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»

Выполнил:
Ст. гр. ПИ-222
Салихов Р.Н.
Бесполитов Д.А.
Иутин А.С.

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	*Салихов Р.Н.* *Бесполитов Д.А* *Иутин А.С.*	Группа	ПИ-222	Консультант	Казанцев А.В.
	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 выносится 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. Настойка 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	

АННОТАЦИЯ

Курсовой проект на тему «Разработка кроссплатформенного программного продукта на языке JAVA с использованием системы контроля версий» по дисциплине «Информационные системы».

Выполнили – Салихов Р.Н.; Бесполитов Д.А.; Иутин А.С.

В данном курсовом проекте разработано web-приложение, которое выполняет расчет стоимости строительства частного дома.

Репозиторий с программным кодом и документацией:
(Salikhov12/calculator: Курсовая: [сайт]. URL: https://github.com/Salikhov12/calculator)

СОДЕРЖАНИЕ

Раздел 1. Описание предметной области 4
1.1 Предметная область 4
1.2 Математическая модель работы ПО 9
Раздел 2. Техническое задание на создание программного продукта 11
Раздел 3. Настройка среды разработки для операционных систем семейств Windows и Linux 12
3.1 Настройка среды разработки для операционных систем на Linux Ubuntu 20.04. 12
3.3 Настройка среды разработки для операционных систем на Windows 10 19
Раздел 4. Настройка среды разработки для подключения к системе контроля версий 24
Раздел 5. Реализация исходного кода по зонам ответственности 29
Раздел 6. Сборка и тестирование программного продукта 30
6.1 Описание UNIT-тестов 30
6.2 Инструкция по сборке проекта 31
Раздел 7. Настройка программной среды для развертывания и запуска программного продукта 40
7.1 Настройка непрерывной интеграции для платформы Heroku 40
7.2 Настройка непрерывной интеграции для веб-сервера 41
7.3 Описание каталогов и файлов работающей программы 43
7.4 Условия работы программы 44
Раздел 8. Руководство пользователя программного продукта 46
Заключение 47
ПРИЛОЖЕНИЕ 1 48
ПРИЛОЖЕНИЕ 2 49
ПРИЛОЖЕНИЕ 3 50
ПРИЛОЖЕНИЕ 4 55
ПРИЛОЖЕНИЕ 5 56
ПРИЛОЖЕНИЕ 6 58
ПРИЛОЖЕНИЕ 7 60
ПРИЛОЖЕНИЕ 8 67
ПРИЛОЖЕНИЕ 9 77
ПРИЛОЖЕНИЕ 10 81
ПРИЛОЖЕНИЕ 11 82
ПРИЛОЖЕНИЕ 12 83
ПРИЛОЖЕНИЕ 13 84
ПРИЛОЖЕНИЕ 14 85
ПРИЛОЖЕНИЕ 15 90
ПРИЛОЖЕНИЕ 16 100
ПРИЛОЖЕНИЕ 17 101
ПРИЛОЖЕНИЕ 18 102

ПРИЛОЖЕНИЕ 19 103
ПРИЛОЖЕНИЕ 20 104
ПРИЛОЖЕНИЕ 21 105
Список использованной литературы 106

Раздел 1. Описание предметной области

1.1 Предметная область

Предметная область — строительство частного дома. Частные дома - индивидуальный городской или сельский малоэтажный (обычно двухэтажный) жилой дом с небольшим участком прилегающей земли для постоянного или временного проживания.

Строительство — создание (возведение) зданий, строений и сооружений.

В широком смысле к строительству (как отрасли) также относят капитальный и текущий ремонт зданий и сооружений, а ещё их реконструкцию, реставрацию и реновацию.

Функции строительства:

1. Создание новых сооружений.

2. Расширение - строительство новых сооружений и увеличение производственных площадей, объектов на территории действующих предприятий.

3. Реконструкция – переустройство существующих объектов основного назначения без расширения этих объектов.

В строительстве подразумевается множество деталей. Туда включается закупка материала, правильный выбор территории и ее подготовка, а также многое другое.

- Первым делом выполняется планирование постройки дома. Выбор места, размера материалов и так далее.

- Далее идет закупка материалов, доставка до места стройки, поиск рабочих и подготовка территории.

- После этого идет постройка дома, а именно заливка фундамента, постройка основы и каркаса крыши.

- Когда основа дома готова то начинают заниматься отделкой. Она бывает «черновая» - установка дверей, окон и оштукатуривание. Так же может быть «под ключ» - уже полностью предназначенная для жилья квартира

- Ну и финальный этап — уборка после строительства.

Нормативные документы:

1)ФЗ №7 от 10.01.02г. "Об охране окружающей среды";

2)ФЗ №184 от 27.12.02г. "О техническом регулировании";

3)ФЗ №315 от 01.12.07г. "О саморегулируемых организациях";

4)СП 53.13330.2019 от 29.12.2004 «Планировка и застройка территории ведения гражданами садоводства»;

5)СанПиН 2.1.6.983-00 от 13.07.2000 «Атмосферный воздух и воздух закрытых помещений, санитарная охрана воздуха. Гигиенические требования к обеспечению качества атмосферного воздуха населенных мест»;

6)СанПиН 2.1.4.027-95 от 10.04.1995 г. «Зоны санитарной охраны источников водоснабжения и водопроводов хозяйственно-питьевого назначения»;

7)СанПиН 2.1.2.2645-10 Санитарно-эпидемиологические требования к условиям проживания в жилых зданиях и помещениях.

Наше программное обеспечение будет писаться для строительной компании, которая занимается исключительно строительством частных домов. Использоваться же данное программное обеспечение будет клиентами этой компании.

Также в нашей программе будет присутствовать возможность компанией изменить созданные шаблоны через специальный аккаунт администратора приложения, в который можно попасть через форму входа.

Сама компания выполняет различные виды работ при строительстве частных домов: кирпичная кладка, устройство фундамента, отделочные работы.

Также эта компания предлагает услуги по строительству жилых сооружений на территории Республики Башкортостан. Компания предлагает такие услуги:

-разработка проектов;

-строительство домов с разными видами отделки;

-готовые шаблоны домов (зимний и летний варианты);

-очистка участка и другие услуги.

Мнемосхема рассматриваемого бизнес-процесса представлена на рисунке 1. Условные обозначения – в таблице 1.

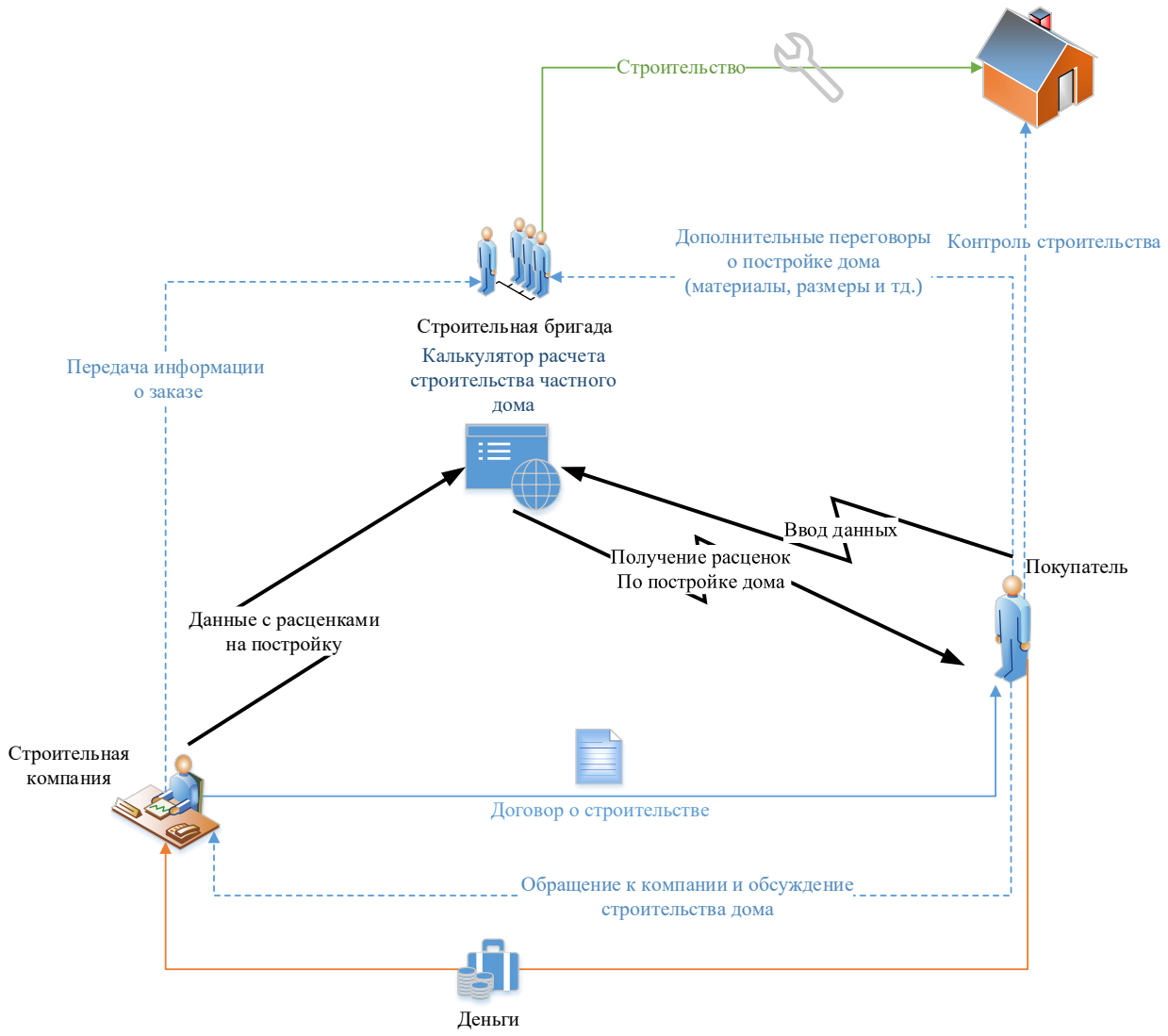


Рис. 1. Мнемосхема бизнес-процесса.

Таблица 1. Условные обозначения.

Условное обозначение	Вид
	Бумажный
	Устный
	Финансовый
	Материальный

Диаграмма вариантов использования, создаваемого нами приложения представлена на рисунке 2, а диаграмма классов данного приложения на рисунке 3.

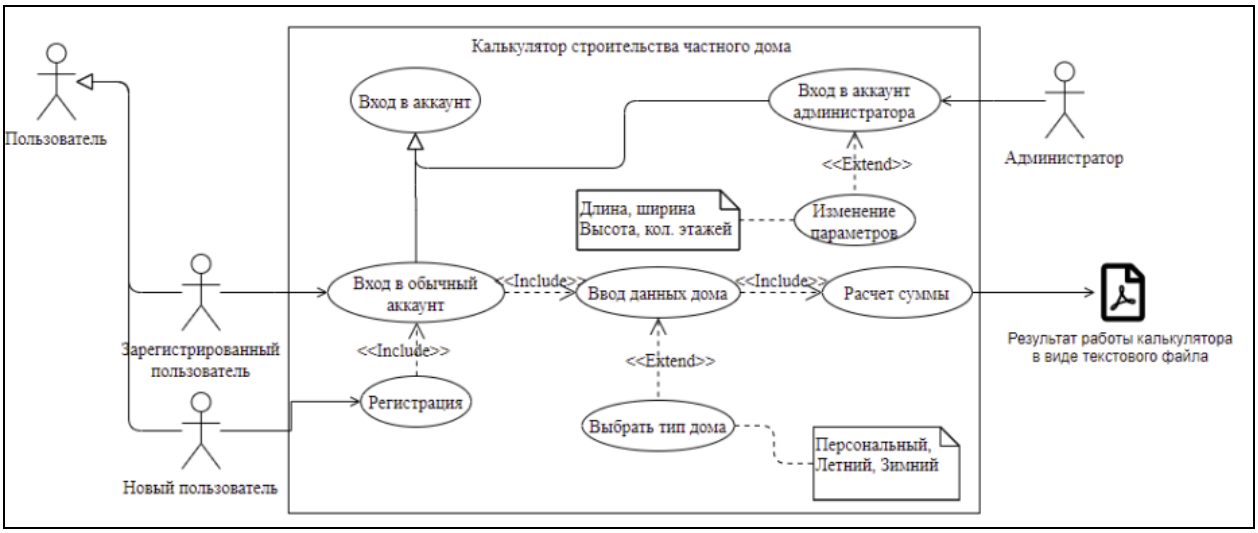


Рис. 2. Диаграмма вариантов использования.

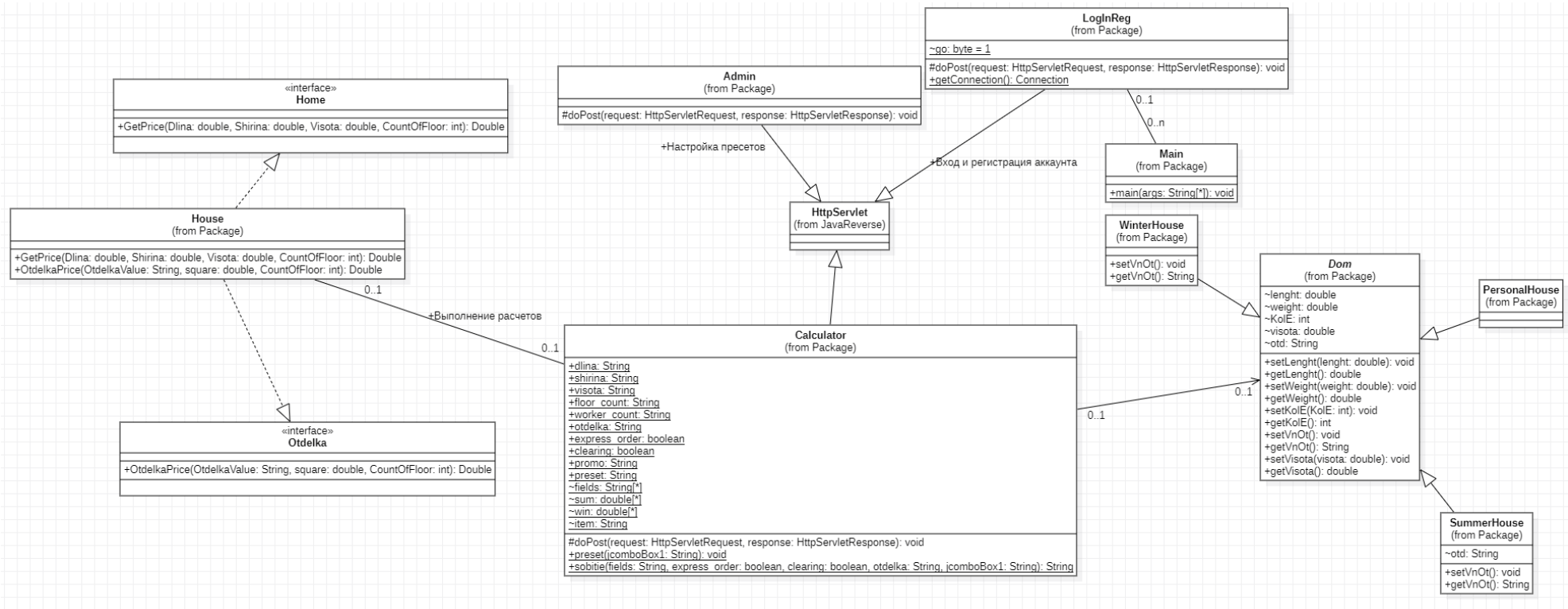


Рис. 3. Диаграмма классов.

1.2 Математическая модель работы ПО

Площадь дома рассчитывается по формуле 1.

$$S = dlina \ x \ shirina \ xCOF \ , (1)$$

где dlina — Длина, м;

shirina — Ширина, м;

COF –Количество этажей, шт;

S – Площадь дома, м^2.

Расчет цены за квадратный метр представлен в формуле 2.

$$Price = S \ x \ POOM \ , (2)$$

где POOM – Цена за квадратный метр, руб;

S – Площадь дома, м^2;

Price – Цена за площадь дома, руб.

Расчет цены за отделку рассчитывается по формуле 3.

$$Value = S \ x \ Const \ , (3)$$

где Value – Стоимость отделки, руб;

S – Площадь дома, м^2;

Const – Цена за отделку, руб.

Расчет итоговой стоимости рассчитывается по формуле 4.

$$Result = ((Value + Price) + Srochno + Rabotnik + Chistka) \ x \ Promo \ , (4)$$

где Value – Стоимость отделки, руб;

Price – Цена за площадь дома, руб;

Srochno – Добавочная стоимость за срочный заказ, руб;

Rabotnik – Добавочная стоимость за дополнительных работников, руб;

Promo – Промокод для уменьшение итоговой стоимости, константа;

Chistka – Добавочная стоимость за чистку участка, руб;

Result — итоговая стоимость работы, руб.

Пример №1.

Длина — 5, ширина – 4, высота – 6, этажей – 2, отделка – «Под ключ» (10000 руб), дополнительных работников – 0, срочный заказ – нет, расчистка участка – да (5000 руб), промокод – отсутствует (1).

Цена за 1 м^2 — 50000 руб.

$$40=4\times5\times2$$

$$2000000=40\times50000$$

$$400000=40\times10000$$

$$2405000=((400000+2000000)+0+0+5000)\times1$$

Пример №2

Длинна — 10, ширина – 10, высота – 6, этажей – 2, дополнительных работников – 1(5000 руб), отделка – «Черновая отделка» (5000 руб), срочный заказ – нет, расчистка участка – нет, промокод – 15% (0,85).

Цена за 1 м^2 — 50000 руб.

$$200=10\times10\times2$$

$$10000000=200\times50000$$

$$1000000=200\times5000$$

$$9354250=((1000000+10000000)+0+5000+0)\times0{,}85$$

Раздел 2. Техническое задание на создание программного продукта

Смотрите Приложение 1.

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

3.1 Настройка среды разработки для операционных систем на Linux Ubuntu 20.04.

1. Для работы необходим jdk. Для его установки откроем терминал и введем команду (Рис.4):

sudo apt-get install default-jdk

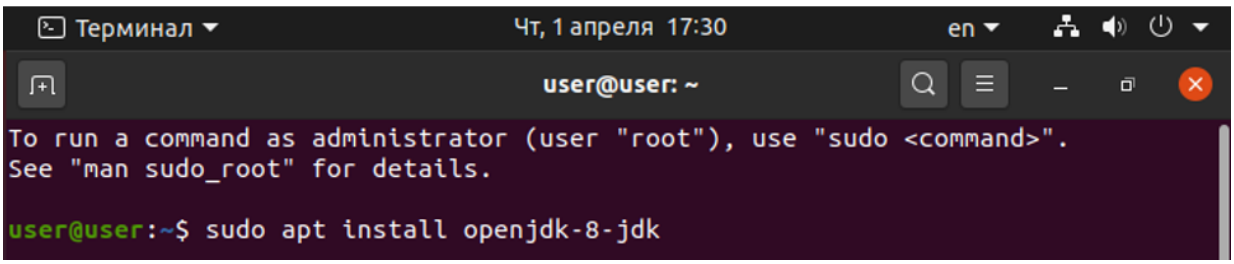


Рис. 4. Установка jdk.

2. Для проверки правильности установки jdk введем следующую команду (Рис.5):

java -version

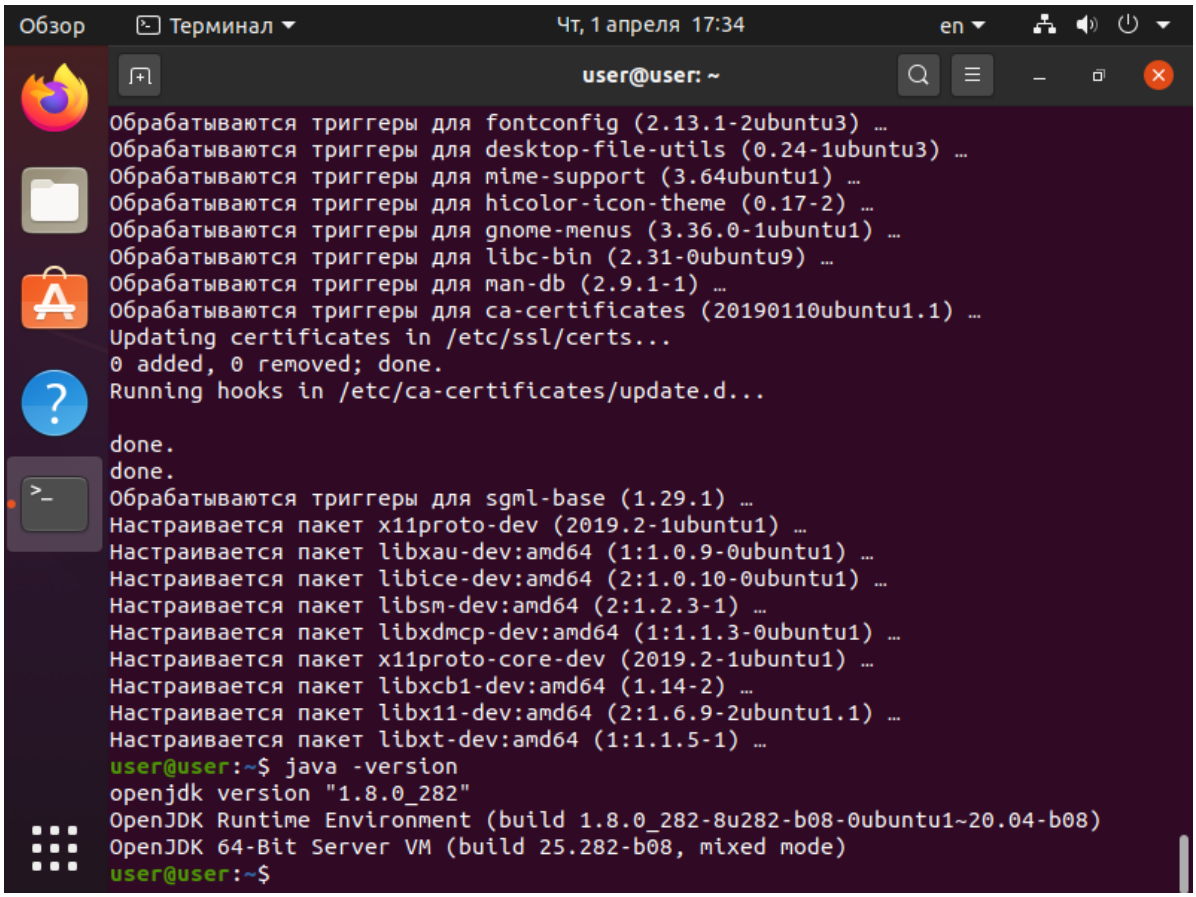


Рис. 5. Проверка правильности установки jdk.

3. Для установки Eclipse IDE for Enterprise Java and Web Developers воспользуемся браузером для загрузки дистрибутива Eclipse с официальной сайта https://www.eclipse.org/downloads/packages/ (Рис.6). Выбираем *Linux x86_64*. Далее следует распаковать дистрибутив в домашнюю папку пользователя и выполнить файл eclipse из каталога eclipse. На рисунке 7 показан загруженный дистрибутив, который находится в архиве, а на рисунке 8 результат распаковки архива.

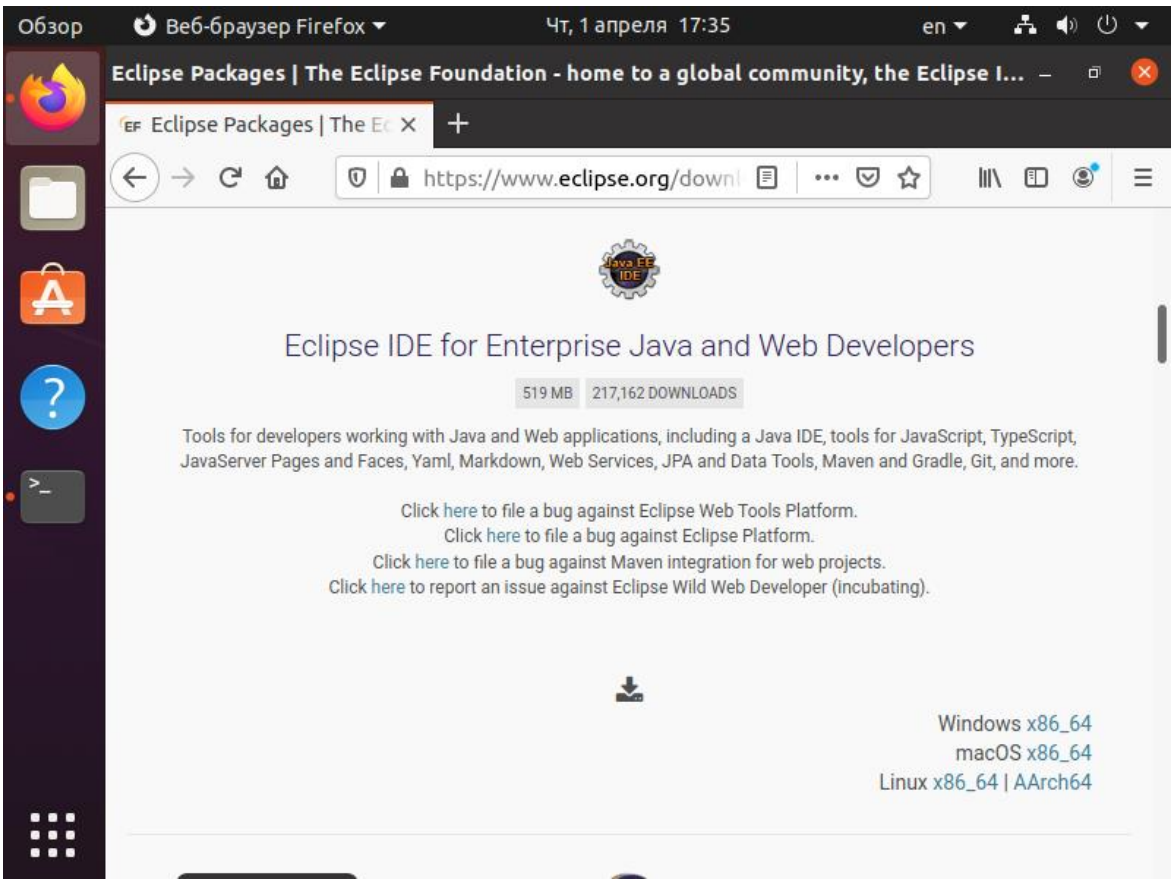


Рис. 6. Сайт Eclipse.

