

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-223
Панаев С.И.
Никулова К.Д.
Дёмин Р.А.

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 Панев С.И. Группа ПИ-223 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.

3. Требования к оформлению:

3.1. Пояснительная записка должна быть оформлена в текстовом процессоре LibreOffice Writer в соответствии с требованиями ГОСТ. В бумажном виде оформляются: титульный лист, задание, календарный план и аннотация, которая содержит ссылку на репозиторий с программным кодом и документацией.

3.2. В пояснительной записке должны содержаться следующие разделы:

Раздел 1. Описание предметной области.

Раздел 2. Техническое задание на создание программного продукта.

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

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

Раздел 5. Реализация исходного кода по зонам ответственности.

Раздел 6. Сборка и тестирование программного продукта.

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

Раздел 8. Руководство пользователя программного продукта.

3.3. В приложение выносятся программный код и код тестов.

4. Графическая часть должна включать:

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

Дата выдачи 6.03.2021 Дата окончания 29.05.2021

Руководитель Казанцев А.В. Студент Панев С.И.

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 Никулова К.Д. Группа ПИ-223 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.

3. Требования к оформлению:

3.1. Пояснительная записка должна быть оформлена в текстовом процессоре LibreOffice Writer в соответствии с требованиями ГОСТ. В бумажном виде оформляются: титульный лист, задание, календарный план и аннотация, которая содержит ссылку на репозиторий с программным кодом и документацией.

3.2. В пояснительной записке должны содержаться следующие разделы:

Раздел 1. Описание предметной области.

Раздел 2. Техническое задание на создание программного продукта.

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

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

Раздел 5. Реализация исходного кода по зонам ответственности.

Раздел 6. Сборка и тестирование программного продукта.

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

Раздел 8. Руководство пользователя программного продукта.

3.3. В приложение выносятся программный код и код тестов.

4. Графическая часть должна включать:

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

Дата выдачи 6.03.2021 Дата окончания 29.05.2021

Руководитель Казанцев А.В. Студент Никулова К.Д.

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 Дёмин Р.А. Группа ПИ-223 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.

3. Требования к оформлению:

3.1. Пояснительная записка должна быть оформлена в текстовом процессоре LibreOffice Writer в соответствии с требованиями ГОСТ. В бумажном виде оформляются: титульный лист, задание, календарный план и аннотация, которая содержит ссылку на репозиторий с программным кодом и документацией.

3.2. В пояснительной записке должны содержаться следующие разделы:

Раздел 1. Описание предметной области.

Раздел 2. Техническое задание на создание программного продукта.

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

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

Раздел 5. Реализация исходного кода по зонам ответственности.

Раздел 6. Сборка и тестирование программного продукта.

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

Раздел 8. Руководство пользователя программного продукта.

3.3. В приложение выносятся программный код и код тестов.

4. Графическая часть должна включать:

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

Дата выдачи 6.03.2021 Дата окончания 29.05.2021

Руководитель Казанцев А.В. Студент Дёмин Р.А.

Содержание

1 Описание предметной области.....	6
2 Техническое задание на создание программного продукта.....	14
3 Настройка среды разработки для операционных систем семейств Windows и Linux.....	15
4 Настройка среды разработки для подключения к системе контроля версий.....	21
5 Реализация исходного кода по зонам ответственности.....	25
6 Сборка и тестирование программного продукта.....	26
7 Настройка программной среды для развертывания и запуска программного продукта.....	31
8 Руководство пользователя программного продукта.....	34
Заключение.....	35
Приложения.....	36
ПРИЛОЖЕНИЕ П-1.....	36
ПРИЛОЖЕНИЕ П-2.....	52
ПРИЛОЖЕНИЕ П-3.....	62
ПРИЛОЖЕНИЕ П-4.....	63
ПРИЛОЖЕНИЕ П-5.....	64
ПРИЛОЖЕНИЕ П-6.....	65
ПРИЛОЖЕНИЕ П-7.....	71
ПРИЛОЖЕНИЕ П-8.....	74
ПРИЛОЖЕНИЕ П-9.....	76
ПРИЛОЖЕНИЕ П-10.....	77
ПРИЛОЖЕНИЕ П-11.....	78
ПРИЛОЖЕНИЕ П-12.....	85
ПРИЛОЖЕНИЕ П-13.....	86
ПРИЛОЖЕНИЕ П-14.....	87
ПРИЛОЖЕНИЕ П-15.....	90
ПРИЛОЖЕНИЕ П-16.....	91
ПРИЛОЖЕНИЕ П-17.....	92
ПРИЛОЖЕНИЕ П-18.....	94
ПРИЛОЖЕНИЕ П-19.....	96
Список использованной литературы.....	99

1 Описание предметной области

Быстровозводимые здания – это сооружения, построенные из металлического каркаса для современной промышленности и гражданского назначения. Под быстровозводимыми зданиями чаще всего подразумевают: каркасные дома на несколько этажей, многоквартирные малоэтажные дома, коммерческие здания и производственные строения, а также промышленные помещения.

Они не рассчитаны на последующие разборки, транспортирование и монтаж, а срок их службы может быть аналогичен сроку службы капитальных домов.

Быстровозводимые ангары – это объект, который собирается наподобие конструктора. Зачастую такие строения можно отнести к временным, что существенно упростит все согласования. Но тут всё во многом зависит от типа фундамента.

Существуют следующие основные технологии возведения быстровозводимых домов [1]:

- Каркасно-щитовые (каркасно-панельные).
- Бескаркасные панельные (сборно-щитовые).
- Каркасно-рамочные.
- Каркасно-монолитные (метод несъемной опалубки).
- Блочно-модульные.

Другие варианты каркасных домов – это либо комбинации перечисленных видов, либо небольшие видоизменения вышеупомянутых каркасов.

Для удобства использования программы “Калькулятор стоимости строительства быстровозводимых ангаров” было решено использовать две основные технологии возведения: каркасно-щитовые и каркасно-рамочные. Рассмотрим эти виды технологий более подробно.

Каркасно-щитовые конструкции производятся на заводе. В соответствии с

проектом изготавливается каркас, панели (щиты), межэтажные перекрытия и даже кровля. Каждый элемент размечен, имеет необходимые отверстия. Полный комплект дома доставляется на строительную площадку, где осуществляется его монтаж. Технология строительства каркасно-щитового дома достаточно проста, но для того, чтобы дом был прочным, теплым и надежным, следует четко соблюдать нормы и правила.

Основными преимуществами каркасно-щитовых быстровозводимых конструкций являются:

- высокая скорость возведения;
- доступная стоимость строительства;
- неподверженность усадке.

Минусами каркасно-щитовых домов являются отсутствие возможности произвести перепланировку, а также потребность в специальной технике (кране) для погрузочно-разгрузочных работ.

Каркасно-рамочный метод строительства осуществляется по-другому: каркас дома собирается на строительной площадке в виде не обшитых рамок. Лишь когда вся конструкция установлена, приступают к ее утеплению и обшивке.

Основные преимущества этого метода:

- технология позволяет вести строительство быстровозводимых каркасных зданий в любое время года;
- отсутствие усадки;
- возможность осуществить самые оригинальные архитектурные проекты на начальном этапе, а также позже, если вдруг потребуется перепланировка.

Минусами являются удлиненные сроки строительства и необходимость обработки древесины на строительной площадке.

Этапы строительства ничем не отличаются от других технологий. Сначала идет устройство фундамента, затем монтируется каркас здания, после осуществляется монтаж стеновых панелей, в завершение кровля.

Все методы и этапы работы должны быть выполнены по нормативам, которые обеспечивают и регламентируют эту деятельность:

- ФЗ РФ от 07.02.1992 № 2300-1 (ред. от 31.07.2020) «О защите прав потребителей» [2];
- ГОСТ 12.0.004-2015. «Межгосударственный стандарт. Система стандартов безопасности труда. Организация обучения безопасности труда.» [3];
- ГОСТ 12.1.005-88. «Межгосударственный стандарт. Система стандартов безопасности труда. Общие санитарно-гигиенические требования к воздуху рабочей зоны» [4];
- ФЗ РФ от 21.12.1994 N 69-ФЗ (ред. от 28.05.2017) «О пожарной безопасности» [5].

Для наглядного отражения исследуемой системы было разработано графическое изображение бизнес-процесса, которое представлено на мнемонической схеме процесса «Строительство быстровозводимого ангара», которое приведено на рисунке 1.

