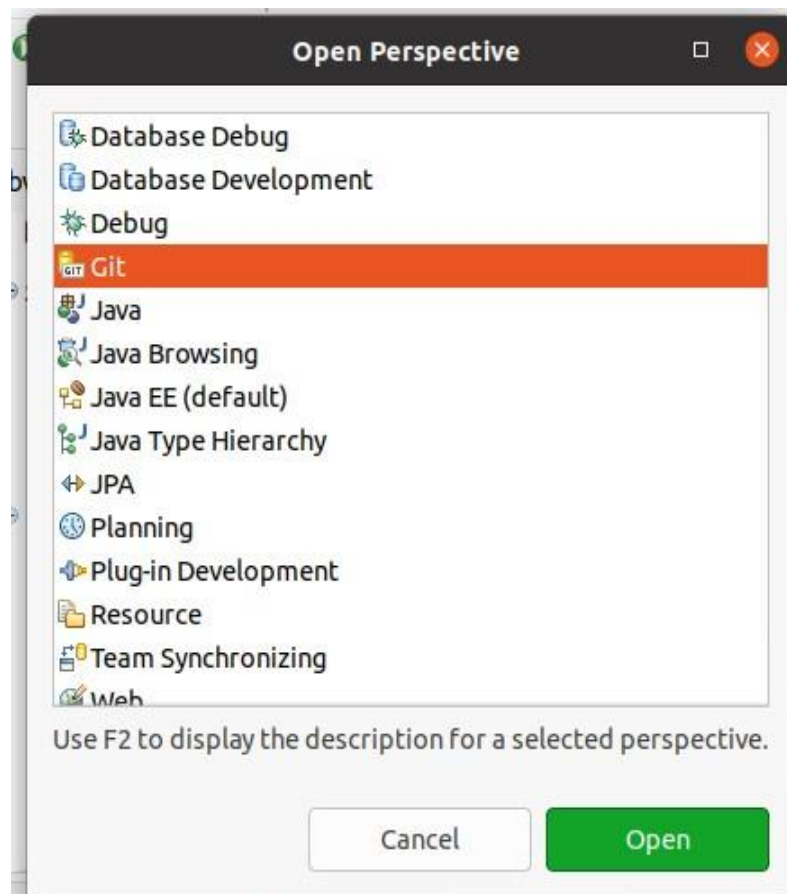


Рисунок 30 — Демонстрация наличия Git

Для создания Maven проекта в Ubuntu 20.04 нужно нажать на File > New > Project... > Maven > Maven Project > Поставить галочку Create a simple project (skip from archetype) после произвести конфигурацию проекта (смотреть рисунок 31–34).

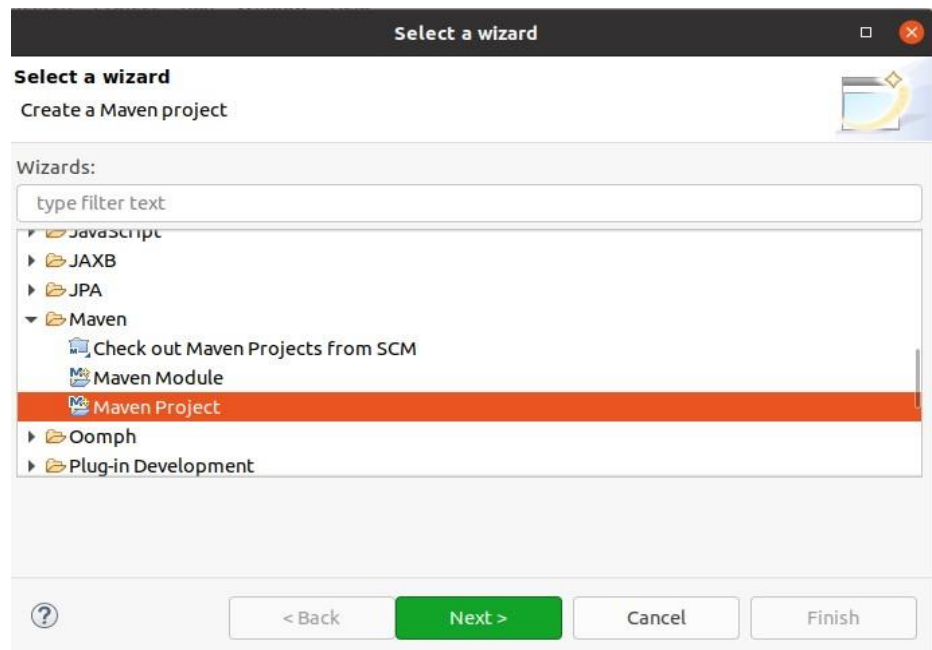


Рисунок 31 — Выбор Maven Project

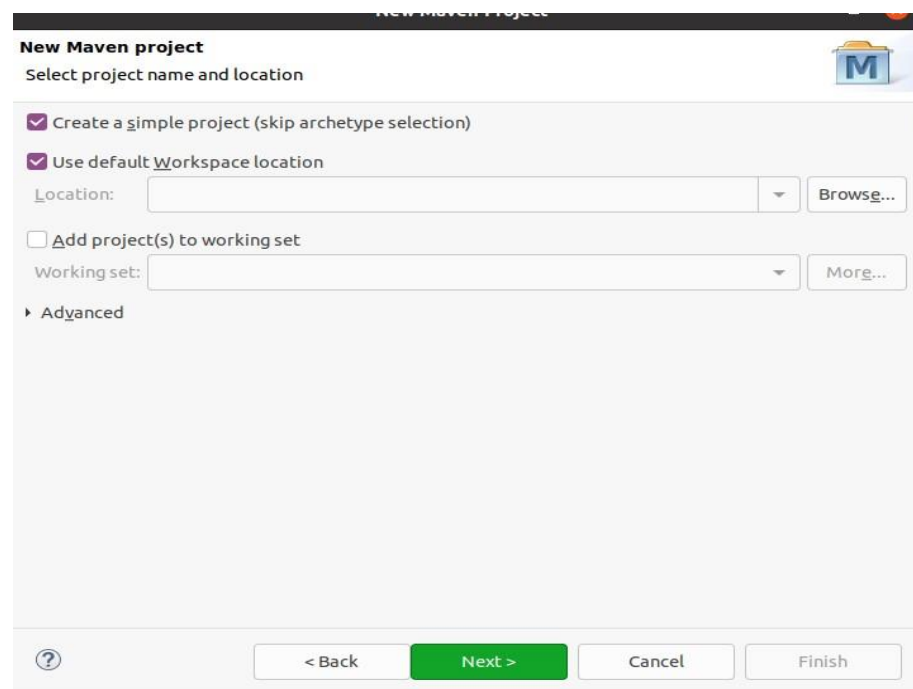


Рисунок 32 — Выбор пути нового Maven Project

New Maven project
Configure project

Artifact

Group Id: HelloStudent

Artifact Id: Kurovaya

Version: 0.0.1-SNAPSHOT

Packaging: jar

Name:

Description:

Parent Project

Group Id:

Artifact Id:

Version:

Browse... Clear

Advanced

< Back Next > Cancel Finish

Рисунок 33 — Конфигурация Maven Project

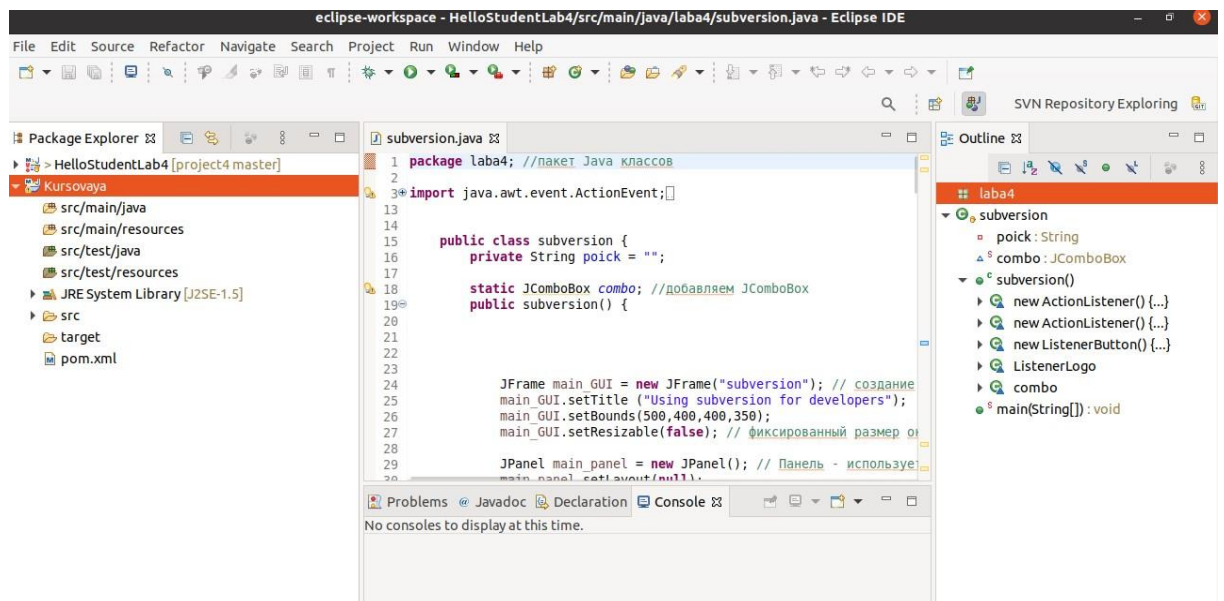


Рисунок 34 — Итог создания Maven Project в Ubuntu 20.04

3.3 Установка и настройка среды разработки Eclipse IDE Enterprise на OpenSUSE Leap 15.2

Для установки JDK на OpenSUSE Leap 15.2 необходимо при установке самой ОС в окне «Настройки установки» нажать на «Программное обеспечение» (смотреть рисунок 35), затем выбрать пункт «Разработка на Java» (смотреть рисунок 36).

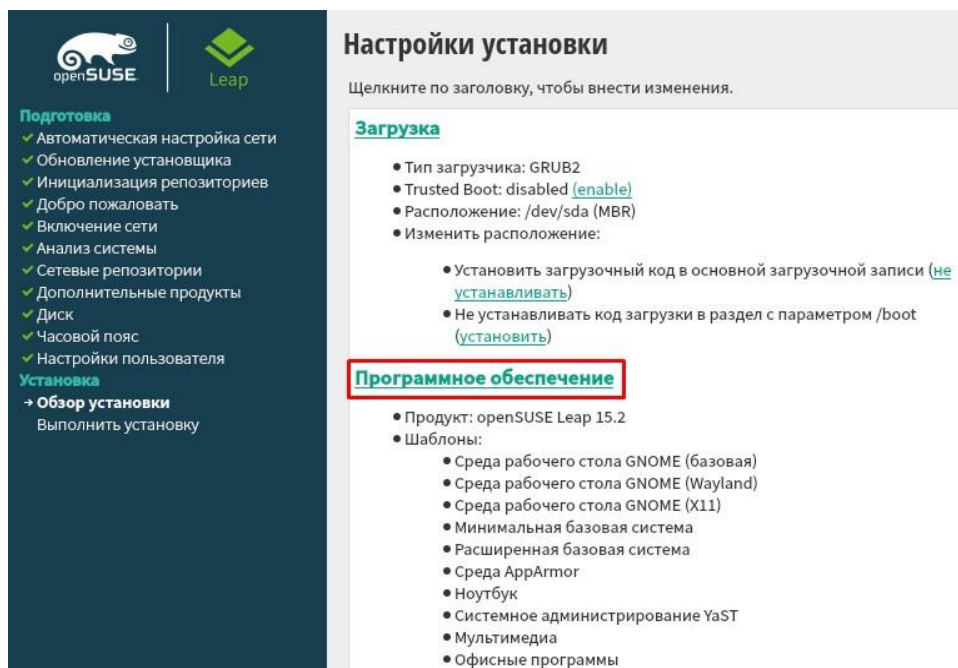


Рисунок 35 — Окно «Настройки установки»

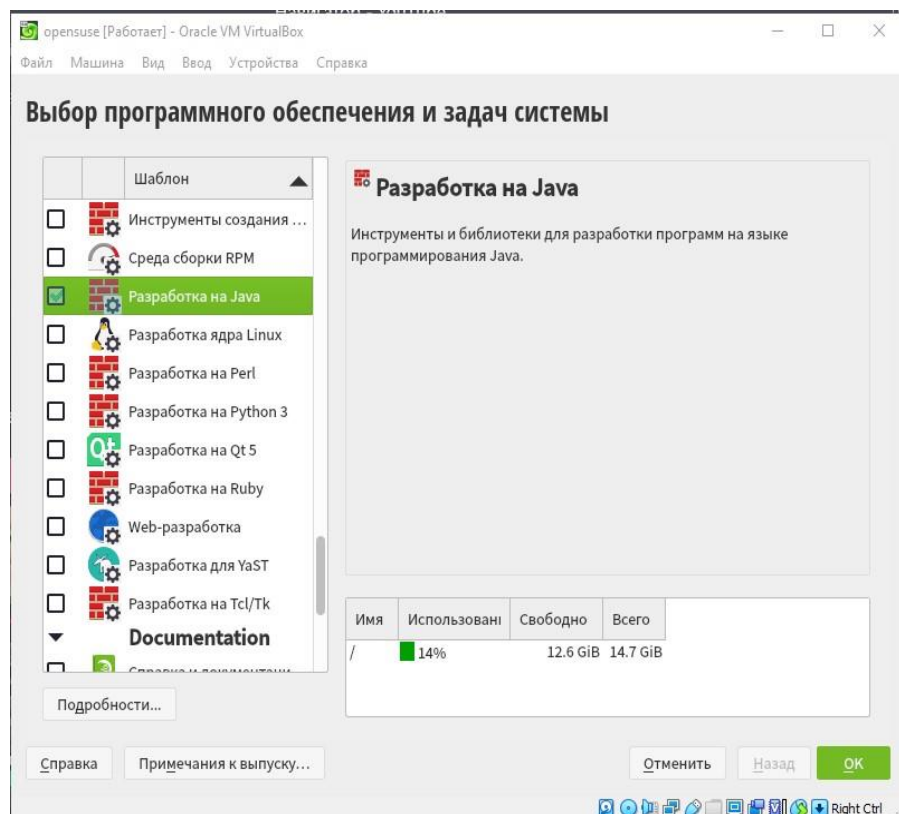


Рисунок 36 — Выбор пункта «Разработка на Java»

Чтобы проверить установку JDK необходимо зайти в терминал. Его можно найти с помощью Обзор > Все приложения > Утилиты > Терминал (смотреть рисунок 37), после написать `java -version` (смотреть рисунок 38).

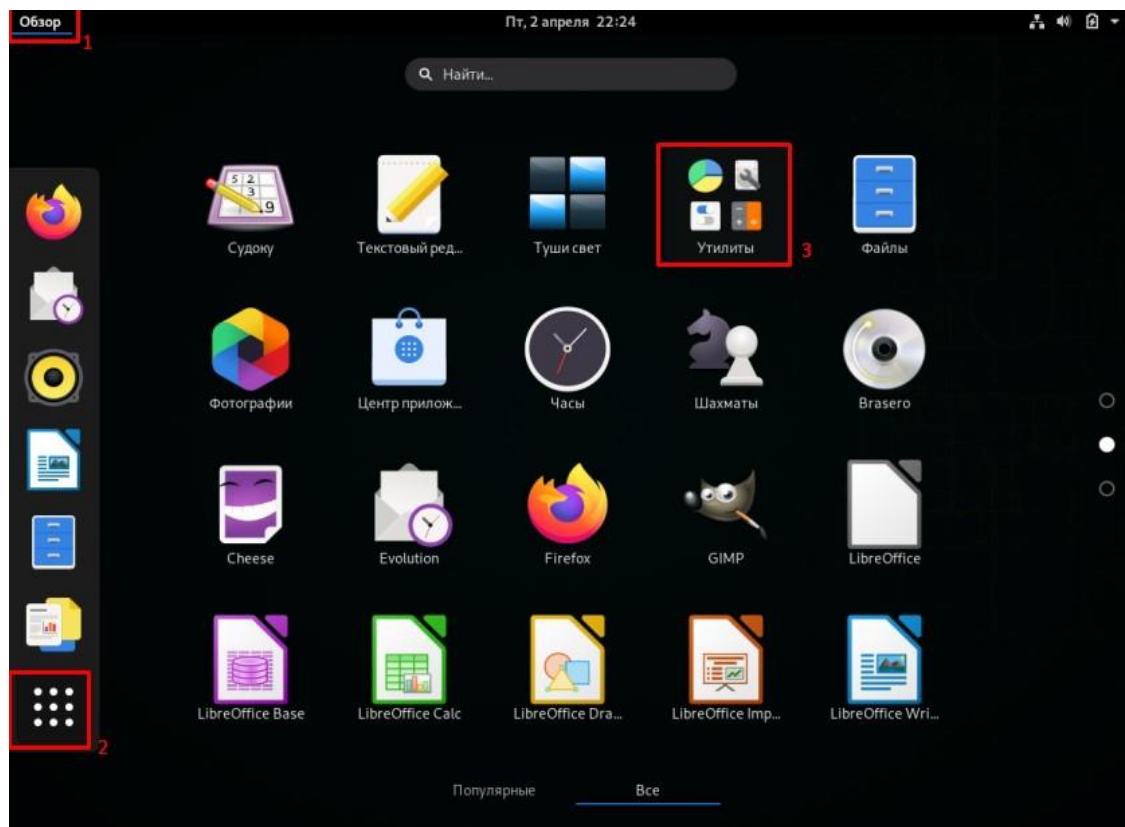


Рисунок 37 — Открытие терминала

```
artem@localhost:~  
artem@localhost:~> java -version  
openjdk version "11.0.10" 2021-01-19  
OpenJDK Runtime Environment (build 11.0.10+9-suse-lp152.2.9.1-x8664)  
OpenJDK 64-Bit Server VM (build 11.0.10+9-suse-lp152.2.9.1-x8664, mixed mode)  
artem@localhost:~>
```

Рисунок 38 — Проверка установки JDK

После успешной установки необходимо скачать и установить среду разработки Eclipse IDE Enterprise. Для её установки нужно перейти на сайт

<https://www.eclipse.org/downloads/>, выбрать Download Packages (смотреть рисунок 39) и выбрать Linux_x86_x64 (смотреть рисунок 40). После скачивания в загрузках появится архив, который нужно будет распаковать (смотреть рисунок 41), затем запустить eclipse-inst (смотреть рисунок 42).