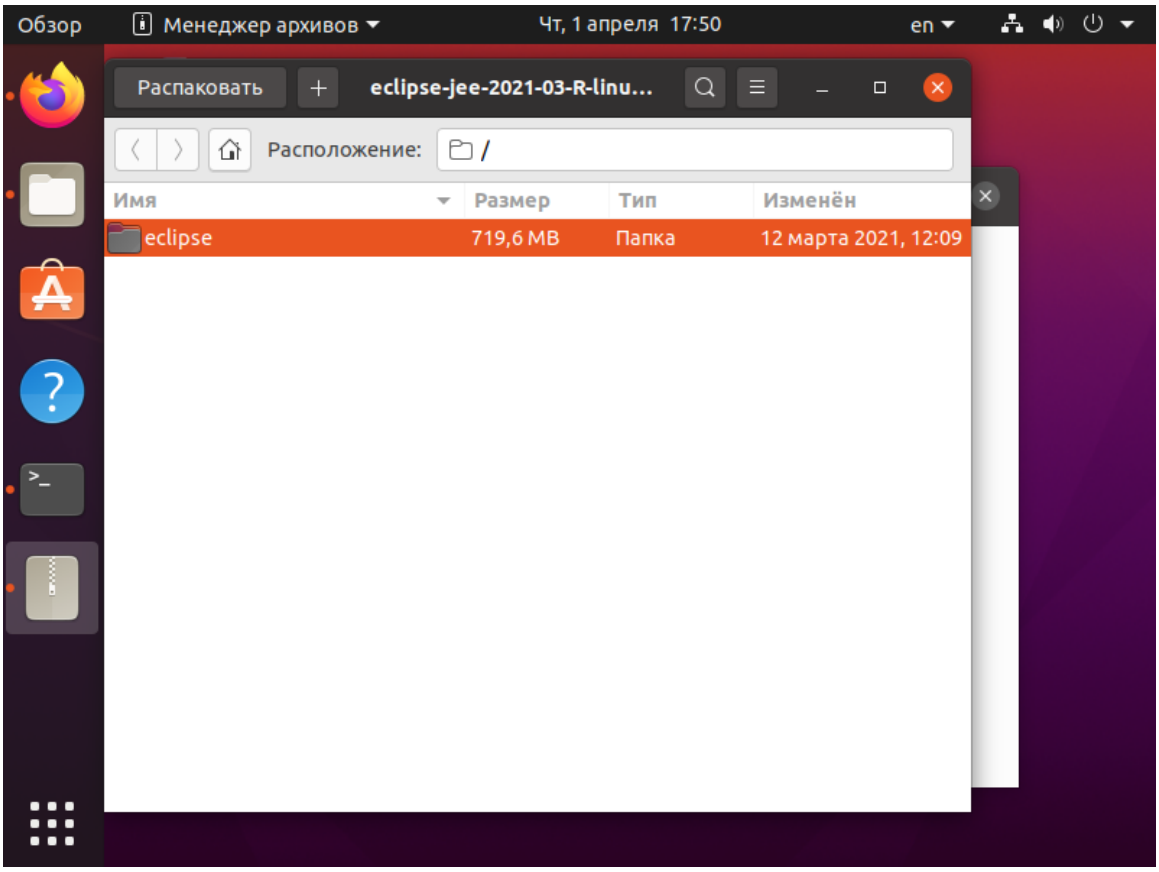


Рис. 7. Загруженный дистрибутив.

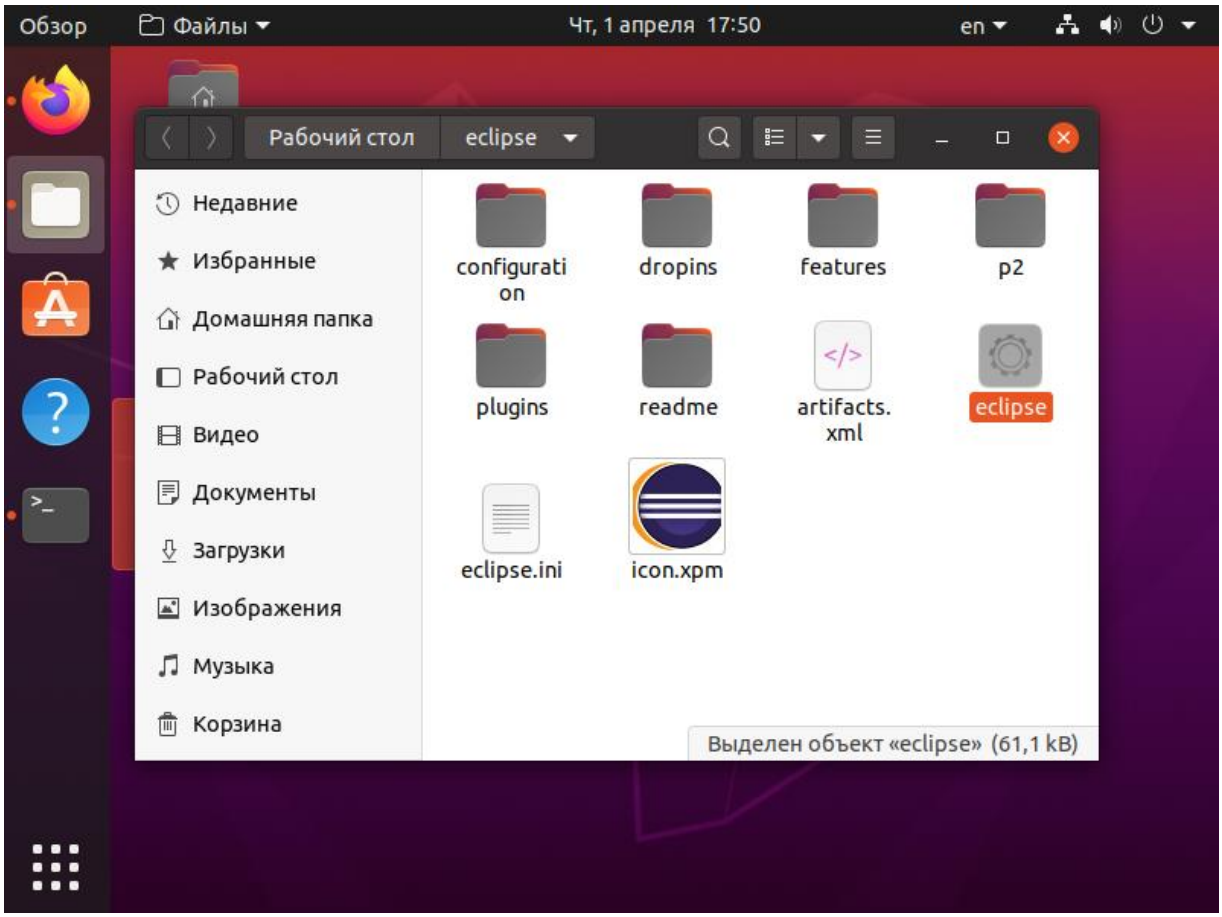


Рис. 8. Результат распаковки архива.

4. Установим и настроим операционную систему для работы с Git. Для установки пропишем в терминале следующую команду (Рис.9):

sudo apt-get install git

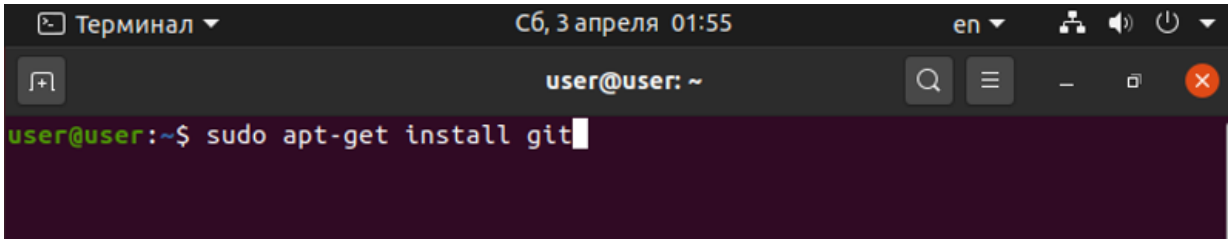


Рис. 9. Установка Git.

5. Настроем Git, для этого введем следующие две команды, для настройки имени пользователя и адреса электронной почты (Рис.10):

git config --global user.name "My Name"

git config --global user.email myEmail@example.com

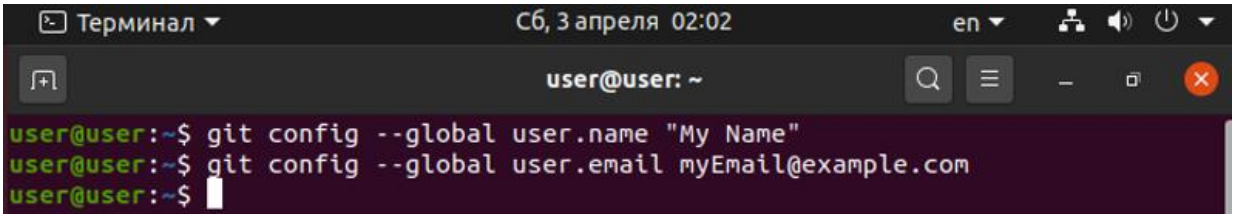


Рис. 10. Настройка Git.

6. Необходимые Git и Maven в Eclipse IDE for Enterprise Java and Web Developers предустановлены, но для работы с Git требуется вывести на экран его перспективу, для этого нажимаем Window -> Perspective -> Open Perspective -> Other и выбираем Git (Рис. 11).

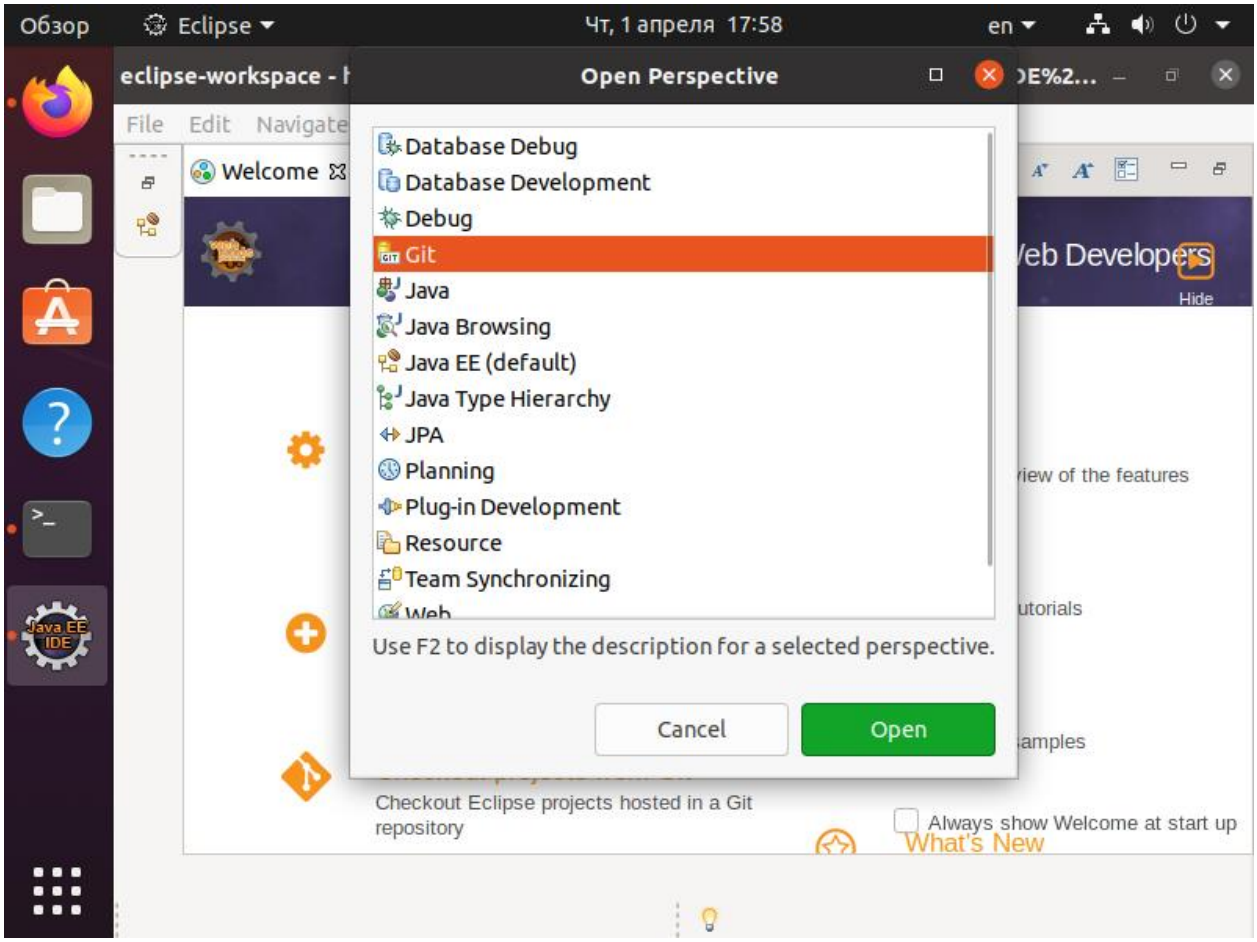


Рис. 11. Перспектива Git.

7. Для использования JUnit тестов будем использовать Maven проект. Для работоспособности JUnit в pom.xml в project необходимо добавить:

```
<dependencies>
    <dependency>
        <groupId>junit</groupId>
        <artifactId>junit</artifactId>
        <version>4.13.1</version>
```

```
    </dependency>
</dependencies>
```

3.2 **Настройка среды разработки для операционных систем на Linux Mint.**

1. Для работы необходим jdk, но перед его установкой обновим систему, введя в терминале команду:

sudo apt-get update

2. После обновления системы установим jdk, введя команду (Рис.12):

sudo apt-get install default-jdk

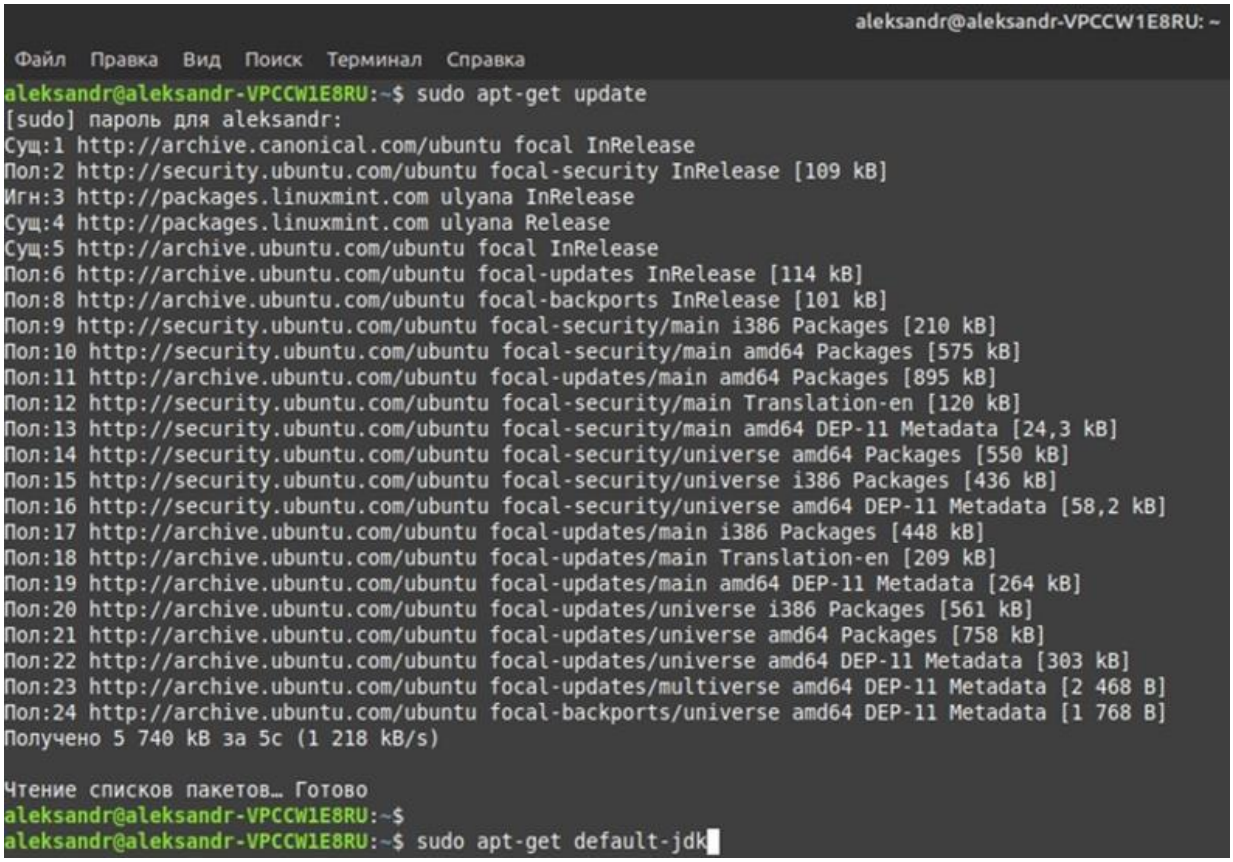


Рис. 12. Установка jdk.

3. Далее требуется установить Eclipse IDE for Enterprise Java and Web Developers. Сделать это можно на официальном сайте: https://www.eclipse.org/downloads/packages/ (Рис. 13). Выбираем *Linux x86_64*. Далее следует распаковать дистрибутив в домашнюю папку пользователя и выполнить файл eclipse из каталога eclipse.

Рис. 13. Сайт Eclipse.

4. Установим и настроим операционную систему для работы с Git. Для установки пропишем в терминале следующую команду (Рис.14):

sudo apt-get install git

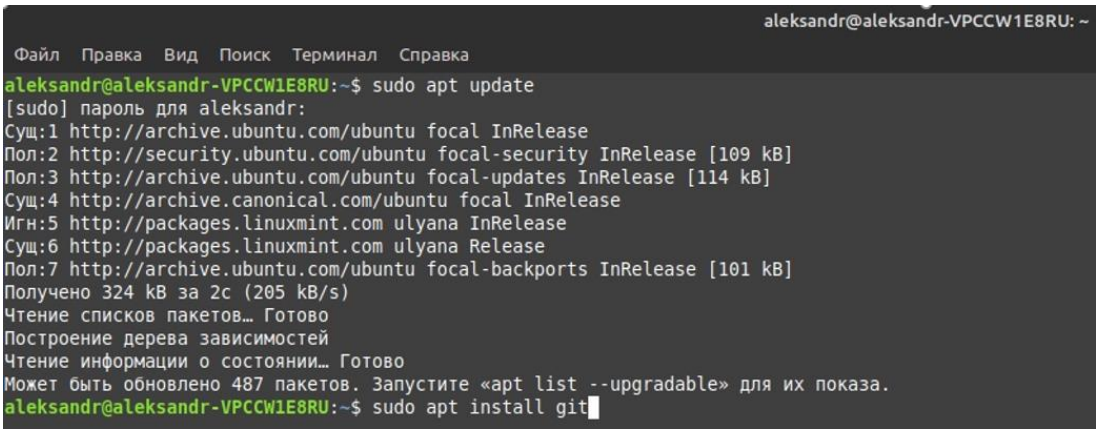


Рис. 14. Установка Git.

5. Настроем Git, для этого введем следующие две команды, для настройки имени пользователя и адреса электронной почты (Рис.15):

git config --global user.name "My Name"

git config --global user.email myEmail@example.com

Рис. 15. Настройка Git.

6. Необходимые Git и Maven в Eclipse IDE for Enterprise Java and Web Developers предустановлены, но для работы с Git требуется вывести на экран его перспективу, для этого нажимаем Window -> Perspective -> Open Perspective -> Other и выбираем Git (Рис.16).

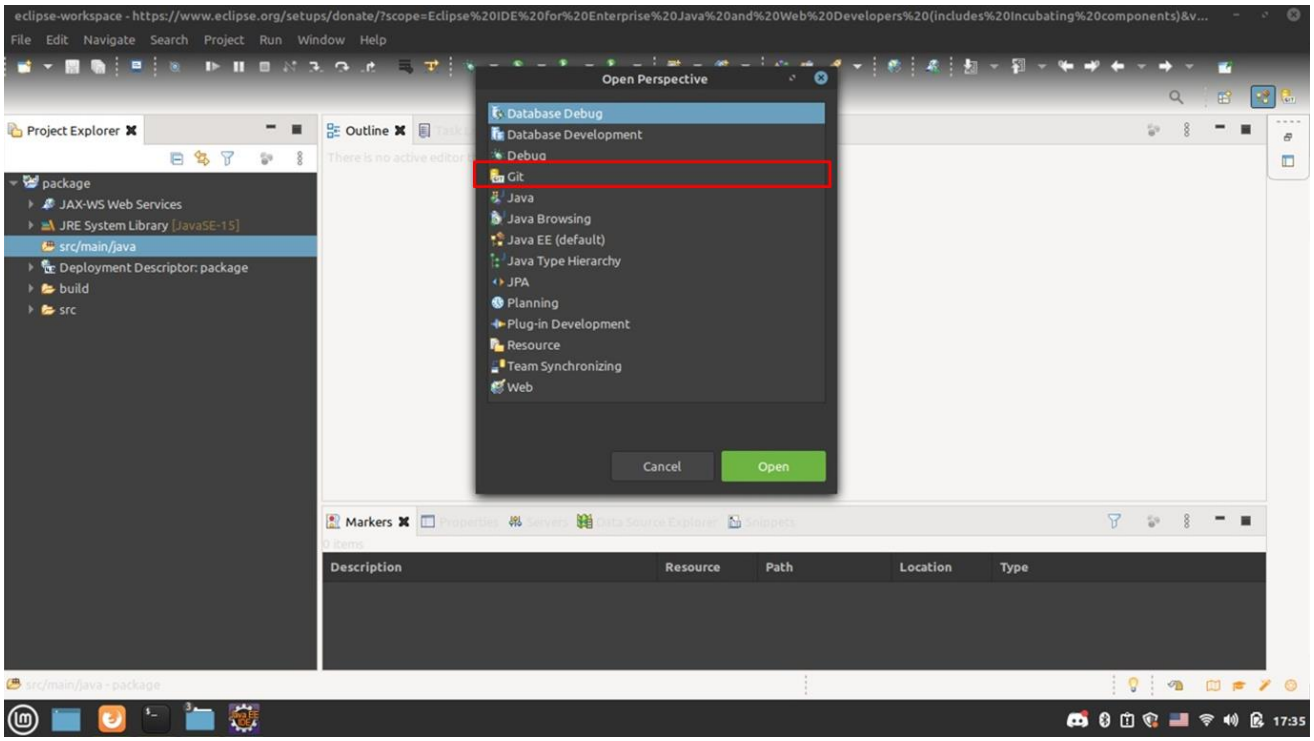


Рис. 16. Перспектива Git.

7. Для использования JUnit тестов будем использовать Maven проект. Для работоспособности JUnit в pom.xml в project необходимо добавить:

```
<dependencies>
    <dependency>
        <groupId>junit</groupId>
        <artifactId>junit</artifactId>
        <version>4.13.1</version>
    </dependency>
</dependencies>
```

3.3 Настройка среды разработки для операционных систем на Windows 10

1. Для работы необходим jdk. Для его установки перейдем по ссылке и скачаем версию windows x64 (Рис. 17):

https://www.oracle.com/ru/java/technologies/javase-jdk15-downloads.html

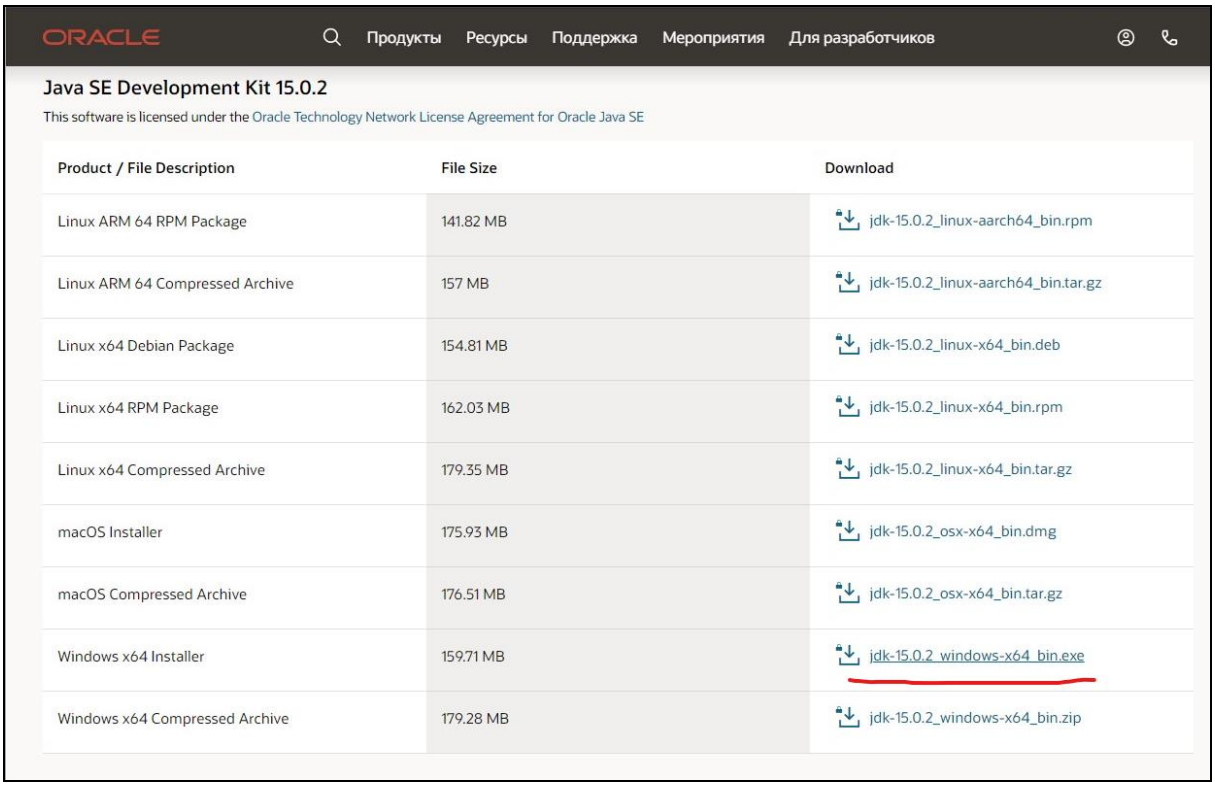


Рис. 17. Скачивание jdk.

2. Запускаем скачанный файл для установки jdk.

3. Для установки Eclipse IDE for Enterprise Java and Web Developers перейдем по ссылке https://www.eclipse.org/downloads/download.php?file=/oomph/epp/2021-03/R/eclipse-inst-jre-win64.exe и нажмем *Download* (Рис.18).

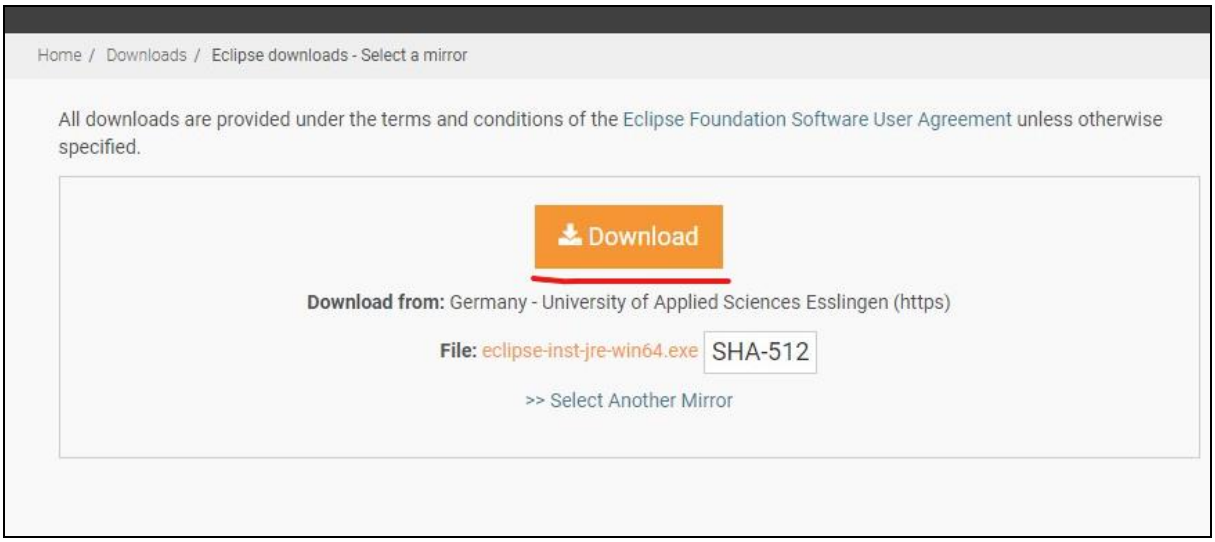


Рис. 18. Скачивание Eclipse IDE for Enterprise Java and Web Developers

4. После скачивания и запуска файла выбираем версию Eclipse IDE for Enterprise Java and Web Developers (Рис.19).