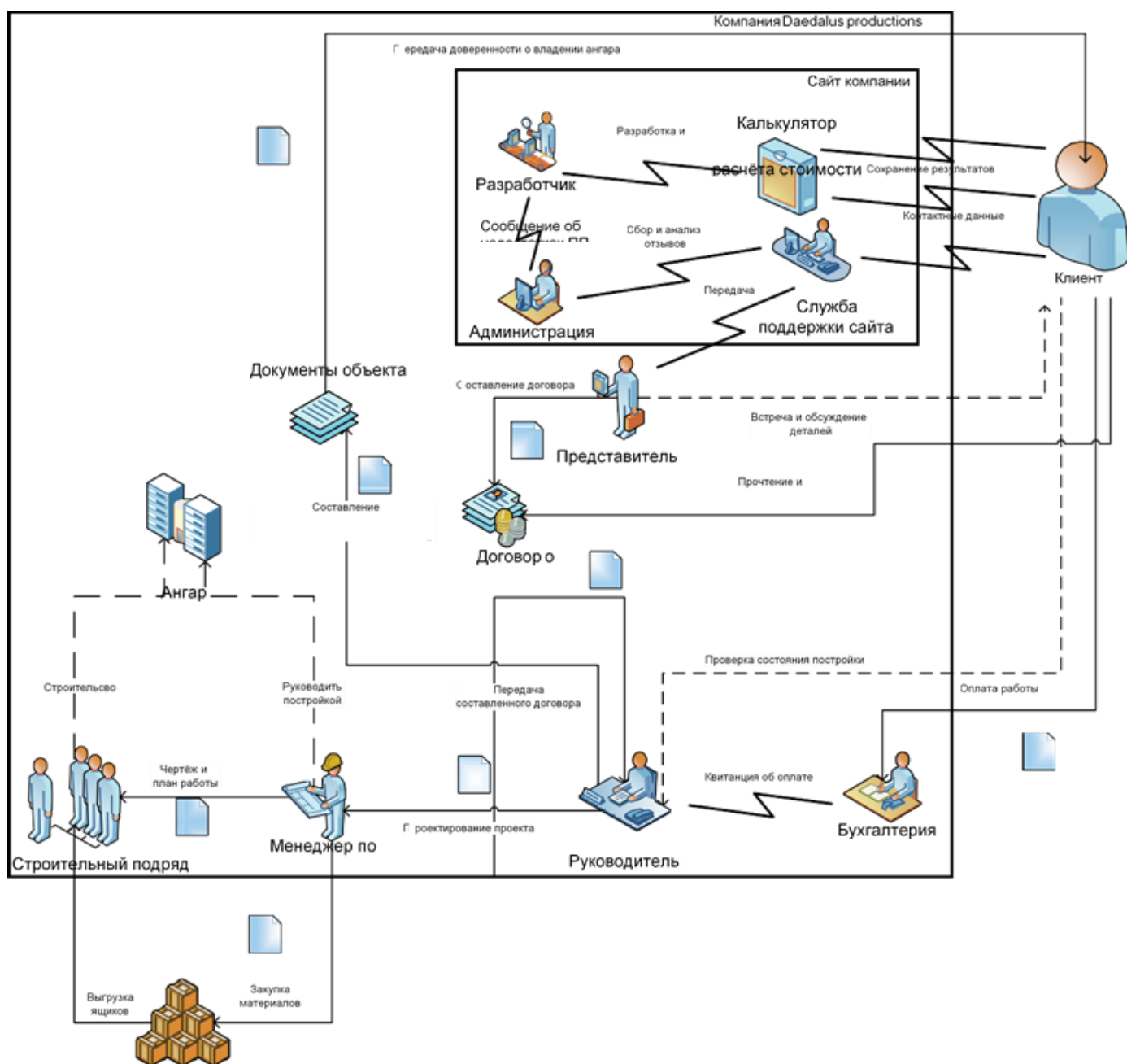


Рисунок 1 – Мнемосхема процесса «Строительство быстровозводимых ангаров»

Для большего понимания данной мнемосхемы, в таблице 1 приведены виды связей объектов и субъектов, используемых в самом процессе.

Таблица 1 – Обозначение взаимодействий

Условное обозначение	Тип взаимодействия
	Бумажный
	Устный
	Электронный

В качестве описания функциональных возможностей программного продукта «Калькулятор стоимости строительства быстровозводимых ангаров» была разработана поведенческая диаграмма, которая приведена на рисунке 2.

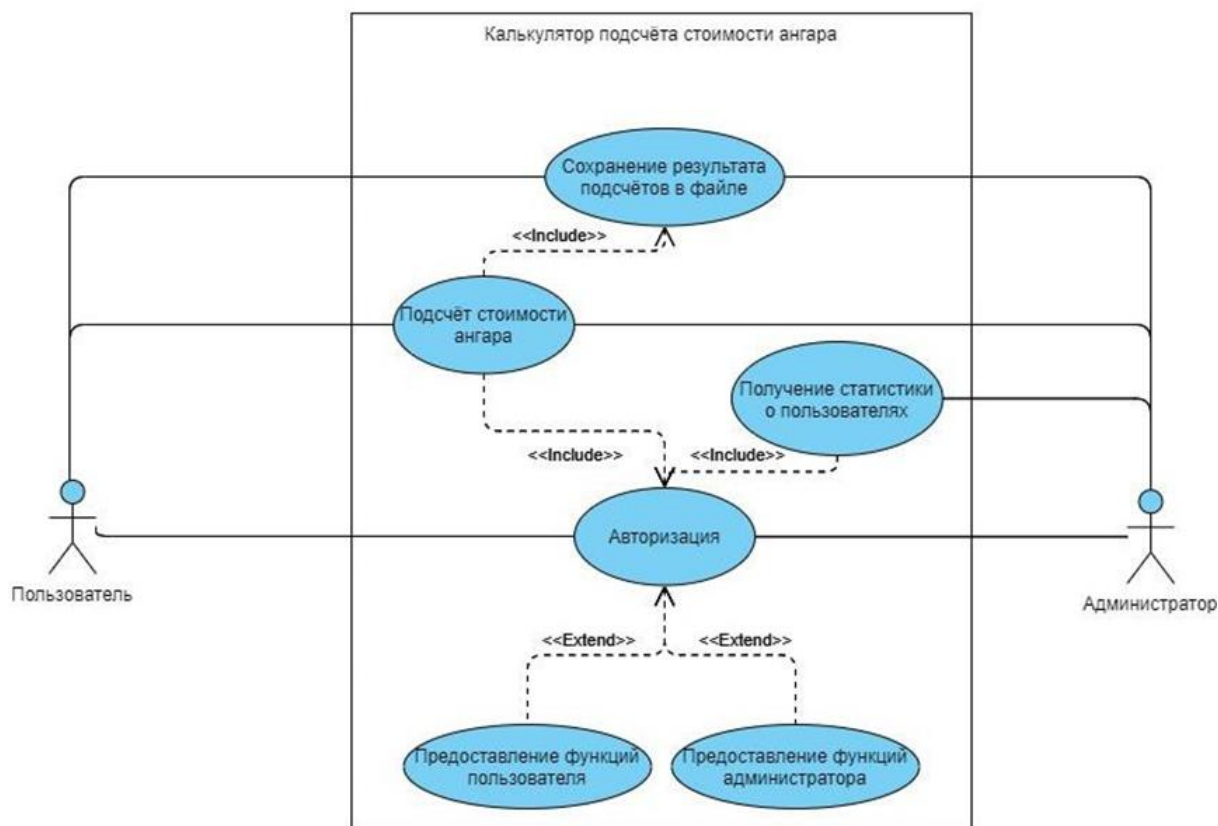


Рисунок 2 – Поведенческая диаграмма

Так же была разработана диаграмма классов, которая демонстрирует структуру классов программы «Калькулятор стоимости строительства быстровозводимых ангаров» и приведена на рисунке 3.

ангара. Всего в нашей работе было выбрано 3 формы ангара. Поэтому у нас будет 3 основные формулы расчета цены основной части ангара.

1) Арочная форма. В случае арочной формы высота (z) считается равной ширине (y).

$$C_{\text{осн}}(x, y, z, C_{\text{пан}}) = \pi \cdot \frac{y}{2} \cdot \left(\frac{y}{2} + x \right) \cdot C_{\text{пан}}, \quad (2)$$

2) Прямостенная форма.

$$C_{\text{осн}}(x, y, z, C_{\text{пан}}) = (2 \cdot (y \cdot z + x \cdot z) + x \cdot y) \cdot C_{\text{пан}}, \quad (3)$$

3) Шатровая форма.

$$C_{\text{осн}}(x, y, z, C_{\text{пан}}) = \\ = 2 \cdot \left(\frac{22}{30} \cdot y \cdot z + x \cdot \sqrt{0,1 \cdot y^2 + \left(\frac{2}{3} \cdot z\right)^2} + \sqrt{\left(\frac{1}{3} \cdot z\right)^2 + (y \cdot 0,4)^2} \cdot x \right) \cdot C_{\text{пан}}, \quad (4)$$

В качестве примера рассмотрим заказ покупателя, которому требуется прямостенный ангар с размерами 5x4x3 на бетонном фундаменте с шторовой дверью. В качестве материала, используемого на стенах, были выбраны «сендвич-панели». Метод постройки ангара был выбран «каркасно-щитовой». Срок постройки ангара – 3 месяца. Промокод у покупателя отсутствует.

На основе этих данных, были выведены следующие значения для расчета стоимости ангара: высота $x = 5$ м, длина $y = 4$ м и ширина $z = 3$ м. Цена сендвич-панелей составила $C_{\text{пан}} = 1600$ руб. за кв.м. Подставив данные в формулу (3) был получен следующий результат:

$$C_{\text{осн}}(x, y, z, C_{\text{пан}}) = (2 \cdot (4 \cdot 3 + 5 \cdot 4) + 5 \cdot 4) \cdot 1600 = 134400 \text{ руб.}$$

Для расчета итоговой стоимости заказа были введены остальные данные. Коэффициент $k_{\text{тип}} = 1$, так как он соответствует каркасно-щитовому типу строительства ангара. Цена бетонного фундамента составила $C_{\text{фунд}} = 1400$ руб. за кв. м. Стоимость шторовой двери составляет $C_{\text{дверь}} = 8400$ руб. за кв. м. И коэффициент $k_{\text{промо}} = 1$, так как покупатель не ввел соответствующий промокод для получения скидки. Приведенные выше данные необходимо подставить в формулу (1) для получения итоговой стоимости:

$$C_{\text{итог}} = (134400 \cdot 1 + 5 \cdot 4 \cdot 1400 + 4 \cdot 3 \cdot 0,65 \cdot 8400) \cdot 1 \cdot 1 = 227920 \text{ руб.}$$

Покупатель обратился с просьбой о расчете арочного ангара с размерами 4х6 для сравнения стоимости. В качестве материала, используемого на стенах, были выбраны «сендвич-панели».

На основе вышеперечисленных данных, были выведены следующие значения для расчета стоимости арочного ангара: высота $x = 6$ м, длина $y = 4$ м. Цена сендвич-панелей составила $C_{\text{пан}} = 1600$ руб. за кв.м. Данные были подставлены в формулу (2) и был получен следующий результат:

$$C_{\text{осн}}(x, y, z, C_{\text{пан}}) = \pi \cdot \frac{4}{2} \cdot \left(\frac{4}{2} + 6 \right) \cdot 1600 = 80384 \text{ руб.}$$

Для определения стоимости шатрового ангара были введены значения высоты $x = 5$ м, длины $y = 4$ м, ширины $z = 3$ м. Был выбран обычный тип сендвич-панелей, цена которых составляет $C_{\text{пан}} = 1600$ руб. за кв. м. Эти данные были подставлены в формулу (4) и получен следующий результат:

$$\begin{aligned} C_{\text{осн}}(x, y, z, C_{\text{пан}}) = \\ = 2 \cdot \left(\frac{22}{30} \cdot 4 \cdot 3 + 5 \cdot \sqrt{(0,1 \cdot 4)^2 + \left(\frac{2}{3} \cdot 3\right)^2} + \sqrt{\left(\frac{1}{3} \cdot 3\right)^2 + (4 \cdot 0,4)^2} \cdot 5 \right) \times \\ \times 1600 = 3200 \cdot (8,8 + 10,2 + 9,4) = 90880 \text{ руб.} \end{aligned}$$

2 Техническое задание на создание программного продукта

Техническое задание размещено в приложении П-1.

3 Настройка среды разработки для операционных систем семейств Windows и Linux

В данном разделе будут описаны инструкции по установке и запуску дистрибутивов. Дистрибутивов будет всего 3 – Windows 10, Ubuntu 20.04 и OpenSUSE Leap 15.2.