Рисунок 39 — Кнопка Download Packages

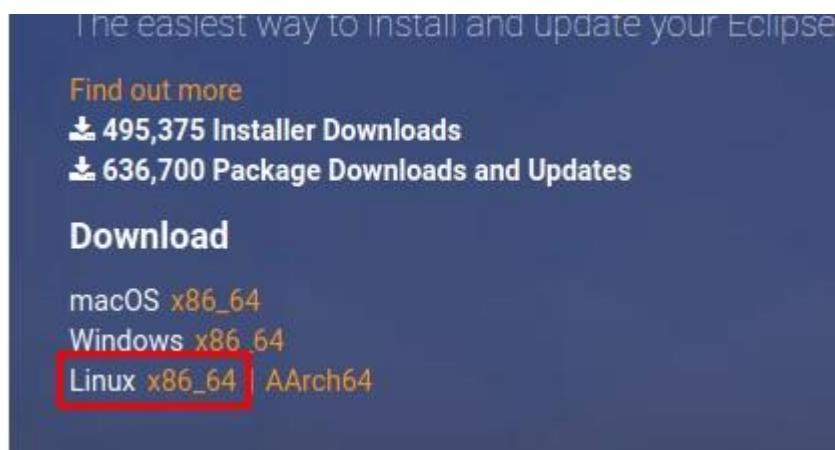


Рисунок 40 — Выбор скачивания Eclipse IDE для Linux

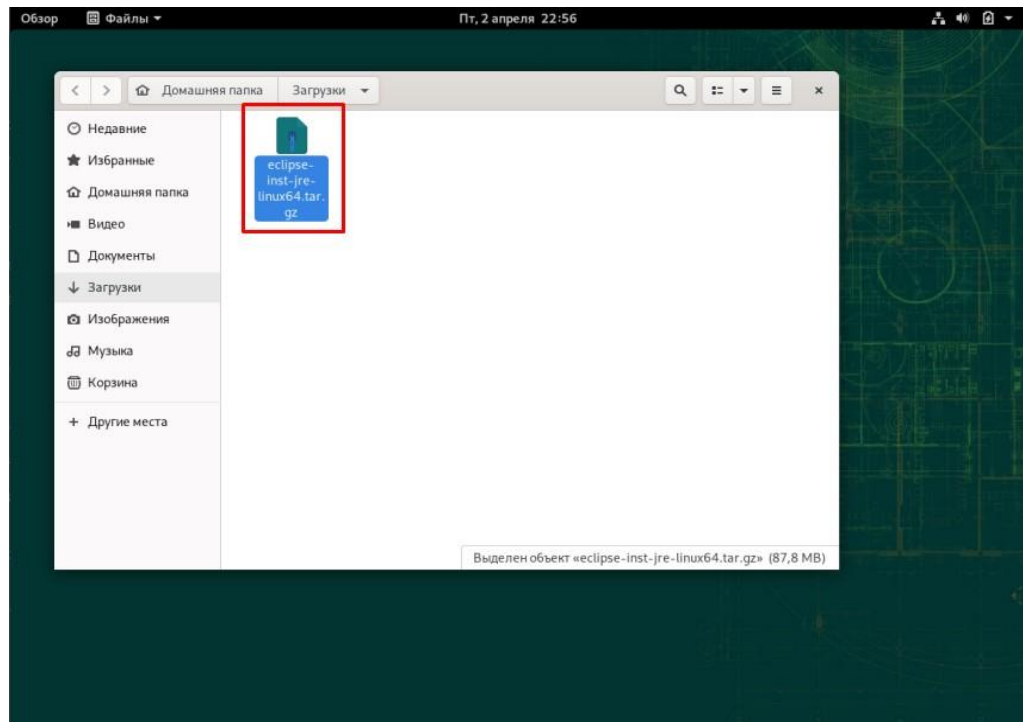


Рисунок 41 — Скачанный архив

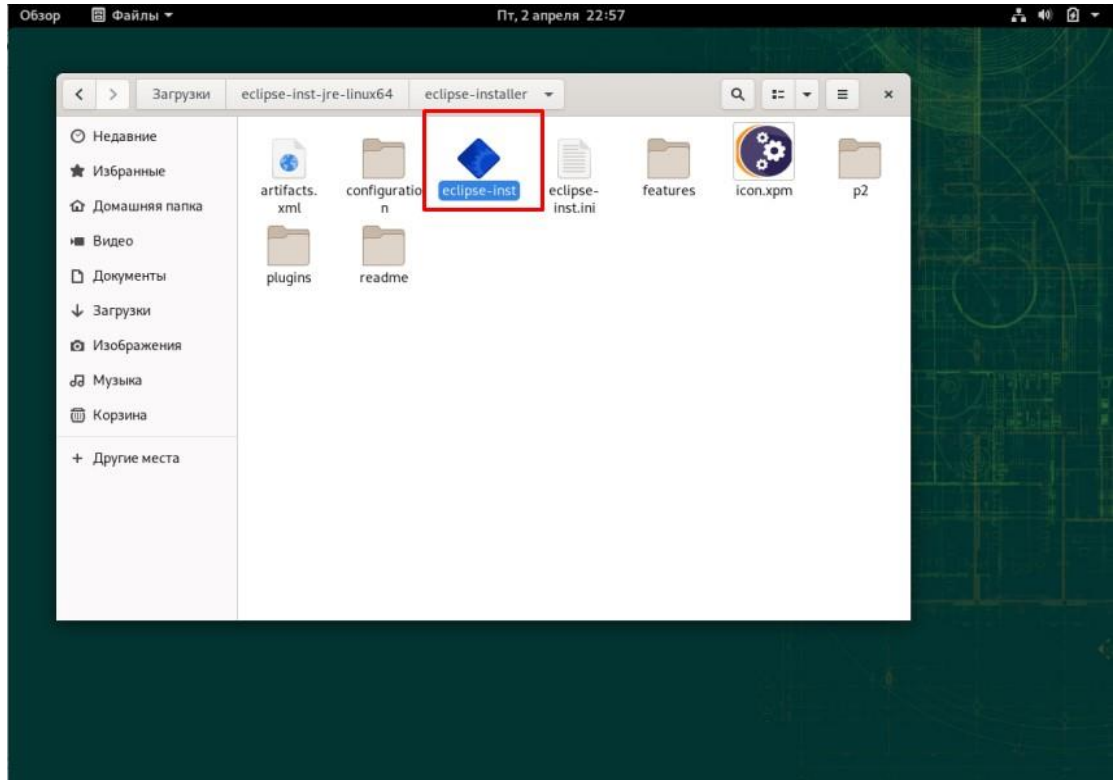


Рисунок 42 — Установочный файл Eclipse IDE

После запуска установочного файла необходимо выбрать пункт “Eclipse IDE for Enterprise Java and Web Developers” (смотреть рисунок 43) и после произвести установку (смотреть рисунок 44–45).

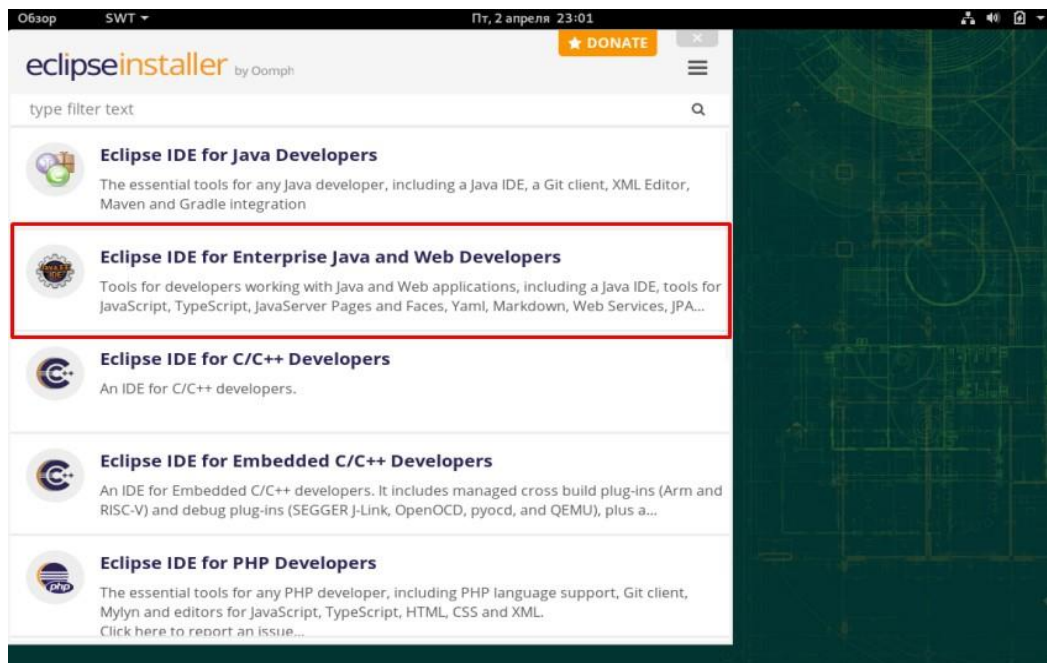


Рисунок 43 – Выбор версии

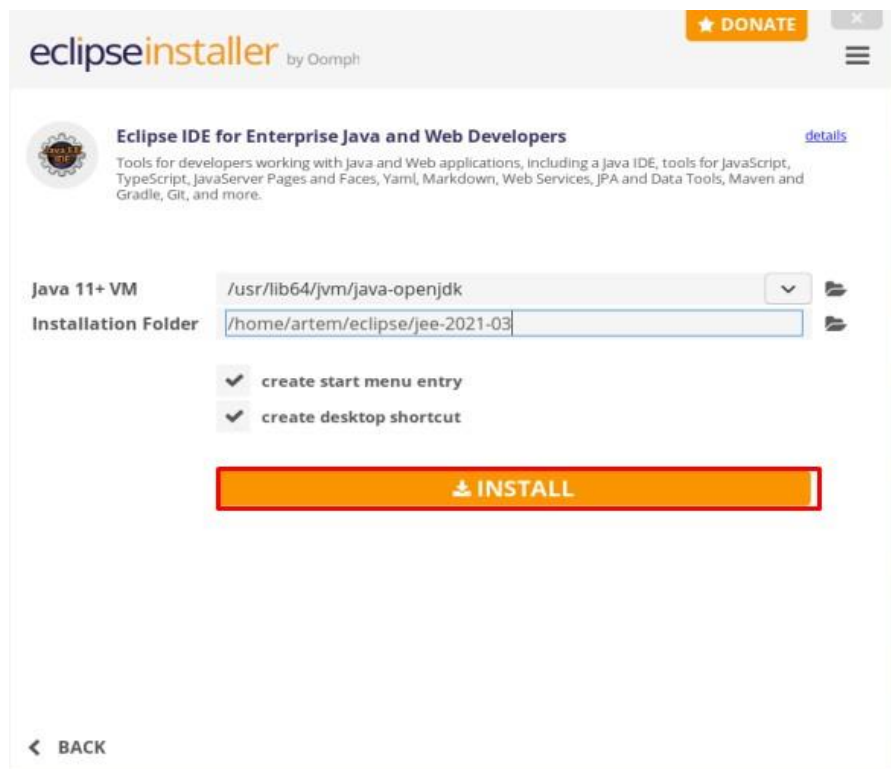


Рисунок 44 — Окно установки с выбором пути

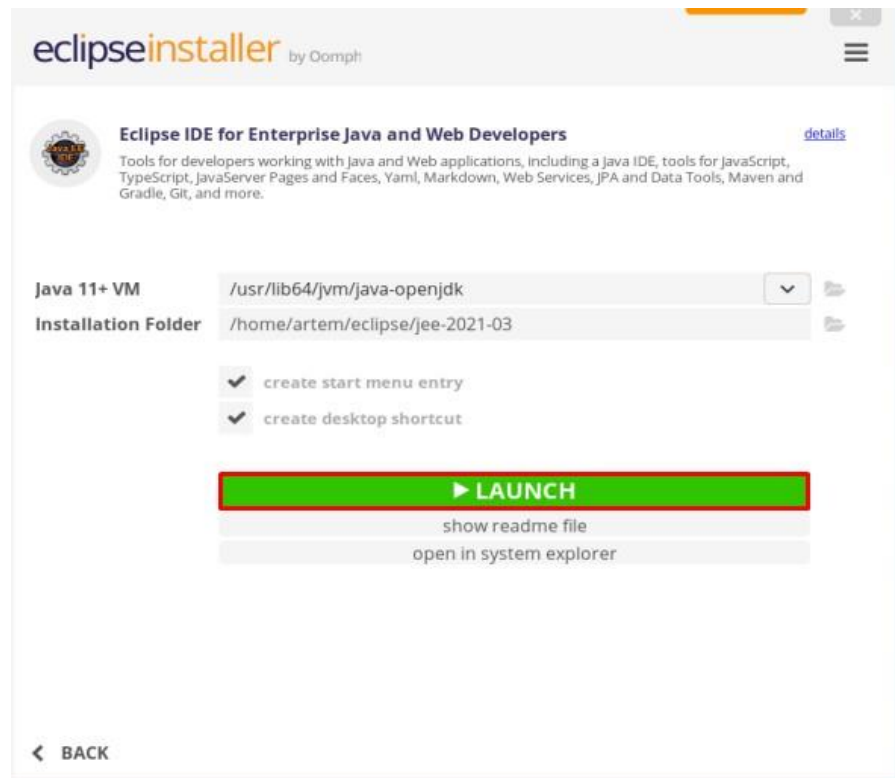


Рисунок 45 — Итоговое окно установки Eclipse IDE

Изначально в Eclipse IDE предварительно установлен Git. Для того, чтобы пользователь смог удостовериться в его наличии, необходимо выполнить следующие действия: нажать на Window > Perspective > Open Perspective > Other. В выпадающем окне выбрать Git (смотреть рисунок 46).

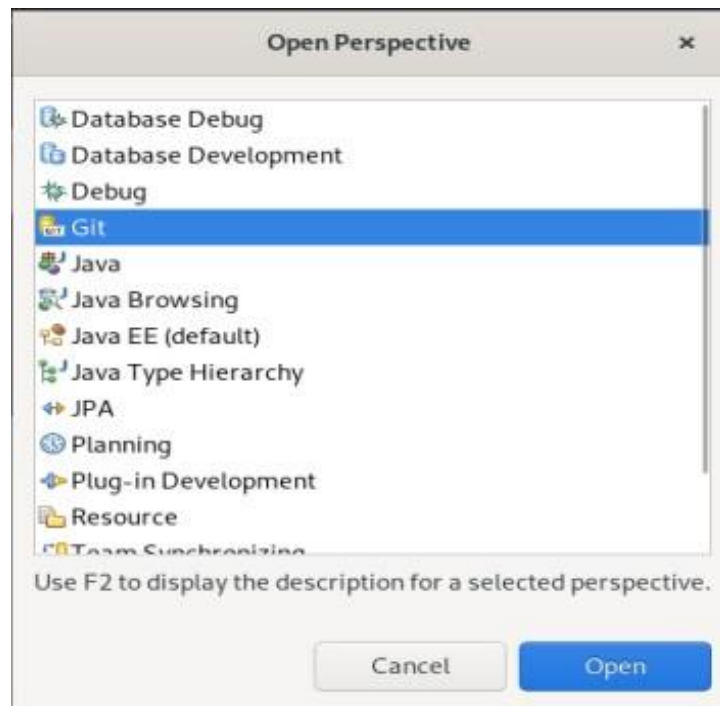


Рисунок 46 — Демонстрация наличия Git

Для создания Maven проекта в OpenSUSE Leap 15.2 нужно нажать на File > New > Project... > Maven > Maven Project > Поставить галочку Create a simple project (skip from archetype) после произвести конфигурацию проекта (смотреть рисунок 47–50).