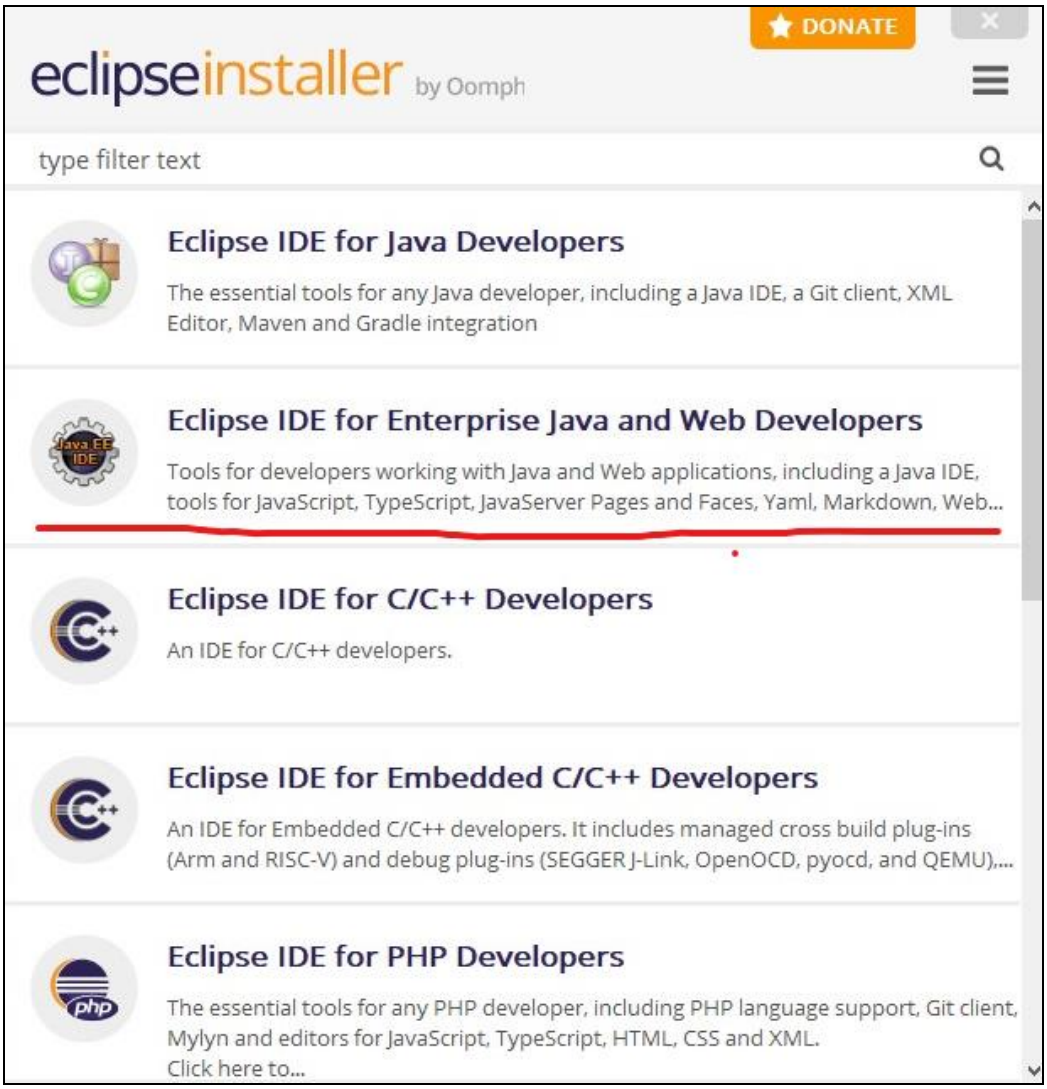


Рис. 19. Установка Eclipse IDE for Enterprise Java and Web Developers.

5. Необходимый Git в Eclipse IDE for Enterprise Java and Web Developers уже предустановлен, для добавления его перспективы нажмем Window -> Perspective -> Open Perspective -> Other и выберем Git (Рис.20), (Рис.21).

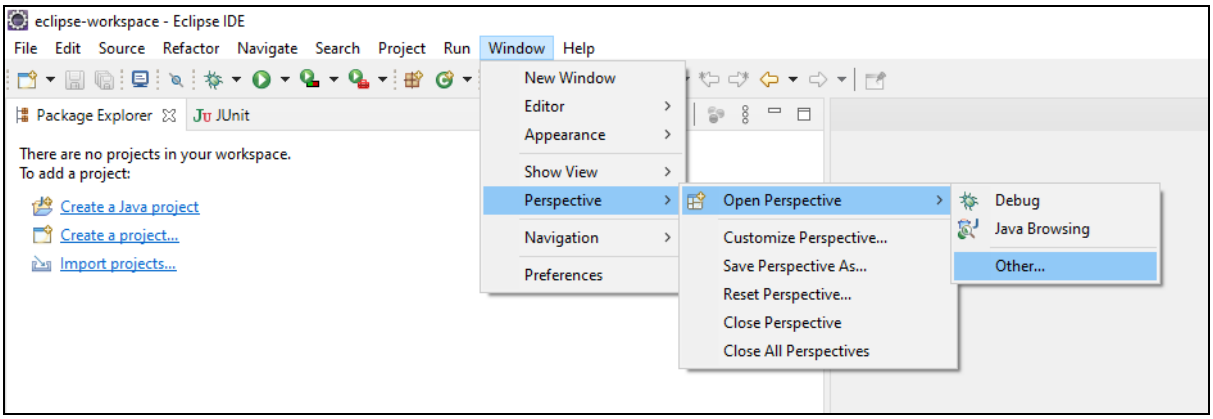


Рис. 20. Выбор перспективы Git.

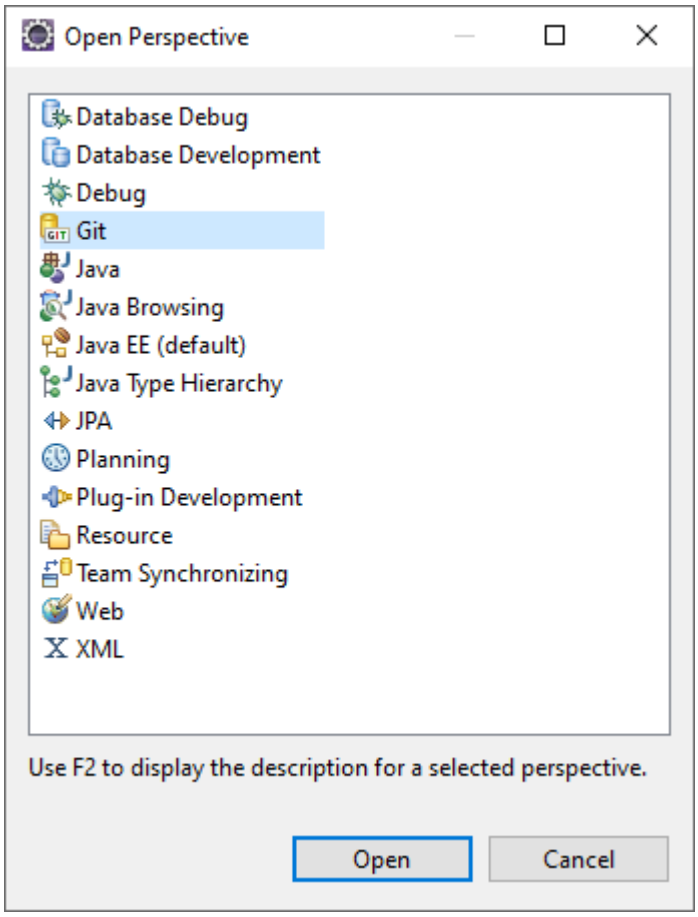
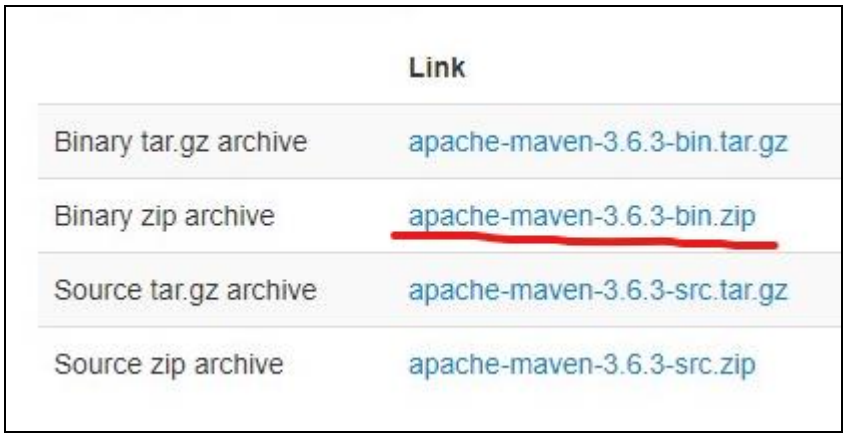


Рис. 21. Перспектива Git.

6. Для загрузки Maven перейдем по ссылке: http://maven.apache.org/download.cgi и скачаем архив (Рис.22).



	Link
Binary tar.gz archive	apache-maven-3.6.3-bin.tar.gz
Binary zip archive	apache-maven-3.6.3-bin.zip
Source tar.gz archive	apache-maven-3.6.3-src.tar.gz
Source zip archive	apache-maven-3.6.3-src.zip

Рис. 22. Скачивание Maven.

7. Распаковываем скаченный архив в любую папку, например, C:\Maven.
8. Создадим переменную в Path, где указываем путь к Maven. Также нужно создать переменную M2_HOME, M2 и если нужно, то JAVA_HOME (Рис.23).

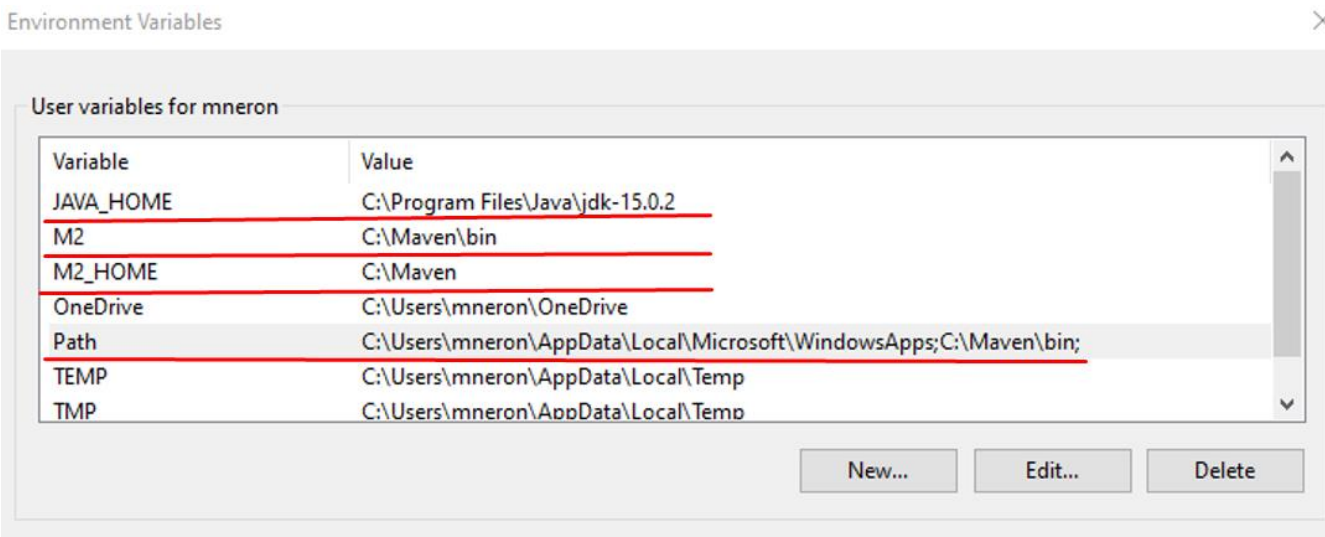


Рис. 23. Созданные и измененные переменные среды.

9. Для проверки работоспособности введем в командную строку команду (Рис.24):

mvn –version

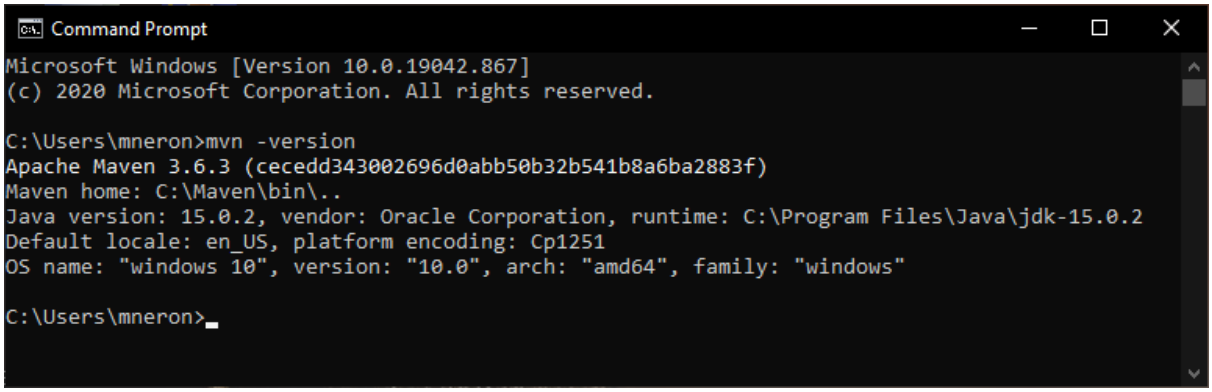


Рис. 24. Проверка работоспособности Maven.

10. Для использования JUnit тестов будем использовать Maven проект. Для работоспособности JUnit в pom.xml в project необходимо добавить:

```
<dependencies>
    <dependency>
        <groupId>junit</groupId>
        <artifactId>junit</artifactId>
        <version>4.13.1</version>
    </dependency>
</dependencies>
```

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

Поскольку среда разработки Eclipse универсальна на всех операционных системах, то будет достаточно показать настройку единожды.

1) Ссылка на репозиторий, в котором ведется работа: https://github.com/Salikhov12/calculator (Рис.25).

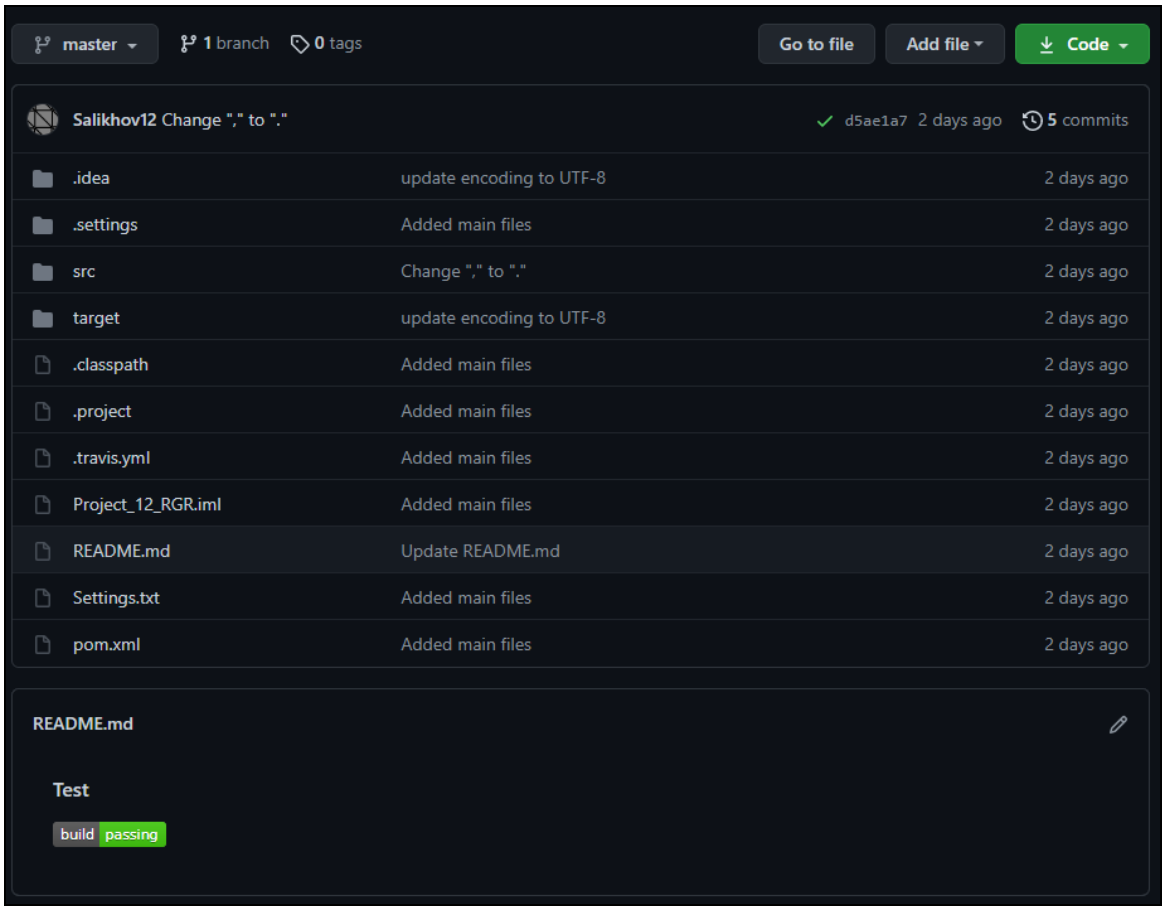


Рис. 25. Вид репозитория в браузере.

2) В последних версиях Eclipse Git уже предустановлен. Добавим перспективу Git. Для этого нажимаем *Window→Perspective→Open Perspective→Other…* (Рис.26). После чего выбираем *Git* (Рис.27).

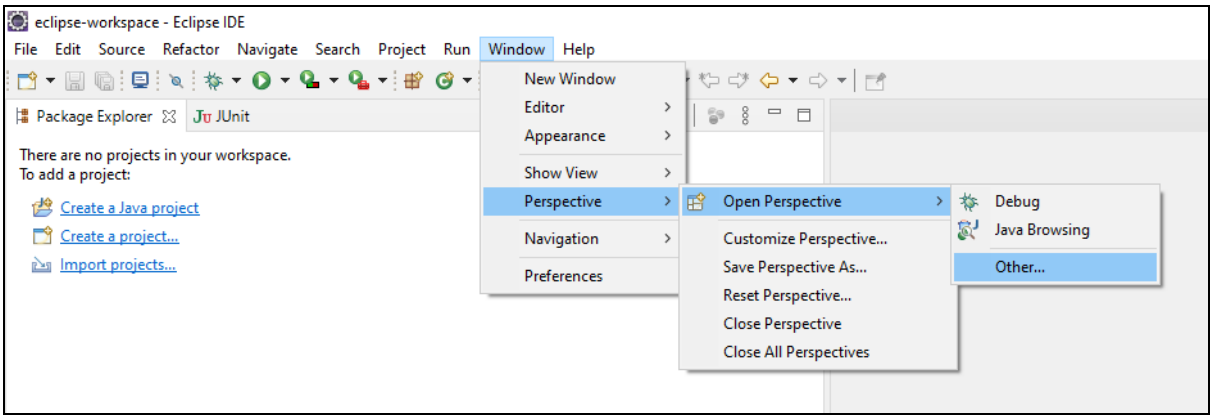


Рис. 26. Открытие окна перспектив, для выбора Git.

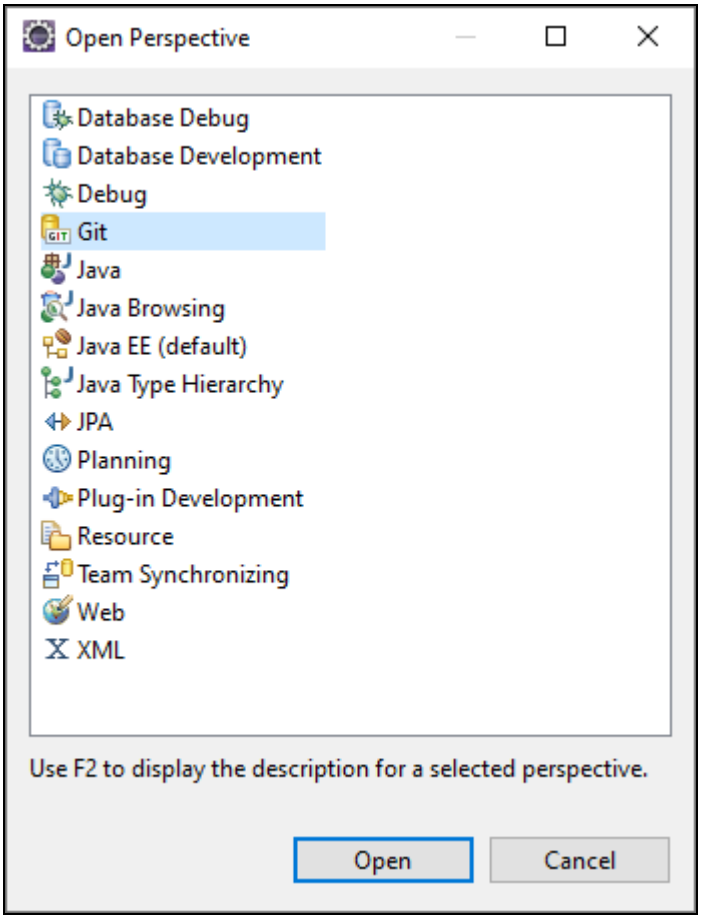


Рис. 27. Выбор перспективы Git.

3) Для клонирования репозитория в перспетиве Git выбираем *Clone a Git repository* (Рис.28).

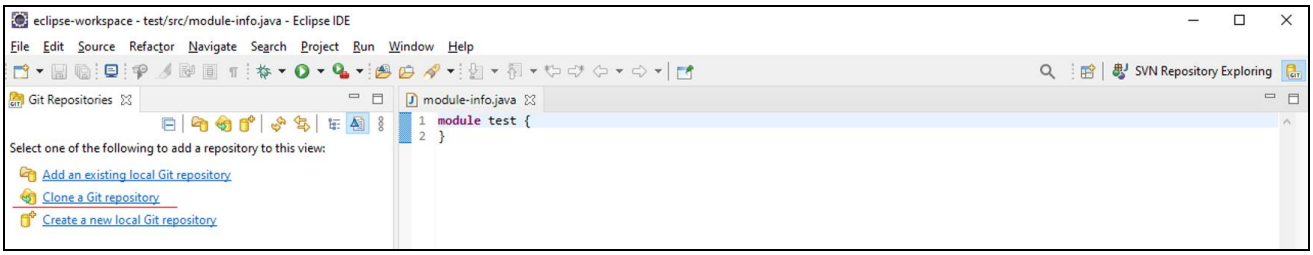


Рис. 28. Выбор Clone a Git Repository.

4) В поле *URI* вводим ссылку на нужный репозиторий, в разделе *Authentication* вводим логин и пароль от своего аккаунта на Github (Рис.29), нажимаем *Next*.

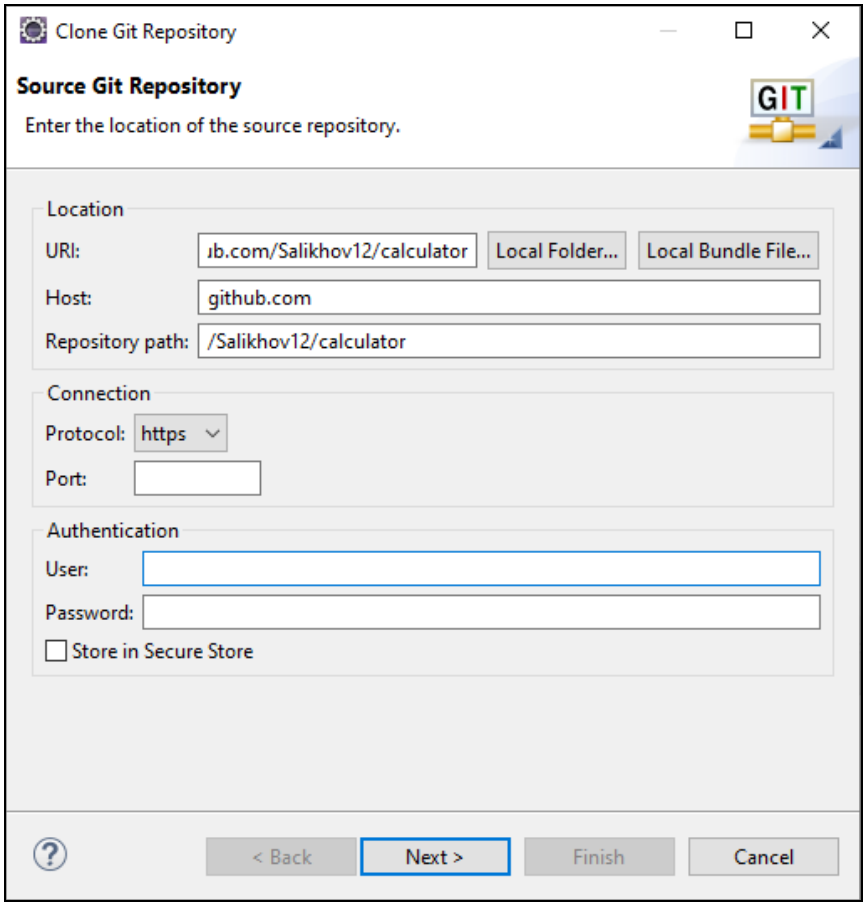


Рис. 29. Ввод данных репозитория и аккаунта.

5) Выбираем необходимые ветки (Рис.30), нажимаем *Next*.