Первоначально был рассмотрен дистрибутив Windows 10. Для разработки на данной ОС, понадобится:

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

Необходимо скачать и установить JDK (который содержит JRE), как показано на рисунке 4 - 5.

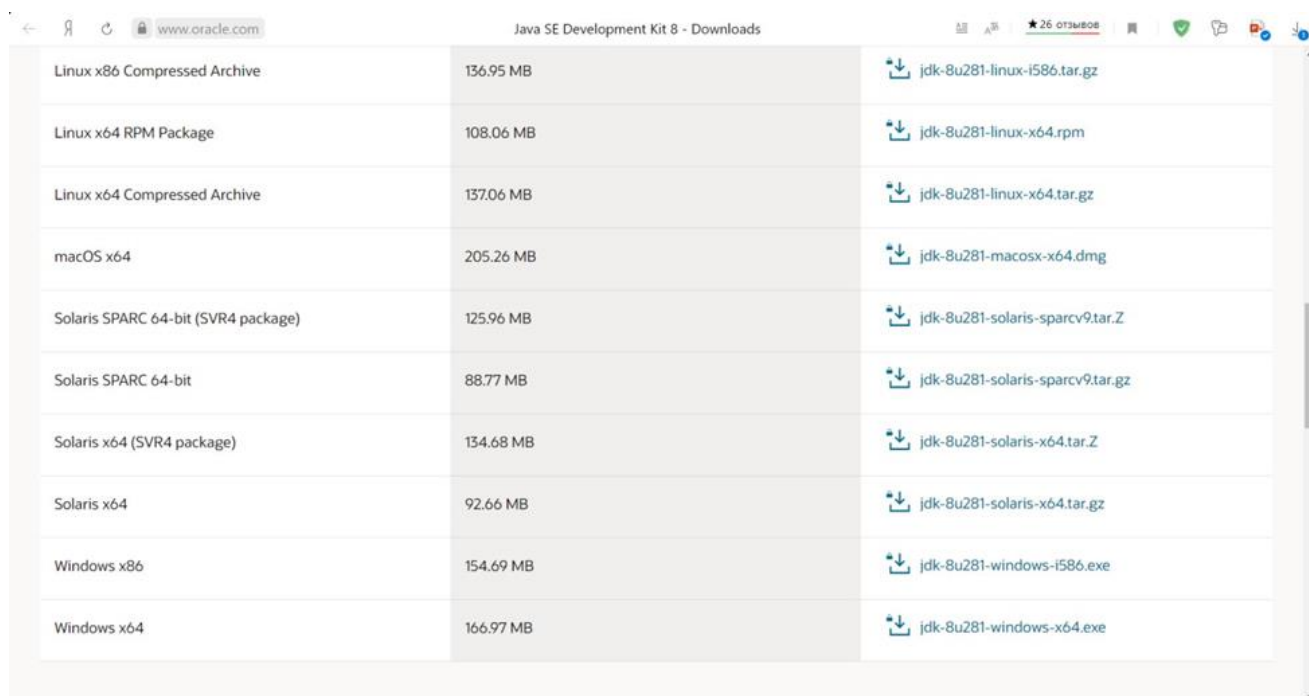


Рисунок 4 – Сайт с установщиком jdk

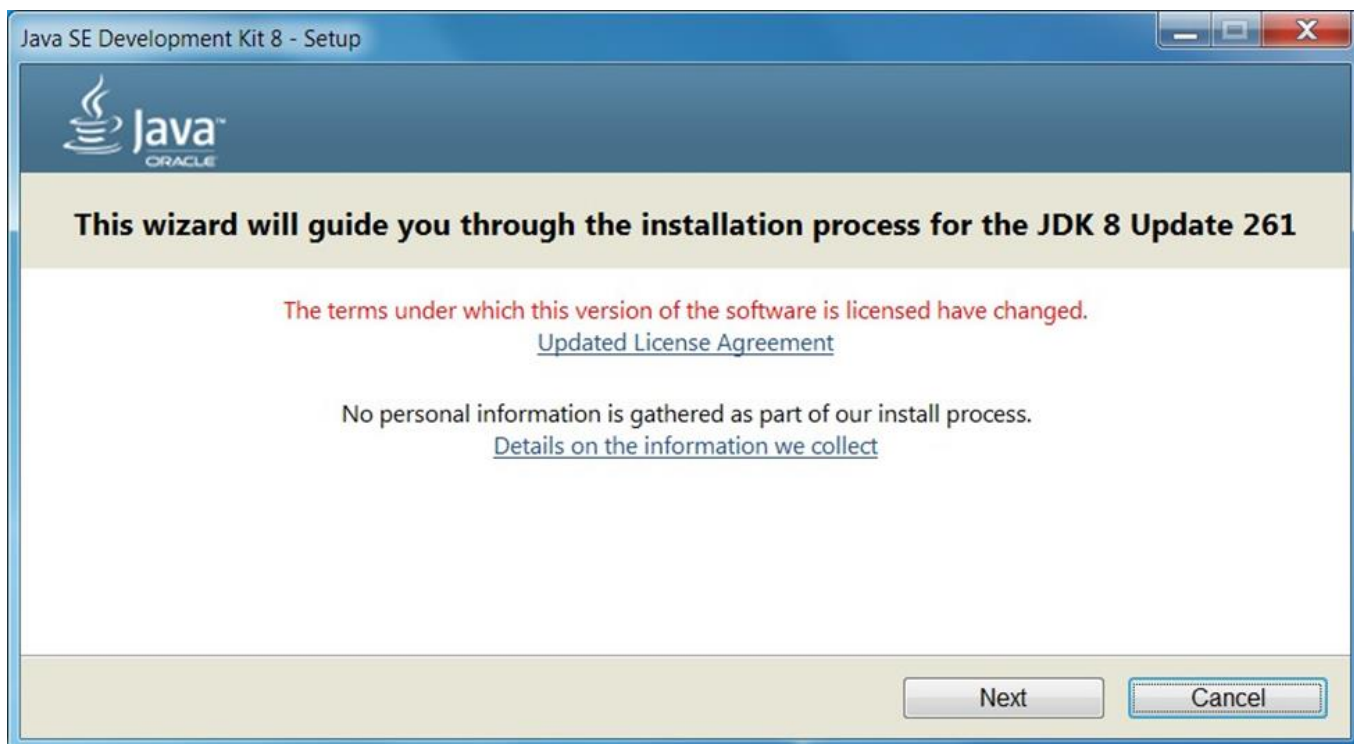


Рисунок 5 – Установка jdk

Следующим этапом будет загрузка установщика Eclipse IDE, как показано на рисунке 6.

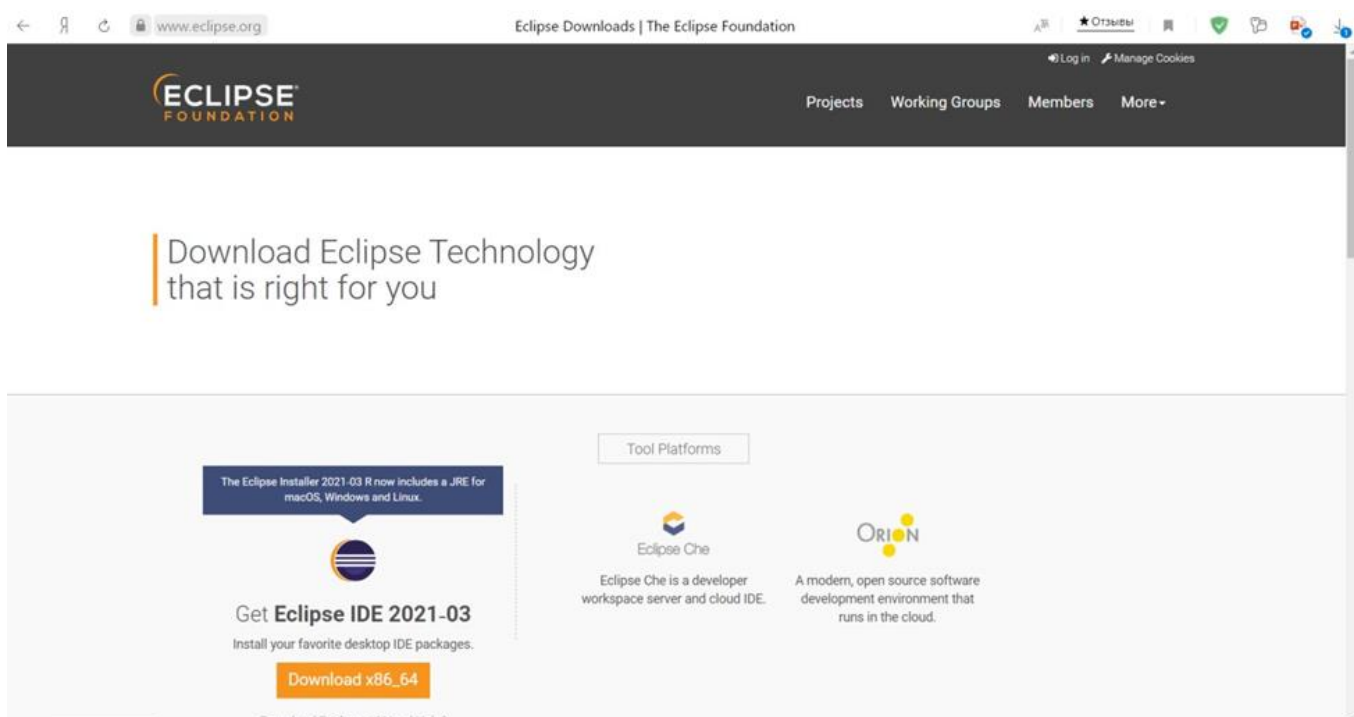


Рисунок 6 – Официальный сайт Eclipse IDE

Далее необходимо выбрать файл «Eclipse IDE for Java Developers», как продемонстрировано на рисунке 7. После завершения установки, необходимо

запустить программу и указать рабочую область.



Рисунок 7 – установщик Eclipse IDE

В установленном Eclipse IDE уже встроен Git, Maven и JUnit, поэтому устанавливать дополнительные модули не требуется.