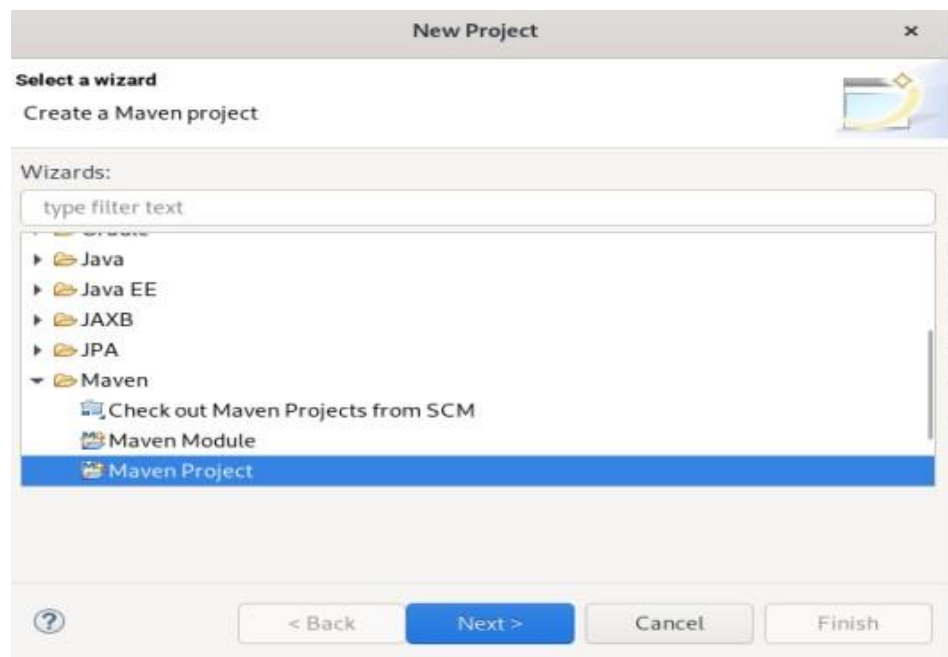


Рисунок 47 — Выбор Maven Project

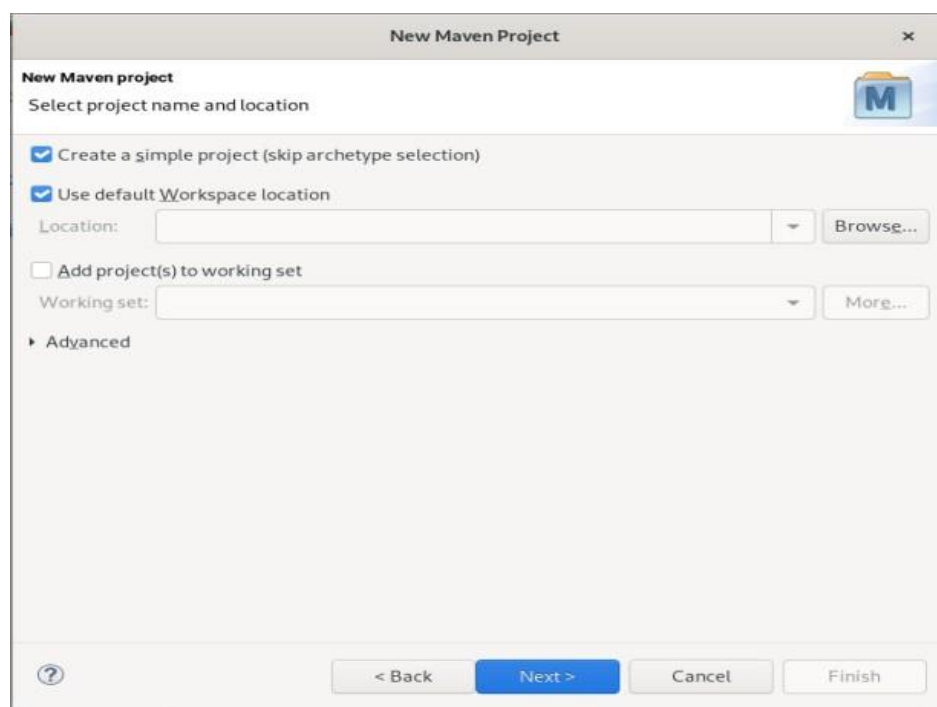


Рисунок 48 – Выбор пути нового Maven Project

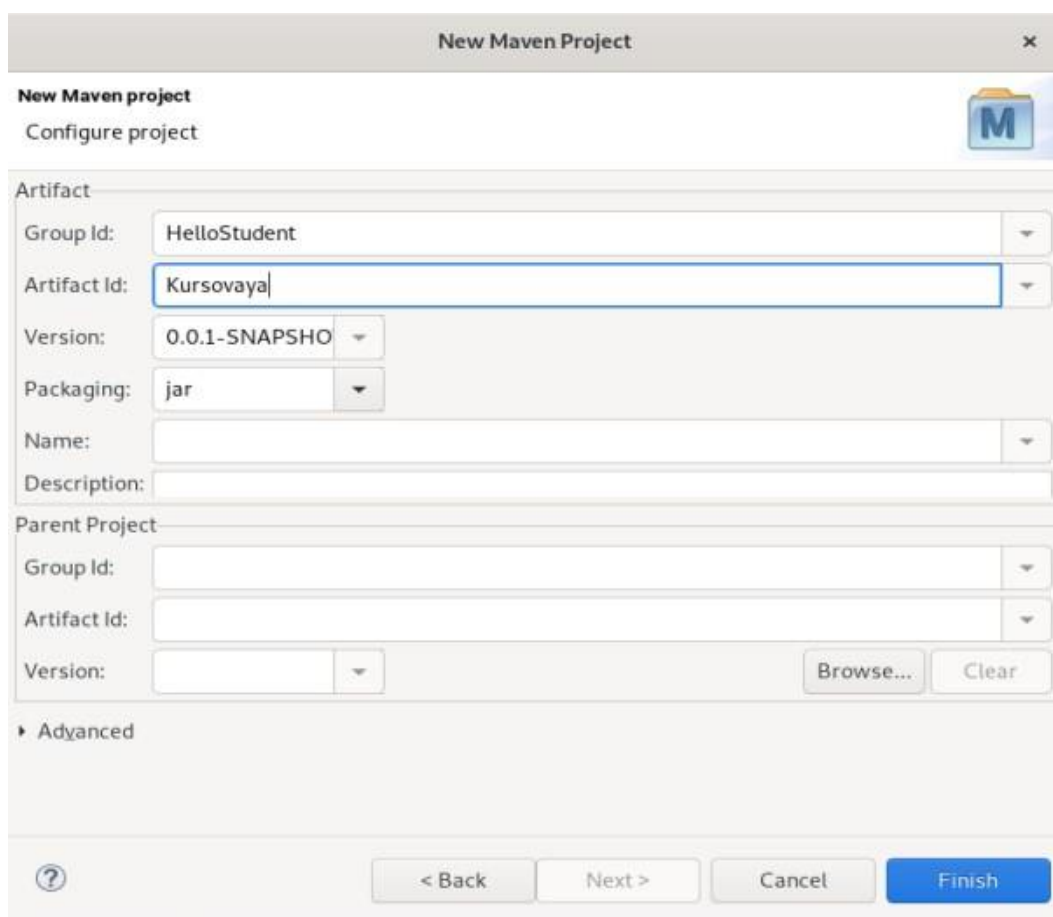


Рисунок 49 — Конфигурация Maven Project

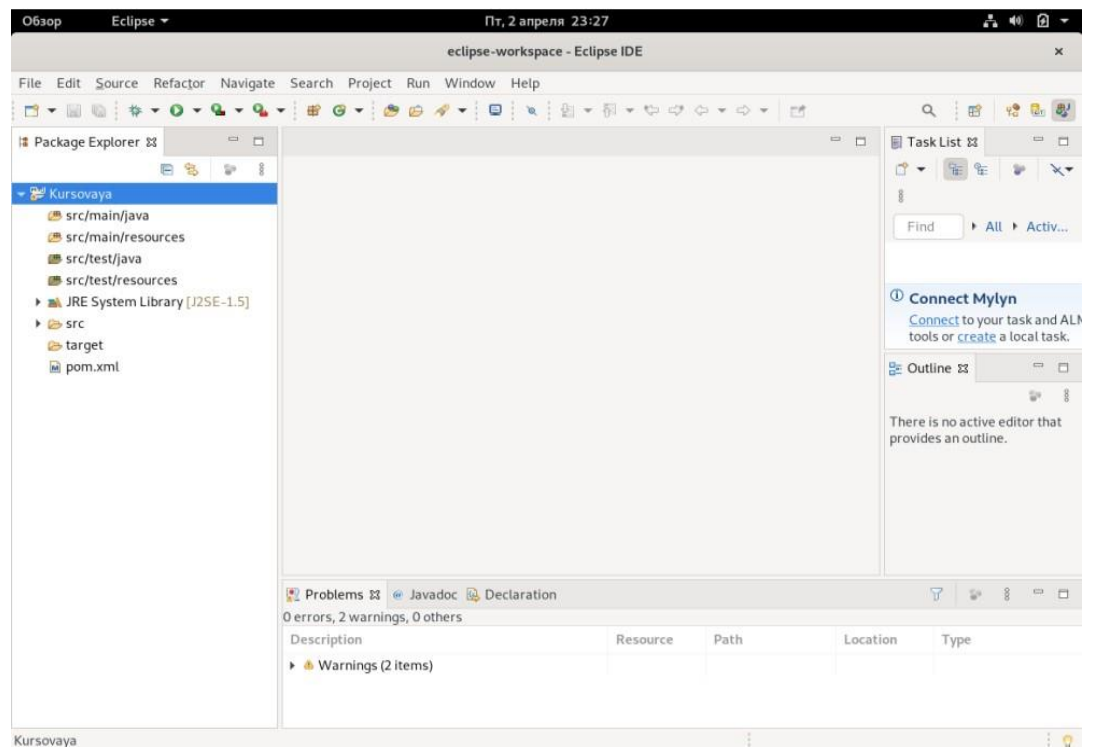


Рисунок 50 — Итог создания Maven Project в OpenSUSE Leap 15.2

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

В данном разделе мы будем рассматривать настройку среды разработки *Eclipse* для подключения к системе контроля версий (СКВ).

Так как *Eclipse* универсален для работы с различными операционными системами, поэтому рассмотрим настройку на ОС Linux Ubuntu 20.04.

После того, как модератор создал репозиторий на Github, необходимо к нему подключиться. Изначально в Eclipse IDE предварительно установлен Git. Для появления окна Git необходимо нажать на Window > Perspective > Open Perspective > Other. В выпадающем окне выбрать Git (смотреть рисунок 51).

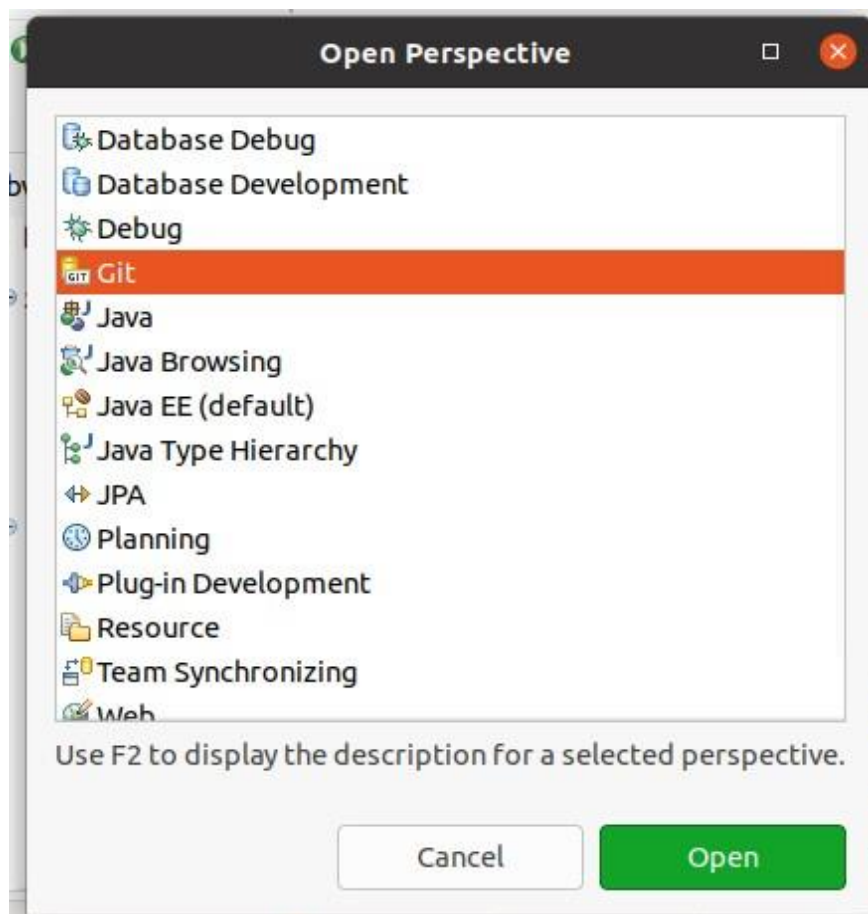


Рисунок 51 – Демонстрация наличия Git.

После открытия окна Git, необходимо клонировать Git репозиторий. Для этого выберем пункт “Clone a git repository” (смотреть рисунок 52).

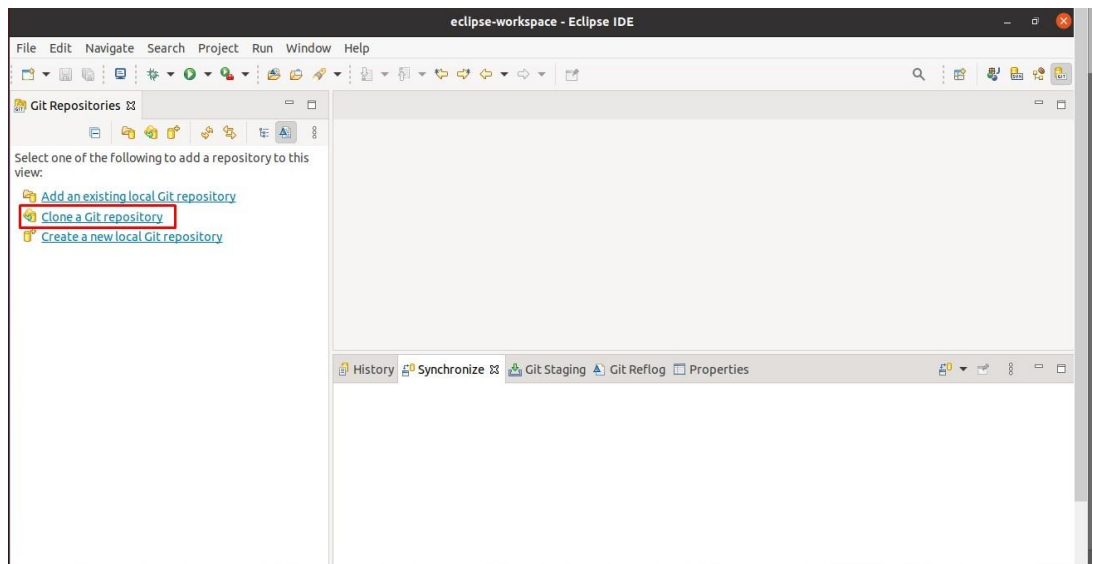


Рисунок 52 – Кнопка клонирования репозитория.

Затем в выпадающем окне необходимо ввести ссылку на Git репозиторий в окне “URL” (в нашем случае <https://github.com/mxrph/pi221team4k>). Также можно ввести логин и пароль в соответствующие окна, если вы получили доступ разработчика от модератора (смотреть рисунок 53).

Рисунок 53 – Окно клонирования Git репозитория.

После ввода ссылки и данных, необходимо выбрать ветку для клонирования (смотреть рисунок 54). Затем выйдет финальное окно с проверкой выбранных параметров. В итоге репозиторий будет клонирован.

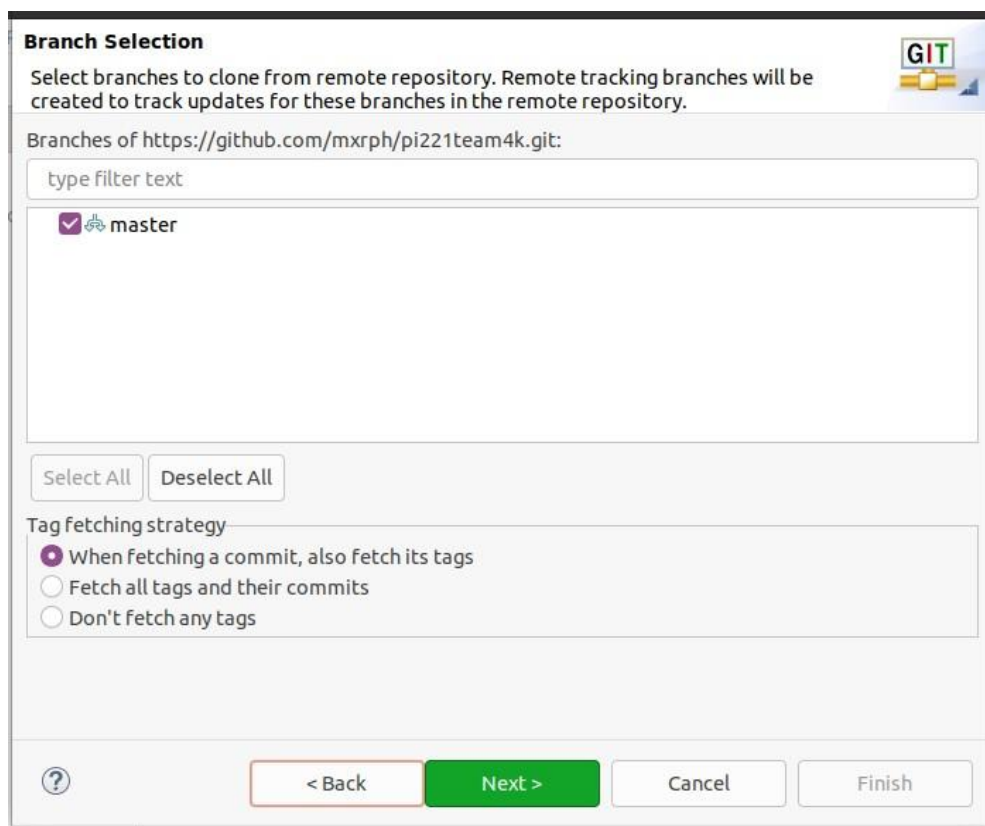


Рисунок 54 – Окно выбора клонирования ветки.

Чтобы выгрузить себе проект в рабочую копию, необходимо проделать следующие шаги: *раскрыть репозиторий > нажать правой клавишей по “Working Tree” > «Import Projects...»* (смотреть рисунок 55).

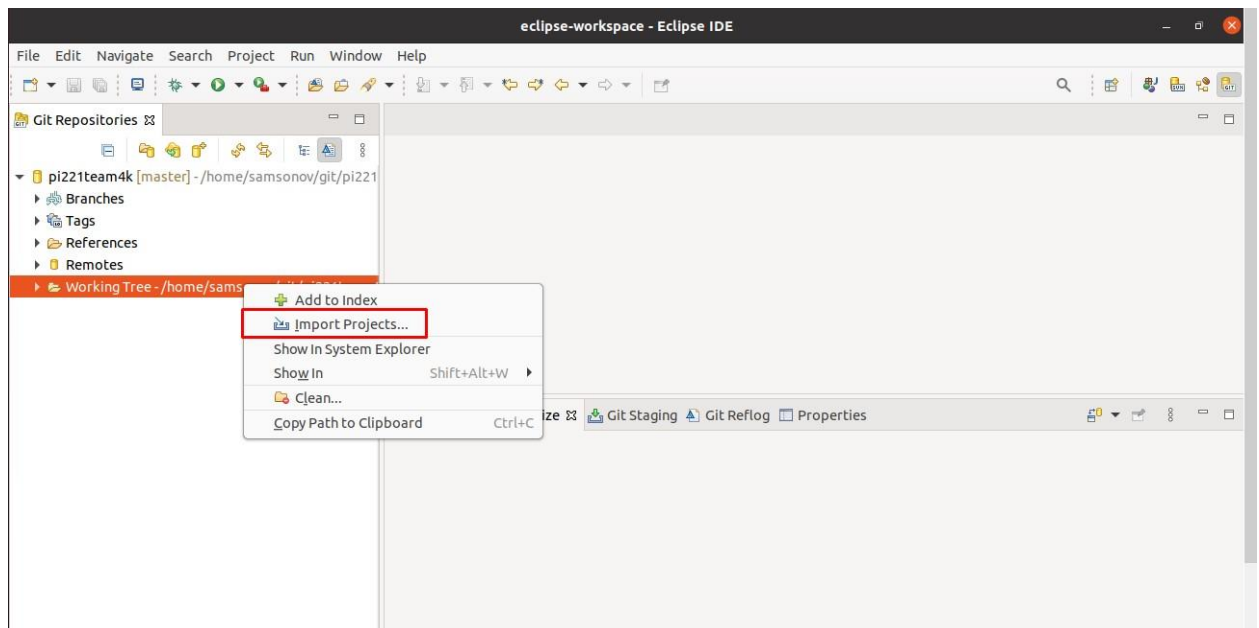


Рисунок 55 – Выгрузка проекта в рабочую копию

Для работы с импортированным проектом необходимо перейти в Project Explorer (смотреть рисунок 56).