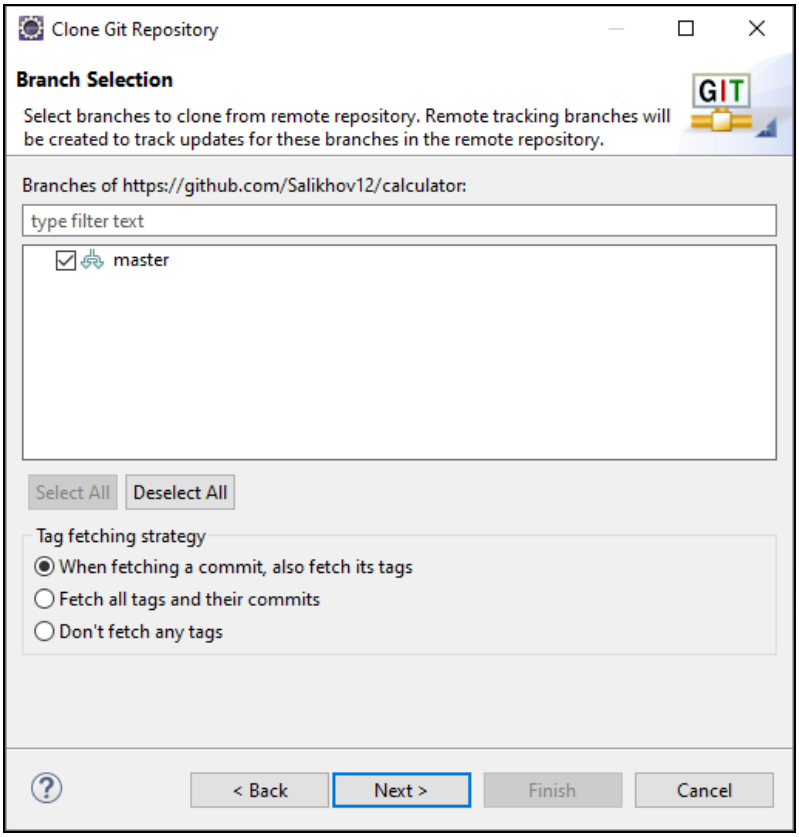


Рис. 30. Выбор необходимых веток.

6) Выбираем в *Directory* расположение репозитория на ПК и нажимаем *Finish* (Рис.31).

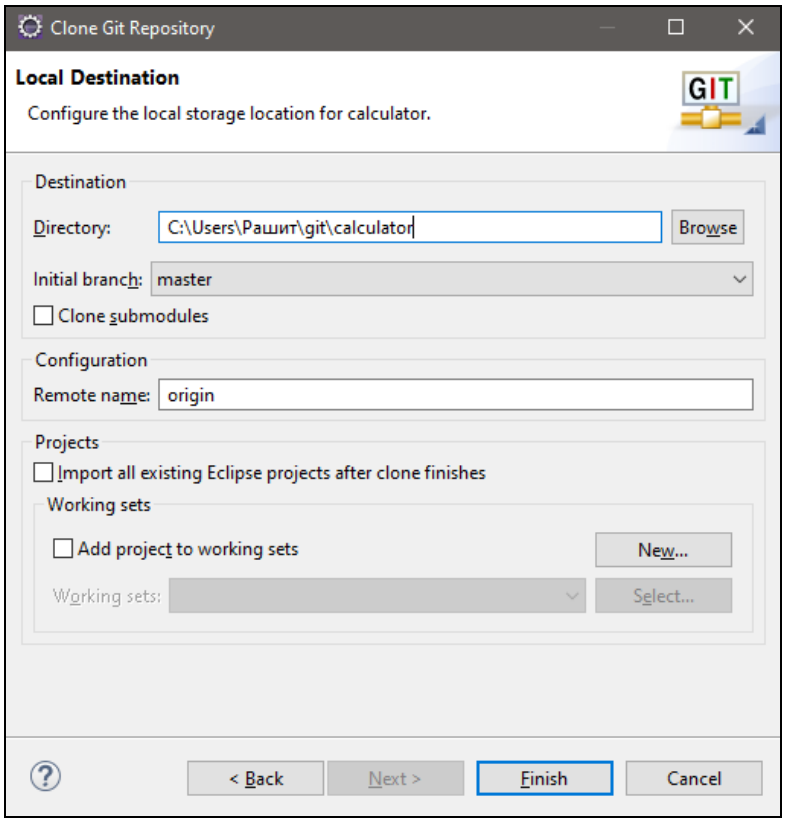


Рис. 31. Выбор расположения репозитория.

7) Для создания рабочей копии нажимаем ПКМ на *Working Tree*, выбираем *Import Projects…* (Рис.32) и нажимаем *Finish*.

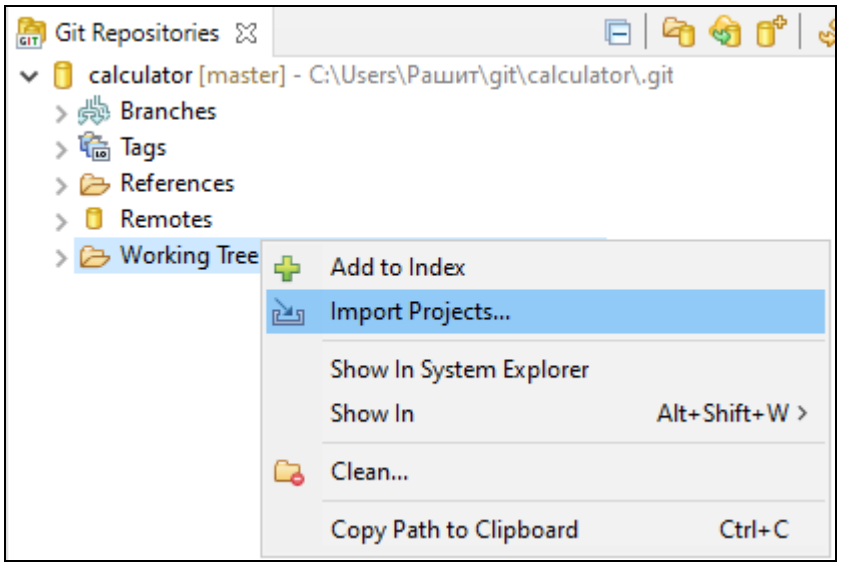


Рис. 32. Импортирование проекта с Github.

Раздел 5. Реализация исходного кода по зонам ответственности

Зона ответственности каждого разработчика представлена в таблице 1.

Таблица 1 – Зоны ответственности

№	ФИО разработчика/модератора	Зона ответственности
1	Салихов Рашит Ниязович	Создал: Классы LogInReg и Main; Jsp файлы LogIn и Reg. (Приложение 3-6)
2	Иутин Александр Сергеевич	Создал: Jsp файлы Form, FormAdm, Result; Стили jsp файлов Form, FormAdm, Result, LogIn, Reg; Абстрактный класс Dom; Классы PersonalHouse, SummerHouse, WinterHouse. (Приложение 7-13)
3	Бесполитов Даниил Анатольевич	Создал: Классы Admin, Calculator, House; Интерфейсы Home и Otdelka. (Приложение 14-18)

Ссылка на репозиторий:
https://github.com/Salikhov12/calculator
Ссылка на веб-приложение выделенного сервера:
http://mikilyanger.ru:49213/Calculator
Ссылка на веб-приложение, развернутое на Heroku:
https://calculator12pi.herokuapp.com/
Аккаунт администратора:
Логин: admin
Пароль: admin
Аккаунт обычного пользователя:
Логин: ivanov
Пароль: password

Раздел 6. Сборка и тестирование программного продукта

6.1 Описание UNIT-тестов

Описание UNIT-тестов и ответственные за их реализацию представлены в таблице 2.

Таблица 2 – Описание UNIT-тестов

№	ФИО разработчика/модератора	Описание UNIT-теста	№ Приложения
1	Салихов Рашит Ниязович	Тест проверяет подключение к базе данных в методе getConnection. Применяется для класса: LogInReg. Исходные данные, которые меняются: con. Ожидаемый результат: true.	Приложение 19
2	Иутин Александр Сергеевич	Тест проверяет верную передачу данных в методах setKolE и setLenght. Применяется для класса: SummerHouse. Исходные данные, которые меняются: KolE, lenght. Ожидаемый результат: true.	Приложение 20
3	Бесполитов Даниил Анатольевич	Тест проверяет расчеты, производимые в методе OtdelkaPrice. Применяется для класса: House. Исходные данные, которые меняются: OtdelkaValue, square, CountOfFloor. Ожидаемый результат: true.	Приложение 21

6.2 Инструкция по сборке проекта

Структура проекта по каталогам представлена на рисунке 33.

Рис. 33. Структура проекта по каталогам

Проект включает в себя:

1. Каталоги src, artifacts, target.

2. Файлы pom.xml, Procfile, README.md, Инструкция по подключению к системе контроля версий.pdf

Каталог src включает в себя:

1. Подкаталоги main/java/Package, test/java/Package и main/webapp

Каталог artifacts включает в себя:

1. Файлы Calculator.war и webapp-runner.jar

Подкаталог main/java/Package включает в себя:

1. Файлы Admin.java, Calculator.java, Dom.java, Home.java, House.java, LogInReg.java, Main.java, Otdelka.java, PersonalHouse.java, SummerHouse.java, WinterHouse.java

Подкаталог test/java/Package включает в себя:

1. Файлы ConnectionDataBaseTest.java, HouseTest.java, ummerHouseTest.java

Подкаталог main/webapp включает в себя:

1. Подкаталог WEB-INF

2. Файлы Form.jsp, FormAdm.jsp, LogIn.jsp, Reg.jsp, Result.jsp

Подкаталог WEB-INF включает в себя:

1. Файл web.xml

Назначение каталогов:

1. src – содержит в себе необходимые для работы проекта файлы типа .java

2. artifacts – содержит в себе файлы, необходимые для работы веб-приложения

3. target – используется фреймворком Maven

Назначение подкаталогов:

1. main/java/Package – содержит в себе .java классы, необходимые для работы проекта

2. test/java/Package – содержит в себе JUnit тесты, необходимые для

тестирования проекта

3. main/webapp – содержит в себе .jsp файлы, необходимые для работы веб-приложения

4. WEB-INF – содержит в себе .xml файл, необходимый для работы веб-приложения

Назначение файлов:

1. pom.xml – содержит в себе описание всей структуры проекта, а также необходимые для работы зависимости и плагины

2. Procfile – файл, необходимый облачной платформе Heroku для развертывания на ней приложения

3. README.md – файл, в которой содержится некая информация о проекте. Используется на github

4. Calculator.war – веб-архив разрабатываемого приложения

5. webapp-runer.jar – архив, используемый для работы Tomcat

6. Admin.java – настройки администратора

7. Calculator.java – основные расчёты

8. Dom.java – абстрактный класс

9. Home.java – интерфейс

10. House.java – класс, имплементирующий классы Otdelka, Home

11. LogInReg.java – вход в аккаунт и регистрация нового

12. Main.java – основной класс

13. Otdelka.java – интерфейс

14. PersonalHouse.java – класс, наследующий класс Dom, персональный дом

15. SummerHouse.java – класс, наследующий класс Dom, летний дом

16. WinterHouse.java – класс, наследующий класс Dom, зимний дом

17. ConnectionDataBaseTest.java – проверка соединения с базой данных

18. HouseTest.java – проверка расчетов в методе OtdelkaPrice

19. SummerHouseTest.java – проверка правильности работы класса SummerHouse

20. Form.jsp – форма расчета для обычных пользователей
21. FormAdm.jsp – форма расчета для админа
22. LogIn.jsp – форма входа
23. Reg.jsp – форма регистрации
24. Result.jsp – форма с результатом расчетов

Описание структуры pom.xml (Рис.34), (Рис.35).

В структуре файла pom.xml можно выделить следующие элементы:

1. groupId – идентификатор некой группы организации
2. artifactId – идентификатор конкретного проекта
3. version – версия проекта
4. packaging – в каком виде будет результат сборки проекта
5. name – название проекта
6. description – описание проекта

Также в данном файле можно добавлять необходимые проекту плагины, например: maven-compiler-plugin, maven-compiler-plugin, maven-war-plugin, а также необходимые зависимости, например: junit, javax.servlet-api, el-api, webapp-runner, mysql-connector-java.

Для работоспособности проекта необходимы все вышеперечисленные плагины и зависимости, пример заполнения представлен на рисунках 2 и 3.

```
<project xmlns="http://maven.apache.org/POM/4.0.0" xmlns:xsi="http://ww
  <modelVersion>4.0.0</modelVersion>
  <groupId>RGR</groupId>
  <artifactId>Project_12_RGR</artifactId>
  <version>1.0</version>
    <packaging>war</packaging>
  <name>Calculator</name>
  <description>Calculator RGR 12</description>
    <build>
      <finalName>Calculator</finalName>
  <defaultGoal>package</defaultGoal>
<pluginManagement>
            <plugins>
                <plugin>
                    <artifactId>maven-compiler-plugin</artifactId>
                    <version>3.3</version>
                    <configuration>
                        <source>1.8</source>
                        <target>1.8</target>
                    </configuration>
                </plugin>

    <plugin>
    <groupId>org.apache.maven.plugins</groupId>
    <artifactId>maven-dependency-plugin</artifactId>
    <version>3.1.2</version>
    <executions>
        <execution>
            <phase>package</phase>
            <goals>
                <goal>copy</goal>
            </goals>
            <configuration>
                <artifactItems>
                    <artifactItem>
                        <groupId>com.heroku</groupId>
                        <artifactId>webapp-runner</artifactId>
                        <version>9.0.31.0</version>
                        <destFileName>webapp-runner.jar</destFileName>
                    </artifactItem>
                </artifactItems>
                <outputDirectory>artifacts</outputDirectory>
            </configuration>
        </execution>
    </executions>
</plugin>
```

Рис. 34.Содержимое pom.xml, первая половина

```
        <plugin>
    <artifactId>maven-war-plugin</artifactId>
    <version>3.3.0</version>
    <configuration>
        <goal>war:inplace</goal>
        <outputDirectory>artifacts</outputDirectory>
    </configuration>
</plugin>
            </plugins>
        </pluginManagement>
  </build>
  <dependencies>
    <dependency>
        <groupId>junit</groupId>
        <artifactId>junit</artifactId>
        <version>4.13.1</version>
    </dependency>

      <dependency>
          <groupId>javax.servlet</groupId>
          <artifactId>javax.servlet-api</artifactId>
          <version>3.1.0</version>
      </dependency>
      <dependency>
          <groupId>javax.el</groupId>
          <artifactId>el-api</artifactId>
          <version>2.2</version>
      </dependency>
      <dependency>
          <groupId>com.github.jsimone</groupId>
          <artifactId>webapp-runner</artifactId>
          <version>9.0.27.1</version>
          <scope>provided</scope>
      </dependency>
      <dependency>
          <groupId>mysql</groupId>
          <artifactId>mysql-connector-java</artifactId>
          <version>8.0.21</version>
      </dependency>
  </dependencies>
</project>
```

Рис. 35. Содержимое pom.xml, вторая половина

Описание структуры web.xml (Рис.36).

В структуре файла web.xml можно выделить следующие элементы:

1. display-name – название, которое будет выводиться на вкладке
2. welcome-file - .jsp файл, который будет открываться при открытии страницы проекта.

Для работоспособности проекта необходимо заполнить файл web.xml, как показано на рисунке 36.

```
1 <?xml version="1.0" encoding="UTF-8"?>
2 <web-app xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xmlns="http://xmlns.jcp.org/xml/ns/javaee"
  xsi:schemaLocation="http://xmlns.jcp.org/xml/ns/javaee http://xmlns.jcp.org/xml/ns/javaee/web-app_4_0.xsd"
  version="4.0">
3   <display-name>Вход</display-name>
4   <welcome-file-list>
5     <welcome-file>LogIn.jsp</welcome-file>
6   </welcome-file-list>
7 </web-app>
```

Рис. 36. Содержимое web.xml

Сборка проекта

Для сборки проекта необходимо выполнить следующие шаги в IDE Eclipse:

1. ПКМ по проекту, затем Run As и Maven build… (Рисунок 37).

2. В поле Goals вводим команду package и жмем Run (Рисунок 38).

3. Дожидаемся выполнения сборки (Рисунок 39).

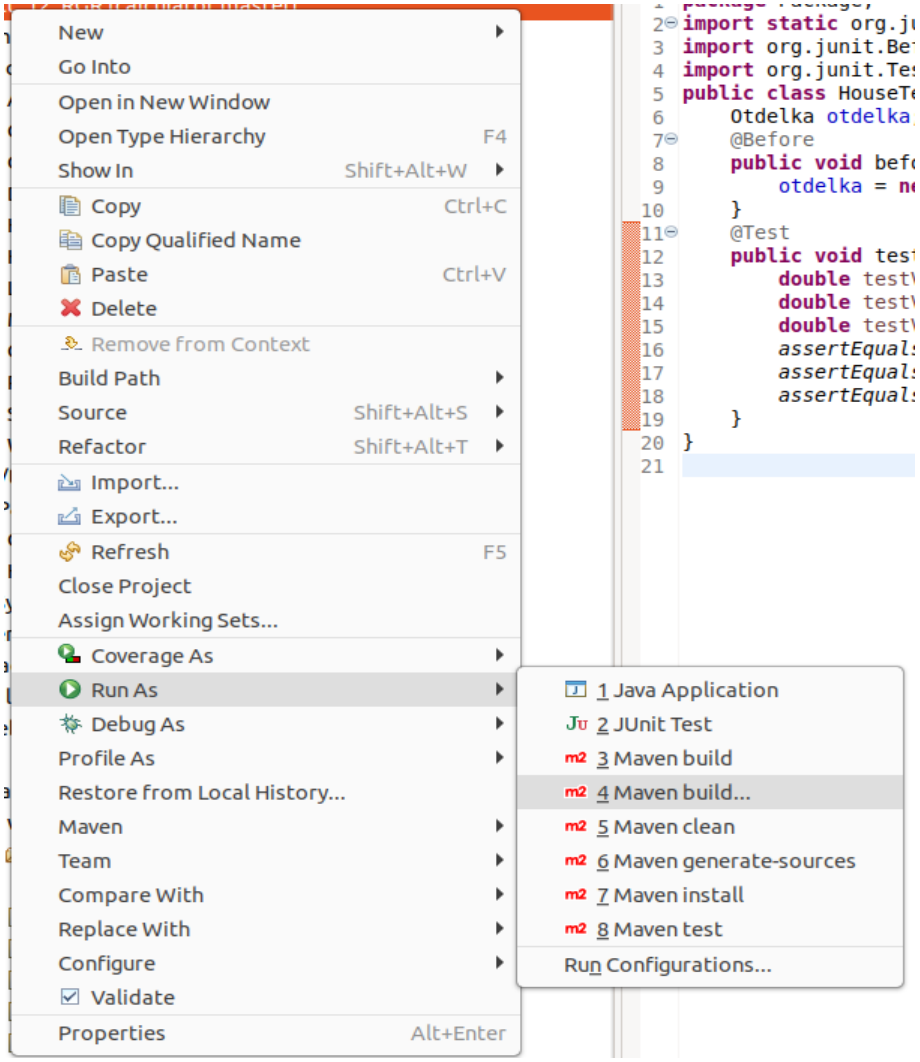


Рис. 37. Начало сборки проекта

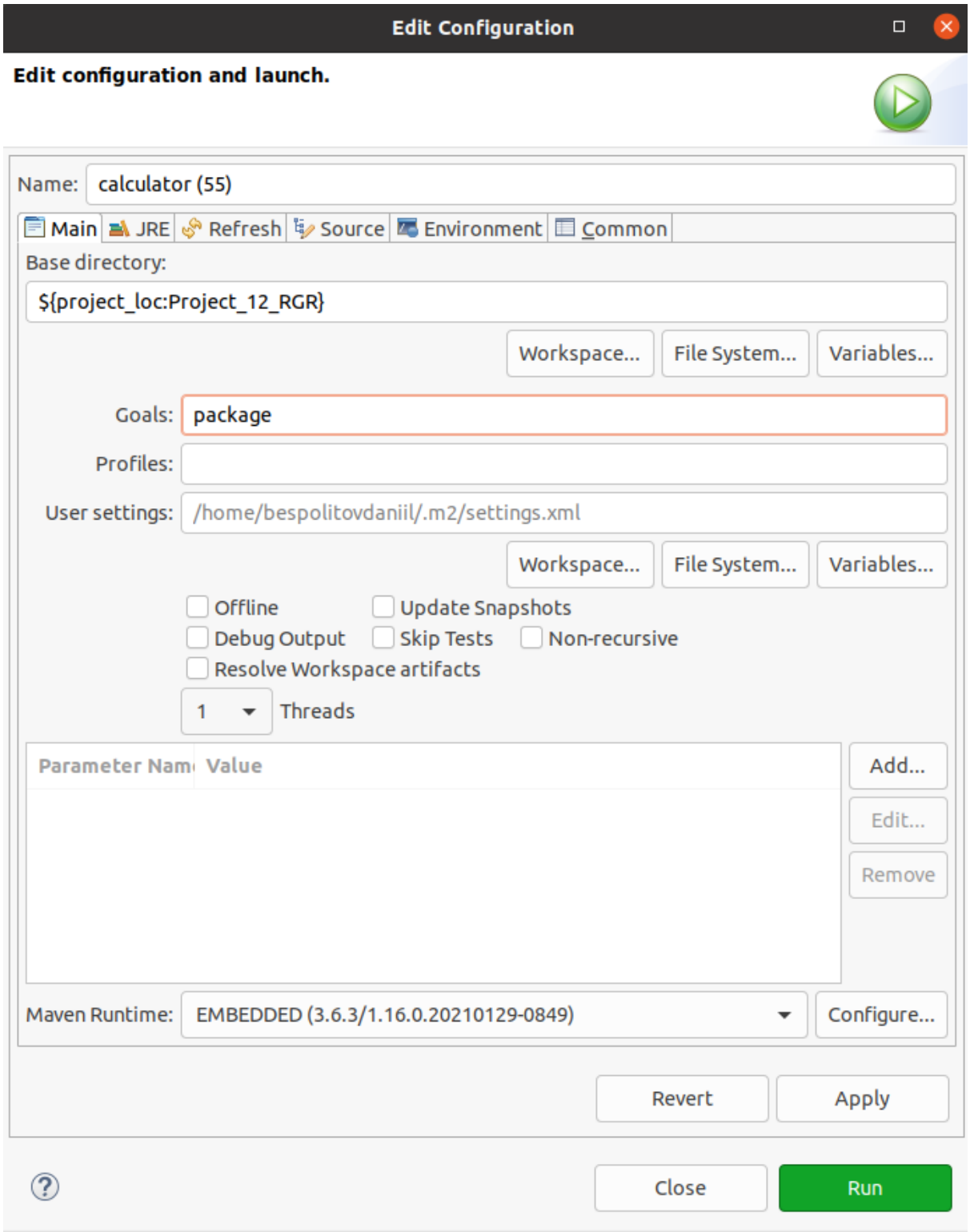


Рис. 38. Ввод команды package в Goals

```
[INFO] --- maven-war-plugin:3.3.0:war (default-war) @ Project_12_RGR ---
[INFO] Packaging webapp
[INFO] Assembling webapp [Project_12_RGR] in [/home/bespolitovdaniil/git/calculator/target/Calculator]
[INFO] Processing war project
[INFO] Copying webapp resources [/home/bespolitovdaniil/git/calculator/src/main/webapp]
[INFO] Building war: /home/bespolitovdaniil/git/calculator/artifacts/Calculator.war
[INFO] ------------------------------------------------------------------------
[INFO] BUILD SUCCESS
[INFO] ------------------------------------------------------------------------
[INFO] Total time:  7.706 s
[INFO] Finished at: 2021-05-13T16:26:18+05:00
[INFO] ------------------------------------------------------------------------
```

Рис. 39. Результат сборки проекта

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

7.1 Настройка непрерывной интеграции для платформы Heroku

Для непрерывной интеграции на Heroku добавим в каталог проекта файл .travis.yml.

В созданный файл добавим дополнение language и его описание java (Рис.40), чтобы дать знать системе, какой язык мы используем.

После этого пропишем script (Рис.40), для выполнения сборки проекта.

```
language: java
script:
- mvn clean package
```

Рис. 40. Файл .travis.yml шаг script

Для деплоя на платформу добавим код, показанный на рисунке 41, где в поле provider – наша платформа, master – название развертываемой ветви проекта, а в поле master – название созданного нами приложения на сайте Heroku CLI.

```
deploy:
  provider: heroku
  app:
    master: calculatorl2pi
  api_key:
    secure:
```

Рис. 41. Файл .travis.yml шаг deploy

Для связи Travis CI и Heroku необходим API ключ, который необходимо зашифровать.

API ключ получается на сайте Heroku CLI в настройках пользователя (Рис.42).

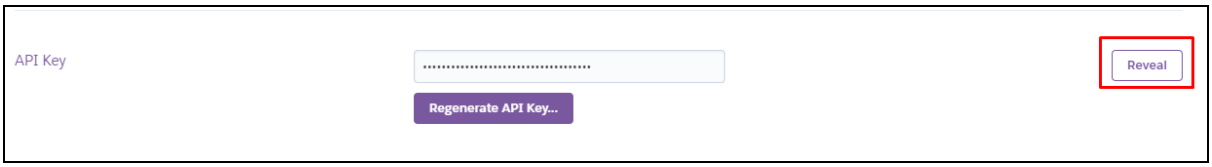


Рис. 42. Получение API ключа на Heroku CLI

Для шифровки ключа воспользуемся следующей командой из каталога проекта:

travis encrypt $(heroku auth:token) --add deploy.api_key --pro

Где в нашем случае вместо *$(heroku auth:token)* введем d97513d9-20d8-4af9-a563-860b5b6b7062.

После данной команды в .travis.yml в поле secure появится зашифрованный ключ.

Теперь после закрепления данных изменений будет доступно автоматическое развертывание и запуск приложения на Heroku.

7.2 Настройка непрерывной интеграции для веб-сервера

Для подключения автоматической развертки через Travis CI на VPS (Virtual Private Server) через ssh необходимо выполнить следующие действия:

Создадим новую папку .travis (для более удобного хранения ключей) командой:

mkdir .travis

Для получения доступа к серверу создадим два ssh ключа командой:

ssh-keygen -t .travis/deploy_rsa -C "deploy_key"

После этого введем их название *id_rsa_deploy* и оставим без пароля.

После команды выше в папке .travis сгенерируются файлы id_rsa_deploy и id_rsa_deploy.pub.

Зашифруем файл id_rsa_deploy командой:

travis encrypt-file .travis/id_rsa_deploy --add --com,

где --add добавляет в файл .travis.yml необходимую команду для дешифровки (Рис.43).

```
before_install:
- openssl aes-256-cbc -K $encrypted_060bfcf8ce8c_key -iv $encrypted_060bfcf8ce8c_iv
  -in .travis/id_rsa_deploy.enc -out .travis/id_rsa_deploy -d
```

Рис. 43. Добавленная команда в файл .travis.yml

Теперь мы должны удалить файл id_rsa_deploy из каталога .travis, поскольку он не зашифрован, командой:

rm .travis/id_rsa_deploy

После выполнения этих действий можем закрепить изменения репозитория.

Файл id_rsa_deploy.pub необходимо добавить в ~/.ssh/authorized_keys на сервере командой:

ssh-copy-id -i deploy_key.pub root@ovz1.iutin1999.pqj7n.vps.myjino.ru.

Также в домашнюю папку необходимо добавить папку .travis и скопировать в неё id_rsa_deploy.enc.