Для работы в ОС Ubuntu 20.04 с инструментальной средой Eclipse необходимо выполнить обновление, которое выполняется команда `sudo apt upgrade` как показано на рисунке 8.

```
dyomin@rondol-VirtualBox: ~
завершено
dyomin@rondol-VirtualBox:~$ sudo apt upgrade
Чтение списков пакетов... Готово
Построение дерева зависимостей
Чтение информации о состоянии... Готово
Расчёт обновлений... Готово
Следующий пакет устанавливался автоматически и больше не требуется:
  linux-hwe-5.8-headers-5.8.0-44
Для его удаления используйте «sudo apt autoremove».
Следующие пакеты будут обновлены:
  bind9-dnsutils bind9-host bind9-libs dnsmasq-base firefox firefox-locale-en firefox-locale-ru git
  git-man gnome-shell gnome-shell-common initramfs-tools initramfs-tools-bin initramfs-tools-core
  isc-dhcp-client isc-dhcp-common libglib2.0-0 libglib2.0-bin libglib2.0-data libgnome-autoar-0-0
  libldb2 libnss-systemd libpam-systemd libssl1.1 libsystemd0 libudev1 libzstd1 linux-firmware
  openssh-client openssl python3-pil python3-update-manager systemd systemd-sysv systemd-timesyncd
  thunderbird thunderbird-gnome-support thunderbird-locale-en thunderbird-locale-en-us
  thunderbird-locale-ru udev update-manager update-manager-core update-notifier update-notifier-common
  wrsuppllicant
Обновлено 46 пакетов, установлено 0 новых пакетов, для удаления отмечено 0 пакетов, и 0 пакетов не обновл
ено.
Необходимо скачать 233 МВ/234 МВ архивов.
После данной операции объём занятого дискового пространства возрастёт на 40,7 МВ.
Хотите продолжить? [Д/н] у
Пол:1 http://ru.archive.ubuntu.com/ubuntu focal-updates/main amd64 udev amd64 245.4-4ubuntu3.5 [1 366 kB]
```

Рисунок 8 – Выполнение команд в ОС Ubuntu

В ОС OpenSUSE Leap15.2 выполнение обновления выполняется командой `sudo zypper update`, что показано в соответствии с рисунком 9.

```
Rondol@localhost.localdomain:~
Rondol@localhost:~$ sudo zypper update
[sudo] пароль для root:
Загрузка данных о репозиториях...
Чтение установленных пакетов...

Будут обновлены 4 пакета:
  tar tar-lang tar-rmt yast2-bootloader

4 пакета для обновления.
Общий размер загрузки: 0 В. Уже кэшировано: 672,8 KiB. После этой операции будет использовано дополнительно 797,0 В.
Продолжить? [у/н/в/...? выводит все параметры] (у): у
В кэше tar-rmt-1.30-lp152.4.3.1.x86_64.rpm (1/4), 40,1 KiB ( 57,4 KiB после распаковки
В кэше yast2-bootloader-4.2.28-lp152.2.9.1.x86_64.rpm (2/4), 104,4 KiB (237,9 KiB после распаковки
В кэше tar-1.30-lp152.4.3.1.x86_64.rpm (3/4), 210,0 KiB (470,6 KiB после распаковки
В кэше tar-lang-1.30-lp152.4.3.1.noarch.rpm (4/4), 318,4 KiB ( 1,9 MiB после распаковки

Проверка на конфликты файлов: .....[готово]
(1/4) Установка: tar-rmt-1.30-lp152.4.3.1.x86_64 .....[готово]
(2/4) Установка: yast2-bootloader-4.2.28-lp152.2.9.1.x86_64 .....[готово]
(3/4) Установка: tar-1.30-lp152.4.3.1.x86_64 .....[готово]
(4/4) Установка: tar-lang-1.30-lp152.4.3.1.noarch .....[готово]

Rondol@localhost:~$
```

Рисунок 9 – Выполнение команд в ОС OpenSUSE

После того как обновление будет завершено, необходимо выполнить установку JDK выполнив команду для Ubuntu: `sudo apt install openjdk-8-jdk`, как показано на рисунке 10, и для OpenSUSE `sudo zypper install java-1_8_0-openjdk`, как на рисунке 11.

```
dyomin@rondol-VirtualBox: ~
Настраивается пакет libnss-systemd:amd64 (245.4-4ubuntu3.5) ...
Настраивается пакет libpam-systemd:amd64 (245.4-4ubuntu3.5) ...
Настраивается пакет update-notifier (3.192.30.6) ...
Настраивается пакет update-manager (1:20.04.10.6) ...
Обрабатываются триггеры для man-db (2.9.1-1) ...
Обрабатываются триггеры для dbus (1.12.16-2ubuntu2.1) ...
Обрабатываются триггеры для desktop-file-utils (0.24-1ubuntu3) ...
Обрабатываются триггеры для mime-support (3.64ubuntu1) ...
Обрабатываются триггеры для hicolor-icon-theme (0.17-2) ...
Обрабатываются триггеры для gnome-menus (3.36.0-1ubuntu1) ...
Обрабатываются триггеры для libc-bin (2.31-0ubuntu9.2) ...
Обрабатываются триггеры для initramfs-tools (0.136ubuntu6.4) ...
update-initramfs: Generating /boot/initrd.img-5.8.0-48-generic
dyomin@rondol-VirtualBox:~$ sudo apt install openjdk-8-jdk
Чтение списков пакетов... Готово
Построение дерева зависимостей
Чтение информации о состоянии... Готово
Уже установлен пакет openjdk-8-jdk самой новой версии (8u282-b08-0ubuntu1~20.04).
Следующий пакет устанавливался автоматически и больше не требуется:
  linux-hwe-5.8-headers-5.8.0-44
для его удаления используйте «sudo apt autoremove».
Обновлено 0 пакетов, установлено 0 новых пакетов, для удаления отмечено 0 пакетов, и 0 пакетов не обновле
но.
dyomin@rondol-VirtualBox:~$
```

Рисунок 10 – Выполнение команд для установки JDK в ОС Ubuntu

```
Rondol@localhost.localdomain:~
Rondol@localhost:~$ sudo zypper update
[sudo] пароль для root:
Загрузка данных о репозиториях...
Чтение установленных пакетов...

Будут обновлены 4 пакета:
tar tar-lang tar-rmt yast2-bootloader

4 пакета для обновления.
Общий размер загрузки: 0 В. Уже кэшировано: 672,8 KiB. После этой операции будет использовано дополнительно 797,0 В.
Продолжить? [y/n/v/...? выводит все параметры] (y): y
В кэше tar-rmt-1.30-lp152.4.3.1.x86_64.rpm (1/4), 40,1 KiB ( 57,4 KiB после распаковки
В кэше yast2-bootloader-4.2.28-lp152.2.9.1.x86_64.rpm (2/4), 104,4 KiB (237,9 KiB после распаковки
В кэше tar-1.30-lp152.4.3.1.x86_64.rpm (3/4), 210,0 KiB (470,6 KiB после распаковки
В кэше tar-lang-1.30-lp152.4.3.1.noarch.rpm (4/4), 318,4 KiB ( 1,9 MiB после распаковки

Проверка на конфликты файлов: .....[готово]
(1/4) Установка: tar-rmt-1.30-lp152.4.3.1.x86_64 .....[готово]
(2/4) Установка: yast2-bootloader-4.2.28-lp152.2.9.1.x86_64 .....[готово]
(3/4) Установка: tar-1.30-lp152.4.3.1.x86_64 .....[готово]
(4/4) Установка: tar-lang-1.30-lp152.4.3.1.noarch .....[готово]

Rondol@localhost:~$
```

Рисунок 11 – Выполнение команд для установки JDK в ОС OpenSUSE

Для установки Eclipse необходимо воспользоваться браузером для загрузки дистрибутива Eclipse с официального сайта. Далее следует распаковать дистрибутив в домашнюю папку пользователя и выполнить установочный файл из каталога eclipse, как показано на рисунке 12-13.