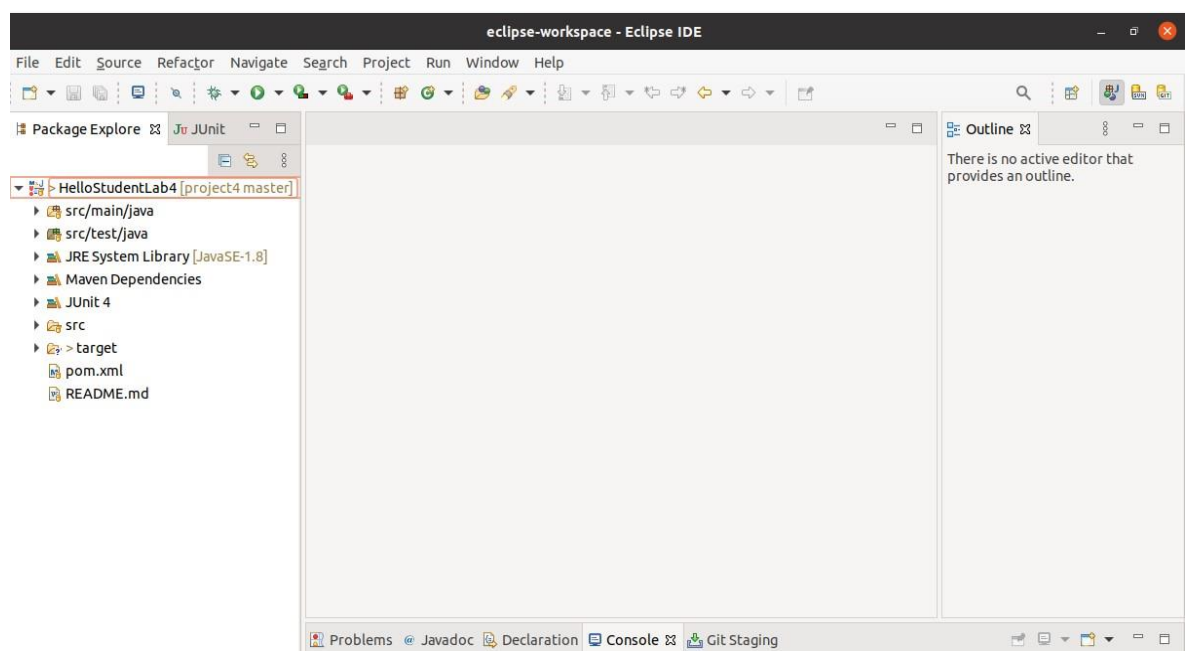


Рисунок 56 – Импортированный проект

Раздел 5. Реализация исходного кода по зонам ответственности

Разработка ПО выполнялась на языке высокого уровня программирования Java с использованием инструментальной среды Eclipse. Калькулятор расчёта потребительского кредита выполнен в виде Web-приложения.

Web-приложение – клиент-серверное приложение, в котором происходит взаимодействие клиента с web-сервером при помощи браузера. В таблице 1 представлено распределение по зонам ответственности.

Ссылка на репозиторий, в котором содержится проект курсовой работы:

<https://github.com/mxrph/pi221team4k>

Таблица 1 – Распределение по зонам ответственности

	ФИО модератора/разработчика	Зона ответственности	Приложение
	Самсонов Артём Андреевич	Создание класса <i>PDF</i> , jsp-форм: <i>Login</i> , <i>Results</i> , <i>User</i> , <i>LoginFalse</i>	см. Приложение В-1 Приложение В-2 Приложение В-3 Приложение В-4 Приложение В-5
	Фахретдинова Кристина Олеговна	Создание абстрактного класса <i>Uslovie</i> и абстрактного метода <i>Calculation</i> . Класс <i>Check</i> , <i>Config</i>	см. Приложение В-6 Приложение В-7 Приложение В-8
	Шейфель Жанна Владимировна	Создание классов <i>DlyaSotrudnikov</i> , <i>DlyaZaiomshikov</i> и <i>Standart</i> на основе абстрактного класса, переопределение абстрактного метода <i>Calculation</i> , jsp-формы: <i>komanda</i>	см. Приложение В-9 Приложение В-10 Приложение В-11 Приложение В-16
	Степанян Степан Арменович	Создание классов <i>Main</i> , <i>Login</i> , <i>Admin</i> , jsp-формы <i>Admin</i>	см. Приложение В-12 Приложение В-13 Приложение В-14 Приложение В-15

В связи с изменением структуры кода на рисунке 57 представлена новая диаграмма классов.

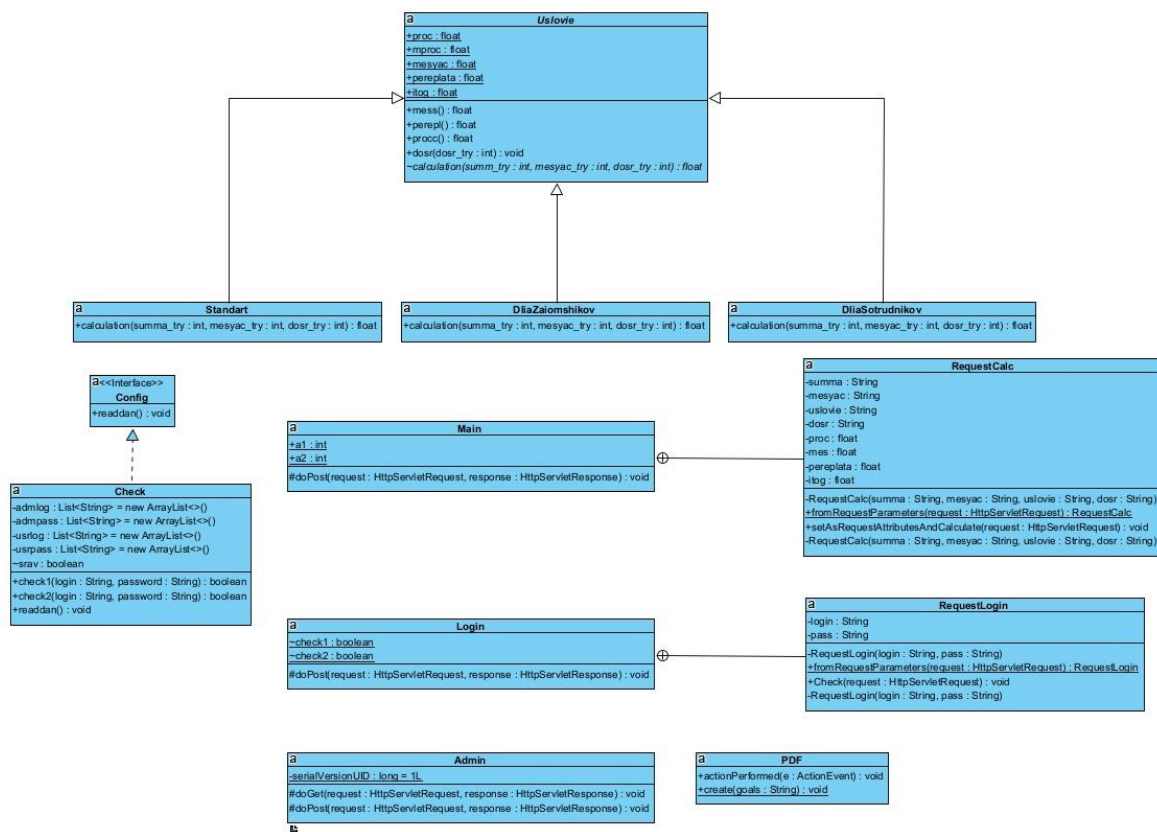


Рисунок 57 – Новая диаграмма классов

Раздел 6. Сборка и тестирование программного продукта

Сборка курсового проекта происходит с использованием *Maven*. Тестирование происходит с использованием библиотеки *JUnit*. В таблице 2 представлено описание UNIT-тестов.

Таблица 2 – Описание UNIT теста

	ФИО модератора/разработчика	Описание UNIT-теста	№ приложения
	Самсонов Артём Андреевич	<i>LoginTest</i> - тестируется проверка ввода логина, выполняется для класса <i>Login</i>	см. Приложение Г-1
	Фахретдинова Кристина Олеговна	<i>UslovieTest</i> – тестируется проверка пункта «Планирую погасить кредит досрочно в первый год» для класса	см. Приложение Г-2
	Шейфель Жанна Владимировна	<i>ProcTest</i> – тестируется установка процента в зависимости от условий в классах <i>Standart</i> , <i>DliaZaiomshikov</i> , <i>DliaSotrudnikov</i>	см. Приложение Г-3
	Степанян Степан Арменович	<i>OgranTest</i> – тестируется ограничение вводимых данных для класса <i>Main</i>	см. Приложение Г-4

Рассмотрим структуру проекта по каталогам. На рисунке 58 видно, что проект *pi221team4k* включает в себя каталоги: *artifacts* для хранения библиотек, в котором находятся файлы *Calculate.war* и *webapp-runner.jar* для того, чтобы программный код отображался на сервере с помощью Apache Tomcat, *src/main/java* для хранения классов, содержащих программный код, *src/test* для содержания JUnit тестов, *src/webapp* для хранения файлов *Admin.jsp*. Проект включает окно администратора, в котором находится окно редактирования значений, *User.jsp*, окно простого пользователя, в котором пользователь производит вычисления, *Results.jsp*, окно вывода, в котором выводятся итоговые

значения в зависимости от введенных пользователем данных и *Login.jsp*, многопользовательское окно авторизации.

В папке *webapp* находится каталог *WEB-INF*, в котором находится *web.xml* – конфигурационный файл для работы с сервером, каталог *src/test* и *target*, а также файлы *Procfile* для связывания программного кода с Heroku, конфигурационный файл *Travis.yml* для работы с непрерывной сборкой *travis-ci*, *adminlogpass.txt*, с данными администратора, *userlogpass.txt* с данными пользователя, *pot.xml* и *Result.pdf*.

В каталоге *src/main/java* находятся классы *Check* для проверки ввода логина и пароля пользователя, интерфейс *Config*, *DliaSotrudnikov* для расчёта при выборе пункта «Являюсь сотрудником компании-партнёров этого банка», *DliaZaiomshikov* для расчёта при выборе пункта «Держу зарплатный проект в данном банке», *Login* для окна авторизации *Login.jsp*, а именно ввод логина и пароля пользователя. *Main*, в котором идёт обработка введенных пользователем данных в *User.jsp* и последующим выводом на *Results.jsp*, *PDF* для генерации PDF файла, *Standart* для расчёта при выборе пункта «Стандартные условия», абстрактный класс *Uslovie*, который переопределён классами *DliaSotrudnikov*, *DliaZaiomshikov* и *Standart*. В каталоге *test* находятся тесты, которые были описаны ранее в таблице 2.

В конфигурационном файле *pot.xml* находятся плагины: *maven-compiler-plugin 3.3* для сборки проекта Maven, *maven-war-plugin 3.3.0* для компилирования *war* файлов; зависимости: *junit 4.12* для создания *JUNIT* тестов, *javax.servlet 3.1.0* для подключения сервлетов *java* с помощью которых можно связать подключение между клиентом и сервером, *javax-el 2.2* для получения доступа к *.jsp* файлам (HTML-формы), *webapp-runner 9.0.27.1* для того, чтобы программный код отображался на сервере с помощью Apache Tomcat, *itextpdf 5.5.13* для генерации pdf файлов.

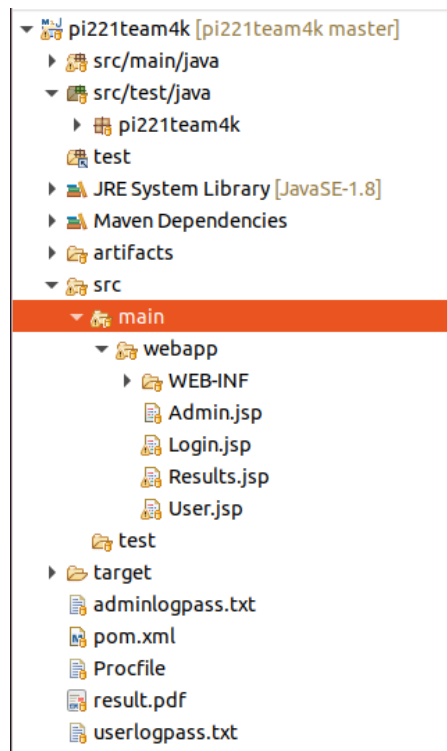


Рисунок 58 – Содержимое проекта *pi221team4k*

Сборка проекта осуществляется следующим образом:

1. Разработчик вносит изменения в программный код и фиксирует изменения на том репозитории ,в котором он работает (смотреть рисунок 59).

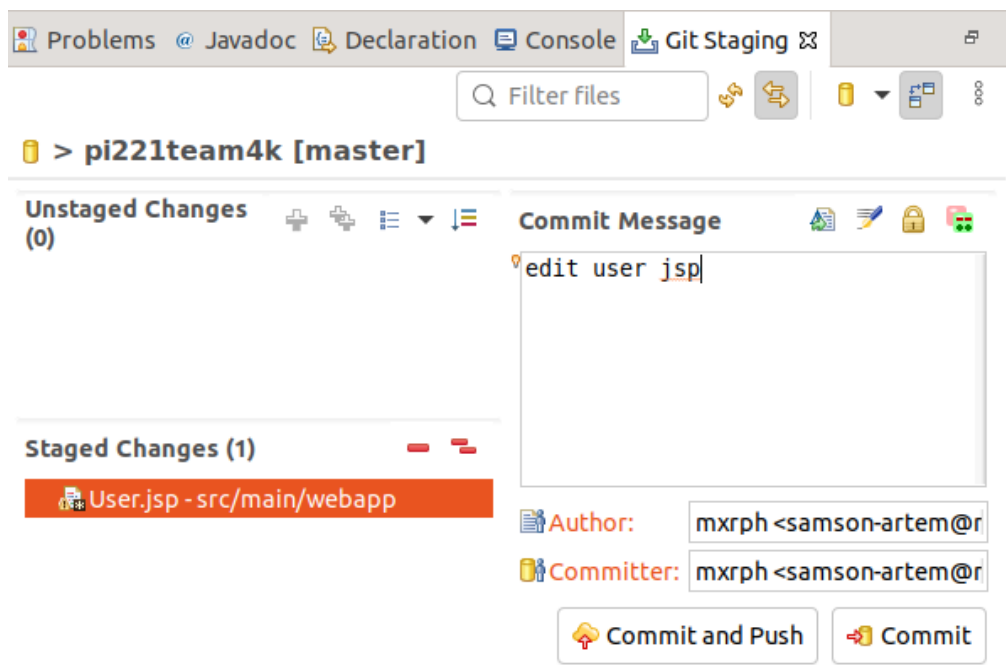


Рисунок 59 – Окно фиксации изменений

2. После отправки фиксации изменений на сервисе непрерывной сборки проектов *travis-ci* происходит сборка проекта (смотреть рисунок 60).

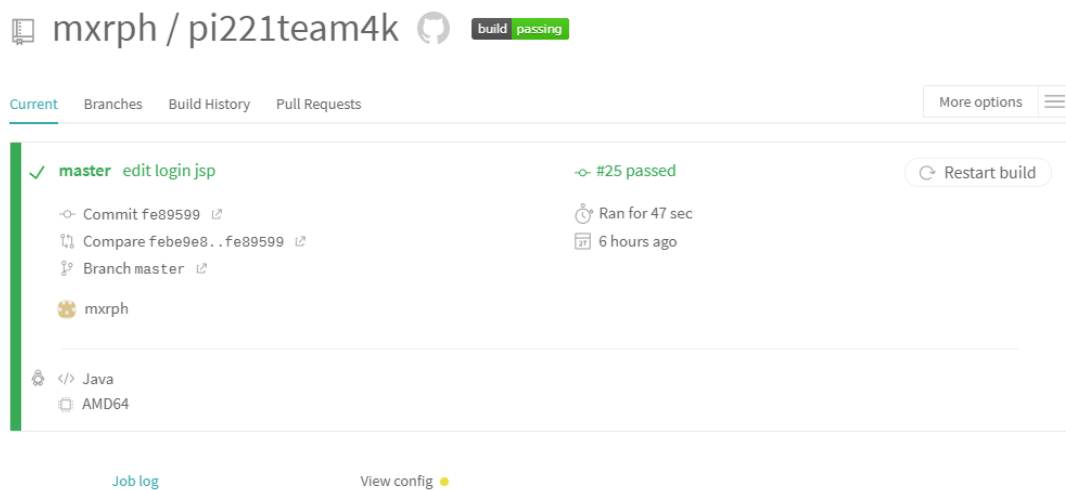


Рисунок 60 – Успешная сборка проекта на *travis-ci*

3. Далее собранный проект автоматически развёртывается на *Heroku* в веб-приложение, в котором пользователь может производить расчёты (смотреть рисунок 61).

Добро пожаловать на сайт калькулятора потребительского кредита!