Чтобы Travis CI мог подключаться к нашему серверу, добавим в .travis.yml ssh_known_hosts (Рис.44).

```
addons:
  ssh_known_hosts: ovz1.iutin1999.pqj7n.vps.myjino.ru:49290
```

Рис. 44. Необходимый для подключения к серверу код

После чего пропишем команды по инициализации нового ключа:

after_success:

- eval "$(ssh-agent -s)"

- chmod 600 .travis/id_rsa_deploy

- ssh-add .travis/id_rsa_deploy

В том же after_success пропишем копирование упакованного .war файла с Travis-CI на наш сервер:

- scp -P 49290 /home/travis/build/Salikhov12/calculator/artifacts/Calculator.war root@ovz1.iutin1999.pqj7n.vps.myjino.ru:/root/tomcat/

А потом команду для удаления прошлого .war файла и перемещение на его место нового:

- ssh -p 49290 root@ovz1.iutin1999.pqj7n.vps.myjino.ru rm /root/tomcat/apache-tomcat-9.0.45/webapps/Calculator.war

- ssh -p 49290 root@ovz1.iutin1999.pqj7n.vps.myjino.ru cp /root/tomcat/Calculator.war /root/tomcat/apache-tomcat-9.0.45/webapps/

- ssh -p 49290 root@ovz1.iutin1999.pqj7n.vps.myjino.ru rm /root/tomcat/Calculator.war

После чего закрепляем изменения.

Итоговый файл .travis.yml можно увидеть на рисунке 45.

```
language: java
addons:
  ssh_known_hosts: ovz1.iutin1999.pqj7n.vps.myjino.ru:49290
before_install:
- openssl aes-256-cbc -K $encrypted_060bfcf8ce8c_key -iv $encrypted_060bfcf8ce8c_iv
  -in .travis/id_rsa_deploy.enc -out .travis/id_rsa_deploy -d
script: mvn clean package
after_success:
  - eval "$(ssh-agent -s)"
  - chmod 600 .travis/id_rsa_deploy
  - ssh-add .travis/id_rsa_deploy
  - scp -P 49290 /home/travis/build/Salikhov12/calculator/artifacts/Calculator.war root@ovz1.iutin1999.pqj7n.vps.myjino.ru:/root/tomcat/
  - ssh -p 49290 root@ovz1.iutin1999.pqj7n.vps.myjino.ru rm /root/tomcat/apache-tomcat-9.0.45/webapps/Calculator.war
  - ssh -p 49290 root@ovz1.iutin1999.pqj7n.vps.myjino.ru cp /root/tomcat/Calculator.war /root/tomcat/apache-tomcat-9.0.45/webapps/
  - ssh -p 49290 root@ovz1.iutin1999.pqj7n.vps.myjino.ru rm /root/tomcat/Calculator.war
deploy:
  provider: heroku
  app:
    master: calculator12pi
  api_key:
    secure: cH8o9wFPGorqvMJhgrcLtg8J1fYbXWCcsXQAOaB33NIpuJUohThqNmobdOs9kD5fcrUL1LouOyUb2NNhV3iMsfKv5Pgz6HuGQB7bUlOqllQsOTolcb8nmacRc0NEW
```

Рис. 45. Итоговый файл .travis.yml

7.3 Описание каталогов и файлов работающей программы.

Структура архива, в котором расположены файлы необходимые для работы программы, представлена на рисунке 46.

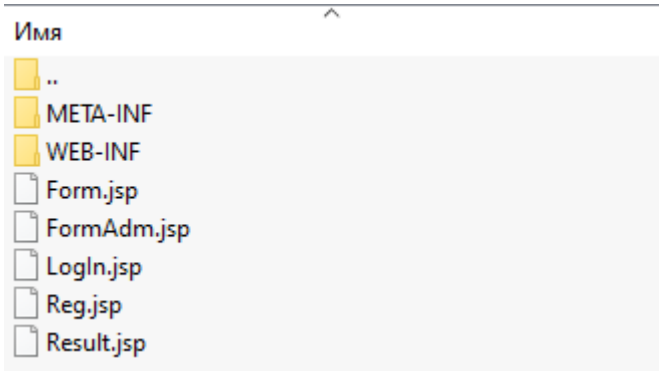


Рис. 46. Структура архива Calculator.war

Каталог META-INF содержит в себе подкаталоги maven, RGR, Project_12_RGR, а также файлы MANIFEST.MF, pom.properties и pom.xml.

Каталог META-INF является каталогом метаданных.

Файл MANIFEST.MF содержит в себе номер своей версии, чем был создан и номер используемого jdk.

Файл pom.properties содержит в себе номер версии, айди группы и айди артефакта проекта.

Остальные файлы были описаны в предыдущем разделе.

7.4 Условия работы программы

Наше веб-приложение работает через браузер, а условием запуска является наличие интернета.

Примеры работы программы: Mozilla Firefox на Ubuntu 20.04 (Рис. 47), Google Chrome на Windows 10 (Рис. 48), Яндекс.Браузер на Android 10 (Рис. 49).

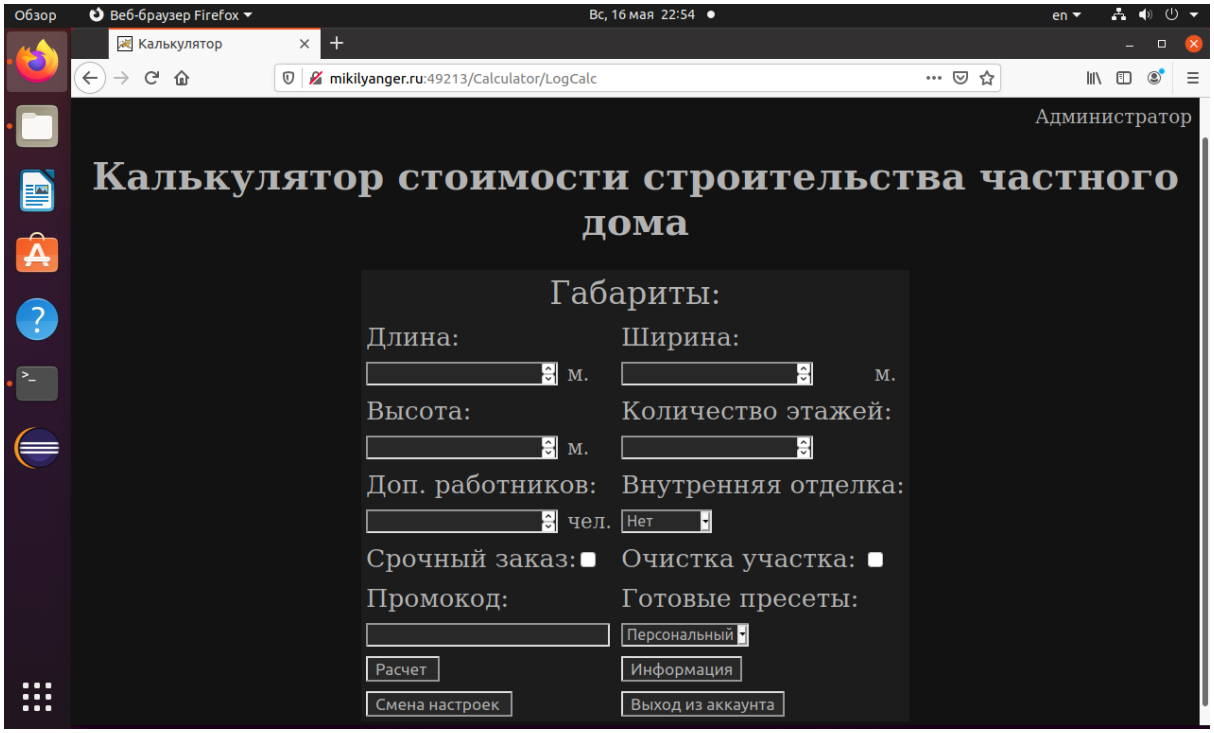


Рис. 47. Работа приложения в браузере Mozilla Firefox на Ubuntu 20.04

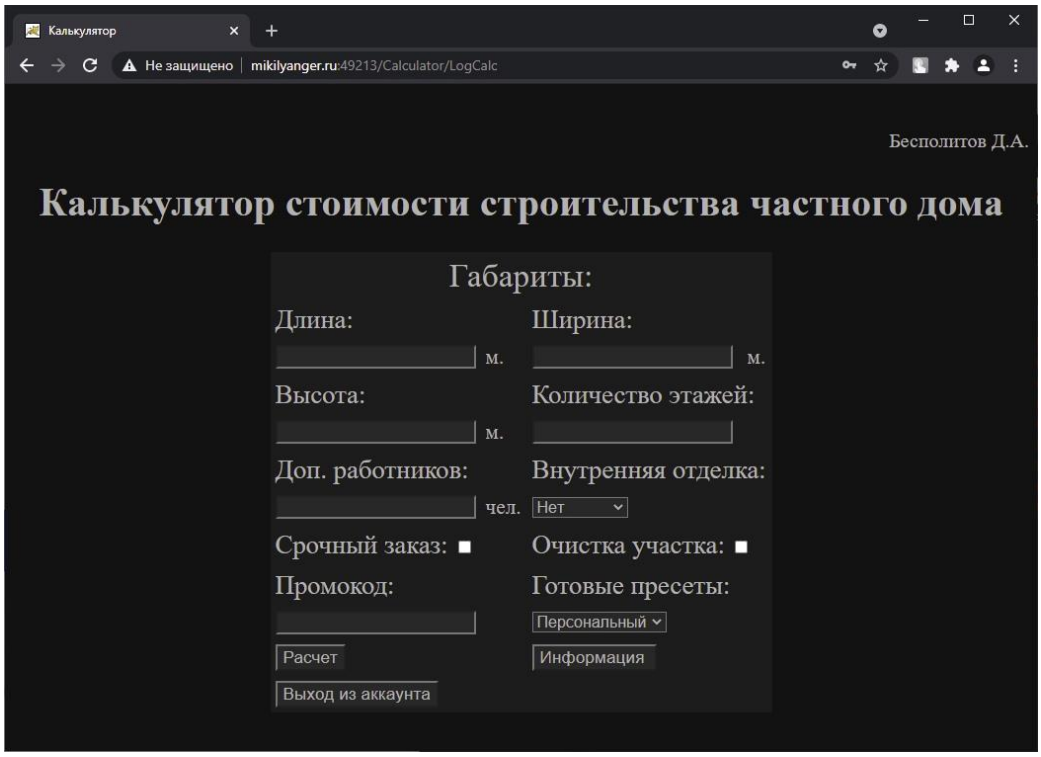


Рис. 48. Google Chrome на Windows 10

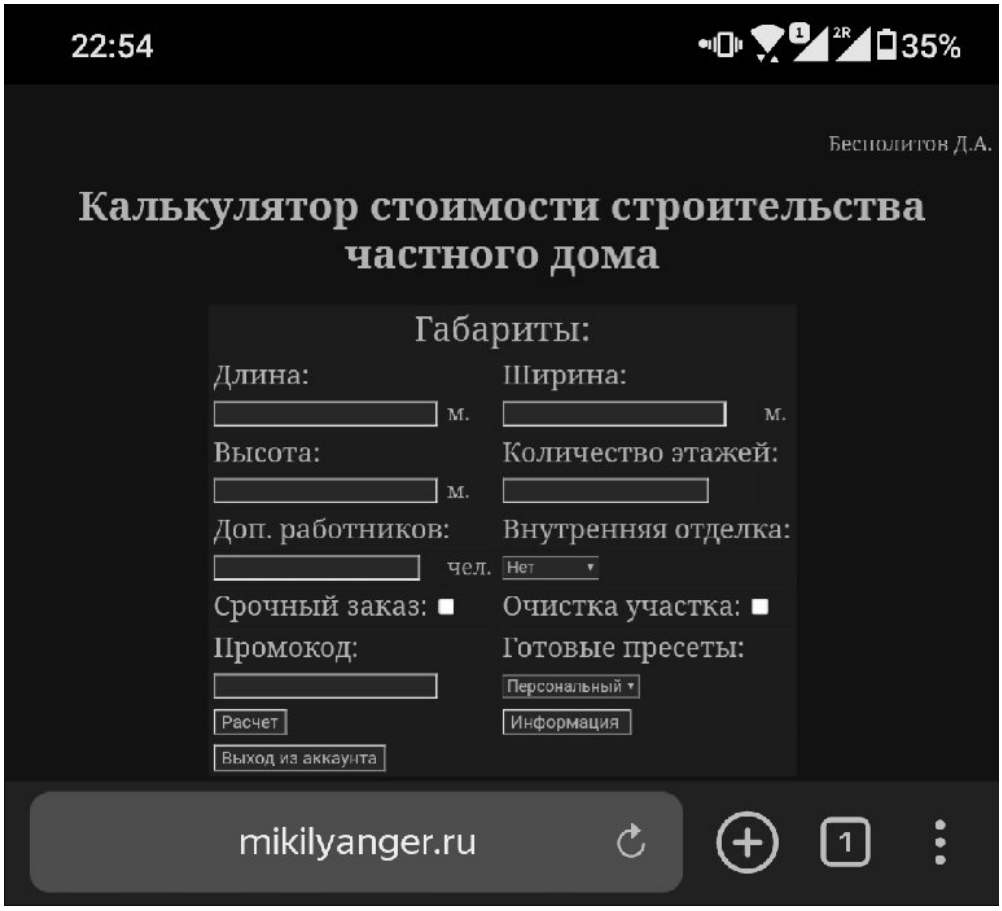


Рис. 49. Работа приложения в браузере Яндекс на Android 10

Раздел 8. Руководство пользователя программного продукта

Смотрите приложение 2.

Заключение

В данной курсовой работе были развиты способности программирования на объектно-ориентированном языке Java, настройки и использования среды Eclipse, создания программных прототипов решения прикладных задач, разработка веб-приложения, развертывание приложения на веб-сервисах, проведение тестирования компонентов программного обеспечения информационных систем, а также освоение навыка совместной разработки программного обеспечения с использованием контроля версий Git.

ПРИЛОЖЕНИЕ 1

УТВЕРЖДЕН
АСУ-1304.300012.000-ЛУ

КАЛЬКУЛЯТОР СТОИМОСТИ СТРОИТЕЛЬСТВА ЧАСТНОГО ДОМА

Техническое задание

АСУ-1304.300012.000

Листов 9

Подп. и дата	*Инв.№дубл.*	*Взам.инв.№*	*Подп. и дата*	*Инв.№подл.*	

Уфа – 2021

СОГЛАСОВАНО		УТВЕРЖДАЮ	
Преподаватель Казанцев А.В.		Преподаватель Казанцев А.В.	
Личная подпись	Расшифровка подписи	Личная подпись	Расшифровка подписи

КАЛЬКУЛЯТОР СТОИМОСТИ СТРОИТЕЛЬСТВА ЧАСТНОГО ДОМА

Техническое задание

ЛИСТ УТВЕРЖДЕНИЯ

АСУ-1304.300012.000-ЛУ

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

Разработчики

Модератор

Личная подпись	Расшифровка подписи

Разработчик 1

Личная подпись	Расшифровка подписи

Разработчик 2

Личная подпись	Расшифровка подписи

Уфа – 2021

1. Введение

1.1. Наименование создаваемого программного обеспечения – «Калькулятор стоимости строительства частного дома».

Предметная область – строительство частного дома. Частные дома – индивидуальный городской или сельский малоэтажный (обычно двухэтажный) жилой дом с небольшим участком прилегающей земли для постоянного или временного проживания.

Строительство – создание (возведение) зданий, строений и сооружений.

В широком смысле к строительству (как отрасли) также относят капитальный и текущий ремонт зданий и сооружений, а ещё их реконструкцию, реставрацию и реновацию.

Функции строительства:

1) Создание новых сооружений.

2) Расширение – строительство новых сооружений и увеличение производственных площадей, объектов на территории действующих предприятий.

3) Реконструкция – переустройство существующих объектов основного назначения без расширения этих объектов.

В строительстве подразумевается множество деталей. Туда включается закупка материала, правильный выбор территории и ее подготовка, а также многое другое.

– Первым делом выполняется планирование постройки дома. Выбор места, размера материалов и так далее.

– Далее идет закупка материалов, доставка до места стройки, поиск рабочих и подготовка территории.

– После этого идет постройка дома, а именно заливка фундамента, постройка основы и каркаса крыши.

– Когда основа дома готова то начинают заниматься отделкой. Она бывает «черновая» - установка дверей, окон и оштукатуривание. Так же может быть «под ключ» - уже полностью предназначенная для жилья квартира

– Ну и финальный этап - уборка после строительства.

Наше программное обеспечение будет писаться для строительной компании, которая занимается исключительно строительством частных домов. Использоваться же данное программное обеспечение будет клиентами этой компании.

Также в нашей программе будет присутствовать возможность компанией изменить созданные шаблоны через специальный аккаунт администратора приложения, в который можно попасть через форму входа.

Сама компания выполняет различные виды работ при строительстве частных домов: кирпичная кладка, устройство фундамента, отделочные работы.

Также эта компания предлагает услуги по строительству жилых сооружений на территории Республики Башкортостан. Компания предлагает такие услуги:

– разработка проектов;

– строительство домов с разными видами отделки;

– готовые шаблоны домов (зимний и летний варианты);

– очистка участка и другие услуги.

2. Основания для разработки

2.1. Основанием для разработки является выполнение курсовой работы на тему «Разработка кроссплатформенного программного продукта на языке JAVA с использованием системы контроля версий» по дисциплине «Информационные системы» по направлению «Прикладная информатика» в ФГБОУ ВО «УГАТУ».

3. Назначение разработки

3.1. Функциональное назначение программы

Функциональным назначением программы является представление информации о стоимости строительства частного дома.

3.2. Эксплуатационное назначение программы

Программа будет использоваться сотрудниками фирмы и клиентами для расчета цены строительства. Также сотрудникам фирмы будут доступны изменения системы расчетов, если это необходимо.

4. Требования к программе

4.1. Требования к разрабатываемому кроссплатформенному программного продукту

4.1.1 Требования к функциональным характеристикам

Входные данные: параметры дома, дополнительные услуги.

Выходные данные: стоимость дома.

4.1.2 Требования к надежности

Для обеспечения надежного функционирования требуется бесперебойный доступ к электричеству и интернету.

4.1.3 Условия эксплуатации

Программа запускается на устройстве с доступом в интернет, посредством перехода по нужной ссылке в любом поддерживаем браузере.

4.1.4 Требования к составу и параметрам технических средств

Требования к серверу:

CPU: 1 ядро 2,5 ГГц

RAM: 1 ГБ

Диск: 10 ГБ

4.1.5 Требования к информационной и программной совместимости

Программа разрабатывается на языке программирования JAVA. Полный список в приложении 1.1.

4.1.6 Требования к маркировке и упаковке

Специальных требований к маркировке и упаковке не предъявляется.

4.1.7 Требования к транспортированию и хранению

Специальных требований к транспортировке и хранению не предъявляется.

4.1.8 Специальные требования

1) Поддержка коллективной работы с исходным текстом и документацией по программному продукту.

2) Наличие интуитивно-понятного графического интерфейса для пользователей.

3) Применение и использование веб-серверной технологии, реализующей многопользовательскую работу.

4) Кроссплатформенность: работа с несколькими аппаратными платформами или операционными системами, а именно: MS Windows 10, Linux Ubuntu 20.04, Android 7.0 Nougat, Android 10

5. Требования к программной документации

5.1. Необходимая программная документация для создания программного обеспечения:

1) Техническое задание;
2) Руководство оператора (пользователя);
3) Программа и методика испытаний;
4) Текст программы.

6. Технико-экономические показатели

6.1. Программа «Калькулятор стоимости строительства частного дома» будет использоваться для облегчения расчета цены постройки частного дома, что позволит клиенту быстрее выбрать нужные характеристики дома, исходя из его бюджета, а также клиенту не нужно будет тратить время на переговоры со строительной компанией. Преимущество нашего калькулятор перед другим в том, что он позволяет клиентам выбирать уже готовые пресеты, что облегчает клиентам выбор параметров дома, также ПО имеет упрощенный для понимания интерфейс. Мы считаем, что наше программное обеспечение экономически выгодно по двум причинам: скорость выбора характеристик дома увеличится, за счет созданных пресетов, а второе – быстрое и легкое изменение системы расчетов ПО.

7. Стадии и этапы разработки

7.1. Стадии и этапы разработки создаваемого продукта представлены в таблице 1.

Таблица 1

Наименование этапа работ	**Трудоемкость выполнения, час**	**Процент к общей трудоемкости выполнения**	**Срок предъявления консультанту**
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

Продолжение таблицы 1

Раздел 3. Настройка среды разработки для операционных систем семейств Windows и Linux	10	10%	31 неделя
Раздел 4. Настойка 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. 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. Демонстрация работы приложения на разных операционных системах.
6. Знание своей зоны ответственности

ПРИЛОЖЕНИЕ 1.1

Необходимые программные средства

Язык программирования JAVA.

Среда разработки Eclipse IDE for Java EE Developers.

Распределённая система управления версиями Git.

Сборщик проектов Apache Maven.

Операционная система Linux Ubuntu 20.04.

Фреймворк для тестирования программного обеспечения JUnit.

ПРИЛОЖЕНИЕ 2

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

РУКОВОДСТВО ПОЛЬЗОВАТЕЛЯ

Аннотация

Данный документ является руководством пользователя для калькулятора по расчету строительства частного дома. Документ разработан в рамках выполнения курсовой работы по теме «Разработка кроссплатформенного программного продукта на языке JAVA с использованием системы контроля версий»

Содержание

1 Назначение программы 4
1.1 Область применения 4
1.2 Краткое описание возможностей 4
1.3 Уровень подготовки пользователя 4
2 Условия выполнения программы 5
2.1 Первоначальная настройка 5
3 Выполнение программы 6
3.1 Регистрация пользователя 6
3.2 Авторизация пользователя 8
3.3 Порядок проверки работоспособности 8
3.4 Описание интерфейса 9
3.5 Описание действий пользователя 9
3.6 Описание действий администратора 10
3.7 Просмотр результатов 11
4 Сообщения оператору 12
4.1 Ошибки 12

1 Назначение программы

Назначение данного калькулятора в том, чтобы считать для пользователя стоимость постройки частного дома.

1.1 Область применения

Калькулятор может использоваться сотрудниками фирмы и клиентами для расчета цены строительства. Также сотрудникам фирмы будут доступны изменения системы расчетов, если это необходимо.

1.2 Краткое описание возможностей

Данный калькулятор способен рассчитать стоимость строительства частного дома. Так же присутствует возможность выбирать готовые пресеты.

1.3 Уровень подготовки пользователя

Для работы с калькулятором пользователь должен иметь базовые навыки использования браузера.

2 Условия выполнения программы

2.1 Первоначальная настройка

Для работы с калькулятором не требуется установка дополнительного программного обеспечения. Взаимодействие пользователя с калькулятором осуществляется посредством веб-интерфейса.

До начала работы необходимо:

− Ознакомиться с настоящим руководством пользователя.

− Получить доступ к веб-приложению.

− Получить технические средства доступа к соответствующим информационным ресурсам.

3 Выполнение программы

3.1 Регистрация пользователя

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

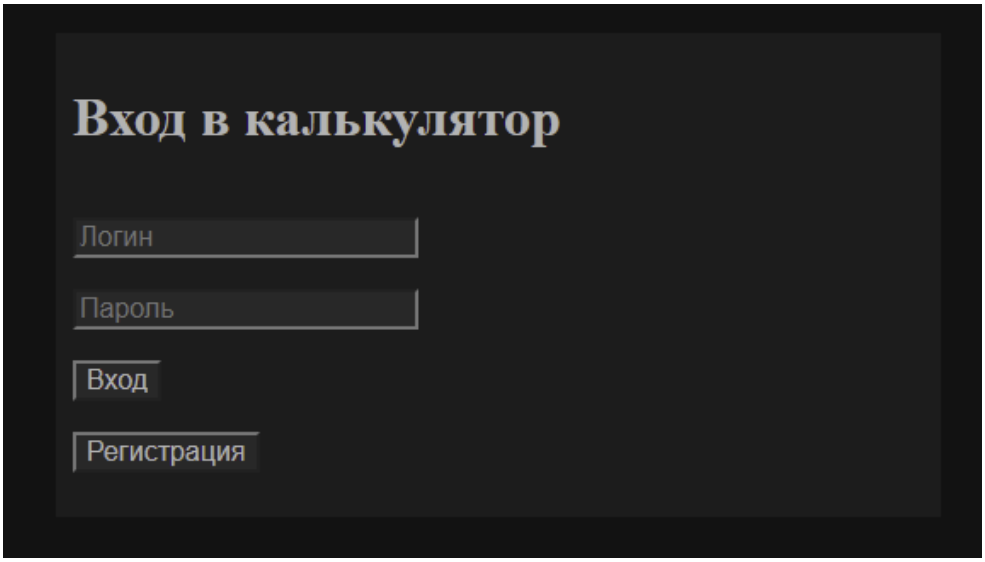


Рис. 1. Окно авторизация

Если вход производится впервые, то требуется нажать кнопку «регистрация».

Для регистрации следует указать личные данные (фамилию, первые буквы имени и отчества), придумать логин и пароль, так же следует подтвердить пароль его повторным написанием (Рис. 2). И далее нажать кнопку «Регистрация». Если же пароли не будут совпадать, то будет выведена надпись: «Пароли не совпадают» (Рис. 3).