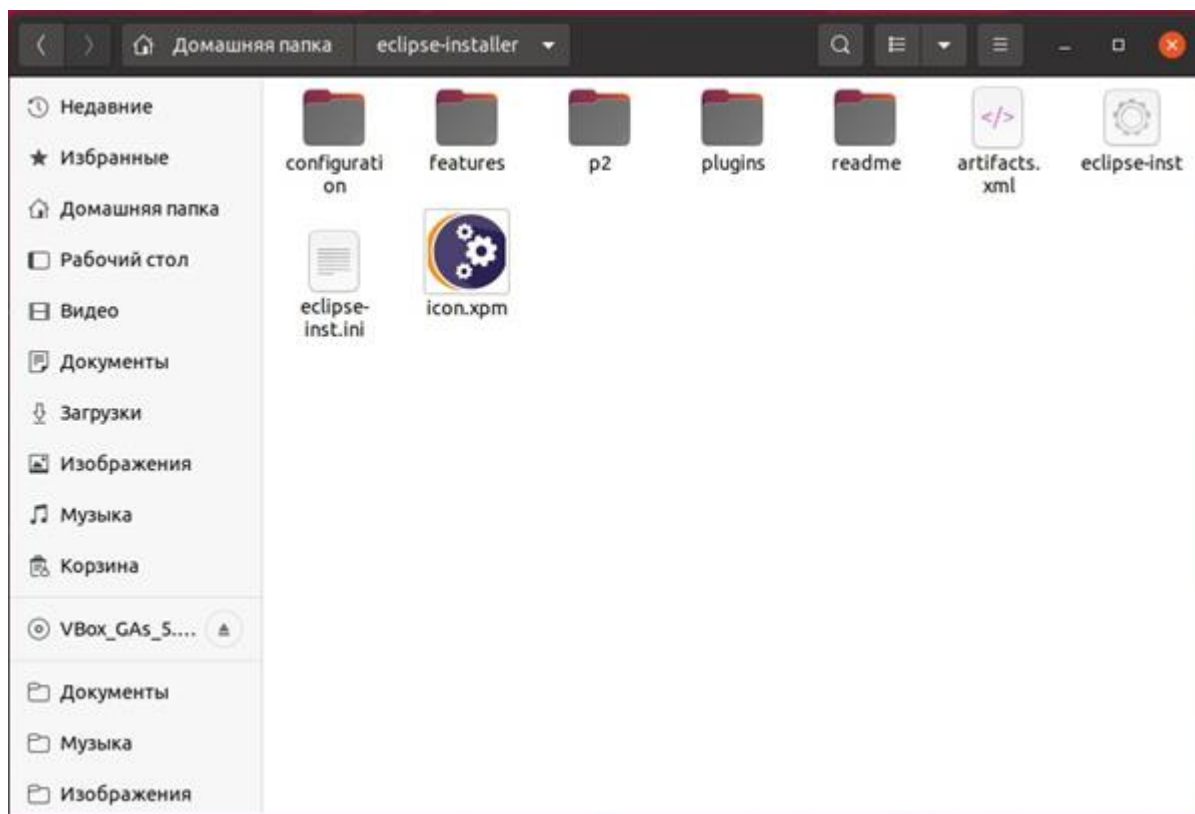


Рисунок 12 – Домашняя папка пользователя с файлом eclipse в ОС Ubuntu

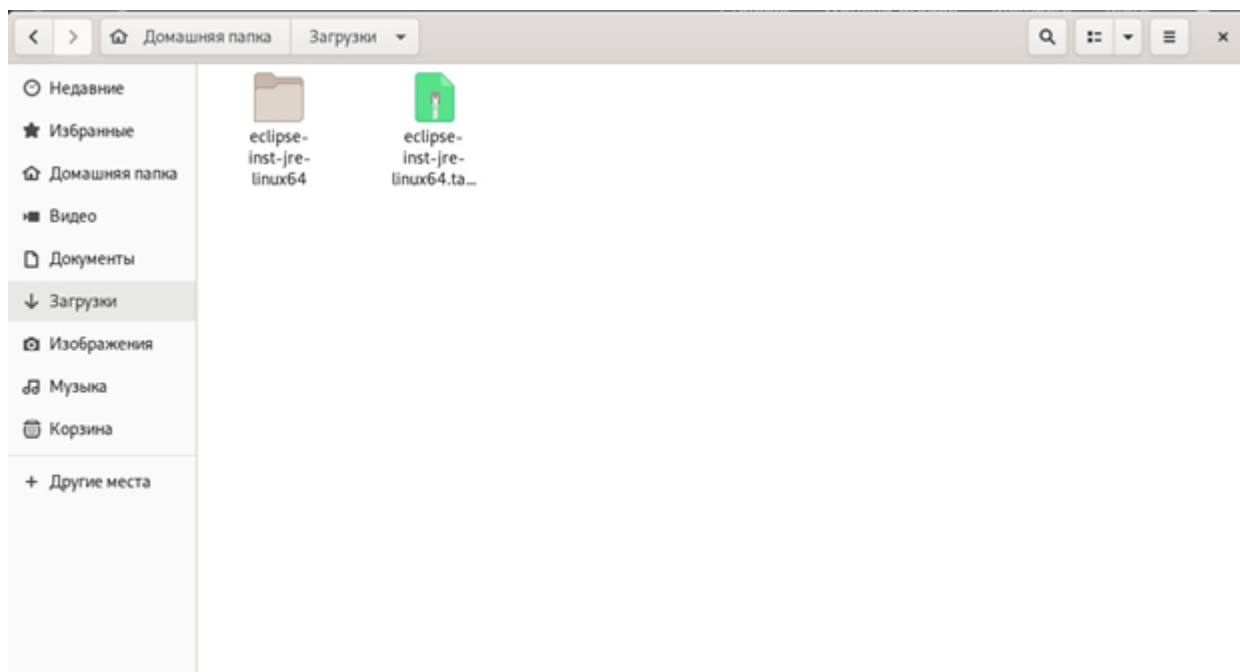


Рисунок 13 – Домашняя папка пользователя с выполняемым файлом в ОС OpenSUSE

После запуска Eclipse откроется окно приветствия, которое нужно закрыть и перейти в Package Explorer. Дальнейшая работа в Eclipse не отличается от работы в операционной системе Windows.

4 Настройка среды разработки для подключения к системе контроля версий

Для работы с Git понадобится установленный Eclipse IDE. После его запуска.

В открывшемся интерфейсе, необходимо нажать на кнопку «Open Perspective», затем выбрать Git, как показано на рисунке 14.

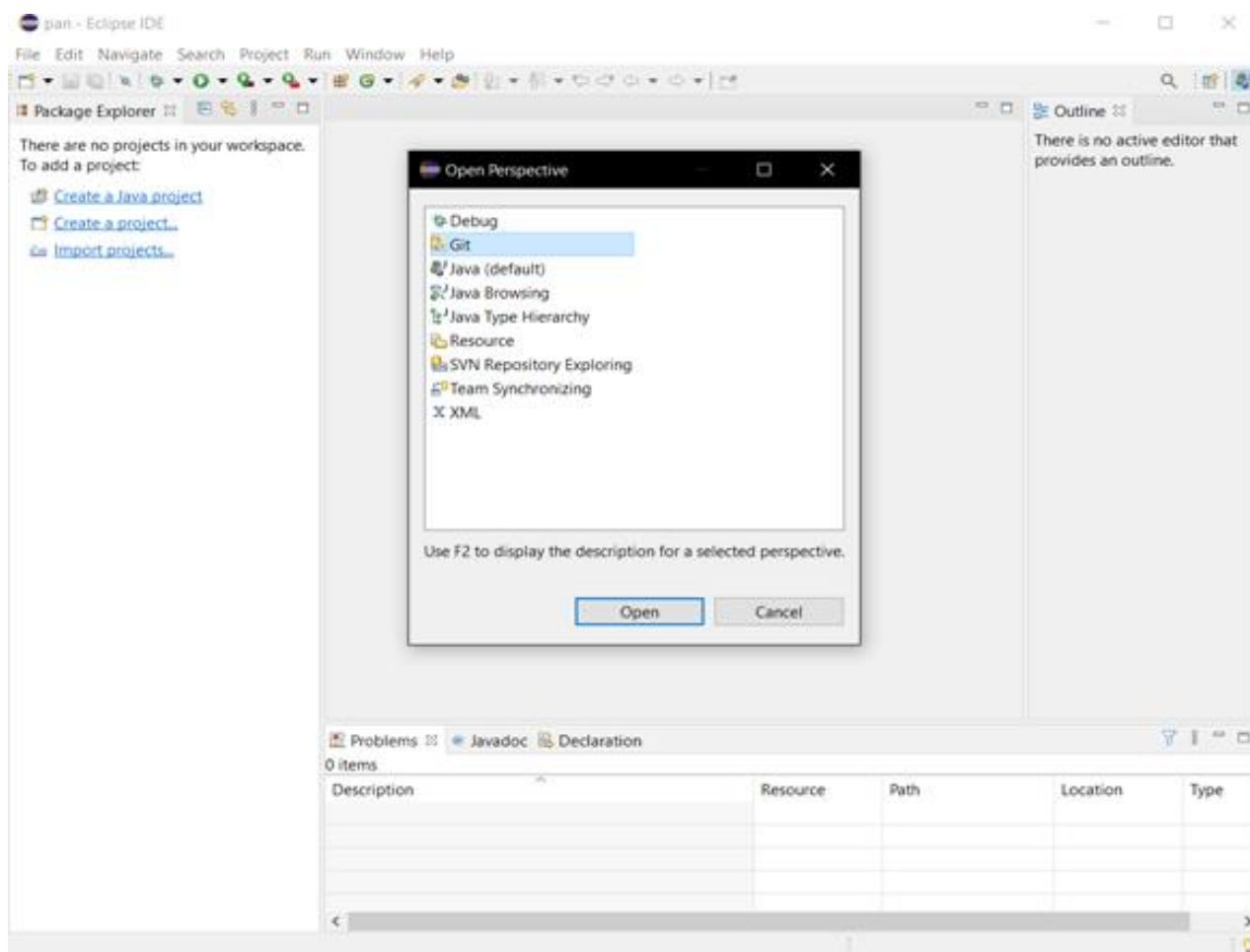


Рисунок 14 – Интерфейс Eclipse IDE

В данной проекции необходимо нажать на «Clone a Git repository», и заполнить все поля:

URI – ссылка репозитория, https://github.com/Panaev27/course_workPI-223-Team-27;

User – логин пользователя GitHub.com;

Password – пароль от аккаунта;

Затем нажать «Next», как показано на рисунке 15.

Clone Git Repository

Source Git Repository

Enter the location of the source repository.

Location

URI: Local Folder... Local Bundle File...

Host:

Repository path:

Connection

Protocol: Port:

Authentication

User:

Password:

☒ Store in Secure Store

? < Back Next > Finish Cancel

Рисунок 15 – Клонирование репозитория

Если всё выполнено верно, Eclipse IDE скопирует все имеющиеся данные с удалённого репозитория в локальный репозиторий. Для работы с самим проектом, так же нужно импортировать проект, как показано на рисунке 15.

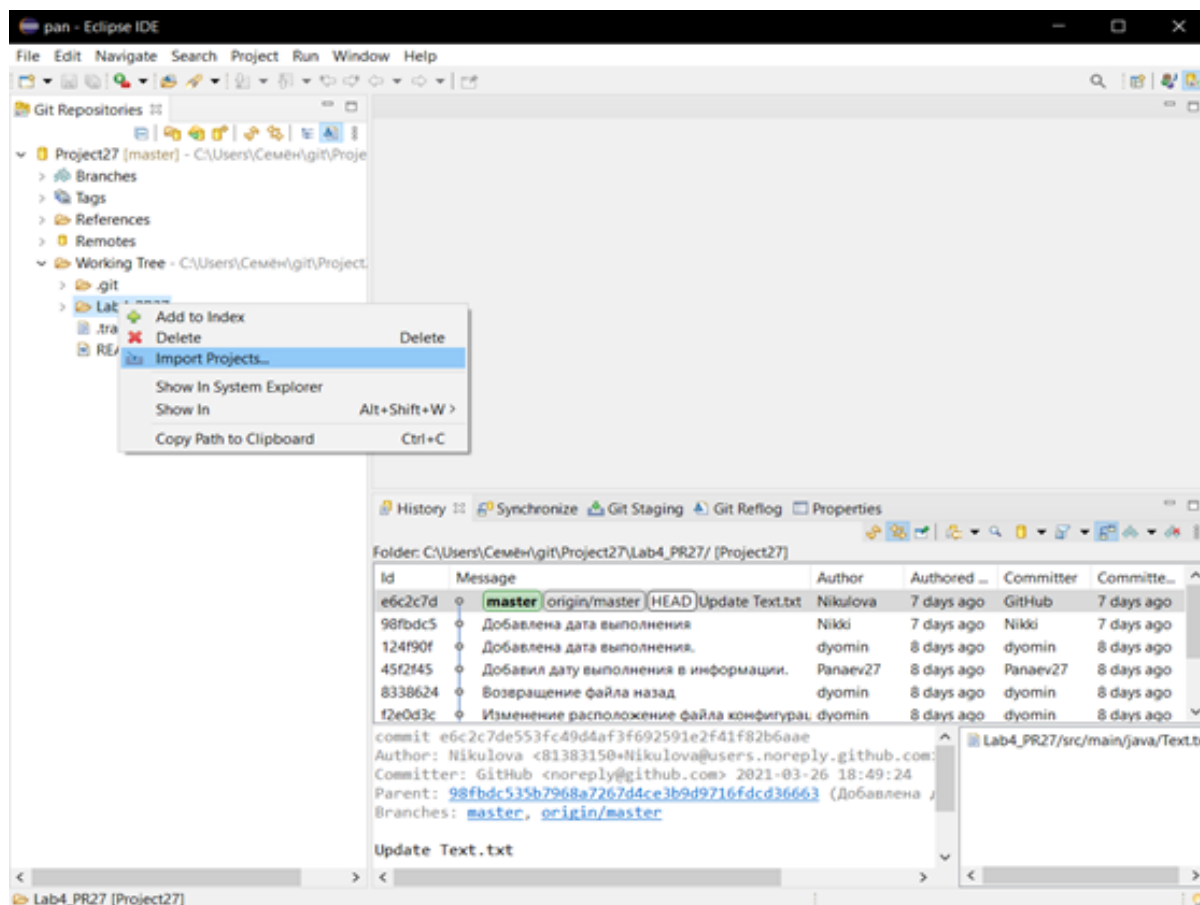


Рисунок 15 – Импорт проекта

После импорта проекта в Eclipse, в разделе «java» появится рабочая копия, как показано на рисунке 16.