Для продолжения, необходима авторизация

Логин
Введите логин

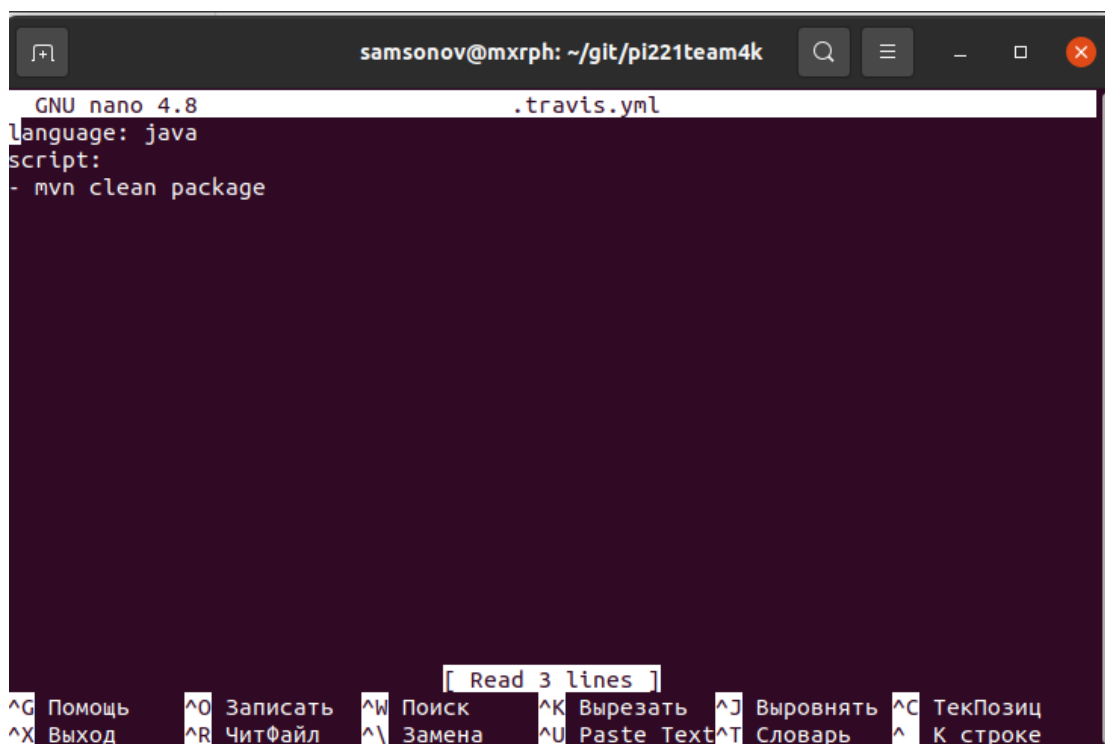
Пароль
Введите пароль

Войти

Рисунок 61 – Реализация развёртывания приложения на *Heroku*

Раздел 7. Настройка программной среды для развёртывания и запуска программного продукта.

При создании Web-приложения «Калькулятор расчёта потребительского кредита» была использована практика непрерывного развёртывания на платформе *Heroku*, за счёт использования системы непрерывной интеграции *Travis CI*. *Travis CI* – распределённый Web-сервис для сборки и тестирования программного обеспечения, использующий *GitHub* в качестве хостинга исходного кода. Конфигурационный файл *.travis.yml* показан на рисунке 62.



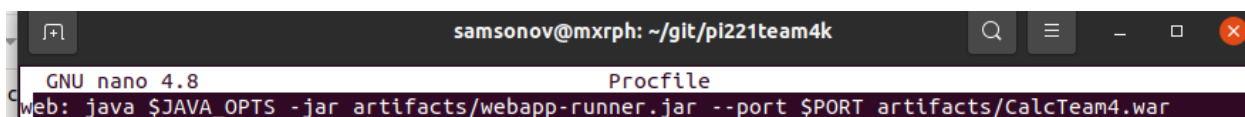
```
GNU nano 4.8 .travis.yml
language: java
script:
- mvn clean package
```

The screenshot shows a terminal window with the title bar 'samsonov@mxrph: ~/git/pi221team4k'. The editor is GNU nano 4.8 editing the file .travis.yml. The content of the file is: language: java, script:, and a list item - mvn clean package. At the bottom, there is a status bar with various keyboard shortcuts in Russian: ^G Помощь, ^O Записать, ^W Поиск, ^K Вырезать, ^J Выводить, ^C ТекПозиц, ^X Выход, ^R ЧитФайл, ^_ Замена, ^U Paste Text, ^T Словарь, and ^_ К строке. A 'Read 3 lines' indicator is also visible.

Рисунок 62 – Содержимое файла *.travis.yml*

В данном файле находятся шаг *script*. Шаг *script* отвечает за сборку проекта “- *mvn clean package*”.

После сборки проекта с помощью системы непрерывной интеграции, проект автоматически начинается разворачиваться на *Heroku*. Для этого есть специальный конфигурационный файл *Procfile* показанный на рисунке 63.



```
GNU nano 4.8 Procfile
web: java $JAVA_OPTS -jar artifacts/webapp-runner.jar --port $PORT artifacts/CalcTeam4.war
```

The screenshot shows a terminal window with the title bar 'samsonov@mxrph: ~/git/pi221team4k'. The editor is GNU nano 4.8 editing the file Procfile. The content of the file is: web: java \$JAVA_OPTS -jar artifacts/webapp-runner.jar --port \$PORT artifacts/CalcTeam4.war.

Рисунок 63 – Содержимое файла *Procfile*

Когда все конфигурационные файлы были настроены после каждой фиксации изменений, проект будет отправлен в систему непрерывной интеграции *travis-ci* (смотреть рисунок 64).

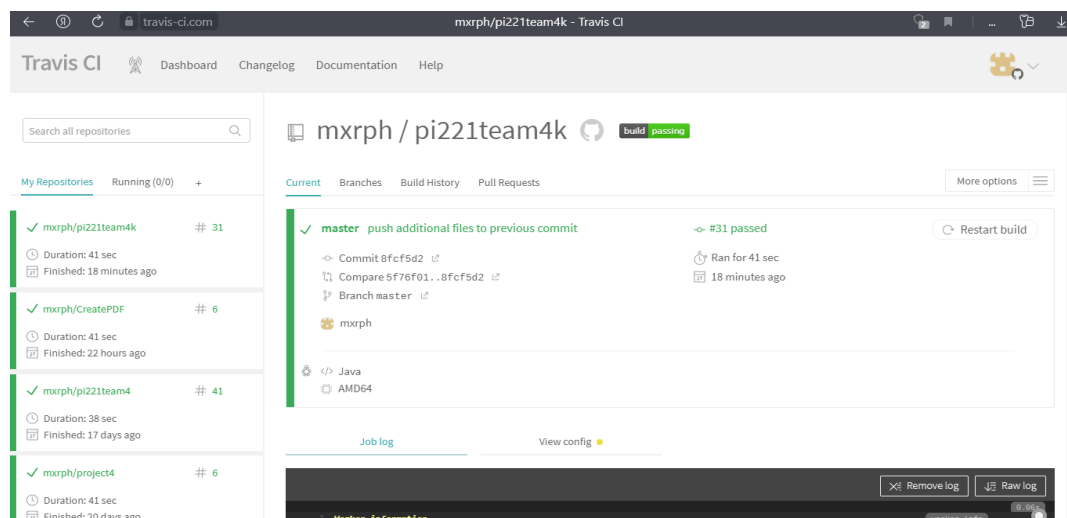


Рисунок 64 – Окно *travis-ci* со сборкой проекта.

Далее, после сборки проекта уже заранее созданное приложение на *Heroku* начинает автоматически разворачиваться. Чтобы сделать автоматическое развёртывание приложение необходимо нажать на само приложение, перейти во вкладку Deploy, связать приложение с репозиторием, где находится программный код. После выбрать в нём ветку и нажать на кнопку “Enable automatic deploys” (смотреть рисунок 65). После данных действий любое изменение в программном коде будет автоматически собираться и развёртываться. В нашем случае ссылка на приложение *team4pi221.herokuapp.com* (смотреть рисунок 66).

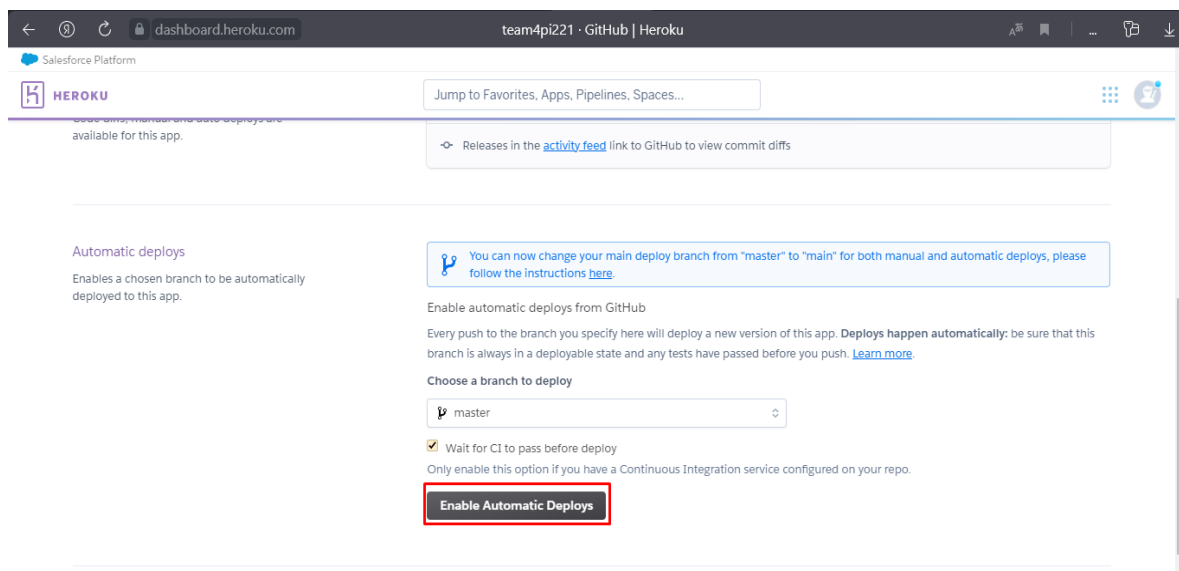
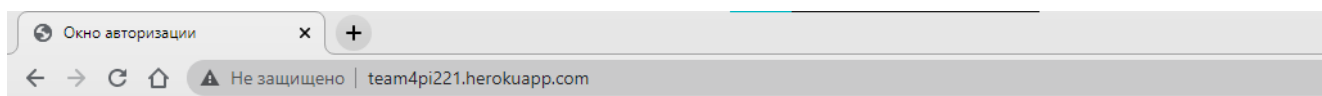


Рисунок 65 – Настройка автоматического развёртывания приложения



Добро пожаловать на сайт калькулятора потребительского кредита!

Для продолжения введите логин и пароль

Логин
Логин

Пароль
Пароль

Войти

Рисунок 66 – Приветственное окно приложения team4pi221

Данная настройка подходит для браузеров операционной системы Windows, таких как Яндекс.Браузер, Google Chrome, Internet Explorer. А также для браузеров на операционной системе Linux, таких как Mozilla Firefox и Яндекс.Браузер.

Раздел 8. Руководство пользователя программного продукта

Руководство пользователя программного продукта, которое включает в себя назначение программы, условия её выполнения, непосредственно само выполнение программы и сообщения оператору представлено в приложении Б-1.

Заключение

В ходе выполнения данной курсовой работы были получены практические навыки установки и настройки инструментальной среды Eclipse в операционных системах *Windows*, *Linux Ubuntu* и *OpenSUSE*, представления о совместной разработке программного обеспечения с использованием системы контроля версий *Git*. Была изучена непрерывная интеграция с использованием сервиса для сборки и тестирования программного обеспечения *Travis CI*. Приобретены навыки использования практики непрерывного развёртывания с помощью платформы *Heroku*. Получены представления о формировании документации при разработке программного обеспечения, а также освоены навыки программирования на объектно-ориентированном языке *Java*, с практической реализацией классов и интерфейсов, сервлетов, JSP-форм.

УТВЕРЖДЕН
АСУ-1304.300004.000 ТУ

КАЛЬКУЛЯТОР ПОТРЕБИТЕЛЬСКОГО КРЕДИТА

Техническое задание
АСУ-1304.300004.000 ТУ

12 Листов

Инд N подп	Подп и дата	Взам инд N	Инд М дубл	Подп и дата

СОДЕРЖАНИЕ

1. Введение.....	3
1.1. Наименование программы.....	3
1.2. Краткая характеристика области применения программы.....	3
2. Основания для разработки.....	4
2.1. Основания для проведения разработки.....	4
2.2. Наименование и условное обозначение темы разработки.....	4
3. Назначение разработки.....	5
3.1. Функциональное назначение.....	5
3.2. Эксплуатационное назначение программы.....	5
4. Требования к программе или программному изделию.....	6
4.1. Требования к функциональным характеристикам.....	6
4.2. Требования к надежности.....	7
4.3. Условия эксплуатации.....	7
4.4. Требования к составу и параметрам технических средств.....	7
4.5. Требования к информационной и программной совместимости.....	8
4.6. Требования к маркировке и упаковке.....	8
4.7. Требования к транспортированию и хранению.....	8
5. Требования к программной документации.....	9
5.1. Предварительный состав программной документации.....	9
6. Техничко-экономические показатели.....	10
7. Стадии и этапы разработки.....	11
7.1. Стадии разработки.....	11
7.2. Этапы разработки.....	11
7.3. Содержание работ по этапам.....	11
8. Порядок контроля и приемки.....	12
8.1. Виды испытаний.....	12
8.2. Общие требования к приемке работы.....	12

1. ВВЕДЕНИЕ

1.1 Наименование программы

Калькулятор потребительского кредита.

1.2 Краткая характеристика области применения программы

Программный продукт предназначен для самостоятельного расчёта ежемесячного платежа по потребительскому кредиту.

2. ОСНОВАНИЯ ДЛЯ РАЗРАБОТКИ

2.1 Основания для проведения разработки

Основанием для проведения разработки является учебный план кафедры автоматизированных систем управления (АСУ), согласно которому студенты учебной группы направления "Прикладная информатика" обязаны разработать кроссплатформенный программный продукт на языке Java для курсовой работы по дисциплине "Информационные системы".

Согласно плану-графику выполнения курсовой работы студенты обязаны продемонстрировать работоспособность программного продукта, предоставить исходный код и документацию к разработанной системе преподавателю данной дисциплины не позднее 29.05.2021.

2.2 Наименование и условное обозначение темы разработки

Наименование темы разработки – «Разработка программного продукта для расчёта потребительского кредита».

Условное обозначение – «Калькулятор ПК».

3. НАЗНАЧЕНИЕ РАЗРАБОТКИ

3.1 Функциональное назначение

Функциональным назначением программного продукта является предоставление информации по выбранным условиям потребительского кредита, быстрый и безошибочный расчёт ежемесячной суммы выплат, процентной ставки по кредиту, общей суммы выплат вместе с переплатой.

3.2 Эксплуатационное назначение программы

Разрабатываемый программный продукт предназначен для использования в некоммерческих целях. Программа выступает в роли помощника: каждый пользователь данной программы может самостоятельно рассчитать сумму выплат, исходя из введенных им данных, и оценить свои возможности перед обращением в кредитную организацию. Программный продукт будет использоваться двумя группами пользователей: пользователь и администратор.

4. ТРЕБОВАНИЯ К ПРОГРАММЕ ИЛИ ПРОГРАММНОМУ ИЗДЕЛИЮ

4.1. Требования к функциональным характеристикам

4.1.1 Требования к составу выполняемых функций

Программный продукт должен обеспечивать возможность выполнения перечисленных ниже функций:

1. Авторизация пользователей;
2. Выбор условий для кредитования;
3. Выбор срока кредитования;
4. Выбор суммы для кредита;
5. Сведения о разработчиках программного продукта по запросу в самой программе;
6. Автоматизированный расчёт по заданным условиям;
7. Запрос на печатную форму с расчётами по введённым данным;
8. Ввод новых значений параметров.

4.1.2 Требования к организации входных данных

Входные данные программного продукта должны быть организованы в виде вводимого в специальную форму текста, выбора нужного пункта флажком, а также в виде выбора одного из вариантов из выпадающего списка. Данные, которые будут вводиться вручную, проверяются на корректность после попытки расчёта.

4.1.3 Требования к организации выходных данных

Выходные данные программного продукта должны быть организованы в виде отчёта в цифровом и бумажном виде, в которых предоставлены все результаты, произведённые, исходя из введённых пользователем данных.

4.1.4 Требования к временным характеристикам

Специальных требований к временным характеристикам программного продукта не предъявляется.

4.2. Требования к надежности

4.2.1 Требования к обеспечению надёжного функционирования

Надёжное функционирование зависит от организации бесперебойного питания технических средств, также пользователю, работающему с программным продуктом через web-браузер должен быть предоставлен непрерывный доступ к нему. Веб-сервис не должен прерывать свою работу. Корректная работа программного продукта напрямую влияет от входной информации (авторизация, сумма кредита, срок кредитования).

4.3. Условия эксплуатации

4.3.1 Климатические условия эксплуатации

Особых требований к климатическим условиям эксплуатации не предъявляется.

4.3.2 Требования к видам обслуживания

В перечень задач, выполняемых системным администратором, должны входить: задача поддержания работоспособности технических средств; задачи своевременной замены необходимых параметров с целью поддержания актуальности программного продукта.

4.4. Требования к составу и параметрам технических средств

В состав технических средств должен входить персональный компьютер.

Персональный компьютер пользователя:

Процессор: частота 1ГГц и выше;

Оперативная память: 1.5 GB ОЗУ;

Мышь, клавиатура, монитор;

Сервер:

CPU 1 ядро по 2ГГц;

Оперативная память: 1.5 GB ОЗУ;

Не менее 512 мб дискового пространства.

4.5. Требования к информационной и программной совместимости

4.5.1 Требования к информационным структурам и методам решения

Пользовательский интерфейс должен быть интуитивно понятным. Отчеты должны содержать лишь интересующую информацию.

4.6. Требования к маркировке и упаковке.

Не предъявляются.

4.7. Требования к транспортированию и хранению

Не предъявляются.

5. ТРЕБОВАНИЯ К ПРОГРАММНОЙ ДОКУМЕНТАЦИИ

5.1 Предварительный состав программной документации

Состав программной документации должен включать :

1. Техническое задание (включает описание применения);
2. Программа и методика испытаний;
3. Руководство администратора;
4. Руководство пользователя ;
5. Текст программы.

6. ТЕХНИКО-ЭКОНОМИЧЕСКИЕ ПОКАЗАТЕЛИ

Данный программный продукт не разрабатывается с целью получения экономической выгоды.

Предполагаемая потребность: данный продукт всегда будет актуален.

В мире сильно развита тенденция взятия кредитов, а для расчёта потребительского кредита, пользователь всегда может воспользоваться нашим программным продуктом и ему не потребуется для этого посещение отделение банка.

Экономические преимущества для данного программного продукта не ставятся, потому что массовый выпуск не планируется.

7. СТАДИИ И ЭТАПЫ РАЗРАБОТКИ

В таблице 1 представлен план-график выполнения курсовой работы по дисциплине «Информационные системы».