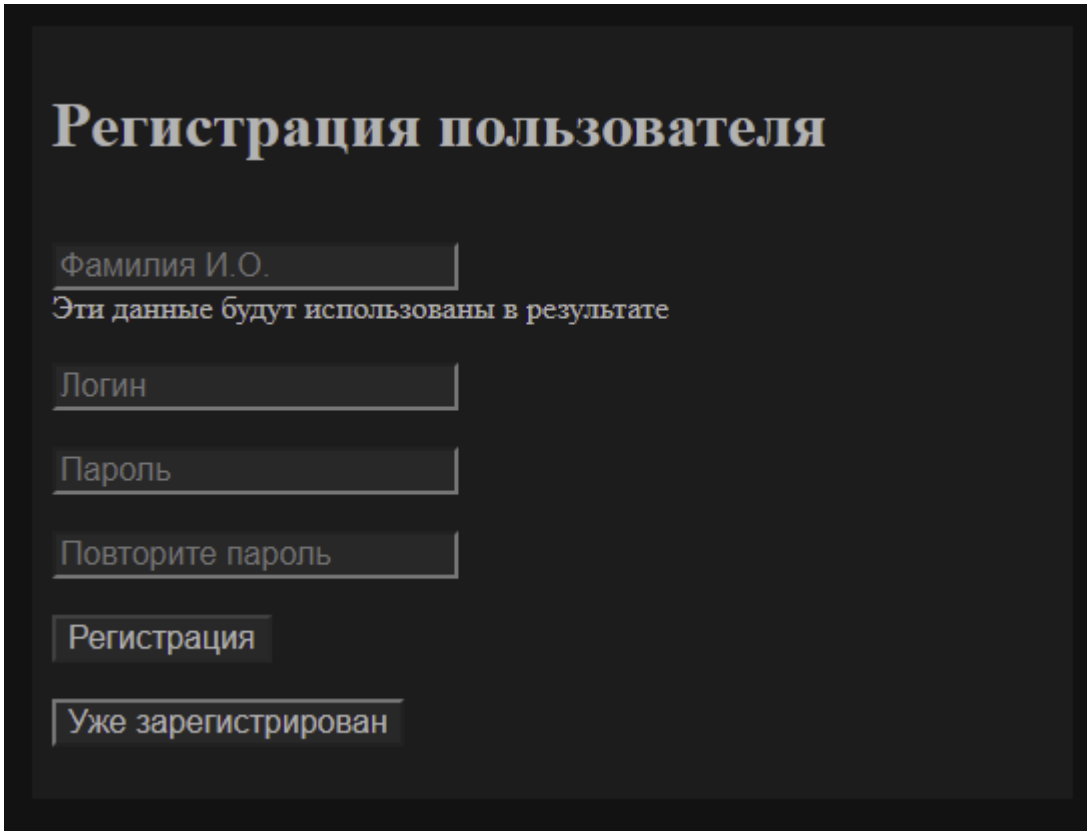


Рис. 1. Регистрирование нового пользователя

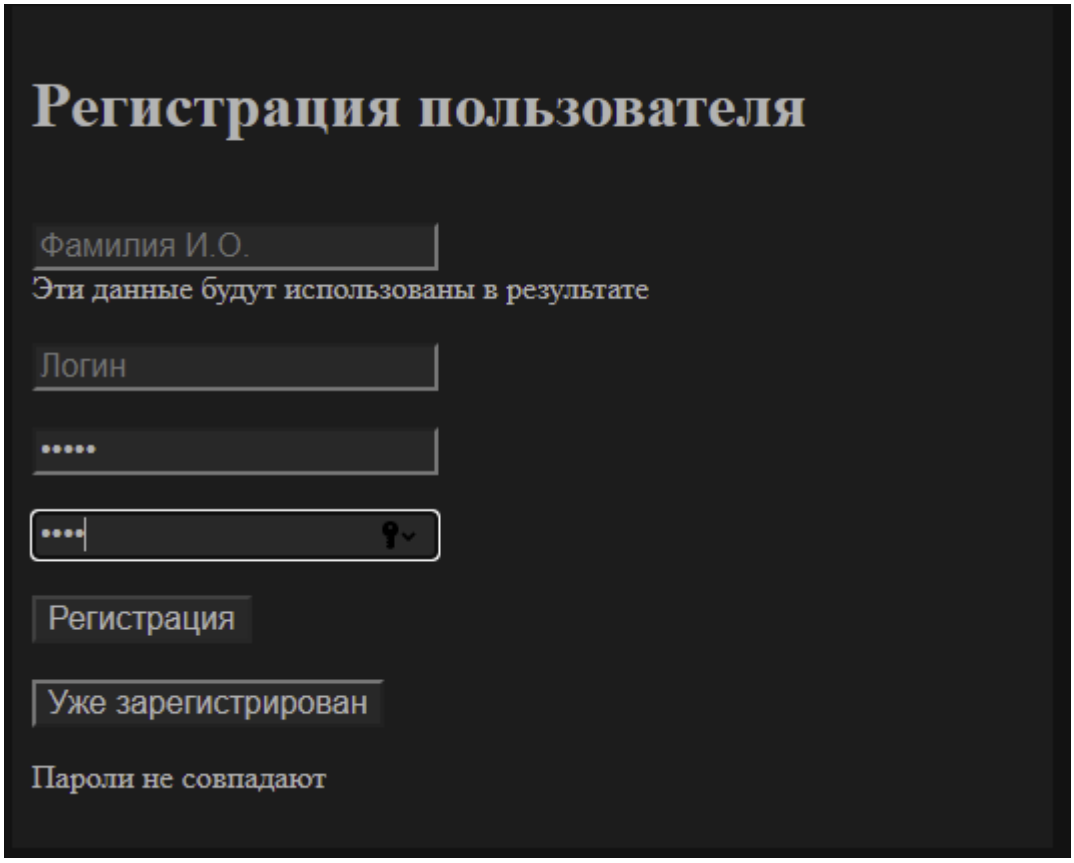


Рис. 2. Не совпадение паролей

3.2 Авторизация пользователя

По завершении регистрации, либо после ввода адреса в рабочем окне браузера появится форма авторизации. Для входа пользователь должен воспользоваться своими данными - логином и паролем.

При вводе логина и пароля стоит убедится в том, что не нажата клавиша CapsLock и выставлена верная раскладка клавиатуры. Так же для безопасности пароль скрыт.

При не корректном вводе данных выдаёт следующее сообщение (Рис. 4):

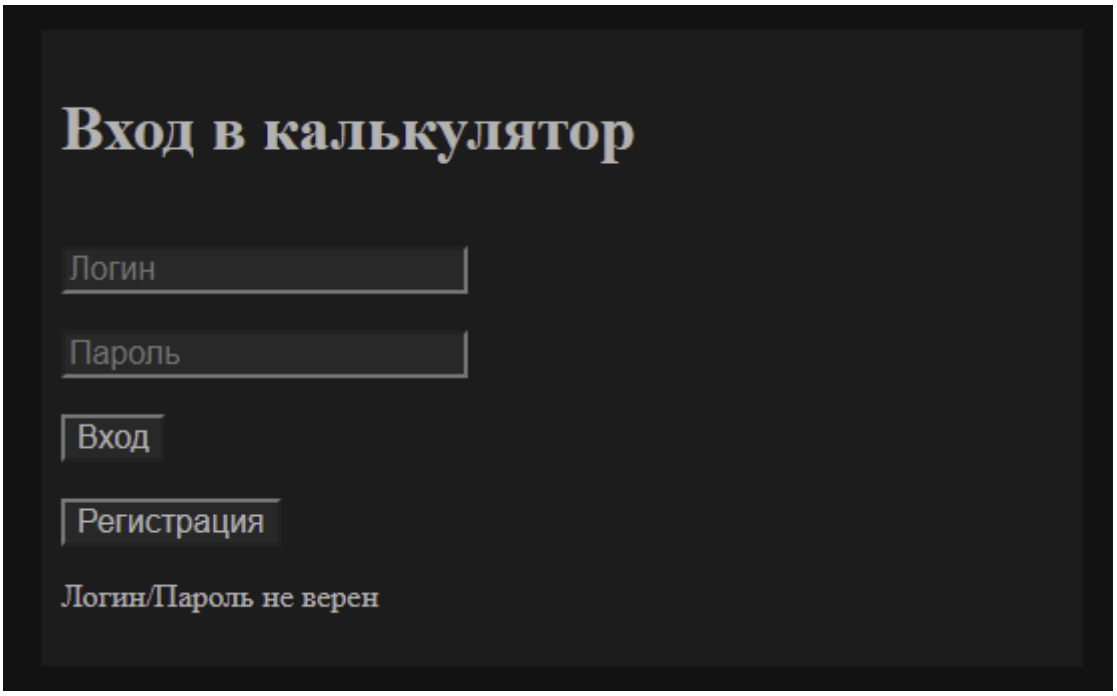


Рис. 4. Неверно введенные данные

В случае если все данные введены верно, то произойдет вход в калькулятор.

3.3 Порядок проверки работоспособности

После прохождения процедуры авторизации на экране монитора в окне браузера должна появиться главная страница калькулятора (Рис. 5). В случае, если этого не произошло, то следует обратится к администратору.

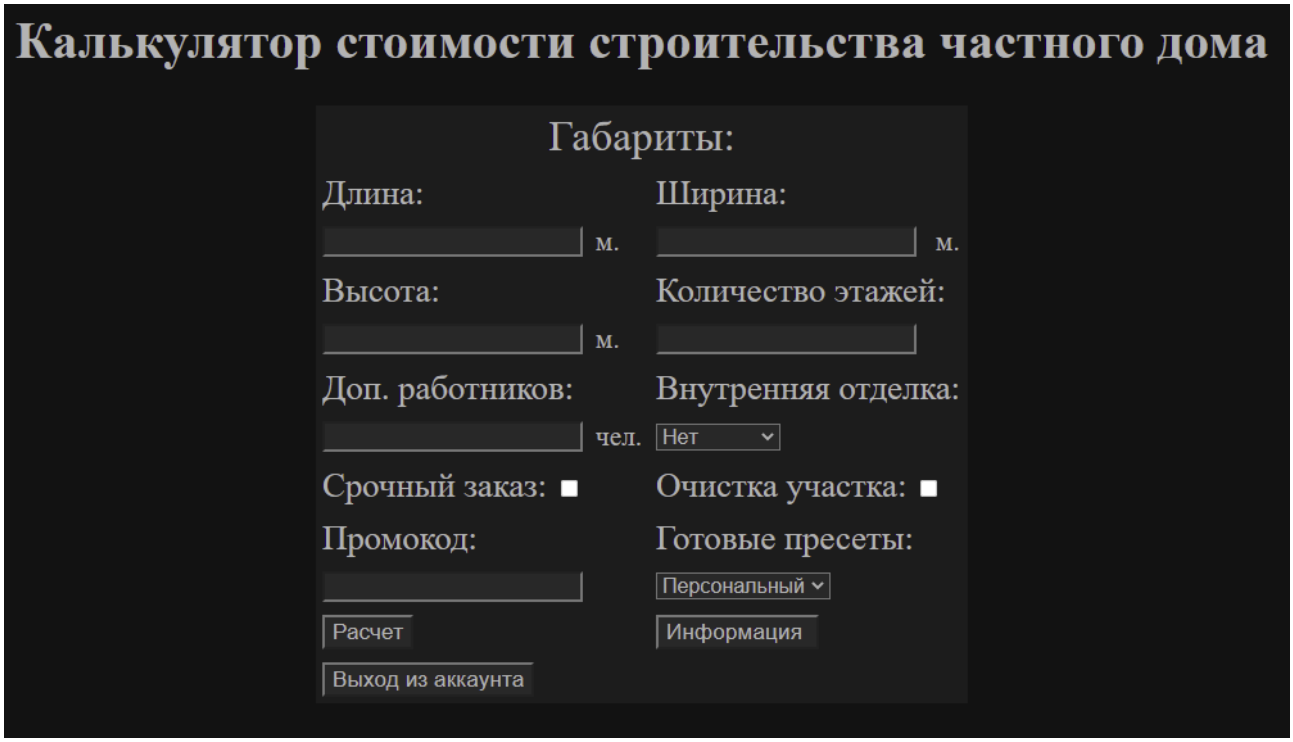


Рис. 3. Главное окно калькулятора

3.4 Описание интерфейса

Данный интерфейс предназначен для обычного пользователя. Представлен он полями для заполнения, выпадающими списками и флаговыми кнопками, так же присутствуют кнопки для расчета, выхода или вывода информации.

Поля «Длина», «Ширина», «Высота», «Количество этажей», «Доп. работников» (Если требуются) и «Промокод» (Если присутствует) заполняются пользователем вручную, также справа у данных полей присутствую стрелочки для увеличения или уменьшения нужных данных.

Поля «Внутренняя отделка» и «Готовые пресеты» выбираются путем нажатия на данное поле и выбора из всех вариантов.

Поля «Срочный заказ» и «Очистка участка» выбираются путем нажатия на них.

3.5 Описание действий пользователя

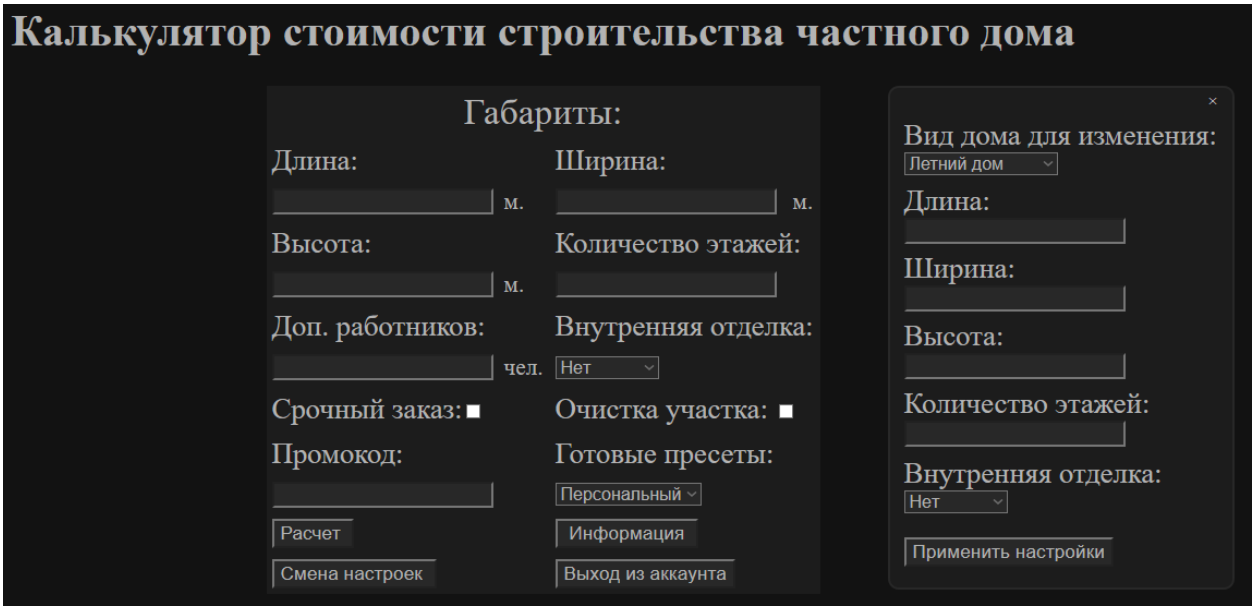
Чтобы воспользоваться калькулятором требуется ввести такие поля как: «Длина», «Ширина», «Высота», «Количество этажей». Они являются

обязательными для заполнения. Остальные поля можно оставить пустыми, если пользователю дополнительно ничего не требуется.

После ввода всех данных требуется нажать «Расчет». Если же функции данного калькулятора не требуются или надо сменить пользователя, то можно нажать «Выход из аккаунта». Если же интересно узнать информацию о тех, кто создавал этот калькулятор, то следует нажать «Информация».

3.6 Описание действий администратора

В данном окне можно производить те же действия, что описаны в пункте «Описание действий пользователя». Главное отличие данного аккаунта от аккаунта пользователя — это то, что здесь доступна кнопка



«Смена настроек» (Рис. 6).

Рис. 6. Окно администратора

При нажатии данной кнопки появится дополнительное окно, где будет возможность изменить настройки пресетов. При нажатии на выпадающий список появятся виды пресетов, которые можно изменить. «Сброс "Летний"» и «Сброс "Зимний"» отвечают за сброс к базовым параметрам данных пресетов. После выбора пресета для изменений вписываются новые длина, ширина, высота, кол-во этажей и из выпадающего списка выбирается вид

отделки, и когда уже все параметры выбраны нажимается кнопка «Применить настройки» для сохранения параметров, выставленных администратором.

3.7 Просмотр результатов

После нажатия кнопки «Расчет» появится следующие окно (Рис. 7)

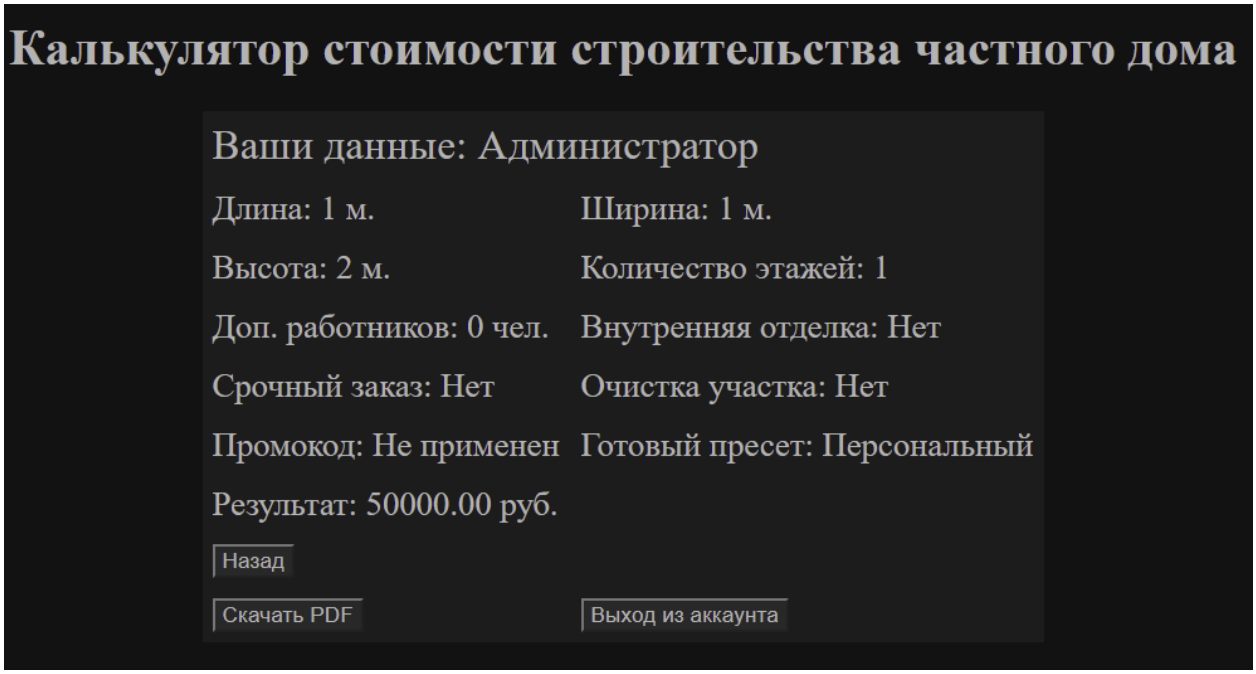


Рис. 7. Результат

Здесь показаны данные, которые вы вводили, и сам результат с посчитанной суммой за заданные параметры дома.

Так же при нажатии кнопки «Скачать PDF» на ваш компьютер будет скачан файл с расширением .pdf, в котором будет указано тоже самое.

4 Сообщения оператору

4.1 Ошибки

В случае возникновения ошибки при вводе данных пользователь получит сообщение с указаниями по её устранению (Рис. 8-10).

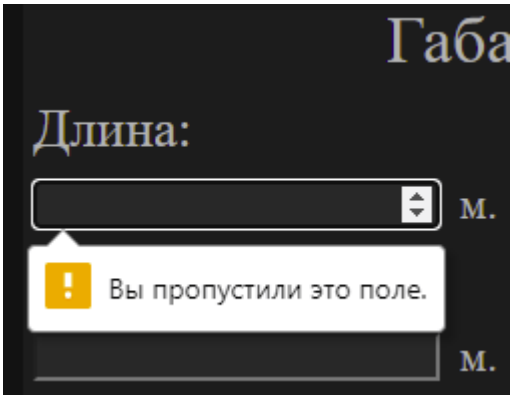


Рис. 8. Ошибка с пустым полем

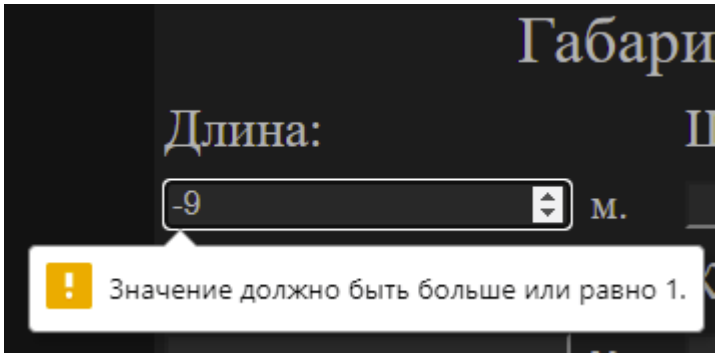


Рис. 9. Ошибка с отрицательным значением

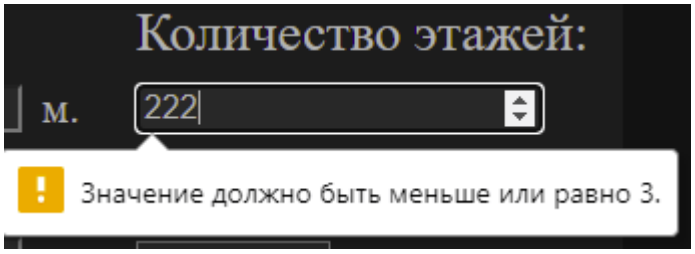


Рис. 10. Ошибка с превышением диапазона

Лист регистрации изменений

Изм.	Номера листов (страниц)				Всего листов (страниц) в доку-менте	Номер доку-мента	Входящий номер сопроводи-тельного документа и дата	Подпись	Дата
	изменен-ных	заменен-ных	новых	аннулиро-ванных					

Подпись и дата | Инв. № дубл. | Взам. инв. № | Подпись и дата | Инв. № подл.

ПРИЛОЖЕНИЕ 3

LogInReg.java

```
package Package;

import java.io.*;
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 java.sql.Connection;
import java.sql.DriverManager;
import java.sql.ResultSet;
import java.sql.SQLException;
import java.sql.Statement;

@WebServlet(name="Log", urlPatterns="/LogCalc")

public class LogInReg extends HttpServlet {

    static byte go = 1;

    protected void doPost(HttpServletRequest request, HttpServletResponse response)
throws ServletException, IOException {
        request.setCharacterEncoding("UTF-8");
        response.setCharacterEncoding("UTF-8");
        RequestCalc Calc = RequestCalc.fromRequestParameters(request);
        Calc.setAsRequestAttributesAndCalculate(request);

        switch (go){ //На какую страницу будет переход при нажатии на кнопку типа
submit
            case 0: request.getRequestDispatcher("/LogIn.jsp").forward(request, response);
break;
            case 1: request.getRequestDispatcher("/Form.jsp").forward(request, response);
break;
            case 2: request.getRequestDispatcher("/Reg.jsp").forward(request, response);
break;
            case 3: request.getRequestDispatcher("/FormAdm.jsp").forward(request,
response); break;
        }
    }

    private static class RequestCalc {
```

```
    private final String log;
    private final String pas;
    private final String page;
    private final String fio;

    private RequestCalc (String first, String second,String page,String fio) {
        this.log = first;
        this.pas = second;
        this.page = page;
        this.fio = fio;
    }

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

    public void setAsRequestAttributesAndCalculate(HttpServletRequest request) {
        Connection con =null;
        Statement stmt=null;
        ResultSet rs=null;
        ResultSet rs1=null;
        String arr[] ={"length","width","height","floor","otdelka"};
        if (page.equals("LogIn")) {
            try {
                // opening database connection to MySQL server
                con = getConnection();
                // getting Statement object to execute query
                stmt = con.createStatement();
                //Проверка соответствия логина и пароля
                String query = "SELECT id from accounts where login like '"+log+"' and pass
like '"+pas+"'";

                rs = stmt.executeQuery(query);

                if (rs.next()){ //Если такой элемент существует
                    go=1;
                    String query1;
```

```
                if (log.equals("admin")) {
                    go = 3;
                }

                for(int i =0;i<5;i++){
                    query1 = "SELECT "+arr[i]+" FROM settings WHERE
type='uptWinter';";
                    rs1 = stmt.executeQuery(query1);
                    if (rs1.next()){
                    request.setAttribute("winter"+i,rs1.getString(arr[i]));
                    }
                    query1 = "SELECT "+arr[i]+" FROM settings WHERE
type='uptSummer';";
                    rs1 = stmt.executeQuery(query1);
                    if (rs1.next()){
                        request.setAttribute("summer"+i,rs1.getString(arr[i]));
                    }
                }
                query1 = "SELECT fio FROM accounts WHERE login='"+log+"';";
                rs1 = stmt.executeQuery(query1);
                if (rs1.next()){
                    request.setAttribute("fio",rs1.getString("fio"));
                }
            }
            else{
                request.setAttribute("result", "Логин/Пароль не верен");
                go = 0;
            }

        }catch (NullPointerException sqlNPE){
            request.setAttribute("result","Error1: "+sqlNPE.getLocalizedMessage());
            go = 0;
        }
        catch (SQLException sqlEx) {
            request.setAttribute("result","Error2: "+sqlEx.getSQLState());
            go = 0;
        }
        finally {
            //close connection ,stmt and resultset
            try {con.close(); } catch (NullPointerException sqn){ /**/}
catch(SQLException se) { /* .*/ }
            try {stmt.close(); } catch (NullPointerException sqn){ /**/}
catch(SQLException se) { /* .*/ }
```

```
                try {rs.close();  } catch (NullPointerException sqn){ /**/}
catch(SQLException se) { /* .*/ }
                try {rs1.close();  } catch (NullPointerException sqn){ /**/}
catch(SQLException se) { /* .*/ }
            }

        }

        else{
            go = 0;
            try {
                // opening database connection to MySQL server
                 con = getConnection();
                // getting Statement object to execute query
                stmt = con.createStatement();
                //Проверка на существование логина
                String query = "SELECT id from accounts where login like '"+log+"'";
                rs = stmt.executeQuery(query);
                if (!rs.next()){
                    //Создание нового пользователя
                    query = "INSERT INTO accounts (login, pass, fio) VALUES ('"+log+'',
'"+pas+'', '"+fio+'');";
                    stmt = con.createStatement();
                    stmt.executeUpdate(query);
                }
                else{
                    request.setAttribute("result", "Такой логин уже существует");
                    go = 2;
                }

            }catch (NullPointerException sqlNPE){
                request.setAttribute("result","1Error: "+sqlNPE.getMessage());
                go = 2;
            }
            catch (SQLException sqlEx) {
                request.setAttribute("result","2Error: "+sqlEx.getSQLState());
                go = 2;
            } finally {
                //close connection ,stmt and resultset
                try {con.close(); } catch (NullPointerException sqn){ /**/}
catch(SQLException se) { /* .*/ }
                try {stmt.close(); } catch (NullPointerException sqn){ /**/}
catch(SQLException se) { /* .*/ }
```

```
                try {rs.close();  } catch (NullPointerException sqn){ /**/}
catch(SQLException se) { /* .*/ }
            }

        }
    }
  }

  public static Connection getConnection() { //Connection to DataBase
    String driver = "com.mysql.cj.jdbc.Driver";
    final String url = "jdbc:mysql://remotemysql.com:3306/smszCuaCce";
    final String user = "smszCuaCce";
    final String password = "fjrxusR9mP";
    try {
      Class.forName(driver);
      java.sql.Connection con = DriverManager.getConnection(url,user,password);
      return con;

    } catch (ClassNotFoundException | SQLException e) {
      System.out.println("Exception " + e);
      return null;
    }
  }

}
```

ПРИЛОЖЕНИЕ 4

Main.java

```
package Package;
import javax.swing.*;
import java.awt.event.ActionEvent;
import java.awt.event.ActionListener;
import java.io.File;
import java.io.FileInputStream;
import java.io.FileWriter;
import java.io.IOException;
import java.util.Scanner;

import Package.Calculator.*;



public class Main {
	public static void main(String[] args) {
		LogInReg l = new LogInReg();
		//l.logWin();
	}


}
```

ПРИЛОЖЕНИЕ 5

LogIn.jsp

```
<%@ page language="java" contentType="text/html; charset=UTF-8"
pageEncoding="UTF-8"%>
<!DOCTYPE HTML>
<html>
<head>
<meta charset="utf-8">
<title>Вход в калькулятор</title>
<style>
body{
        background: #121212;
}
  .blok-center {
   width: 500px; /* Ширина блока */
   height: auto; /* Высота блока */
   margin: auto; /* Отступ от блока */
    /* Цвет фона блока */
  background: #1c1c1c;
   padding: 10px; /* Отступ внутри блока */
  }
  .text{
  color: #b3b3b3;
  }
  .combo{
  background-color: #282828;
  color: #b3b3b3;
  font-size : 12pt;
  }
</style>
</head>
<body>
<div class="blok-center">
<h1 class="text">Вход в калькулятор</h1>
<form action="${pageContext.request.contextPath}/LogCalc" method="post">
<input type="hidden" name="page" id="LogIn" value="LogIn"/>
<input type="hidden" name="fio" id="fio" value=""/>
<br>
<input type="text" placeholder="Логин" class="combo" name="login" id="login"
value="${login}" required title="Введите ваш логин">
<!--<strong>м</strong> pattern="(^[1-9](\d*)?([.]\d+)?)|(^0[.](\d*)?(?=[1-9])\d+)"-->
</br>
<br>
```

```
<input  class="combo" placeholder="Пароль" type="password" name="pass" id="pass"
value="${pass}" required  title="Введите ваш пароль">
<!--<strong>м</strong> pattern="(^[1-9](\d*)?([.]\d+)?)|(^0[.](\d*)?(?=[1-9])\d+)"-->
</br>
<br><input  class="combo" type="submit" name="sign" value="Вход"></br>
<br><input class="combo" type="button" name="gtreg" value="Регистрация"
onclick="location.href='Reg.jsp'"></br>
<p class="text">${result}</p>
</form>
</div>
</body>
</html>
```