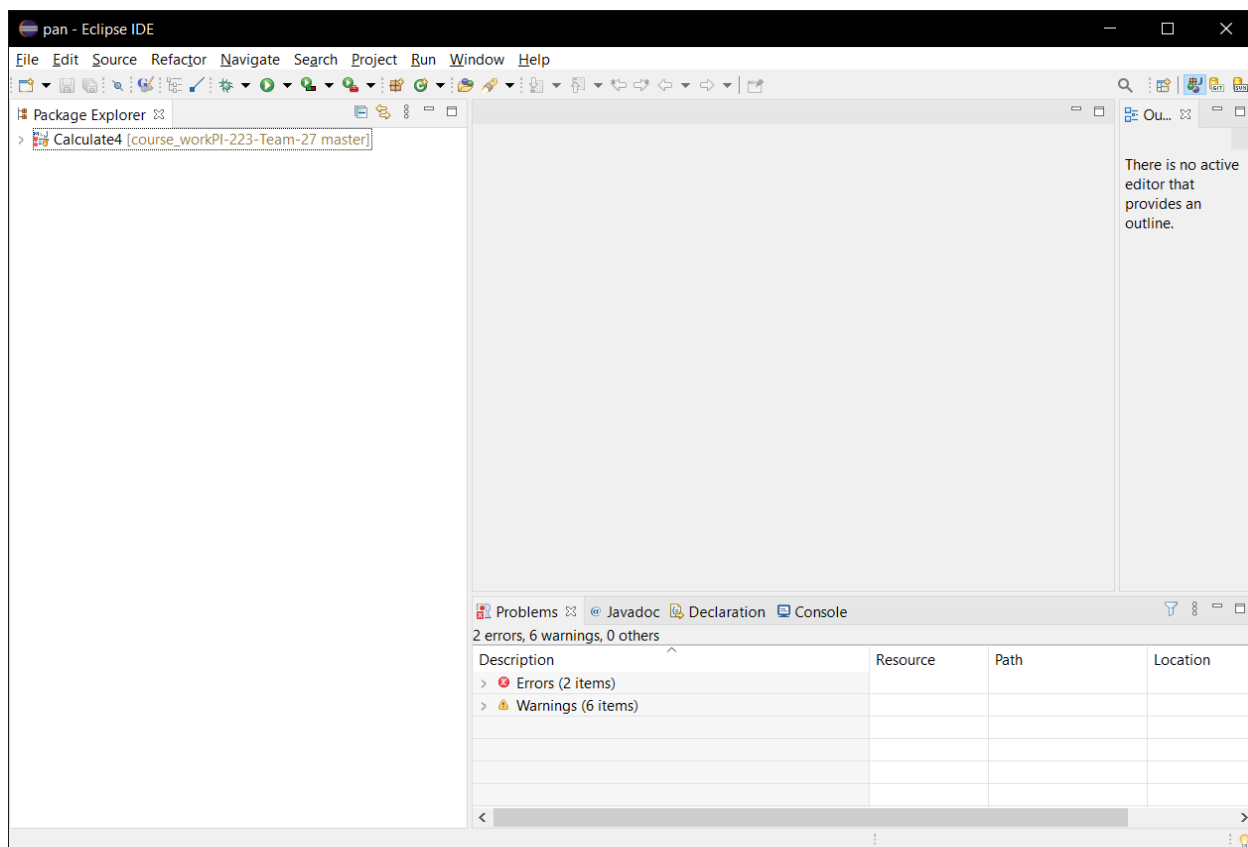


Рисунок 16 – рабочая копия проекта

5 Реализация исходного кода по зонам ответственности

Программное обеспечение (калькулятор стоимости быстровозводимых ангаров) реализуется в виде веб-приложения через размещение на Heroku.

Так же для правильного выполнения работы будут разделены зоны ответственности для каждого из участников разработки в таблице 1.

Таблица 2 – Зона ответственности

№	ФИО разработчика/модератора	Зона ответственности
1	Панаев Семён Игоревич	Работа с интерфейсом: WriterInFile. Работа с классами: ArcHangar, TentHangar, VoxHangar. Работа с конфигурацией: web.xml, pom.xml, Procfile, .travis.yml.
2	Никулова Ксения Дмитриевна	Работа с абстрактными классами: Hangar. Работа с JSP страницами: Form.jsp, Results.jsp, Authorization.jsp, Edit.jsp. Работа со стилем: style.css
3	Дёмин Роман Алексеевич	Работа с классами: Calculator, PDFWriter, Authorization, AuthBaseController, AuthManager.

Ссылка на репозиторий с программным кодом:
https://github.com/Panaev27/course_workPI-223-Team-27

Программный код, используемый для создания ПО размещён в приложении.

6 Сборка и тестирование программного продукта

Данный раздел содержит 3 пункта:

1. Таблица 1 с описанием unit-тестов и отвечающих за них разработчиков
2. Описание структуры проекта по каталогам, включая описание pom.xml.
3. Инструкция по сборке проекта.

Таблица 3 – Описание UNIT-тестов

№	ФИО разработчика/ модератора	Описание UNIT- теста	№ приложения
1	Панаев Семён Игоревич	Тестирует методы GetHangarArea классов-наследников Hangar на соответствие математической модели.	см. Приложение П-3
2	Никулова Ксения Дмитриевна	Тестирует методы абстрактного класса Hangar на соответствие математической модели.	см. Приложение П-4
3	Дёмин Роман Алексеевич	Тестирует класс PDFWriter на то, создаётся ли файл Hangar.pdf	см. Приложение П-5

Описание структуры проекта по каталогам, как показано на рисунке 17, проект включает в себя такие каталоги как:

- artifacts – содержит артефакты (результат разработки). В данном проекте артефактом является файл Calculate.war. В данном каталоге так же содержится контейнер сервлетов webapp-runner.java;
- src/main/java – содержит исходный программный код, разложенный по пакетам в виде файлов с расширением java (WriterInFile.java,

ArcHangar.java, AuthBaseController.java, AuthManager.java,
Authorization.java, BoxHangar.java, Calculator.java, Hangar.java,
PDFWriter.java, TentHangar.java);

- src/main/webapp – содержит компоненты, относящиеся к веб-части;
- src/test – содержит unit-тесты;
- target – содержит выходную информацию при сборке проекта.

Такой каталог как src/main/webapp содержит следующие каталоги, которые перечислены ниже:

- Каталог css, содержащий css стиль под названием style.css, определяющий стиль веб-страниц.
- Каталог WEB-INF. В нём расположен файл web.xml, определяющий конфигурацию веб-приложения при развёртывании.
- Jsp-страницы Authorization.jsp, Edit.jsp, Form.jsp, Results.jsp. Эти файлы определяют содержание веб-страниц.

Так же данный проект включает в себя файлы, которые предоставлены ниже:

- README.md – описание проекта.
- pom.xml – конфигурация сборки Maven проекта.
- PROCFILE – конфигурация для сервиса Heroku.
- src/main/AuthBase.txt – файл с первоначальной базой учётных записей пользователей.

В файле pom.xml описывается конфигурация для сборки проекта Maven'ом.

В описание включается:

- Описание проекта (его название, версия и т.д.).
- После идёт описание всех плагинов: плагин компиляции (maven-compiler-plugin), плагины maven (maven-compiler-plugin) с изменением конфигурации в виде добавления артефакта com.heroku.webapp-runner, плагин для компиляции war-файлов (maven-war-plugin).
- Так же включено описание зависимостей: модульное тестирование (junit),

сервлеты (javax.servlet), Unified Expression Language (el-api), создание pdf файлов (com.itextpdf), контейнер сервлетов (com.github.jsimone.webapp-runner).