Таблица 1 – План-график выполнения курсовой работы

Наименование этапа работ	Трудоемкость выполнения, час	Процент к общей трудоемкости выполнения	Срок предъявления консультанту	Отметка о выполнении
Получение и согласование задания	1,7	1,7%	27 неделя	
Раздел 1. Описание предметной области	20	20%	29 неделя	
Раздел 2. Техническое задание на создание программного продукта	10	10%	30 неделя	
Раздел 3. Настройка среды разработки для операционных систем семейств Windows и Linux	10	10%	31 неделя	
Раздел 4. Настройка среды разработки для подключения к системе контроля версий	7	7%	32 неделя	
Раздел 5. Реализация исходного кода по зонам ответственности	23	23%	34 неделя	
Раздел 6. Сборка и тестирование программного продукта	8	8%	35 неделя	
Раздел 7. Настройка программной среды для развертывания и запуска программного продукта	10	10%	36 неделя	
Раздел 8. Руководство пользователя программного продукта	10	10%	37 неделя	
Защита	0,3	0,3%	38 неделя	

8. ПОРЯДОК КОНТРОЛЯ И ПРИЕМКИ

8.1 Виды испытаний

На данном этапе осуществляется функциональное тестирование программы, визуальная проверка интерфейса программы на соответствие пункту 4.1 настоящего технического задания.

8.2 Общие требования к приемке работы

Прием программного продукта будет утвержден при корректной работе программного продукта и выполнении поставленных задач в соответствии с настоящим техническим заданием.

Процедура защиты курсовой работы предполагает следующие этапы:

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.

Калькулятор потребительского кредита

Руководство пользователя

Листов 11

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

АННОТАЦИЯ

Данный документ является руководством пользователя для web-приложения “Калькулятор потребительского кредита”.

СОДЕРЖАНИЕ

1. Назначение программы.....	4
2. Условия, необходимые для работы приложения.....	5
3. Выполнение программы.....	6
3.1 Авторизация простого пользователя.....	6
3.2 Авторизация пользователя как администратора.....	9
4. Сообщения оператору.....	11

1. Назначение программы

Программный продукт предоставляет информацию по выбранным условиям потребительского кредита, быстрый и безошибочный расчёт ежемесячной суммы выплат, процентной ставки по кредиту и общей суммы выплат вместе с переплатой.

2. Условия, необходимые для работы приложения

- В состав технических средств должен входить персональный компьютер.
- ОС: Windows 7,8.1, Windows 10, Linux Ubuntu 20.04.2
- Процессор: частота 1ГГц и выше
- Оперативная память: 1.5 GB ОЗУ
- Наличие доступа к сети Интернет
- Наличие браузера Google Chrome, Firefox, Yandex

3. Выполнение программы

3.1 Авторизация простого пользователя

После ввода адреса в рабочем окне браузера появится форма авторизации (смотреть рисунок 1).

Добро пожаловать на сайт калькулятора потребительского кредита!

Для продолжения, необходима авторизация

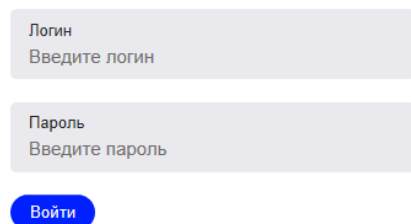
The image shows a login form with two input fields. The first field is labeled 'Логин' (Login) and contains the placeholder text 'Введите логин' (Enter login). The second field is labeled 'Пароль' (Password) and contains the placeholder text 'Введите пароль' (Enter password). Below these fields is a blue button with the text 'Войти' (Login).

Рисунок 1 – Окно авторизации

Для того, чтобы пользователь смог воспользоваться функционалом калькулятора, ему потребуется воспользоваться своими авторизационными данными – логином и паролем.

При вводе имени пользователя и пароля необходимо убедиться в правильности установки регистров клавиатуры (клавиша CapsLock). Поскольку для безопасности вводимый пароль подменяется на экране «звездочками» (или иными символами в зависимости от используемого браузера), ошибки при наборе часто связаны так же с неверным выбором языка. При копировании пароля необходимо убедиться, что копируются только символы пароля без пробела до или после них.

В случае, если пользователь ввёл верный логин и пароль он получит доступ к калькулятору (смотреть рисунок 2).

The image shows a digital form for a consumer credit calculator. At the top right, there is a link 'О нас' and the number '6'. The title of the form is 'Калькулятор потребительского кредита'. The form contains five numbered input fields: 1. 'Введите сумму кредита от 123 до 123 рублей' with a sub-label 'Сумма'; 2. 'Введите срок выплат от 12 до 60 месяцев' with a sub-label 'Месяц'; 3. 'Выберите условия кредита' with a dropdown menu showing 'Стандартные условия'; 4. 'Собираетесь ли вы досрочно погашать кредит?' with radio buttons for 'Да' and 'Нет'; 5. A blue button labeled 'Вычислить'. A hand is shown pointing at the 'Вычислить' button. The background of the form is a blurred image of a calculator.

Рисунок 2 – Форма для ввода данных

Содержимое основной страницы:

1. В данном поле пользователь указывает сумму, на которую хочет взять кредит;
2. В данном поле пользователь указывает срок, на который он хочет взять кредит;
3. Из выпадающего списка пользователь выбирает условия кредитования;
4. В этом пункте пользователь указывает, собирается ли он досрочно погашать кредит.
5. Кнопка вычислить, при нажатии на которую пользователь перейдёт на форму с расчётами по введенным данным.
6. При нажатии на кнопку «О нас» появляется информация о разработчиках (смотреть рисунок 3).

Разработчики команды №4

Модератор: Самсонов Артём Андреевич

Разработчик №1: Фахретдинова Кристина Олеговна

Разработчик №2: Шейфель Жанна Владимировна

Разработчик №3: Степанян Степан Арменович

[Вернуться к калькулятору](#) 1

Рисунок 3 – Информация о разработчиках

1. При нажатии на кнопку «Вернуться», пользователь возвращается к калькулятору.

Для того чтобы расчёты выполнились, пользователь должен заполнить пункты 1–4.

При верно введенных данных пользователь увидит на экране расчёт (смотреть рисунок 4).

Результат

Ваша процентная ставка: 23.49%

Ежемесячный платеж составляет: 11622.28 руб.

Переплата по кредиту составит: 16236.36 руб.

Общая сумма: 139467.36 руб.

1

Ввести новые данные

2

Открыть PDF-файл

3

Скачать PDF файл

Рисунок 4 – Результат расчётов

Содержимое страницы с результатами расчётов:

1. Возвращает пользователя на основную страницу;
2. Открытие PDF файла в браузере без скачивания;
3. Экспорт pdf файла с результатами расчётов.

При выборе варианта “Скачать PDF файл” пользователю потребуется выбрать место, где он хочет сохранить файл.

При выборе варианта “открыть” пользователю в новой вкладе откроется pdf файл с расчётами. Образец PDF файла показан на рисунке 5.

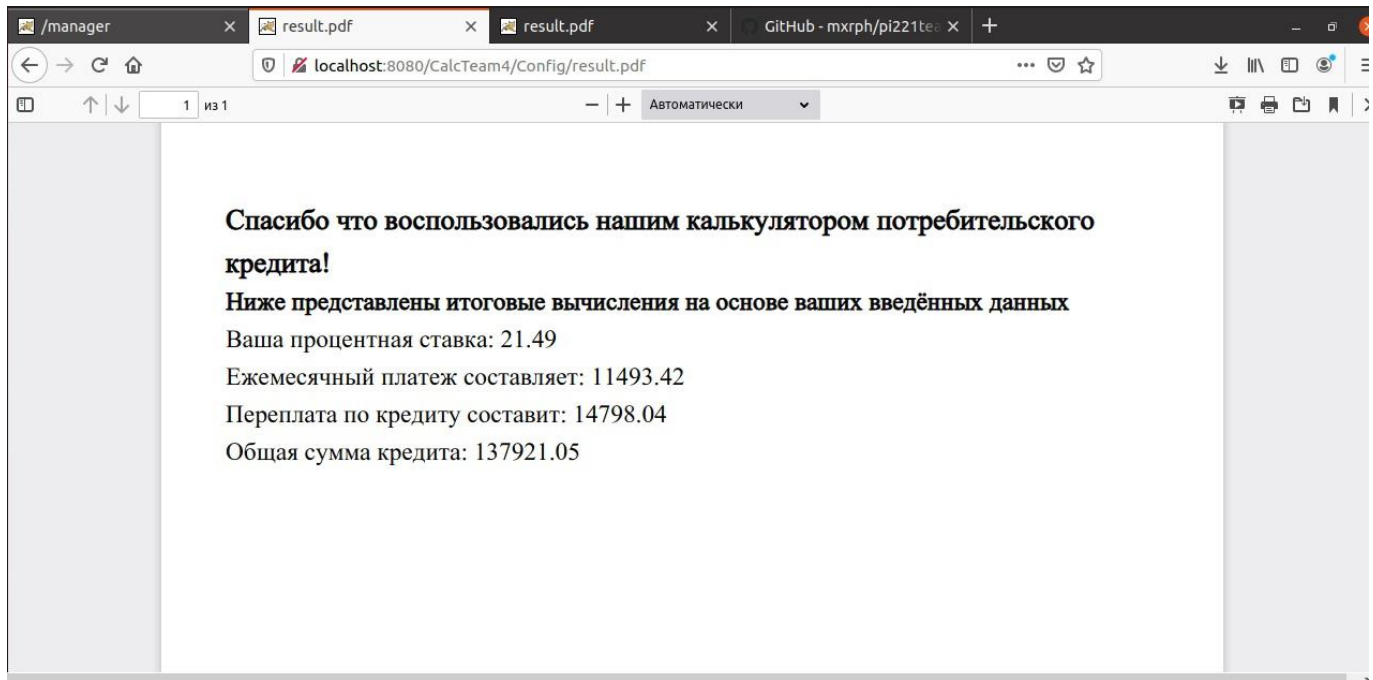


Рисунок 5 – Открытие pdf файла в браузере

3.2 Авторизация пользователя как администратора

Для того чтобы авторизоваться как администратор, пользователь должен авторизоваться с соответствующими данными.

После успешного прохождения авторизации администратор перейдёт на форму, где будут изменяться коэффициенты (смотреть рисунок 6).

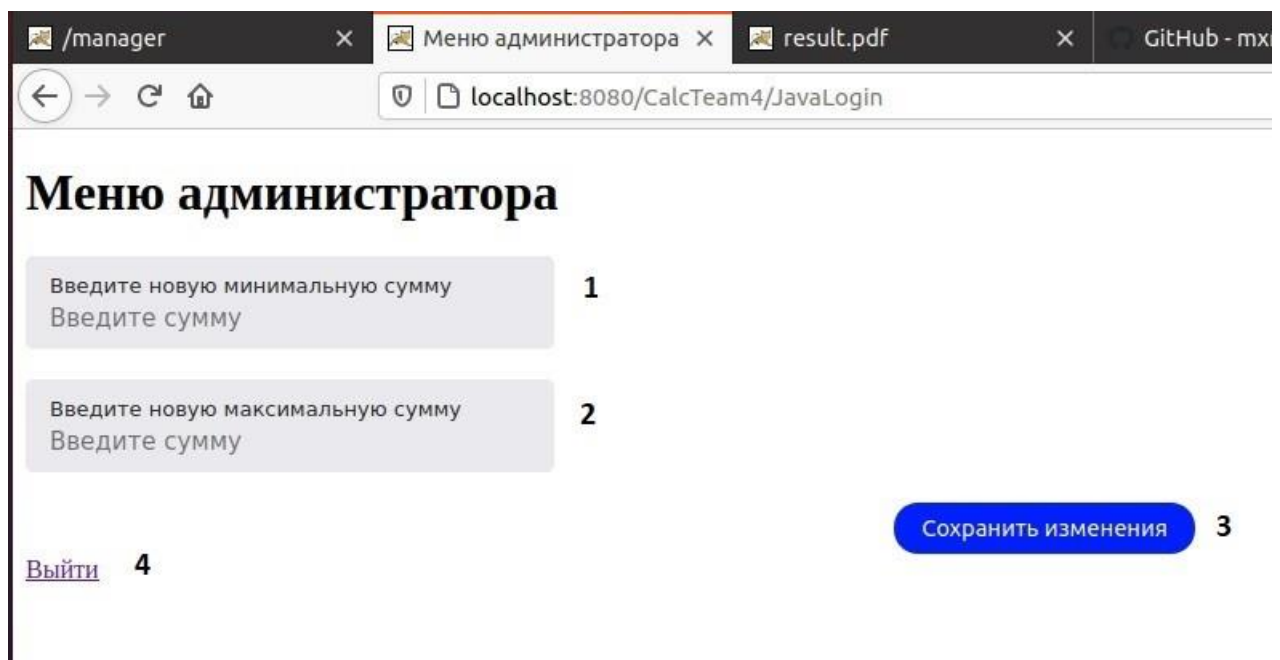


Рисунок 6. Меню администратора

Содержимое страницы администратора:

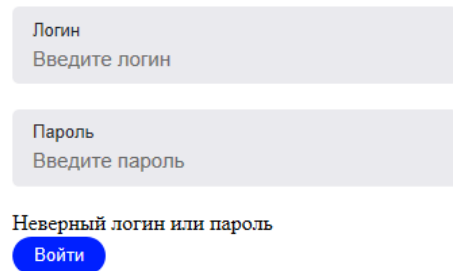
1. При помощи данного поля администратор может изменить минимально возможную сумму, которую может ввести пользователь для произведения расчёта кредита;
2. При помощи данного поля администратор может изменить максимально возможную сумму, которую может ввести пользователь для произведения расчёта кредита;
3. Для сохранения внесённых изменений, администратору требуется нажать на данную кнопку;
4. Для возврата в окно авторизации, администратору требуется нажать эту кнопку.

4. Сообщения оператору

При неверно введённых данных в форме авторизации пользователю выйдет текстовое оповещение об этом (смотреть рисунок 7).

Добро пожаловать на сайт калькулятора потребительского кредита!

Для продолжения, необходима авторизация



Логин
Введите логин

Пароль
Введите пароль

Неверный логин или пароль

Войти

Рисунок 7 – Форма с информированием о неверно введенных данных

Если пользователь ввёл данные для расчётов, не входящих в диапазон, указанный на сайте, то вместо расчёта ему выйдет окно, в котором расчёты не произведены (смотреть рисунок 8).

Результат

Ваша процентная ставка: 25.11%

Ежемесячный платеж составляет: 0.0 руб.

Переплата по кредиту составит: 0.0 руб.

Общая сумма: 0.0 руб.

Рисунок 8 – Форма без расчёта

КЛАСС PDF.java

```

package team4;
import java.awt.event.*;
import java.io.File;
import java.io.FileNotFoundException;
import java.io.FileOutputStream;
import java.io.IOException;

import com.itextpdf.text.Document;
import com.itextpdf.text.DocumentException;
import com.itextpdf.text.Font;
import com.itextpdf.text.Paragraph;
import com.itextpdf.text.pdf.BaseFont;
import com.itextpdf.text.pdf.PdfWriter;

public class PDF implements ActionListener {

    public static void create(String goals) throws IOException {
        Document document = new Document();
        String filepath = new File("").getCanonicalPath();
        String[] parsfilepath = filepath.split("/");

        int lengthpath = parsfilepath.length;
        String abspath="";
        for(int i=0;i<(lengthpath-1);i++) {
            abspath=abspath+parsfilepath[i]+"/";
        }
        filepath=abspath+"webapps/CalcTeam4/Config/result.pdf";
        String fontpath =abspath+"/webapps/CalcTeam4/fonts/times.ttf";
        try {
            PdfWriter.getInstance(document, new
FileOutputStream(filepath));
        } catch (FileNotFoundException | DocumentException e) {
            e.printStackTrace();
        }

        document.open();

        BaseFont times = null;
        try {
            times = BaseFont.createFont(fontpath, "cp1251",
BaseFont.EMBEDDED);

```

```

    } catch (DocumentException | IOException e) {
        e.printStackTrace();
    }

    String ty = "Ñiàñèái ÷ôi âîñîîëüçîââèèñü íàøèì èâëüéóëÿòîðîî
îîððââèèòâëüñîîîî êðââèèòà!";
    Paragraph paragraph = new Paragraph();
    paragraph.add(new Paragraph(ty, new Font(times,16, Font.BOLD)));

    String lower = "Íèæå ððââñòââèâíû èòîîîîûâ âû÷èñëèèëü íà îñíîââ âàøèò
ââââ,ííûð ââííûð";
    paragraph.add(new Paragraph(lower, new Font(times,14, Font.BOLD)));

    String string_pdf = "Âàøà ððîâîîíîîâ ñòââèè: " + Uslovie.proc + "%";
    paragraph.add(new Paragraph(string_pdf, new Font(times,14)));

    String string_pdf2 = "Åæîîîñü÷íúé îëàòâæ ñîñòââèèâò: " +
Uslovie.mesyac + "ðóá.";
    paragraph.add(new Paragraph(string_pdf2, new Font(times,14)));

    String string_pdf3 = "Ïððîîèàòà ïî êðââèèòò ñîñòââèèò: " +
Uslovie.pereplata + "ðóá.";
    paragraph.add(new Paragraph(string_pdf3, new Font(times,14)));

    String string_pdf4 = "Íáùàÿ ñîíîâ êðââèèòà: " + Uslovie.itog + "ðóá.";
    paragraph.add(new Paragraph(string_pdf4, new Font(times,14)));

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

}

@Override
public void actionPerformed(ActionEvent e) {
    // TODO Auto-generated method stub

}

}

```

КЛАСС Result.jsp

```

%@ page language="java" contentType="text/html; charset=UTF-8"
    pageEncoding="UTF-8"%>
<!DOCTYPE html>
<html>
<head>
<meta charset="UTF-8">
<title>Результат:</title>
</head>
<body>
<style>
.center{
text-align:center;
}
.font {
font-family: "Times New Roman", Times,serif }
input[type=submit] {
color: #fff; /* цвет текста */
text-decoration: none; /* убирать подчёркивание у ссылок */
user-select: none; /* убирать выделение текста */
background: rgb(0,32,255); /* фон кнопки */
padding: .5em 1.2em; /* отступ от текста */
outline: none; /* убирать контур в Mozilla */
border: none; /* убираем некрасивую тень */
border-radius:15px;
}
input[type=submit]:hover {background: rgb(0,96,255); }
input[type=submit]:active { background: rgb(0,64,0); }
a.kнопка {
color: #fff; /* цвет текста */
text-decoration: none; /* убирать подчёркивание у ссылок */
user-select: none; /* убирать выделение текста */
background: rgb(0,32,255); /* фон кнопки */
padding: .5em 1.3em; /* отступ от текста */
outline: none; /* убирать контур в Mozilla */
border-radius:15px;
}
a.kнопка:hover { background: rgb(0,96,255); } /* при наведении курсора
мышки */
a.kнопка:active { background: rgb(0,64,0); } /* при нажатии */
</style>
<div class ="font">