ПРИЛОЖЕНИЕ 6

Reg.jsp

```
<%@ page language="java" contentType="text/html; charset=UTF-8"
pageEncoding="UTF-8"%>
<!DOCTYPE HTML>
<html>
<head>
<meta charset="utf-8">
<title>Регистрация</title>
<style>
body{
        background: #121212;
}
  .blok-center {
   width: 500px; /* Ширина блока */
   height: auto; /* Высота блока */
   margin: auto; /* Отступ от блока */
    /* Цвет фона блока */
  background: #1c1c1c;
   padding: 10px; /* Отступ внутри блока */
  }
  .text{
  color: #b3b3b3;
  }
  .combo{
  background-color: #282828;
  color: #b3b3b3;
  font-size : 12pt;
  }
</style>
</head>
<body>
<div class="blok-center">
<h1 class="text">Регистрация пользователя</h1>
<form action="${pageContext.request.contextPath}/LogCalc" method="post">
<input type="hidden" name="page" id="Reg" value="Reg"/>
<br>
<input class="combo" placeholder="Фамилия И.О." type="text" name="fio" id="fio"
minlength="4" value="${fio}" required  title="Введите ваши ФИО"><nobr class="text">
Эти данные будут использованы в результате</nobr>
</br>
<br>
```

```
<input class="combo" placeholder="Логин" type="text" name="login" id="login"
value="${login}" required title="Введите логин">
<!--<strong>м</strong> pattern="(^[1-9](\d*)?([.]\d+)?)|(^0[.](\d*)?(?=[1-9])\d+)"-->
</br>
<br>
<input class="combo" placeholder="Пароль" type="password" name="pass" id="pass"
value="${pass}" required  title="Введите пароль">
<!--<strong>м</strong> pattern="(^[1-9](\d*)?([.]\d+)?)|(^0[.](\d*)?(?=[1-9])\d+)"-->
</br>
<br>
<input class="combo" placeholder="Повторите пароль" type="password" name="pass1"
id="pass1" required  title="Повторно введите пароль" oninput="passR()">
</br>
<br><input class="combo" type="submit" name="sign" id="sign" value="Регистрация"
disabled="true"></br>
<br><input class="combo" type="button" name="gtlog" value="Уже зарегистрирован"
onclick="location.href='LogIn.jsp'"></br>
<p class="text" id="result" >${result}</p>
<script>
function passR(){
var pass = document.getElementById('pass').value;
var pass1 = document.getElementById('pass1').value;
if(pass!==pass1){
	document.getElementById('result').innerHTML = 'Пароли не совпадают';
	document.getElementById('sign').disabled = true;
}
else{
	document.getElementById('result').innerHTML = '';
	document.getElementById('sign').disabled = false;
}
}
</script>
</form>
</div>
</body>
</html>
```

ПРИЛОЖЕНИЕ 7

Form.jsp

```
<%@ page language="java" contentType="text/html; charset=UTF-8"
pageEncoding="UTF-8"%>
<!DOCTYPE HTML>
<html>
<head>
<meta charset="utf-8">
<title>Калькулятор</title>
<style>
body{
  background: #121212;
}
table {
    margin: auto;
  display: inline-table
  text-align: center;
   }
   td {
     background: #1c1c1c;

   }
   input{
    text-align: left;
   }
   p {
    font-size: 20pt; /* Размер шрифта в пунктах */
    background: #1c1c1c;
    text-align: left;
   }
   .cent{
    color: #b3b3b3;
  font-size: 25pt;
   }
   .text{
   color: #b3b3b3;
   }
   .combo{
   background-color: #282828;
   color: #b3b3b3;
   font-size : 12pt;
   }
   #okno {
```

```
        width: auto; /* Ширина и высота окна*/
        height: auto;
        padding: 0px 10px;
        border: 2px solid #282828;
    display: none;
        background: #1c1c1c;
        position: absolute;
        top: auto;
    border-radius: 10px;
        right: 34.5%;
        margin: 10px;
      }
     #okno:target {display: block;}

</style>
</head>
<body onload="onload();">
  <form action="${pageContext.request.contextPath}/Calculator" method="post">
    <input type="hidden" name="lenghtwin" id="winter0" value="${winter0}"/>
    <input type="hidden" name="widthwin" id="winter1" value="${winter1}"/>
    <input type="hidden" name="heightwin" id="winter2" value="${winter2}"/>
    <input type="hidden" name="floorwin" id="winter3" value="${winter3}"/>
    <input type="hidden" name="otdelkawin" id="winter4" value="${winter4}"/>
    <input type="hidden" name="lenghtsum" id="summer0" value="${summer0}"/>
    <input type="hidden" name="widthsum" id="summer1" value="${summer1}"/>
    <input type="hidden" name="heightsum" id="summer2" value="${summer2}"/>
    <input type="hidden" name="floorsum" id="summer3" value="${summer3}"/>
    <input type="hidden" name="otdelkasum" id="summer4" value="${summer4}"/>
    <input type="hidden" name="fio" id="fio" value="${fio}"/>
    <p style="font-size : 15pt; color: #b3b3b3; background: #121212; text-align: right"
>${fio}</p>

  <div style="position: relative; text-align:center;">
    <h1 style="font-size : 30pt; color: #b3b3b3; " >Калькулятор стоимости
строительства частного дома<h1>
  <div style="position: relative; align:center; display:inline-table;">

  <table width=auto height=auto cellspacing="0" cellpadding="5" cols = "4"
style="display: inline-table;">
    <tr>
      <td colspan="4" align="center" class="cent">Габариты:</td>
    </tr>
    <tr>
```

```
      <td colspan="2"><p align="left"  class="text">Длина:</p></td>
      <td colspan="2"><p align="left" class="text">Ширина:</p></td>
    </tr>
    <tr>
      <td>
        <input type="number" autocomplete="off" required min="1" step="0.1" id = "length"
name="length" value= "${length}" size="20" class="combo">
      </td>
      <td>
        <p align="left"  class="text" style="font-size: 16pt;">м.</p>
      </td>
      <td>
        <input type="number" autocomplete="off" required min="1" step="0.1" id = "width"
name="width"  value= "${width}" size="20"class="combo">
      </td>
      <td>
        <p align="left"  class="text" style="font-size: 16pt;">м.</p>
      </td>
    </tr>
    <tr>
      <td colspan="2">
        <p align="left" class="text">Высота:</p>
      </td>
      <td colspan="2">
        <p align="left" class="text">Количество этажей:</p>
      </td>
    </tr>
    <tr>
      <td>
        <input type="number" autocomplete="off" required min="2" step="0.1" id = "height"
name="height" value= "${height}" size="20"class="combo">
      </td>
      <td>
        <p align="left"  class="text" style="font-size: 16pt;">м.</p>
      </td>
      <td>
        <input type="number" autocomplete="off" required min="1" max="3" step="1" id =
"floor" name="floor"  value= "${floor}" size="20"class="combo" style="width: 203px">
      </td>
      <td colspan="2">
        <p align="left"  class="text" style="font-size: 16pt;"></p>
      </td>
    </tr>
```

```
<tr>
  <td colspan="2">
    <p align="left" class="text">Доп. работников: </p>
  </td>
  <td colspan="2">
    <p align="left" class="text">Внутренняя отделка:</p>
  </td>
</tr>
<tr>
  <td>
    <input type="number" autocomplete="off" min="0" max="5" step="1"
name="worker"  value= "${worker}" size="20"class="combo" style="width: 203px">
  </td>
  <td>
    <p align="left"  class="text" style="font-size: 16pt;">чел.</p>
  </td>
  <td colspan="2">
    <select id = "otdelka" name="otdelka" value= "${otdelka}" class="combo">
      <option value="1">Нет</option>
      <option value="2">Черновая</option>
      <option value="3">Под ключ</option>
    </select>
  </td>
</tr>
<tr>
  <td colspan="2">
    <p align="left" class="text">Срочный заказ:
      <input type="checkbox"  name="order" value= "${order}"></p>
  </td>
  <td colspan="2">
    <p align="left" class="text"> Очистка участка: <input type="checkbox"
name="clearing" value= "${clearing}"></p>
  </td>
</tr>
<tr>
  <td colspan="2">
    <p align="left" class="text" placeholder="Промокод не задан">Промокод:</p>
  </td>
  <td colspan="2">
    <p align="left" class="text">Готовые пресеты:</p>
  </td>
</tr>
<tr>
```

```
      <td colspan="2">
        <input type="text" name="promo" value= "${promo}" size="20"class="combo" >
      </td>
      <td colspan="2">
        <select name="preset" id="preset" value= "${preset}" class="combo">
          <option value="1">Персональный</option>
          <option value="2">Летний дом</option>
          <option value="3">Зимний дом</option>
        </select>
      </td>
    </tr>
    <tr>
      <td colspan="2">
        <p align="left"> <input type="submit" name="calculation" value="Расчет"
class="combo" style="font-size : 13pt;" ></p>
      </td>
      <td colspan="2">
      <p align="left" >
      <!-- Открытие и закрытие окна с информацией на одну кнопку-->
      <input id="info" type="button" name="info" value="Информация " class="combo"
onclick="document.getElementById('okno').style.display='block';
        document.getElementById('info').style.display='none';
document.getElementById('example_2_2').style.display='block'; "style=" font-size : 13pt;"
>
      <input type="button" value=" Информация " class="combo" id="example_2_2"
onclick=" document.getElementById('okno').style.display='none';
      document.getElementById('example_2_2').style.display='none';
document.getElementById('info').style.display='block';" style=" font-size : 13pt;
display:none;"></p>
      </td>
    </tr>
  </form>
  <tr>
    <td colspan="4">
      <p align="left"><input type="button" name="exit" value="Выход из аккаунта"
onclick="location.href='LogIn.jsp'" class="combo" style="font-size : 13pt;">
    </td>
  </tr>
</table>
</div>
</div>

<form action="${pageContext.request.contextPath}/Admin" method="post">
```

```
  <div id="okno" style=" color: #b3b3b3;">
    <table class = "text" >
      <tr>
        <td>
          <p style="background: #1c1c1c;font-weight: normal;font-size: 18pt;text-align
:left">Разработчики: </p>
        </td>
      </tr>
  </div>
  <div align="right"><a href="#" style="text-decoration: none;text-align: right;
color:#b3b3b3;">&#215;</a>
      <tr>
        <td style="text-align :left">
          <p style="font-size: 16pt">Салихов Рашит Ниязович<br>
                                Иутин Александр Сергеевич<br>
                                Бесполитов Даниил Анатольевич</p>
        </td>
      </tr>
    </table>
  </div>
</form>
 <script>

     var s = document.getElementById('preset');
     s.addEventListener('change', setPre);
     function setPre(){
     var ch = s.value;
     if (ch=='1'){
     document.getElementById('length').readOnly = false;
     document.getElementById('width').readOnly = false;
     document.getElementById('height').readOnly = false;
     document.getElementById('floor').readOnly = false;
     document.getElementById('otdelka').readOnly = false;
     }
     if (ch=='3'){

     document.getElementById('length').value =
document.getElementById('winter0').value;
     document.getElementById('length').readOnly = true;
     document.getElementById('width').value = document.getElementById('winter1').value;
     document.getElementById('width').readOnly = true;
     document.getElementById('height').value =
document.getElementById('winter2').value;
```

```
    document.getElementById('height').readOnly = true;
    document.getElementById('floor').value = document.getElementById('winter3').value;
    document.getElementById('floor').readOnly = true;
    document.getElementById('otdelka').value =
document.getElementById('winter4').value;;
    document.getElementById('otdelka').readOnly = true;
    }
    if (ch=='2'){
    document.getElementById('length').value =
document.getElementById('summer0').value;
    document.getElementById('width').value =
document.getElementById('summer1').value;
    document.getElementById('height').value =
document.getElementById('summer2').value;
    document.getElementById('floor').value =
document.getElementById('summer3').value;
    document.getElementById('otdelka').value =
document.getElementById('summer4').value;
    document.getElementById('length').readOnly = true;
    document.getElementById('width').readOnly = true;
    document.getElementById('height').readOnly = true;
    document.getElementById('floor').readOnly = true;
    document.getElementById('otdelka').readOnly = true;
    }
    }

   function onload() {
    if (document.getElementById('fio').value.length<4){
       location.href='LogIn.jsp';
    }
    }

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

ПРИЛОЖЕНИЕ 8

FormAdm.jsp

```
<%@ page language="java" contentType="text/html; charset=UTF-8"
pageEncoding="UTF-8"%>
<!DOCTYPE HTML>
<html>
<head>
<meta charset="utf-8">
<title>Калькулятор</title>
<style>
body{
  background: #121212;
}
table {
    margin: auto;
  display: inline-table
  text-align: center;
   }
   td {
     background: #1c1c1c;

   }
   input{
    text-align: left;
   }
   p {
    font-size: 20pt; /* Размер шрифта в пунктах */
    background: #1c1c1c;
    text-align: left;
   }
   .cent{
    color: #b3b3b3;
  font-size: 25pt;
   }
   .text{
   color: #b3b3b3;
   }
   .combo{
   background-color: #282828;
   color: #b3b3b3;
   font-size : 12pt;
   }
   #okno {
```

```
        width: auto; /* Ширина и высота окна*/
        height: auto;
        padding: 0px 10px;
        border: 2px solid #282828;
    display: none;
        background: #1c1c1c;
        position: absolute;
        top: auto;
    border-radius: 10px;
        right: 34.5%;
        margin: 10px;
      }
      #okno:target {display: block;}
      #dop {
        width: auto; /* Ширина и высота окна*/
        height: auto;
    border-radius: 10px;
        text-align: left;
        padding: 0px 10px;
        border: 2px solid #282828;
    background: #1c1c1c;
        color: #1c1c1c;
        display: none;
        position: absolute;
        top: 0px;
        right: -380px;
        margin: 0px;
      }
      #dop:target {display: block;}
}
</style>
</head>
  <body onload="onload();">
  <form action="${pageContext.request.contextPath}/Calculator" method="post">
    <input type="hidden" name="lenghtwin" id="winter0" value="${winter0}"/>
    <input type="hidden" name="widthwin" id="winter1" value="${winter1}"/>
    <input type="hidden" name="heightwin" id="winter2" value="${winter2}"/>
    <input type="hidden" name="floorwin" id="winter3" value="${winter3}"/>
    <input type="hidden" name="otdelkawin" id="winter4" value="${winter4}"/>
    <input type="hidden" name="lenghtsum" id="summer0" value="${summer0}"/>
    <input type="hidden" name="widthsum" id="summer1" value="${summer1}"/>
    <input type="hidden" name="heightsum" id="summer2" value="${summer2}"/>
    <input type="hidden" name="floorsum" id="summer3" value="${summer3}"/>
```

```
    <input type="hidden" name="otdelkasum" id="summer4" value="${summer4}"/>
    <input type="hidden" name="fio" id="fio" value="${fio}"/>
    <p style="font-size : 15pt; color: #b3b3b3; background: #121212; text-align: right"
>${fio}</p>

  <div style="position: relative; text-align:center;">
    <h1 style="font-size : 30pt; color: #b3b3b3; " >Калькулятор стоимости
строительства частного дома<h1>

  <div style="position: relative; align:center; display:inline-table;">
    <table width=auto height=auto cellspacing="0" cellpadding="5" cols = "4"
style="display: inline-table;">
      <tr>
        <td colspan="4" align="center" class="cent">Габариты:</td>
      </tr>
      <tr>
        <td colspan="2"><p align="left"  class="text">Длина:</p>
        </td>
        <td colspan="2">
          <p align="left" class="text">Ширина:</p>
        </td>
      </tr>
      <tr>
        <td>
          <input type="number" autocomplete="off" required min="1" step="0.1"
max="1000" id = "length" name="length" value= "${length}" size="20"
class="combo"style="width: 203px">
        </td>
        <td>
        <p align="left"  class="text" style="font-size: 16pt;">м.</p>
        </td>
        <td>
          <input type="number" autocomplete="off" required min="1" step="0.1"
max="1000" id = "width" name="width"  value= "${width}" size="20"class="combo"
style="width: 203px">
        </td>
        <td>
          <p align="left"  class="text" style="font-size: 16pt;">м.</p>
        </td>
      </tr>
      <tr>
        <td colspan="2"><p align="left" class="text">Высота:</p>
        </td>
```

```
      <td colspan="2"><p align="left" class="text">Количество этажей:</p>
      </td>
    </tr>
    <tr>
      <td>
      <input type="number" autocomplete="off" required min="2" step="0.1" max="30" id
= "height" name="height" value= "${height}" size="20"class="combo" style="width:
203px">
      </td>
      <td>
        <p align="left"  class="text" style="font-size: 16pt;">м.</p>
      </td>
      <td>
        <input type="number" autocomplete="off" required min="1" max="3" step="1" id =
"floor" name="floor"  value= "${floor}" size="20"class="combo" style="width: 203px">
      </td>
      <td colspan="2">
        <p align="left"  class="text" style="font-size: 16pt;"></p>
      </td>
    </tr>
    <tr>
      <td colspan="2">
        <p align="left" class="text">Доп. работников: </p>
      </td>
      <td colspan="2">
        <p align="left" class="text">Внутренняя отделка:</p>
      </td>
    </tr>
    <tr>
      <td>
        <input type="number" autocomplete="off" min="0" max="5" step="1"
name="worker"  value= "${worker}" size="20"class="combo" style="width: 203px">
      </td>
      <td>
        <p align="left"  class="text" style="font-size: 16pt;">чел.</p>
      </td>
      <td colspan="2">
        <select id = "otdelka" name="otdelka" value= "${otdelka}" class="combo">
          <option value="1">Нет</option>
          <option value="2">Черновая</option>
          <option value="3">Под ключ</option>
        </select>
      </td>
```

```
    </tr>
    <tr>
      <td colspan="2">
        <p align="left" class="text">Срочный заказ:<input type="checkbox"
name="order" value= "${order}"></p>
      </td>
      <td colspan="2">
        <p align="left" class="text"> Очистка участка: <input type="checkbox"
name="clearing" value= "${clearing}"> </p>
      </td>
    </tr>
    <tr>
      <td colspan="2">
        <p align="left" class="text">Промокод:</p>
      </td>
      <td colspan="2">
        <p align="left" class="text">Готовые пресеты:</p>
      </td>
    </tr>
    <tr>
      <td colspan="2">
        <input type="text" name="promo" value= "${promo}" size="20"class="combo">
      </td>
      <td colspan="2">
        <select name="preset" id="preset" value= "${preset}" class="combo">
          <option value="1">Персональный</option>
          <option value="2">Летний дом</option>
          <option value="3">Зимний дом</option>
        </select>
      </td>
    </tr>
    <tr>
      <td colspan="2">
        <p align="left">
          <input type="submit" name="calculation" value="Расчет " class="combo"
style="font-size : 13pt;" >
        </p>
      </td>
      <td colspan="2">
       <p align="left">
        <!-- Открытие и закрытие окна с информацией на одну кнопку-->
        <input id="info" type="button" name="info" value=" Информация "
class="combo" onclick="document.getElementById('okno').style.display='block';
```

```
        document.getElementById('info').style.display='none';
document.getElementById('example_2_2').style.display='block'; "style=" font-size :
13pt;">
        <input type="button" value=" Информация " class="combo" id="example_2_2"
onclick=" document.getElementById('okno').style.display='none';
document.getElementById('example_2_2').style.display='none';
document.getElementById('info').style.display='block';" style=" font-size : 13pt;
display:none;">
      </p>
      </td>
    </tr>
  </form>
  <form action="${pageContext.request.contextPath}/Admin" method="post">
    <tr>
      <td colspan="2">
        <p align="left">
          <input id="settings" type="button" name="settings" value="Смена настроек "
class="combo" style="font-size : 13pt;" onclick="location.hash ='dop'" >
        </p>
      </td>
      <td colspan="2">
        <p align="left">
          <input type="button" name="exit" value="Выход из аккаунта"
onclick="location.href='LogIn.jsp'" class="combo" style="font-size : 13pt;">
      </td>
    </tr>
   </table>
  <div id="dop" class="okno" style="font-size : 14pt; color: #b3b3b3; " >
   <table>
    <tr>
      <td>
        <div align="right" style="position:static;">
          <a href="#" style="text-decoration: none;text-align: right;
color:#b3b3b3;">&#215;</a>
        </div>
      </td>
    </tr>
    <tr>
      <td style="height:60px;">
        <p style="text-align:left;"  class="text">Вид дома для изменения:<br>
          <select name="type" value= "${type}" class="combo">
            <option value="1">Летний дом</option>
            <option value="2">Зимний дом</option>
```

```
            <option value="3">Сброс "Летний"</option>
            <option value="4">Сброс "Зимний"</option>
          </select>
        </p>
      </td>
    </tr>
    <tr>
      <td style="height:60px;"><p align="left"  class="text">Длина:<br>
        <input id="length" type="number" min="1" max="1000" name="length" value=
"${length}" size="20" class="combo"style="width: 203px"> </p>
      </td>
    </tr>
    <tr>
      <td style="height:60px;"><p align="left" class="text">Ширина:<br>
        <input type="number" min="1" max="1000" name="width"  value= "${width}"
size="20"class="combo" style="width: 203px"> </p>
      </td>
    </tr>
    <tr>
      <td style="height:60px;"><p align="left" class="text">Высота:<br>
        <input type="number" min="2"max="30" name="height" value= "${height}"
size="20"class="combo" style="width: 203px"></p>
      </td>
    </tr>
    <tr>
      <td style="height:60px;"><p align="left" class="text">Количество этажей:<br>
        <input type="number" min="1" max="3" name="floor"  value= "${floor}"
size="20"class="combo" style="width: 203px"></p>
      </td>
    </tr>
    <tr>
      <td style="height:60px;">
        <p align="left" class="text">Внутренняя отделка:<br>
          <select name="otdelka" value= "${otdelka}" class="combo">
            <option value="1">Нет </option>
            <option value="2">Черновая</option>
            <option value="3">Под ключ</option>
          </select>
        </p>
      </td>
    </tr>
    <tr>
      <td style="height:60px;">
```

```
        <p align="left">
          <input type="submit" name="ready" value="Применить настройки"
class="combo" style="font-size : 13pt;">
        </p>
      </td>
    </tr>
  </table>
 </div>
 </div>
 </div>

 <div id="okno" style=" color: #b3b3b3;">
  <table class = "text" >
    <tr>
      <td>
        <p style="background: #1c1c1c;font-weight: normal;font-size: 18pt;text-align
:left">Разработчики: </p>
      </td>
      <td>
        <div align="right">
          <a href="#" style="text-decoration: none;text-align: right;
color:#b3b3b3;">&#215;</a>
        </div>
      </td>
    </tr>
    <tr>
      <td style="text-align :left">
        <p style="font-size: 16pt">Салихов Рашит Ниязович<br>
                                Иутин Александр Сергеевич<br>
                                Бесполитов Даниил Анатольевич</p> </td></tr>
  </table>
 </div>


 <script>
    function setV(a){
    if (a==1){
      document.getElementById('length').disabled = true ; }
      else{
      document.getElementById('settings').type = 'hidden';
      }
    }
    var s = document.getElementById('preset');
```

```
    s.addEventListener('change', setPre);
    function setPre(){
    var ch = s.value;
    if (ch=='1'){
    document.getElementById('length').readOnly = false;
    document.getElementById('width').readOnly = false;
    document.getElementById('height').readOnly = false;
    document.getElementById('floor').readOnly = false;
    document.getElementById('otdelka').readOnly = false;
    }
    if (ch=='3'){

    document.getElementById('length').value =
document.getElementById('winter0').value;
    document.getElementById('length').readOnly = true;
    document.getElementById('width').value = document.getElementById('winter1').value;
    document.getElementById('width').readOnly = true;
    document.getElementById('height').value =
document.getElementById('winter2').value;
    document.getElementById('height').readOnly = true;
    document.getElementById('floor').value = document.getElementById('winter3').value;
    document.getElementById('floor').readOnly = true;
    document.getElementById('otdelka').value =
document.getElementById('winter4').value;;
    document.getElementById('otdelka').readOnly = true;
    }
    if (ch=='2'){
    document.getElementById('length').value =
document.getElementById('summer0').value;
    document.getElementById('width').value =
document.getElementById('summer1').value;
    document.getElementById('height').value =
document.getElementById('summer2').value;
    document.getElementById('floor').value =
document.getElementById('summer3').value;
    document.getElementById('otdelka').value =
document.getElementById('summer4').value;
    document.getElementById('length').readOnly = true;
    document.getElementById('width').readOnly = true;
    document.getElementById('height').readOnly = true;
    document.getElementById('floor').readOnly = true;
    document.getElementById('otdelka').readOnly = true;
    }
```

```
    }
  function onload() {
    if (document.getElementById('fio').value.length<4){
        location.href='LogIn.jsp';
    }
    }
    </script>