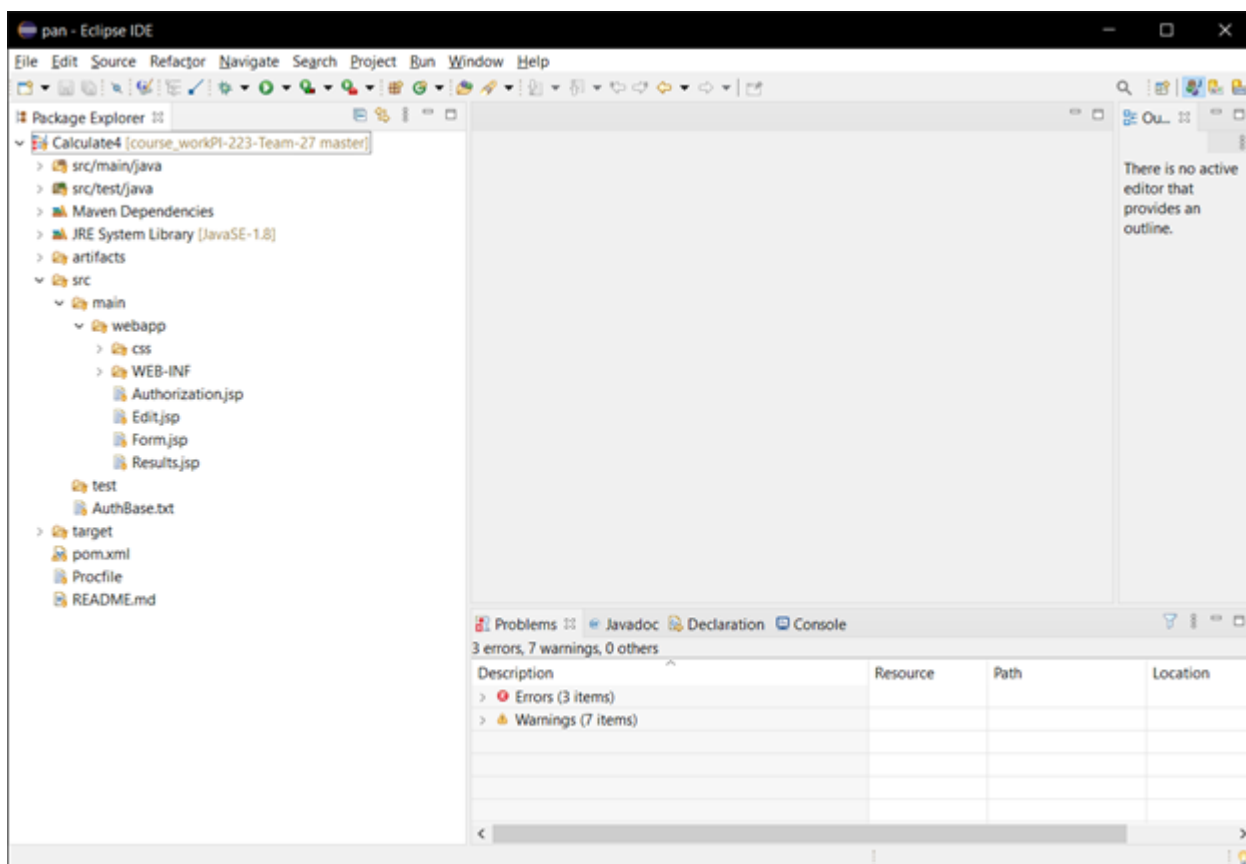


Рисунок 17 – Структура проекта Calculate4

Процесс сборки проекта:

Сборка проекта будет проходить через сборщик проектов Maven. Для начала нужно запустить Eclipse IDE и скачать проект с репозитория. Необходимо нажать на весь проект, найти вкладку «Run as» и выбрать «Maven build...», как показано на рисунке 18.

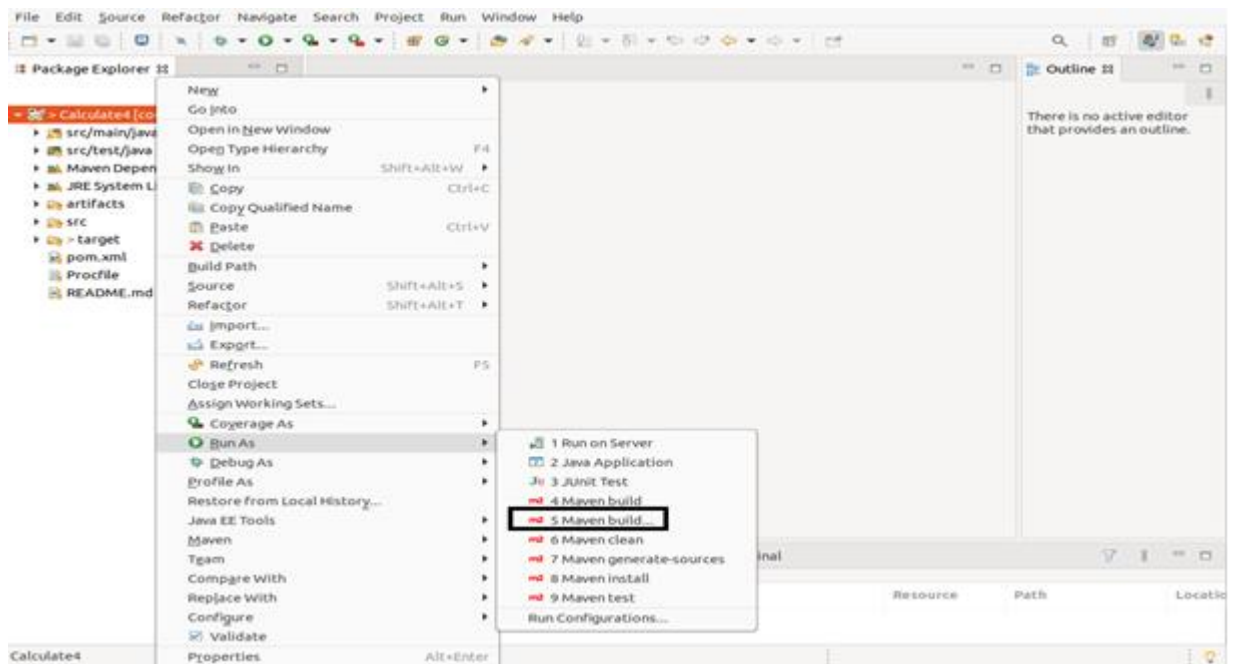


Рисунок 18 – Работа в Eclipse

В открывшемся окне вводим в поле Goals «package» и нажимаем «Run», как показано на рисунке 19.

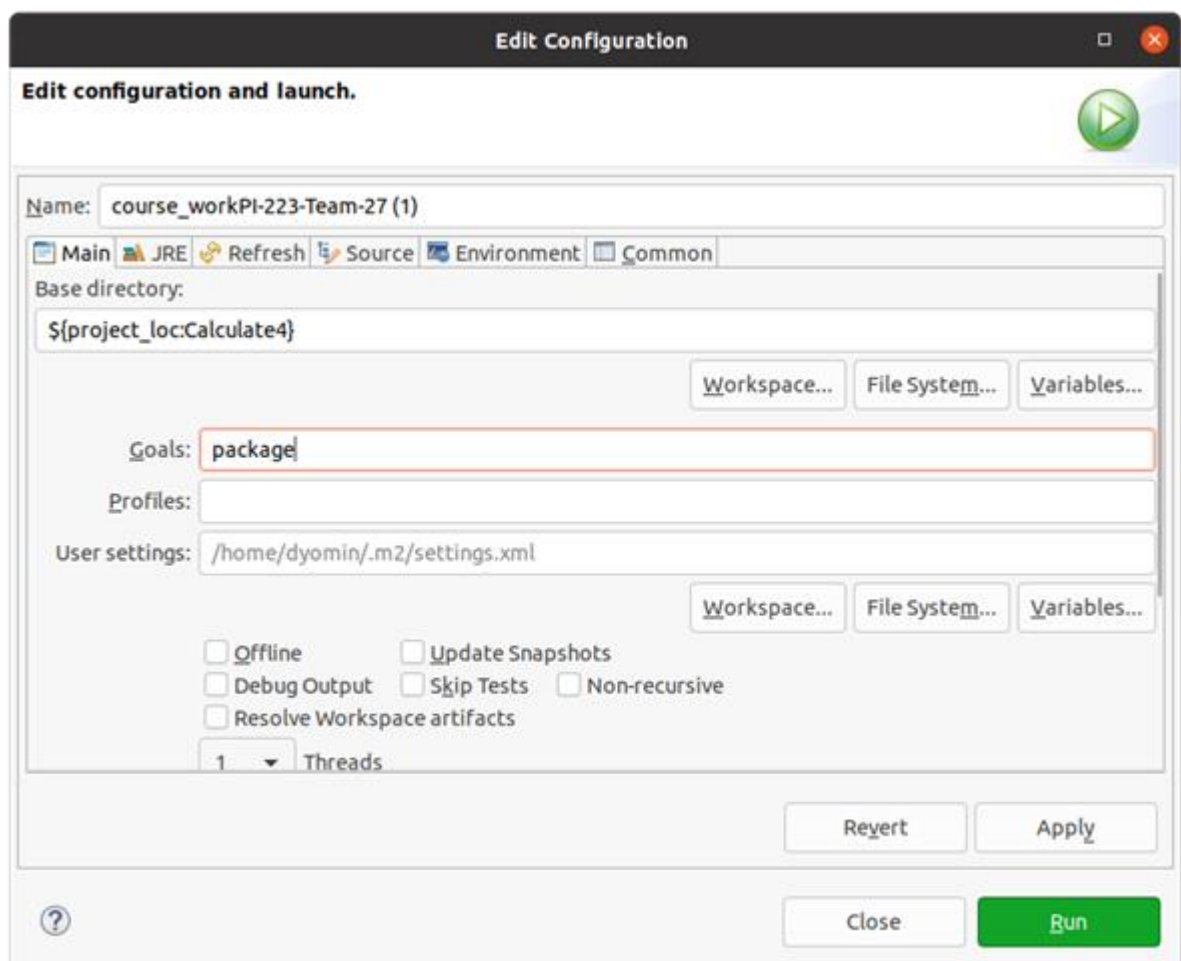


Рисунок 19 – Сборка проекта

В результате получаем файл, который будет храниться в artifacts, как продемонстрировано на рисунке 20.