```

<h1>Результат</h1>

<p>Ваша процентная ставка: \${proc}%</p>

<p>Ежемесячный платеж составляет: \${mes}
руб.</p>

<p>Переплата по кредиту составит: \${pereplata}
руб.</p>

<p>Общая сумма: \${itog} руб.</p>

<div class="center"><form
action="\${pageContext.request.contextPath}/User.jsp">
<input type="submit" name="sign" value="Ввести новые
данные">

</form>

Открыть PDF-файл

<a type="submit" href="/CalcTeam4/Config/result.pdf" class="knopka"
download>Скачать PDF файл</div>

</div>

</body>

</html>

КЛАСС Login.jsp

```

<%@ page language="java" contentType="text/html; charset=UTF-8"
    pageEncoding="UTF-8"%>
<!DOCTYPE html>
<html>
<head>
<meta charset="UTF-8">
<title>Окно авторизации</title>
</head>
<body>
<style>
input[type=submit] {
    color: #fff; /* цвет текста */
    text-decoration: none; /* убирать подчёркивание у ссылок */
    user-select: none; /* убирать выделение текста */
    background: rgb(0,32,255); /* фон кнопки */
    padding: .5em 1.2em; /* отступ от текста */
    outline: none; /* убирать контур в Mozilla */
    border-radius: 15px;
    border: none; /* убираем некрасивую тень */
}
input[type=submit]:hover {background: rgb(0,96,255); }
input[type=submit]:active { background: rgb(0,64,255); }
.form-fields-group {
    position: relative;
}
.form-fields-group-label {
    position: absolute;
    font-size: 13px;
    color: #27252a;
    font-family: 'Roboto', sans-serif;
    top: 10px;
    left: 15px;
}
.form-fields-input {
    width: 300px;
    padding: 30px 15px 10px;
    border: none;
    background: #EAEAEA;
    border-radius: 5px;
    font-family: 'Roboto', sans-serif;

```

```

    font-size: 15px;
}
input:focus ~ label, label:focus {
    color: blue;
}
.font {
font-family: "Times new roman", Times, serif
}
</style>
<div class="a">
<div class="font">
<h1>Добро пожаловать на сайт калькулятора потребительского
кредита!</h1>
<div>
<h3>Для продолжения введите логин и пароль</h3>
    <div class="form">
        <form
method="post"
            action="{pageContext.request.contextPath}/JavaLogin"
            >
            <div class="form-fields-group">
                <label class="form-fields-group-label" for="login">Логин</label>
                <input type="text" name="login" id="login" class="form-fields-input"
placeholder="Логин" value="{login}" required><br><br>
            </div>
            <div class="form-fields-group">
                <label class="form-fields-group-label" for="pass">Пароль</label>
                <input type="password" name="pass" id="pass" class="form-fields-input"
placeholder="Пароль" value="{pass}" required><br><br>
                <input type="submit" name="sign" value="Войти">
            </div>
        </form>
    </div>
</div>
</div>
</body>
</html>

```

КЛАСС User.jsp

```

<% @ page language="java" contentType="text/html; charset=UTF-8"
    pageEncoding="UTF-8"%>
<!DOCTYPE html>
<html>
<head>
<meta charset="UTF-8">
<title>Калькулятор потребительского кредита</title>
</head>
<body>
<style>
    .box {
        border:0;
        padding:5px;
        border-radius:15px;
    }
    input[type=submit] {
        color: #fff; /* цвет текста */
        text-decoration: none; /* убирать подчёркивание у ссылок */
        user-select: none; /* убирать выделение текста */
        background: rgb(0,32,255); /* фон кнопки */
        padding: .5em 1.2em; /* отступ от текста */
        outline: none; /* убирать контур в Mozilla */
        border-radius:15px;
        border: none; /* убираем некрасивую тень */
    }
    input[type=submit]:hover { background: rgb(0,96,255); }
    input[type=submit]:active { background: rgb(0,64,255); }
    .form-fields-group {
        position: relative;
    }
    .form-fields-group-label {
        position: absolute;
        font-size: 13px;
        color: #27252a;
        font-family: 'Roboto', sans-serif;
        top: 10px;
        left: 15px;
    }
    .form-fields-input {
        width: 400px;
        padding: 30px 15px 10px;

```

```

border: none;
background: #EAEAEA;
border-radius: 5px;
font-family: 'Roboto', sans-serif;
font-size: 15px;
}
input:focus ~ label, label:focus {
  color: blue;
}

.new-select-style-wpandyou select {
border-radius: 0;
background: #EAEAEA;
height: 34px;
padding: 5px;
border: none;
outline: none;
font-size: 16px;
line-height: 1;
-webkit-appearance: none;
width: 400px;
}

.select {
width: 100%;
padding: 5px;
border-radius: 10px;
}

.font {
font-family: "Times New Roman", Times, serif }

.a {
  text-align :center;
}

.custom-radio>input {
position: absolute;
z-index: -1;
opacity: 0;
}

/* для элемента label связанного с .custom-radio */
.custom-radio>span {
display: inline-flex;
align-items: center;
user-select: none;
}

/* создание в label псевдоэлемента before со следующими стилями */

```



```

.custom-radio>span::before {
  content: "";
  display: inline-block;
  width: 1em;
  height: 1em;
  flex-shrink: 0;
  flex-grow: 0;
  border: 1px solid #adb5bd;
  border-radius: 50%;
  margin-right: 0.5em;
  background-repeat: no-repeat;
  background-position: center center;
  background-size: 50% 50%;
}
/* стили при наведении курсора на радио */
.custom-radio>input:not(:disabled):not(:checked)+span:hover::before {
  border-color: #b3d7ff;
}
/* стили для активной радиокнопки (при нажатии на неё) */
.custom-radio>input:not(:disabled):active+span::before {
  background-color: #b3d7ff;
  border-color: #b3d7ff;
}
/* стили для радиокнопки, находящейся в фокусе */
.custom-radio>input:focus+span::before {
  box-shadow: 0 0 0 0.2rem rgba(0, 123, 255, 0.25);
}
/* стили для радиокнопки, находящейся в фокусе и не находящейся в
состоянии checked */
.custom-radio>input:focus:not(:checked)+span::before {
  border-color: #80bdff;
}
/* стили для радиокнопки, находящейся в состоянии checked */
.custom-radio>input:checked+span::before {
  border-color: #0b76ef;
  background-color: #0b76ef;
}
.custom-radio>input:disabled+span::before {
  background-color: #e9ecef;
}
body {
  padding: 20px 50px;
}

```

```

    }
    .radio {
        margin-bottom: 0.2em;
    }
    .topright {
        float:center;
    }
</style>
<body background="{pageContext.request.contextPath}/Config/123.jpg">
<div class = "font">
    <div class = "topright">
        <div class="a"><div class = "input"> <a href="komanda.jsp"
role="button">О нас</a></div></div>
        </div>
    <h1>Калькулятор потребительского кредита</h1>
    <form action="{pageContext.request.contextPath}/JavaCalc"
method="post">
        <div class="form-fields-group">
            <label class=form-fields-group-label for="summ">Введите сумму
кредита от ${a1} до ${a2} рублей</label>
            <input class="form-fields-input" type="text" name="summa" id="summa"
placeholder="Сумма" value="{summa}" required pattern="^[0-9]*$"></div><br>
        </div>
        <div class="form-fields-group">
            <label class=form-fields-group-label for="second">Введите срок выплат
от 12 до 60 месяцев </label>
            <input class="form-fields-input" type="text" name="mesyac" id="mesyac"
placeholder="Месяц" value="{mesyac}" required pattern="^[0-9]*$"></div><br>
        </div>
        <label for="uslovie">Выберите условия кредита</label>
        <div class="new-select-style-wpandyou"><select id="uslovie"
name="uslovie">
            <option value="0">Стандартные
условия</option>
            <option value="1">Держу зарплатный проект в
данном банке</option>
            <option value="2">Являюсь сотрудником
компании партнёров этого банка</option>
        </select></div><br><br>
        <label for="name">Собираетесь ли вы досрочно погашать
кредит?</label><br>

        <div class="radio">

```

```
<label class="custom-radio">
  <input type="radio" name="dosr" value="1">
  <span>Да</span>
</label>
</div>
<div class="radio">
  <label class="custom-radio">
    <input checked type="radio" name="dosr" value="-1">
    <span>Нет</span>
  </label>
</div>
<input type="submit" name="sign" value="Вычислить"></div>
</div>
</form>
</body>
</html>
```

КЛАСС LoginFalse.jsp

```
<% @ page language="java" contentType="text/html; charset=UTF-8"
    pageEncoding="UTF-8"%>
<!DOCTYPE html>
<html>
<head>
<meta charset="UTF-8">
<title>Окно авторизации</title>
</head>
<body>
<style>
input[type=submit] {
    color: #fff; /* цвет текста */
text-decoration: none; /* убирать подчёркивание у ссылок */
user-select: none; /* убирать выделение текста */
background: rgb(0,32,255); /* фон кнопки */
padding: .5em 1.2em; /* отступ от текста */
outline: none; /* убирать контур в Mozilla */
border-radius: 15px;
border: none; /* убираем некрасивую тень */
}
input[type=submit]:hover { background: rgb(0,96,255); }
input[type=submit]:active { background: rgb(0,64,255); }
.form-fields-group {
    position: relative;
}
.form-fields-group-label {
    position: absolute;
    font-size: 13px;
    color: #27252a;
    font-family: 'Roboto', sans-serif;
    top: 10px;
    left: 15px;
}
.form-fields-input {
    width: 300px;
    padding: 30px 15px 10px;
    border: none;
    background: #EAEAEA;
    border-radius: 5px;
    font-family: 'Roboto', sans-serif;
```

```

    font-size: 15px;
}
input:focus ~ label, label:focus {
    color: blue;
}
.font {
font-family: "Times new roman", Times, serif
}
</style>
<div class="a">
<div class ="font">
<h1>Добро пожаловать на сайт калькулятора потребительского
кредита!</h1>
<div>
<h3>Для продолжения введите логин и пароль</h3>
    <div class ="form">
        <form
method="post"
            action="{pageContext.request.contextPath}/JavaLogin"
            <div class="form-fields-group">
                <label class="form-fields-group-label" for="login">Логин</label>
                <input type="text" name="login" id="login" class="form-fields-input"
placeholder="Логин" value="{login}" required><br><br>
            </div>
            <div class="form-fields-group">
                <label class="form-fields-group-label" for="pass">Пароль</label>
                <input type="password" name="pass" id="pass" class="form-fields-input"
placeholder="Пароль" value="{pass}" required><br><br>
                <h4>Неверный логин и/или пароль</h4>
                <input type="submit" name="sign" value="Войти">
            </div>
        </form>
    </div>
</div>
</div>
</body>
</html>

```

КЛАСС Uslovie

```

package team4;

import java.io.IOException;

public abstract class Uslovie {

    public static float proc;
    public static float mproc;
    public static float mesyac;
    public static float pereplata;
    public static float itog;

    public float mess() {
        mesyac = (int)(mesyac*100); // Округление в целое
        mesyac = mesyac/100; // Перевод в десятичное
        return mesyac;
    }

    public float perepl() {
        pereplata = (int)(pereplata*100);
        pereplata = pereplata/100;
        return pereplata;
    }

    public float procc() {
        proc = (int)(proc*100); // Округление процента в целое
        proc /= 100; // Перевод числа в десятичную форму для последующих
        ВЫЧИСЛЕНИЙ
        return proc;
    }

    public void dosr(int dosr_try) {
        if (dosr_try == 1) {
            proc -= 2;
        }
    }

    abstract float calculation(int summ_try, int mesyac_try, int dosr_try)
    throws IOException;
}

```

КЛАСС Check.java

```
package team4;

import java.io.BufferedReader;
import java.io.File;
import java.io.FileInputStream;
import java.io.FileReader;
import java.io.IOException;
import java.io.InputStreamReader;
import java.util.ArrayList;
import java.util.List;

public class Check implements Config {

    private List<String> admlog = new ArrayList<>();
    private List<String> admpass = new ArrayList<>();

    private List<String> usrlog = new ArrayList<>();
    private List<String> usrpass = new ArrayList<>();

    boolean srav;

    public boolean check1(String login, String password) throws IOException {
        srav = false;
        readdan();
        for (int i = 0; i < admlog.size(); i++) {
            if (login.equals(admlog.get(i)) && password.equals(admpass.get(i))) {
                srav = true;
                break;
            }
        }
        return srav;
    }

    public boolean check2(String login, String password) throws IOException {
        srav = false;
        for (int i = 0; i < usrlog.size(); i++) {
            if (login.equals(usrlog.get(i)) && password.equals(usrpass.get(i))) {
                srav = true;
                break;
            }
        }
    }
}
```

```

    return srav;
}

public void readan() throws IOException {
    String filepath = new File("").getCanonicalPath();
    String[] parsfilepath = filepath.split("/");

    int lengthpath = parsfilepath.length;
    String abspath="";
    for(int i=0;i<(lengthpath-1);i++) {
        abspath=abspath+parsfilepath[i]+"/";
    }
    filepath=abspath+"webapps/CalcTeam4/Config/";
    File file1 = new File(filepath);

    FileReader fr1 = new FileReader(file1+"/adminlogpass.txt");
    BufferedReader reader1 = new BufferedReader(fr1);
    String line;
    while((line = reader1.readLine()) != null) {
        int end = line.indexOf(' ');
        admlog.add(line.substring(0, end));
        line = line.substring(end+1);
        admpass.add(line.substring(0));
    }
    reader1.close();

    File file2 = new File(filepath);
    FileReader fr2 = new FileReader(file2+"/userlogpass.txt");
    BufferedReader reader2 = new BufferedReader(fr2);
    while((line = reader2.readLine()) != null) {
        int end = line.indexOf(' ');
        usrlog.add(line.substring(0, end));
        line = line.substring(end+1);
        usrpass.add(line.substring(0));
    }
    reader2.close();

}
}

```


КЛАСС Config.java

```
package team4;

import java.io.IOException;

public interface Config {
    void readdan() throws IOException;
}
```

КЛАСС DliaSotrudnikov.java

```

package team4;

    public class DliaSotrudnikov extends Uslovie {

        @Override
        public float calculation(int summa_try, int mesyac_try, int dosr_try)
        {
            float procMin; // Объявление переменной минимального
процента
            float procMax; // Объявление переменной максимального
процента
            float procRaznica; // Объявление переменной, которая
рассчитывает разницу
            float procMes; // Объявление переменной месячного
процента

            if(summa_try < 250001) { //Условие расчёта процента,
который зависит от суммы
                procMin = (float) 15.99;
                procMax = (float) 23.49;
                procRaznica = procMax - procMin;// Расчёт разницы
            } else if (summa_try < 700001) { //Условие расчёта
процента, которая зависит от суммы
                procMin = (float) 13.99;
                procMax = (float) 18.99;
                procRaznica = procMax - procMin;// Расчёт разницы
            } else {
                procMin = (float) 11.99;
                procMax = (float) 18.99;
                procRaznica = procMax - procMin;// Расчёт разницы
            }

            procMes = procRaznica/48;// Вычисление ежемесячного
процента, для вычисления конечного процента
            proc = (float) (procMax - procMes*(mesyac_try - 12));
            dosr(dosr_try);
            this.mproc = proc/1200; // Вычисление ежемесячного
процента

```

```

        this.mesyac = (float)
(summa_try*(mproc+(mproc/(((Math.pow(1 + mproc, mesyac_try)))-1))))); // Формула
вычисления аннуитетного платежа
        this.pereplata = mesyac * mesyac_try - summa_try; //
Вычисление переплаты
        this.itog = summa_try + pereplata;
        itog = (int)(itog*100);
        itog = itog/100;
        return itog;

    }
}

```

КЛАСС DliaZaiomshikov.java

```

package team4;

public class DliaZaiomshikov extends Uslovie {

    @Override
    public float calculation(int summa_try, int mesyac_try, int dosr_try)
    {
        float procMin;// Объявление переменной минимального
процента
        float procMax;// Объявление переменной максимального
процента
        float procRaznica;// Объявление переменной, которая
рассчитывает разницу
        float procMes; // Объявление переменной месячного
процента

        if(summa_try < 250001) { //Условие расчёта процента,
который зависит от суммы
            procMin = (float) 13.99;
            procMax = (float) 23.49;
            procRaznica = procMax - procMin;// Расчёт разницы
        } else if (summa_try < 700001) { //Условие расчёта
процента, которая зависит от суммы
            procMin = (float) 12.99;
            procMax = (float) 18.99;
            procRaznica = procMax - procMin;// Расчёт разницы
        } else {
            procMin = (float) 11.99;
            procMax = (float) 18.99;
            procRaznica = procMax - procMin;// Расчёт разницы
        }
        procMes = procRaznica/48;// Вычисление ежемесячного
процента, для вычисления конечного процента
        proc = (float) (procMax - procMes*(mesyac_try - 12));
        dosr(dosr_try);
        mproc = proc/1200; // Вычисление ежемесячного процента
        mesyac = (float) (summa_try*(mproc+(mproc/(((Math.pow(1 +
mproc, mesyac_try))))-1))))); // Формула вычисления аннуитетного платежа
        pereplata = mesyac * mesyac_try - summa_try; // Вычисление
переплаты

        itog = summa_try + pereplata; // Общая сумма кредита
    }
}

```

```
        itog = (int)(itog*100);  
        itog = itog/100;  
        return itog;  
    }  
}
```

КЛАСС Standart.java

```

package team4;
public class Standart extends Uslovie {
    @Override
        public float calculation(int summa_try, int mesyac_try, int dosr_try)
{
    float procMin; // Объявление переменной минимального процента
    float procMax; // Объявление переменной максимального процента
    float procRaznica; // Объявление переменной рассчитывающей
разницу
    float procMes; // Объявление переменной месячного процента

    if(summa_try < 250001) { //Условие расчёта процента, который
зависит от суммы
        procMin = (float) 16.99;
        procMax = (float) 23.49;
        procRaznica = procMax - procMin; // Расчёт разницы
    } else if (summa_try < 700001) { //Условие расчёта процента, которая
зависит от суммы
        procMin = (float) 14.99;
        procMax = (float) 18.99;
        procRaznica = procMax - procMin; // Расчёт разницы
    } else {
        procMin = (float) 11.99;
        procMax = (float) 18.99;
        procRaznica = procMax - procMin; // Расчёт разницы
    }
    procMes = procRaznica/48; // Вычисление ежемесячного процента, для
вычисления конечного процента
    proc = (float) (procMax - procMes*(mesyac_try - 12));
    dosr(dosr_try);
        mproc = proc/1200; // Вычисление ежемесячного процента
        mesyac = (float) (summa_try*(mproc+(mproc/(((Math.pow(1 +
mproc, mesyac_try)))-1)))); // Формула вычисления аннуитетного платежа
        pereplata = mesyac * mesyac_try - summa_try; // Вычисление
переплаты

        itog = summa_try + pereplata; // Общая сумма кредита
        itog = (int)(itog*100);
        itog = itog/100;
        return itog;
    }
}