</html>
</body>
```

ПРИЛОЖЕНИЕ 9

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>
<style>
body{
  background: #121212;
}
table {
    margin: auto;
   }
   td {
    text-align: center;
     background: #1c1c1c;
   }
   p {
    font-size: 20pt; /* Размер шрифта в пунктах */
    background: #1c1c1c;
   }
   .cent{
    color: #b3b3b3;
  font-size: 25pt;
   }
   .text{
   color: #b3b3b3;
   }
   .combo{
   background-color: #282828;
   color: #b3b3b3;
   font-size : 12pt;
   }
</style>
</head>
<body>
  <p style="font-size : 15pt; color: #b3b3b3; background: #121212; text-align: right"
>${fio}</p>
  <h1 style="font-size : 30pt; color: #b3b3b3; text-align:center; valign:
top;">Калькулятор стоимости строительства частного дома<h1>
```

```
<table width=auto height=auto cellspacing="0" cellpadding="8" cols = "2">
   <tr>
     <td colspan="2" style= "text-align: left;"; class="cent" id="fio">Ваши данные:
${fio}</td>
   </tr>
   <tr>
     <td>
       <p align="left"  class="text"id="length">Длина: ${result_length} м.</p>
     </td>
     <td>
       <p align="left" class="text" id="width">Ширина: ${result_width} м.</p>
    </td>
   </tr>
   <tr>
     <td>
       <p align="left" class="text" id="height">Высота: ${result_height} м.</p>
     </td>
     <td>
       <p align="left" class="text" id="floor">Количество этажей: ${result_floor}</p>
     </td>
   </tr>
   <tr>
     <td>
       <p align="left" class="text" id="worker">Доп. работников: ${result_worker}
чел.</p>
     </td>
     <td>
       <p align="left" class="text" id="otdelka">Внутренняя отделка: ${result_otdelka}
</p>
     </td>
   </tr>
   <tr>
     <td>
       <p align="left" class="text" id="order">Срочный заказ: ${result_order}</p>
     </td>
     <td>
       <p align="left" class="text"id="clearing"> Очистка участка: ${result_clearing}
</p>
     </td>
   </tr>
   <tr>
     <td>
       <p align="left" class="text" id="promo">Промокод: ${result_promo}</p>
```

```
      </td>
      <td>
        <p align="left" class="text" id="preset" id="preset">Готовый пресет:
${result_preset}</p>
      </td>
    </tr>
    <tr>
      <td colspan="2" align="left"><p class="text" align = "left" id="result">Результат:
${result}</p></td>
    </tr>
      <tr>
        <td>
          <p align="left"> <input type="button" name="Cancel" value="Назад"
class="combo" onclick="history.back()" style="font-size : 13pt;"></p>
        </td>
        <td></td>
      </tr>
        <td>
          <p align="left"> <input id="button"  type="submit" name="button"
onclick="asd();" value="Скачать PDF" class="combo"  style="font-size : 13pt;"/> </p>
        </td>
        <td>
          <p align="left"> <input type="button" name="exit" value="Выход из аккаунта"
onclick="location.href='LogIn.jsp'" class="combo" style="font-size : 13pt;"> </p>
        </td>
</table>

<script
src="https://cdnjs.cloudflare.com/ajax/libs/pdfmake/0.1.71/pdfmake.min.js"></script>
<script src="https://cdnjs.cloudflare.com/ajax/libs/pdfmake/0.1.71/vfs_fonts.js"></script>
<script>
function asd(){
var docDefinition = {
    info: {
        title: "Результат",
          },

pageSize:'A4',
pageOrientation:'portrait',
pageMargins:[50,50,30,60],
header:{
    text:'Итог расчетов:',
    fontSize: 17,
```

```
    bold: true,
    alignment:'left',
    margin:[50,30,10,50],

    }
,
content: [
  {
  text:[
  '\n',
  document.getElementById('fio').textContent, '\n',
  document.getElementById('length').textContent,'\n',
  document.getElementById('width').textContent,'\n',
  document.getElementById('height').textContent,'\n',
  document.getElementById('floor').textContent,'\n',
  document.getElementById('worker').textContent,'\n',
  document.getElementById('otdelka').textContent,'\n',
  document.getElementById('order').textContent,'\n',
  document.getElementById('clearing').textContent,'\n',
  document.getElementById('promo').textContent,'\n',
  document.getElementById('preset').textContent,'\n \n',
  document.getElementById('result').textContent
  ],
  style: 'bigger'
  }
],
styles: {

        bigger: {
            fontSize: 16,
        lineHeight: 1.2
        }
    }
  }
pdfMake.createPdf(docDefinition).download('Результат.pdf');
}
</script>
</body>
</html>
```

ПРИЛОЖЕНИЕ 10

Dom.java

```
package Package;
public abstract class Dom {
    double lenght;
    double weight;
    int KolE;
    double visota;
    String otd;
    public void setLenght(double lenght){
        this.lenght = lenght;
    }
    public double getLenght(){
        return lenght;
    }
    public void setWeight(double weight){
        this.weight = weight;
    }
    public double getWeight(){
        return weight;
    }

    public void setKolE(int KolE){
        this.KolE = KolE;
    }
    public int getKolE(){
        return KolE;
    }

    public void setVnOt(){

    }
    public String getVnOt(){
        return otd;
    }
    public void setVisota(double visota){
        this.visota = visota;
    }
    public double getVisota(){
        return visota;
    }

}
```

ПРИЛОЖЕНИЕ 11

PersonalHouse.java

```
package Package;

public class PersonalHouse extends Dom {

}
```

ПРИЛОЖЕНИЕ 12

SummerHouse.java

```
package Package;

public class SummerHouse extends Dom {
    String otd;
    @Override
    public void setVnOt(){//double a,int f
        otd = "Нет";
        //Otdelka otdelka = new House();
        //otdelka.OtdelkaPrice("Нет",a,f);
    }
    @Override
    public String getVnOt(){
        return otd;
    }
}
```

ПРИЛОЖЕНИЕ 13

WinterHouse.java

```
package Package;

public class WinterHouse extends Dom {
    @Override
    public void setVnOt(){//double a,int f
        otd = "Под ключ";
        //Otdelka otdelka = new House();
        //otdelka.OtdelkaPrice("Нет",a,f);
    }
    @Override
    public String getVnOt(){
        return otd;
    }
}
```

ПРИЛОЖЕНИЕ 14

Admin.java

```
package Package;

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 javax.swing.*;



import java.io.File;
import java.io.FileWriter;
import java.io.IOException;
import java.sql.Connection;
import java.sql.DriverManager;
import java.sql.ResultSet;
import java.sql.SQLException;
import java.sql.Statement;
import java.util.Scanner;
@WebServlet(name="Adm", urlPatterns="/Admin")
public class Admin extends  HttpServlet {
	protected void doPost(HttpServletRequest request, HttpServletResponse response)
throws ServletException, IOException {
		RequestCalc Calc = RequestCalc.fromRequestParameters(request);
		Calc.setAsRequestAttributesAndCalculate(request);

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

	}

	private static class RequestCalc {
		private final String length;
		private final String width;
		private final String height;
		private final String fl_count;
		private final String otdelka;
		private final String type;
		private String fio;
```

```
		private RequestCalc (String dlina, String shirina, String visota, String
floor_count, String otdelka, String type) {
			this.length = dlina;
			this.width = shirina;
			this.height = visota;
			this.fl_count = floor_count;
			this.otdelka = otdelka;
			this.type = type;
		}
		public static RequestCalc fromRequestParameters (HttpServletRequest
request) {
			return new RequestCalc(
				request.getParameter("length"),
				request.getParameter("width"),
				request.getParameter("height"),
				request.getParameter("floor"),
				request.getParameter("otdelka"),
				request.getParameter("type")
				);
			}

		public void setAsRequestAttributesAndCalculate(HttpServletRequest request)
{
			Connection con =null;
	Statement stmt=null;
	con = getConnection();
	try {
	stmt = con.createStatement();
	String query;
	Double try_length = 0.0;
	Double try_width = 0.0;
	Double try_height = 0.0;
		int try_fl_count = 0;
		int try_otdelka = 0;
	try {
		try_length=Double.parseDouble(length);
		try_width=Double.parseDouble(width);
		try_height=Double.parseDouble(height);
		try_fl_count = Integer.parseInt(fl_count);
		try_otdelka = Integer.parseInt(otdelka);
			}
			catch (NumberFormatException e) {
```

```
                }
        ResultSet rs1=null;
        String arr[] ={"length","width","height","floor","otdelka"};
        String query1;
        request.setAttribute("fio", "Администратор");
        switch (type) {
        case "1": {
                query = "UPDATE settings SET length ='"+try_length+"' WHERE type =
'uptSummer'";
                stmt.executeUpdate(query);
                query = "UPDATE settings SET width ='"+try_width+"' WHERE type =
'uptSummer'";
                stmt.executeUpdate(query);
                query = "UPDATE settings SET height ='"+try_height+"' WHERE type =
'uptSummer'";
                stmt.executeUpdate(query);
                query = "UPDATE settings SET floor ='"+try_fl_count+"' WHERE type =
'uptSummer'";
                stmt.executeUpdate(query);
                query = "UPDATE settings SET otdelka ='"+try_otdelka+"' WHERE type =
'uptSummer'";
                stmt.executeUpdate(query);
                break;
        }
        case "2":{
                query = "UPDATE settings SET length ='"+try_length+"' WHERE type =
'uptWinter'";
                stmt.executeUpdate(query);
                query = "UPDATE settings SET width ='"+try_width+"' WHERE type =
'uptWinter'";
                stmt.executeUpdate(query);
                query = "UPDATE settings SET height ='"+try_height+"' WHERE type =
'uptWinter'";
                stmt.executeUpdate(query);
                query = "UPDATE settings SET floor ='"+try_fl_count+"' WHERE type =
'uptWinter'";
                stmt.executeUpdate(query);
                query = "UPDATE settings SET otdelka ='"+try_otdelka+"' WHERE type =
'uptWinter'";
                stmt.executeUpdate(query);
                break;
        }
        case "3":{
```

```
                query = "UPDATE settings SET length = (SELECT length FROM
default_settings WHERE type = 'summer') WHERE type = 'uptSummer'";
                stmt.executeUpdate(query);
                query = "UPDATE settings SET width = (SELECT width FROM
default_settings WHERE type = 'summer') WHERE type = 'uptSummer'";
                stmt.executeUpdate(query);
                query = "UPDATE settings SET height = (SELECT height FROM
default_settings WHERE type = 'summer') WHERE type = 'uptSummer'";
                stmt.executeUpdate(query);
                query = "UPDATE settings SET floor = (SELECT floor FROM
default_settings WHERE type = 'summer') WHERE type = 'uptSummer'";
                stmt.executeUpdate(query);
                query = "UPDATE settings SET otdelka = (SELECT otdelka FROM
default_settings WHERE type = 'summer')WHERE type = 'uptSummer'";
                stmt.executeUpdate(query);
                break;
            }
            case "4":{
                query = "UPDATE settings SET length = (SELECT length FROM
default_settings WHERE type = 'winter') WHERE type = 'uptWinter'";
                stmt.executeUpdate(query);
                query = "UPDATE settings SET width = (SELECT width FROM
default_settings WHERE type = 'winter') WHERE type = 'uptWinter'";
                stmt.executeUpdate(query);
                query = "UPDATE settings SET height = (SELECT height FROM
default_settings WHERE type = 'winter') WHERE type = 'uptWinter'";
                stmt.executeUpdate(query);
                query = "UPDATE settings SET floor = (SELECT floor FROM
default_settings WHERE type = 'winter') WHERE type = 'uptWinter'";
                stmt.executeUpdate(query);
                query = "UPDATE settings SET otdelka = (SELECT otdelka FROM
default_settings WHERE type = 'winter')WHERE type = 'uptWinter'";
                stmt.executeUpdate(query);
                break;
            }
            }
            for(int i =0;i<5;i++){
               query1 = "SELECT "+arr[i]+" FROM settings WHERE type='uptWinter';";
               rs1 = stmt.executeQuery(query1);
               if (rs1.next()){
               request.setAttribute("winter"+i,rs1.getString(arr[i]));
               }
               query1 = "SELECT "+arr[i]+" FROM settings WHERE type='uptSummer';";
```

```
            rs1 = stmt.executeQuery(query1);
            if (rs1.next()){
                request.setAttribute("summer"+i,rs1.getString(arr[i]));
            }
        }
                }catch (SQLException sqlEx ){
                        }
                }




        public static Connection getConnection() {
        String driver = "com.mysql.jdbc.Driver";
        final String url = "jdbc:mysql://remotemysql.com:3306/smszCuaCce";
        final String user = "smszCuaCce";
        final String password = "fjrxusR9mP";
        try {
            Class.forName(driver);
            java.sql.Connection con = DriverManager.getConnection(url,user,password);
            return con;

        } catch (ClassNotFoundException | SQLException e) {
            System.out.println("Exception " + e);
            return null;
        }
    }

        }
}
```

ПРИЛОЖЕНИЕ 15

Calculator.java

```
package Package;
import java.awt.event.ActionEvent;
import java.awt.event.ActionListener;
import java.io.File;
import java.io.IOException;
import java.util.Scanner;

import Package.House.*;

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 javax.swing.JButton;
import javax.swing.JCheckBox;
import javax.swing.JComboBox;
import javax.swing.JFrame;
import javax.swing.JLabel;
import javax.swing.JPanel;
import javax.swing.JTextField;

@WebServlet(name="Calc", urlPatterns="/Calculator")
public class Calculator extends  HttpServlet   {
	public static String dlina;
	public static String shirina;
	public static String visota;
	public static String floor_count;
	public static String worker_count;
	public static String otdelka;
	public static boolean express_order;
	public static boolean clearing;
	public static String promo;
	public static String preset;
	static String[] fields;
	protected void doPost(HttpServletRequest request, HttpServletResponse response)
throws ServletException, IOException {
		request.setCharacterEncoding("UTF-8");
	response.setCharacterEncoding("UTF-8");
		RequestCalc Calc = RequestCalc.fromRequestParameters(request);
		Calc.setAsRequestAttributesAndCalculate(request);
```

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

	}
	private static class RequestCalc {
		private final String dl;
		private final String sh;
		private final String vis;
		private final String fl_count;
		private  String rab_count;
		private  String otd;
		private final String ord;
		private final String clear;
		private  String pr;
		private  String pres;
		private  String shirina;
		private String fio;

		private String result;

		private RequestCalc (String dlina, String shirina, String visota, String
floor_count, String worker_count, String otdelka, String order, String clearing, String
promo, String preset, String fio) {
			this.dl = dlina;
			this.sh = shirina;
			this.vis = visota;
			this.fl_count = floor_count;
			this.rab_count = worker_count;
			this.otd = otdelka;
			this.ord = order;
			this.clear = clearing;
			this.pr = promo;
			this.pres = preset;
			this.fio = fio;

		}
		public static RequestCalc fromRequestParameters (HttpServletRequest
request) {
			return new RequestCalc(
				request.getParameter("length"),
				request.getParameter("width"),
				request.getParameter("height"),
				request.getParameter("floor"),
```

```
                        request.getParameter("worker"),
                        request.getParameter("otdelka"),
                        request.getParameter("order"),
                        request.getParameter("clearing"),
                        request.getParameter("promo"),
                        request.getParameter("preset"),
                        request.getParameter("fio")
                        );
                }

        public void setAsRequestAttributesAndCalculate(HttpServletRequest request)
{
                switch (otd){
        case "1": otd = "Нет"; break;
        case "2": otd = "Черновая"; break;
        case "3": otd = "Под ключ"; break;
        }

                switch (pres){
        case "1": pres = "Персональный"; break;
        case "2": pres = "Летний дом"; break;
        case "3": pres = "Зимний дом"; break;
        }
                if (rab_count.equals("")) {
                        rab_count = "0";
                }
                request.setAttribute("result_length", dl);
                request.setAttribute("result_width", sh);
                request.setAttribute("result_height", vis);
                request.setAttribute("result_floor", fl_count);
                request.setAttribute("result_worker", rab_count);
                request.setAttribute("result_otdelka", otd);
                request.setAttribute("fio", fio);
                if (ord != null){
    request.setAttribute("result_order", "Да");
    express_order = true;
                }
        else{
    request.setAttribute("result_order", "Нет");
    express_order = false;
        }
        if (clear != null){
    request.setAttribute("result_clearing", "Да");
```

```
    clearing = true;
}
else{
    request.setAttribute("result_clearing", "Нет");
    clearing = false;
}
switch (pr) {
                case "PROMO5":
                        pr = "PROMO5";
                        break;
                case "PROMO10":
                        pr = "PROMO10";
                        break;
                case "PROMO15":
                        pr = "PROMO15";
                        break;
                default:
                        pr = "Не применен";
                        break;
}
                request.setAttribute("result_promo", pr);
                request.setAttribute("result_preset", pres);

                dlina = dl;
                shirina = sh;
                visota = vis;
                floor_count = fl_count;
                worker_count = rab_count;
                otdelka = otd;
                promo = pr;
                preset = pres;

                /*double length_try;
                double width_try;
                double height_try;
                double floor_try;
                double worker_try;*/

                fields = new String[6];
                fields[0]=dlina;
                fields[1]=shirina;
                fields[2]=visota;
                fields[3]=floor_count;
```

```
                    fields[4]=worker_count;
                    fields[5]=promo;

                    result = sobitie(fields,express_order,clearing,otdelka,preset);
                    request.setAttribute("result", result);
                    /*try {
                    length_try=Double.parseDouble(dl);
                    width_try=Double.parseDouble(sh);
                    height_try=Double.parseDouble(vis);
                    floor_try=Double.parseDouble(fl_count);
                    worker_try=Double.parseDouble(rab_count);



                    }
                    catch (NumberFormatException e) {
                            length_try = 0;
                            width_try = 0;
                            height_try = 0;
                            floor_try = 0;
                            worker_try = 0;
                    }*/

            }

        }



static double[] sum;
static double[] win;


//public static void Window() {



static Dom dom;


public static void preset(String jcomboBox1){
        //setUp();
        switch (preset){
```

```
            /*case "Персональный":{
                dom = new PersonalHouse();
                break;

            }*/
            case "Летний дом":{
                dom = new SummerHouse();
                dom.setLenght(Double.parseDouble(fields[0]));
                dom.setWeight(Double.parseDouble(fields[1]));
                dom.setKolE(Integer.parseInt(fields[3]));
                dom.setVisota(Double.parseDouble(fields[2]));
                dom.setVnOt();
                //close();
                break;
            }
            case "Зимний дом":{
                dom = new WinterHouse();
                dom.setLenght(Double.parseDouble(fields[0]));
                dom.setWeight(Double.parseDouble(fields[1]));
                dom.setKolE(Integer.parseInt(fields[3]));
                dom.setVisota(Double.parseDouble(fields[2]));
                dom.setVnOt();
                //close();
                break;
            }
        }
}

/*public static void close(){
        int a=0;

        switch(dom.getVnOt()){
            case "Нет": {a=0;break;}
            case "Под ключ": {a=2;break;}
        }

        fields[0] = Double.toString(dom.getLenght());
        fields[1] = Double.toString(dom.getWeight());
        fields[2] = Double.toString(dom.getVisota());
        fields[3] = Integer.toString(dom.getKolE());

}*/
```

```
        static String item;

public static String sobitie (String fields[], boolean express_order, boolean clearing, String
otdelka,String jcomboBox1 ){//получение данных
        int summa = 5000;
        double srochno = 0.0;
        double chistka = 0.0;
        double promo = 1.0;

        switch (fields[5]) {
        case "PROMO5":
                promo = 0.95;
                break;
        case "PROMO10":
                promo = 0.90;
                break;
        case "PROMO15":
                promo = 0.85;
                break;

        }

        if (express_order ) {
                srochno = 10000.0;
        }
        if (clearing ) {
                chistka = 5000.0;
        }

    try{ // отлавливание ошибки при переводе строки в число
        double a = Double.parseDouble(fields[0]); // преобразование строк в тип double
        double b = Double.parseDouble(fields[1]);
        double c = Double.parseDouble(fields[2]);
        int f = Integer.parseInt(fields[3]); // преобразование строк в тип int
        int rabotnik = Integer.parseInt(fields[4]);
        if (rabotnik < 0 || rabotnik > 5) {

          return "ERROR: неверное кол-во работников"; //Возвращение значения
        }else {
```

```
if (a < 1 || b < 1  ) { //проверка правильности ввода чисел

  return "ERROR: неверны размеры длины или ширины";
}
else {
 if (c < 2) {

   return "ERROR: неверная высота дома";
 }else {
   if(f < 1 || f > 3) {

        return "ERROR: недопустимое количество этажей";
   }else {
            if (jcomboBox1.equals("Персональный")){
                                dom = new PersonalHouse();

                                dom.setLenght(a);
                                dom.setWeight(b);
                                dom.setKolE(f);
                                dom.setVisota(c);

                        }

            rabotnik = rabotnik * summa;
            Home home = new House();
            Otdelka otdelka_type = new House();
            String item = (String)otdelka;
            //System.out.print(dom.getWeight());

            preset(jcomboBox1);


            String result
=String.format("%.2f",(home.GetPrice(dom.getLenght(),dom.getWeight(),dom.getVisota(
),dom.getKolE())+

otdelka_type.OtdelkaPrice(otdelka,dom.getLenght()*dom.getWeight(),dom.getKolE())+
srochno + chistka + rabotnik)*promo);// Расчет итоговой стоимости
                        return result + " руб.";
   }
 }

}
```

```
    }
    }
    catch (NumberFormatException nfe) {
      // labels[10].setText("ERROR: Введите число"); //вывод сообщения в случае
ошибки
        return "ERROR: Введите число";
    }

}

/*static void setUp(){
        sum = new double[4];
        win = new double[4];

        int last;
        String s;
        String subs;
        try{
                File file = new File("./src/main/resources","Settings.txt");
                if (file.exists()){
                        Scanner scanner = new Scanner(file);
                        while (scanner.hasNextLine()) {
                                s = scanner.nextLine();
                                if (s.s.dl = dlina;
                        this.sh = shirina;
                        this.vis = visota;
                        this.fl_count = floor_count;
                        this.rab_count = worker_count;
                        this.otd = otdelka;
                        this.ord = order;
                        this.clear = clearing;
                        this.pr = promo;
                        this.pres = preset;startsWith("$sum")) {
                                        for (int i =3;i>=0; i--){
                                                last = s.lastIndexOf(" ");
                                                subs = s.substring(last,s.length());
                                                sum[i] = Double.parseDouble(subs);
                                                s = s.substring(0,last);
                                        }
                                }
                                if (s.startsWith("$win")){
                                        for (int i =3;i>=0; i--){
                                                last = s.lastIndexOf(" ");
```

```
                                        subs = s.substring(last,s.length());
                                        win[i] = Double.parseDouble(subs);
                                        s = s.substring(0,last);
                                }
                        }
                }
                scanner.close();
        }
        else{
                file.createNewFile();
                Admin adm = new Admin();
                adm.defaultSet();
                setUp();
        }
    }
    catch (IOException e){System.out.print(e);}
}*/

}
```

ПРИЛОЖЕНИЕ 16

House.java

```
package Package;

public class House implements Home, Otdelka {

	@Override
	public Double GetPrice(double Dlina, double Shirina, double Visota, int
CountOfFloor) {

		Double Price, Square, PriceOfOneMeter = 50000.0;
		Square = Dlina*Shirina;
		Price = Square * PriceOfOneMeter * CountOfFloor;
		return Price;
	}
	@Override
	public Double OtdelkaPrice(String OtdelkaValue, double square, int CountOfFloor)
{
		double price=0;
		switch (OtdelkaValue) {
		case  ("Нет"):
			 price=0;
			 break;
		case  ("Черновая отделка"):
			 price = square * 5000 * CountOfFloor;
			 break;
		case  ("Под ключ"):
			 price = square * 10000 * CountOfFloor;
			 break;
		}
		return price;
	}
}
```

ПРИЛОЖЕНИЕ 17

Home.java

```
package Package;

public interface Home {
 public Double GetPrice(double Dlina, double Shirina, double Visota, int CountOfFloor);
}
```

ПРИЛОЖЕНИЕ 18

Otdelka.java

```
package Package;

public interface Otdelka {
	public Double OtdelkaPrice(String OtdelkaValue, double square, int CountOfFloor);
}
```

ПРИЛОЖЕНИЕ 19

ConnectionDataBaseTest.java

```
package Package;

import static org.junit.Assert.*;

import org.junit.After;
import org.junit.Before;
import org.junit.Test;

import java.sql.Connection;
import java.sql.SQLException;

public class ConnectionDataBaseTest {

    LogInReg cDB;
    Connection con;

    @Before
    public void beforeTest(){
        cDB = new LogInReg();
    }

    @Test
    public void test(){
        con = cDB.getConnection();
        assertNotNull(con);
    }

    @After
    public void afterTest(){
        try{
            if (con!=null){
                con.close();
            }
        }
        catch (SQLException sqEx){
            System.out.println(sqEx.getErrorCode());
        }
    }

}
```

ПРИЛОЖЕНИЕ 20

SummerHouseTest.java

```
package Package;

import static org.junit.Assert.*;

import org.junit.Before;
import org.junit.Test;

public class SummerHouseTest {
SummerHouse dom;

@Before
public void before() {
dom = new SummerHouse();
dom.setKolE(4);
dom.setLenght(3);
}


@Test
public void test() {
assertEquals("4",Integer.toString(dom.getKolE()));
assertEquals("3.0", Double.toString(dom.getLenght()));
}

}
```

ПРИЛОЖЕНИЕ 21

HouseTest.java

```
package Package;
import static org.junit.Assert.*;
import org.junit.Before;
import org.junit.Test;
public class HouseTest {
	Otdelka otdelka;
	@Before
	public void before() {
		otdelka = new House();
	}
	@Test
	public void test() {
		double testValue1 = otdelka.OtdelkaPrice("Нет", 20, 1);
		double testValue2 = otdelka.OtdelkaPrice("Черновая отделка", 20, 1);
		double testValue3 = otdelka.OtdelkaPrice("Под ключ", 20, 1);
		assertEquals("0.0",Double.toString(testValue1));
		assertEquals("100000.0",Double.toString(testValue2));
		assertEquals("200000.0",Double.toString(testValue3));
	}
}
```

Список использованной литературы

1. Федеральный закон об охране окружающей среды РФ [Электронный ресурс]. Режим доступа: http://www.consultant.ru/document/cons_doc_LAW_34823/.

(Дата обращения 25.12.2020)

2. Федеральный закон о техническом регулировании РФ [Электронный ресурс].

Режим доступа: http://www.consultant.ru/document/cons_doc_LAW_40241/

(Дата обращения 25.12.2020)

3. Федеральный закон о саморегулировании организациях РФ [Электронный ресурс]. Режим доступа: http://www.consultant.ru/document/cons_doc_LAW_72967/ (Дата обращения 25.12.2020)

4. Планировка и застройка территории ведения гражданами садоводства. [Электронный ресурс]. Режим доступа: http://docs.cntd.ru/document/564412858 (Дата обращения 25.12.2020)

5. Атмосферный воздух и воздух закрытых помещений, санитарная охрана воздуха. [Электронный ресурс]. Режим доступа: http://docs.cntd.ru/document/1200006803

(Дата обращения 25.12.2020)

6. Зоны санитарной охраны источников водоснабжения и водопроводов хозяйственно-питьевого назначения. [Электронный ресурс]. Режим доступа: http://docs.cntd.ru/document/1400014 (Дата обращения 25.12.2020)

7. СанПиН 2.1.2.2645-10 Санитарно-эпидемиологические требования к условиям проживания в жилых зданиях и помещениях. [Электронный ресурс]. Режим доступа: http://docs.cntd.ru/document/902222351

(Дата обращения 25.12.2020)

8. СНиПы, ГОСТы и градостроительные нормы. [Электронный ресурс]. Режим доступа: https://idr-group.ru/vazhno-znat/snipy-gosty-i-gradostroitelnye-normy/

(Дата обращения 25.12.2020)

9. Абстрактные классы в Java на конкретных примерах. [Электронный ресурс]. Режим доступа: https://javarush.ru/groups/posts/1973-abstraktnihe-klassih-v-java-na-konkretnihkh-primerakh

(Дата обращения 25.12.2020)

10. МЕЖГОСУДАРСТВЕННЫЙ СТАНДАРТ [Электронный ресурс]. – Режим доступа: https://docs.cntd.ru/document/1200007647. (Дата обращения: 17.05.2020)

11. Habr [Электронный ресурс]. – Режим доступа: https://habr.com/ru/. (Дата обращения: 17.05.2020)

12. Travis: Deploy over SSH [Электронный ресурс]. – Режим доступа: https://www.juev.org/2016/09/03/travis-deploy-over-ssh/. (Дата обращения: 17.05.2020)

13. «Джино» — хостинг, конструктор сайтов, регистрация доменов и многое другое. [Электронный ресурс]. – Режим доступа: https://jino.ru/. (Дата обращения: 17.05.2020)

14. Apache Tomcat® - Welcome! [Электронный ресурс]. – Режим доступа: https://tomcat.apache.org/. (Дата обращения: 17.05.2020)