```

КЛАСС Main.java

```
package team4;

import java.io.BufferedReader;
import java.io.File;
import java.io.FileInputStream;
import java.io.IOException;
import java.io.InputStreamReader;
import java.io.PrintWriter;

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 team4.DliaSotrudnikov;
import team4.DliaZaiomshikov;
import team4.Standart;

@WebServlet(name="Calc", urlPatterns="/JavaCalc")

public class Main extends HttpServlet{

    public static int a1;
    public static int a2;

    protected void doPost(HttpServletRequest request, HttpServletResponse
response) throws ServletException, IOException {
        RequestCalc Calc = RequestCalc.fromRequestParameters(request);
        Calc.setAsRequestAttributesAndCalculate(request);
        request.getRequestDispatcher("/Results.jsp").forward(request,
response);

        PDF pdf1 = new PDF();
        String goals = "";
        pdf1.create(goals);
    }

    private static class RequestCalc {
```

```

        private final String summa;
        private final String mesyac;
        private final String uslovie;
        private final String dosr;

        private float proc;
        private float mes;
        private float pereplata;
        private float itog;

        private RequestCalc (String summa, String mesyac, String uslovie,
String dosr) {
            this.summa = summa;
            this.mesyac = mesyac;
            this.uslovie = uslovie;
            this.dosr = dosr;
        }

        public static RequestCalc
fromRequestParameters(HttpServletRequest request) {
            return new RequestCalc(
                request.getParameter("summa"),
                request.getParameter("mesyac"),
                request.getParameter("uslovie"),
                request.getParameter("dosr"));
        }

        public void
setAsRequestAttributesAndCalculate(HttpServletRequest request) throws IOException
        {
            int summa_try;
            int mesyac_try;
            int uslovie_try;
            int dosr_try;

            try {
                String filepath = new File("").getCanonicalPath();
                String[] parsfilepath = filepath.split("/");

                int lengthpath = parsfilepath.length;
                String abspath="";

```



```

        for(int i=0;i<(lengthpath-1);i++) {
            abspath=abspath+parsfilepath[i]+"/";
        }

        filepath=abspath+"webapps/CalcTeam4/Config/koef.txt";
        BufferedReader reader= new BufferedReader(new
InputStreamReader(new FileInputStream(filepath)));
        a1 = Integer.parseInt(reader.readLine());
        a2 = Integer.parseInt(reader.readLine());
        reader.close();
    }
    catch (Exception ex) {
        ex.printStackTrace();
    }

    try {
        summa_try = Integer.parseInt(summa);
        mesyac_try = Integer.parseInt(mesyac);
        uslovie_try = Integer.parseInt(uslovie);
        dosr_try = Integer.parseInt(dosr);
    }
    catch (NumberFormatException e) {
        summa_try = 0;
        mesyac_try = 0;
        uslovie_try = 0;
        dosr_try = 0;
    }
    if (summa_try < a1 || summa_try > a2 || mesyac_try < 12||
mesyac_try > 60) {

        summa_try = 0;
        mesyac_try = 0;
        uslovie_try = 0;
        dosr_try = 0;
    }
    if (uslovie_try == 0) {
        Standart uslovie = new Standart();

        itog  =  uslovie.calculation(summa_try,  mesyac_try,
dosr_try);

        proc = uslovie.procc();
        mes = uslovie.mess();
        pereplata = uslovie.perepl();
    }
}

```

```

        request.setAttribute("itog", itog);
        request.setAttribute("proc", proc);
        request.setAttribute("mes", mes);
        request.setAttribute("pereplata", pereplata);
    }
    else if (uslovie_try == 1) {
        DliaZaiomshikov uslovie = new DliaZaiomshikov();
        itog = uslovie.calculation(summa_try, mesyac_try,
dosr_try);

        proc = uslovie.procc();
        mes = uslovie.mess();
        pereplata = uslovie.perepl();
        request.setAttribute("itog", itog);
        request.setAttribute("proc", proc);
        request.setAttribute("mes", mes);
        request.setAttribute("pereplata", pereplata);
    }
    else if (uslovie_try == 2) {
        DliaSotrudnikov uslovie = new DliaSotrudnikov();
        itog = uslovie.calculation(summa_try, mesyac_try,
dosr_try);

        proc = uslovie.procc();
        mes = uslovie.mess();
        pereplata = uslovie.perepl();
        request.setAttribute("itog", itog);
        request.setAttribute("proc", proc);
        request.setAttribute("mes", mes);
        request.setAttribute("pereplata", pereplata);
    }
    }
}
}
}

```

КЛАСС Login.java

```

package team4;
import java.io.BufferedReader;
import java.io.File;
import java.io.FileInputStream;
import java.io.IOException;
import java.io.InputStreamReader;

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="Login", urlPatterns="/JavaLogin")
public class Login extends HttpServlet {
    static boolean check1;
    static boolean check2;
    protected void doPost(HttpServletRequest request, HttpServletResponse
response) throws ServletException, IOException {
        RequestLogin auth = RequestLogin.fromRequestParameters(request);
        auth.Check(request);
        if (check1) {
            request.getRequestDispatcher("/Admin.jsp").forward(request, response);
        } else if (check2) {
            try {
                String filepath = new File("").getCanonicalPath();
                String[] parsfilepath = filepath.split("/");

                int lengthpath = parsfilepath.length;
                String abspath="";
                for(int i=0;i<(lengthpath-1);i++) {
                    abspath=abspath+parsfilepath[i]+"/";
                }

                filepath=abspath+"webapps/CalcTeam4/Config/koef.txt";
                BufferedReader reader= new BufferedReader(new
InputStreamReader(new FileInputStream(filepath)));
                Main.a1 = Integer.parseInt(reader.readLine());
                Main.a2 = Integer.parseInt(reader.readLine());
                reader.close();
            }

```

```

        catch (Exception ex) {
            ex.printStackTrace();
        }
        request.setAttribute("a1", Main.a1);
        request.setAttribute("a2", Main.a2);
        request.getRequestDispatcher("/User.jsp").forward(request, response);
    } else {
        request.getRequestDispatcher("/LoginFalse.jsp").forward(request,
response);
    }
}

private static class RequestLogin {
    private final String login;
    private final String pass;

    private RequestLogin (String login, String pass) {
        this.login = login;
        this.pass = pass;
    }

    public static RequestLogin fromRequestParameters(HttpServletRequest
request) {
        return new RequestLogin(
            request.getParameter("login"),
            request.getParameter("pass")
        );
    }

    public void Check(HttpServletRequest request) throws IOException {
        String login_try;
        String pass_try;
        login_try = login;
        pass_try = pass;
        Check check = new Check();
        check1 = check.check1(login_try, pass_try);
        check2 = check.check2(login_try, pass_try);
    }
}

```

Класс Admin.java

```

package team4;

import java.io.File;
import java.io.FileWriter;
import java.io.IOException;
import java.io.PrintWriter;

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

/**
 * Servlet implementation class Admin
 */
@WebServlet("/Admin")
public class Admin extends HttpServlet {
    private static final long serialVersionUID = 1L;

    /**
     * @see HttpServlet#doGet(HttpServletRequest request,
     HttpServletResponse response)
     */
    protected void doGet(HttpServletRequest request, HttpServletResponse
    response) throws ServletException, IOException {
        // TODO Auto-generated method stub
        int a1 = Integer.parseInt(request.getParameter("a1"));
        int a2 = Integer.parseInt(request.getParameter("a2"));

        try {
            String filepath = new File("").getCanonicalPath();
            String[] parsfilepath = filepath.split("/");

            int lengthpath = parsfilepath.length;
            String abspath="";
            for(int i=0;i<(lengthpath-1);i++) {
                abspath=abspath+parsfilepath[i]+"/";
            }
            filepath=abspath+"webapps/CalcTeam4/Config/koef.txt";

```

```

        FileWriter fileWriter = new FileWriter(filepath);
        PrintWriter printWriter = new PrintWriter(fileWriter);
        printWriter.println(a1);
        printWriter.print(a2);
        response.sendRedirect("Admin.jsp");
        printWriter.close();
    }
    catch(Exception ex) {
        ex.printStackTrace();
    }

}

/**
 *      @see      HttpServlet#doPost(HttpServletRequest request,
HttpServletResponse response)
 */
protected void doPost(HttpServletRequest request, HttpServletResponse
response) throws ServletException, IOException {
    // TODO Auto-generated method stub
    doGet(request, response);
}

}

```

КЛАСС Admin.jsp

```
<% @ page language="java" contentType="text/html; charset=UTF-8"
    pageEncoding="UTF-8"%>
<!DOCTYPE html>
<html>
<head>
<meta charset="UTF-8">
<title>Меню администратора</title>
</head>
<body>
<style>
    .box {
        border:0;
        padding:5px;
        border-radius:15px;
    }
    input[type=submit] {
        color: #fff; /* цвет текста */
        text-decoration: none; /* убирать подчёркивание у ссылок */
        user-select: none; /* убирать выделение текста */
        background: rgb(0,32,255); /* фон кнопки */
        padding: .5em 1.2em; /* отступ от текста */
        outline: none; /* убирать контур в Mozilla */
        border-radius:15px;
        border: none; /* убираем некрасивую тень */
    }
    input[type=submit]:hover { background: rgb(0,96,255)); }
    input[type=submit]:active { background: rgb(0,64,0); }
    .form-fields-group {
        position: relative;
    }
    .form-fields-group-label {
        position: absolute;
        font-size: 13px;
        color: #27252a;
        font-family: 'Roboto', sans-serif;
        top: 10px;
        left: 15px;
    }
    .form-fields-input {
        width: 300px;
        padding: 30px 15px 10px;
```

```

border: none;
background: #EAEAEA;
border-radius: 5px;
font-family: 'Roboto', sans-serif;
font-size: 15px;
}
input:focus ~ label, label:focus {
color: blue;
}

.new-select-style-wpandyou select {
border-radius: 0;
background: #EAEAEA;
height: 34px;
padding: 5px;
border: none;
outline: none;
font-size: 16px;
line-height: 1;
-webkit-appearance: none;
width: 400px;
}

.select {
width: 100%;
padding: 5px;
border-radius: 10px;
}

.font {
font-family: "Times New Roman", Times, serif }
.a {
text-align :center;
}
</style>
<body bgcolor="FFFFFF">
<div class = "font">
<h1>Меню администратора</h1>
<form action="{pageContext.request.contextPath}/Admin" method="post">
<div class="form-fields-group">
<input type="text" id="name" class="form-fields-input"
placeholder="Введите сумму" name="a1" required>
<label class="form-fields-group-label" for="name">Введите новую
минимальную сумму</label>
</div><br>
<div class="form-fields-group">

```



```

        <input      type="text"      id="name"      class="form-fields-input"
placeholder="Введите сумму" name="a2" required>
        <label    class="form-fields-group-label"    for="name">Введите    новую
максимальную сумму</label>
    </div><br>
    <div    class="a"><input    type="submit"    name="sign"    value="Сохранить
изменения"></div>
    <div class ="input"><a href="Login.jsp" role="button">Выйти</a></div>

    </div>
    </form>
    </body>
</html>

```

КЛАСС komanda.jsp

```

<% @ page language="java" contentType="text/html; charset=UTF-8"
pageEncoding="UTF-8"%>
<!DOCTYPE html>
<html>
<head>
<meta charset="UTF-8">
<title>Информация о разработчиках</title>
</head>
<body>
<style>
.text {
margin: auto;
text-align: center;
vertical-align:center;
font-family: "Times New Roman", Times,serif
}
input[type=submit] {
color: #fff; /* цвет текста */
text-decoration: none; /* убирать подчёркивание у ссылок */
user-select: none; /* убирать выделение текста */
background: rgb(0,32,255); /* фон кнопки */
padding: .5em 1.2em; /* отступ от текста */
outline: none; /* убирать контур в Mozilla */
border-radius:15px;
border: none; /* убираем некрасивую тень */
}
input[type=submit]:hover { background: rgb(0,96,255)); }
input[type=submit]:active { background: rgb(0,64,255); }
.a {
text-align :center;
}
</style>
<div class = text>
<h1>Разработчики команды №4</h1>
<p>Модератор: Самсонов Артём Андреевич</p>
<p>Разработчик №1: Фахретдинова Кристина Олеговна</p>
<p>Разработчик №2: Шейфель Жанна Владимировна</p>
<p>Разработчик №3: Степанян Степан Арменович</p>
<div class="a"><div class = "input"> <a href="User.jsp"
role="button">Вернуться к калькулятору</a></div></div></div>

```

```
</body>  
</html>
```

КЛАСС LoginTest.java

```
package pi221team4k;

import static org.junit.Assert.*;
import org.junit.Before;
import org.junit.Test;

public class LoginTest {

    String login;
    boolean prov1;

    @Before
    public void prisvoenieTest() {

        login = "user"; // òãñò 1
        prov1 = false;
    }

    @Test
    public void logTest() {

        if (login == "user") {
            prov1=true;
        }

        assertTrue(prov1);

    }
}
```

КЛАСС UslovieTest.java

```
package pi221team4k;

import static org.junit.Assert.*;
import org.junit.Before;
import org.junit.Test;

public class UslovieTest {

    protected double proc;
    protected int dosr_try;

    @Before
    public void procTest() {

        proc = 23.49; // тест №1
        dosr_try = 1;
    }

    @Test
    public void provercaprocTest() {

        if (dosr_try == 1) {
            proc -= 2;
        }
        assertEquals(21, (int)proc);
        System.out.println(proc);
    }
}
```

КЛАСС ProcTest.java

```
package pi221team4k;

import static org.junit.Assert.*;

import org.junit.Before;

import org.junit.Test;

    public class ProcTest {

        private int summ;
        private int mes;

    @Before

        public void prisvoenieTest() {

            summ = 2500000; // њїїї 1
            mes = 36;
        }

    @Test

        public void porcentTest() {

            double procMin;
            double procMax;
            double procRaznica;
            double procMes;
            double proc;

            if(summ < 250001) {
                procMin = 16.99;
                procMax = 23.49;
                procRaznica = procMax - procMin;

            } else if (summ < 700001) {
                procMin = 14.99;
                procMax = 18.99;
                procRaznica = procMax - procMin;
```

```
    } else {  
        procMin = 11.99;  
        procMax = 18.99;  
        procRaznica = procMax - procMin;  
  
    }  
  
    procMes = procRaznica/48;  
    proc = procMax - procMes*(mes - 12);  
    assertEquals(15, (int)proc);  
    System.out.println(proc);  
  
    }  
  
}
```

КЛАСС OgranTest.java

```
package pi221team4k;

import static org.junit.Assert.*;
import org.junit.Before;
import org.junit.Test;

public class OgranTest {

    protected int summ;
    protected int mes;

    @Before

    public void prisvoenieTest() {

        summ = 1234567; // òåñò 1
        mes = 60;
    }

    @Test

    public void raschetTest() {

        assertFalse(summ < 50000 || summ >= 3000000 || mes < 12 || mes > 60 );
    }

}
```


Список используемой литературы

1. Васильев, А.Н. Java. Объектно-ориентированное программирование: учеб. Пособие / А.Н. Васильев, Ю.В. Сергиенко – СПб.: Питер, 2013. – 198 с.
2. Гуськова, О. И. Объектно-ориентированное программирование в Java: учебное пособие / О. И. Гуськова. — Москва : МПГУ, 2018. — 240 с. — URL: <https://e.lanbook.com/book/122311> (дата обращения: 21.05.2021)
4. Даниленко, С.А. Банковское потребительское кредитование: учеб. Пособие / С.А. Даниленко, М.В. Комиссарова. – Москва : Юстицинформ, 2011. – 37 с. URL: <https://e.lanbook.com/book/10546> (дата обращения: 21.05.2021)
5. Коузен, К. Современный Java: рецепты программирования [Электронный ресурс] / Коузен К. — Москва: ДМК Пресс, 2018. — 275 с. URL: <https://e.lanbook.com/book/116121> (дата обращения: 21.05.2021)
6. Кулаков, К.А. Основы тестирования программного обеспечения [Электронный ресурс]: учебное электронное пособие для обучающихся Института математики и информационных технологий / К. А. Кулаков, В. М. Димитров; М-во образования и науки Рос. Федерации, Федер. гос. бюджет. образоват. учреждение высш. образования Петрозав. гос. ун-т. - Электрон. текст. дан. – Петрозаводск: Изд-во ПетрГУ, 2018.
7. Лаврушин, О.И. Современная система кредитования: учеб.пособие / О.И. Лаврушин, О.Н. Афанасьева, С.Л. Корниенко. – Москва: КНОРУС, 2007. – 75 с.
8. Совместная разработка программного обеспечения: метод. Рекомендации / сост. А.В. Казанцев — Уфа: РИК УГАТУ, 2020. — 24 с.
9. Лекционный материал по дисциплине «Объектно-ориентированное моделирование и программирование» преподавателя кафедры АСУ Казанцева А.В.
10. Лекционный материал по дисциплине «Информационные системы» преподавателя кафедры АСУ Казанцева А.В.
11. Федеральный закон "О потребительском кредите (займе)" от 21.12.2013 N 353-ФЗ (последняя редакция)

URL: http://www.consultant.ru/document/cons_doc_LAW_155986/

12. ГОСТ 19.106–78 ЕСПД. Требования к программным документам, выполненным печатным способом (с Изменением N 1), 1980–14 с.
13. Javarush - онлайн-курс обучения программированию на Java. [Киев], 2012–2021. URL: <https://javarush.ru/> (дата обращения: 18.05.2021)
14. Хабр. [Никосия], 2003–2021. URL: <https://habr.com/ru/> (дата обращения: 20.05.2021)
15. Артемов, М. А. Введение в Linux : учебно-методическое пособие / составители М. А. Артемов [и др.]. — Воронеж : ВГУ, 2016. — 44 с. URL: <https://e.lanbook.com/book/165430> (дата обращения: 21.05.2021)
16. Иванько, А. Ф. Системное программное обеспечение информационных мультимедиа-систем : учебное пособие / А. Ф. Иванько, М. А. Иванько. — Санкт-Петербург: Лань, 2020. — 80с. — URL: <https://e.lanbook.com/book/139325> (дата обращения: 21.05.2021)