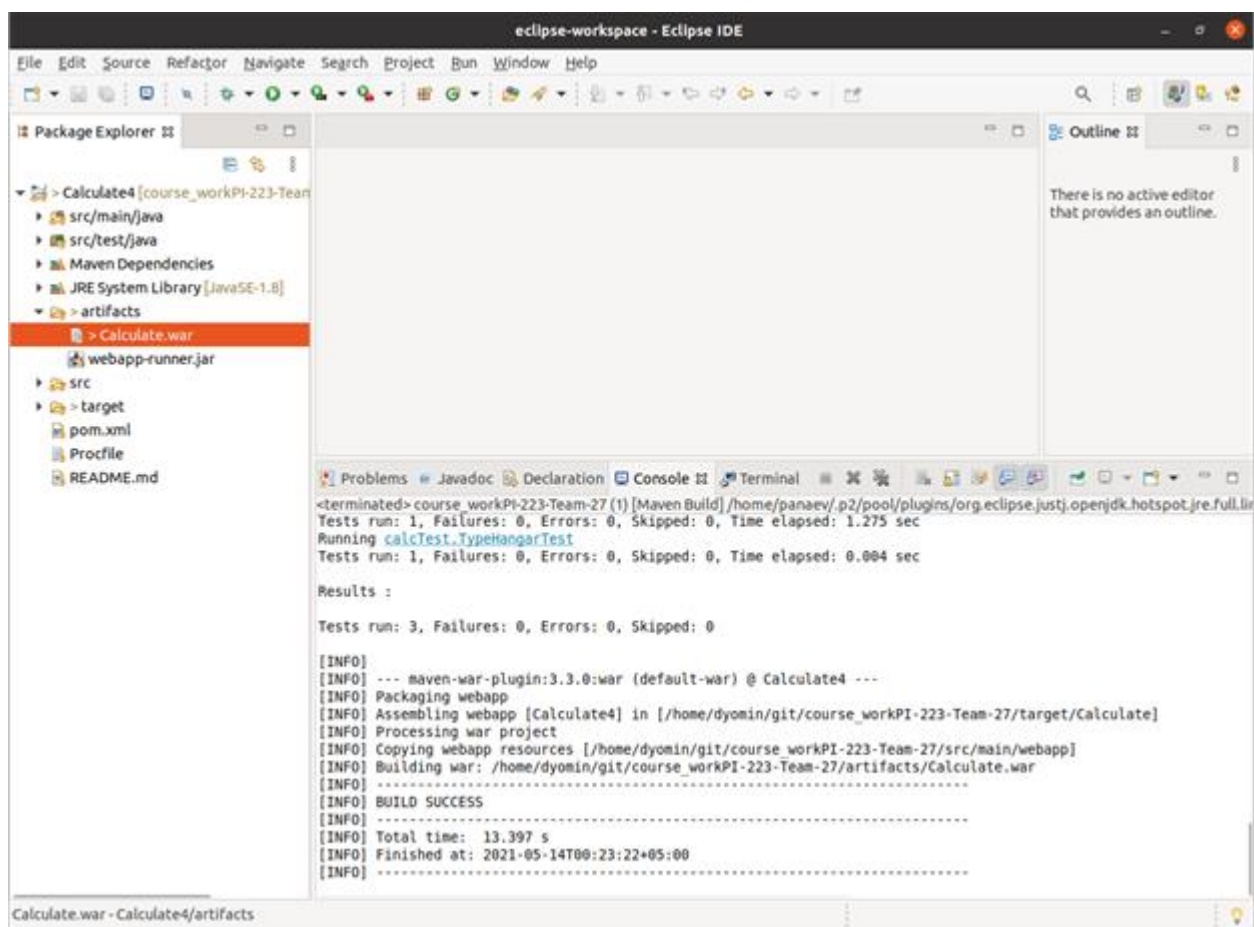


Рисунок 20 – Собранный проект

7 Настройка программной среды для развертывания и запуска программного продукта

Для развёртывания и запуска программного продукта «Калькулятор стоимости быстровозводимых ангаров» используется конвейер (pipeline) Travis CI+Heroku.

Перед настройкой была совершена регистрация на сервисах Travis CI и Heroku. Так же была выполнена синхронизация учётных записей этих сервисов с сервисом Github.

Для настройки работы сервиса непрерывной интеграции Travis CI в программном продукте был создан конфигурационный файл `.travis.yml`. В него были записаны настройки для сборки проекта и дальнейшего развёртывания. Как видно на рисунке 21, в этом файле определяется используемый язык и действия в каждой фазе.

В промежуточных фазах `after_script`, `after_success`, `after_failure` выводятся сообщения о результатах фазы, в `before_script` выводится сообщение о начале фазы `script`.

В фазе `script` запускается выполнения сборки с помощью Maven.

Фаза `deploy` определяет то, как будет происходить развёртывание проекта. В качестве конечного пункта развёртывания выбрано приложение сервиса Heroku под названием `sw-pi223-27`. Развёртывание происходит из репозитория `Panaev27/course_work-223-Team-27`. Для успешного развёртывания указан зашифрованный `api`-ключ.

```

1  language: java
2  before_script:
3  - echo "Starting build"
4  script:
5  - mvn clean package
6  - mvn package
7  after_script:
8  - echo "Script finished"
9  after_success:
10 - echo "Build ready"
11 after_failure:
12 - echo "Build failure"
13 deploy:
14   provider: heroku
15   api_key:
16     secure: WaHagJWJYAMQKQ72Db44n8OV0Z0evdaAH/uFjn9kXaVEHaR1B2UG9XA
17   app: cw-pi223-27
18   on:
19     repo: Panaev27/course_workPI-223-Team-27

```

Рисунок 21 – Конфигурационный файл

Во время работы веб приложения на сервере работают следующие файлы:

- Файлы классов WriterInFile.class, ArcHangar.class, AuthBaseController.class, AuthManager.class, Authorization.class, BoxHangar.class, Calculator.class, Calculator\$RequestCalc.class Hangar.class, PDFWriter.class, TentHangar.class;
- Библиотеки el-api-2-2.jar, itextpdf-5.5.13.jar, javax.servlet-api-3.1.0.jar;
- Файл с конфигурацией веб страниц web.xml;
- Страницы Authorization.jsp, Edit.jsp, Form.jsp, Results.jsp.
- Стилль style.css.

Для работы с программным продуктом пользователю потребуется компьютер, способный поддерживать браузеры Google Chrome, Mozilla Firefox, Yandex Browser, а также с доступом в интернет. Пример успешного запущенного приложения через браузер Google Chrome приведён на рисунке 22.

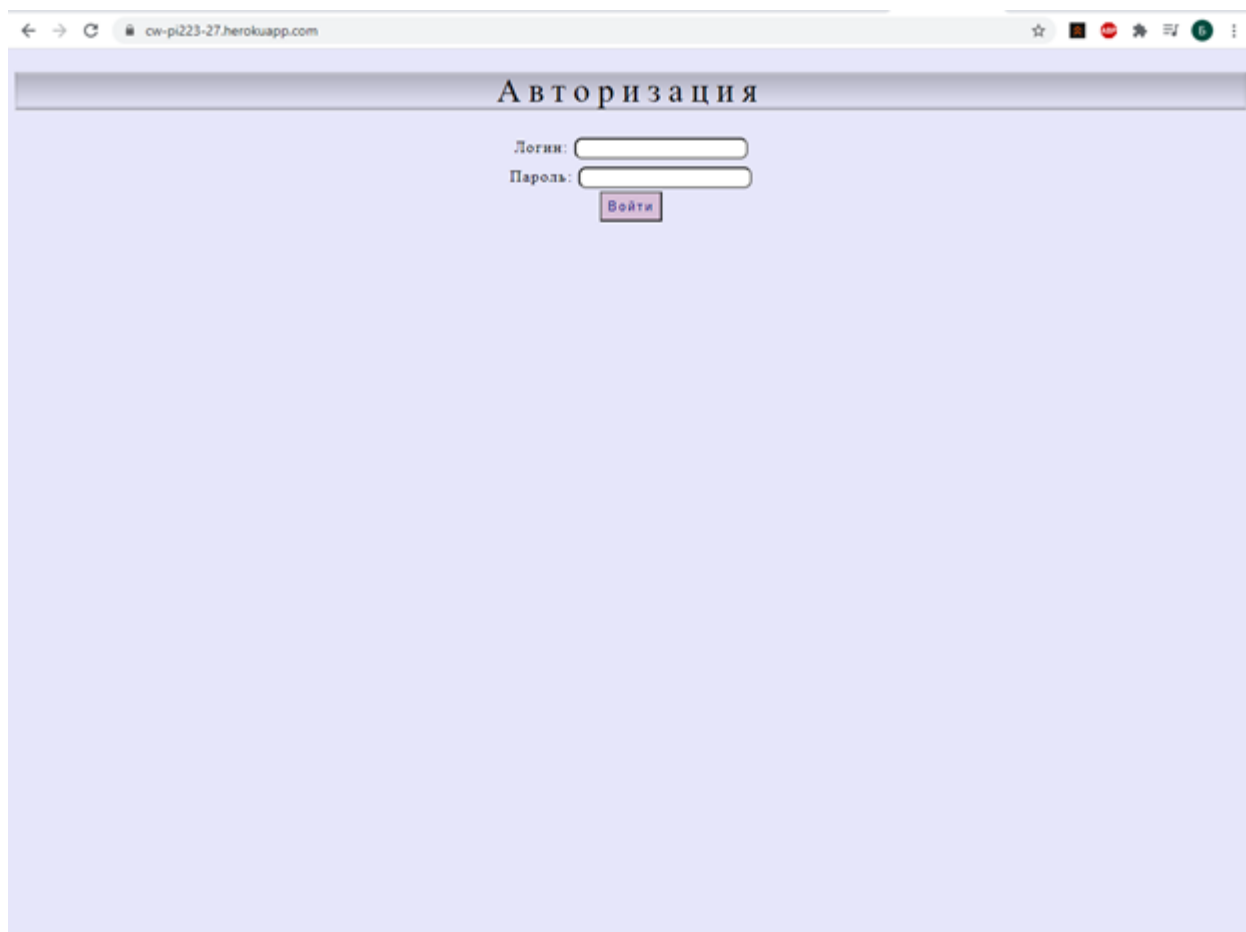


Рисунок 22 – Пример запущенного приложения

8 Руководство пользователя программного продукта

Руководство пользователя размещено в приложении П-2.

Заключение

В ходе написания курсовой работы были раскрыты задачи, которые необходимо решить, используя кроссплатформенный программный продукт.

При написании курсовой работы были изучены методические указания, а также приобретены навыки работы со средой разработки, веб-сервисом Travis CI, облачной платформой для развертывания приложений Heroku и разработан кроссплатформенный программный продукт на языке Java с использованием системы контроля версий.

Для отлаженной разработки программного продукта были разработаны этапы работы и их сроки.

Поставленные цели курсовой работы были достигнуты, поставленные задачи выполнены.

Приложения

ПРИЛОЖЕНИЕ П-1

Техническое задание

ФБГОУ ВО УФИМСКИЙ ГОСУДАРСТВЕННЫЙ АВИАЦИОННЫЙ ТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ

СОГЛАСОВАНО

Доцент кафедры АСУ ФГОУ
УГАТУ
Казанцев А.В.

Личная Расшифровка

подпись подписи

27.03.2021

УТВЕРЖДАЮ

Студент группы ПИ-223
ФИРТ ФГОУ УГАТУ,
модератор Панаев С.И.

Личная Расшифровка

подпись подписи

26.03.2021

Калькулятор стоимости строительства быстровозводимых ангаров

Техническое задание

ЛИСТ УТВЕРЖДЕНИЯ

581.45338009.25777-01 90 01-ЛУ

(Электронный)

Листов 16

СОГЛАСОВАНО

Доцент кафедры АСУ ФГОУ
УГАТУ
Казанцев А.В.

Личная Расшифровка

подпись подписи

27.03.2021

УТВЕРЖДАЮ

Студент группы ПИ-223
ФИРТ ФГОУ УГАТУ,
модератор Панаев С.И.

Личная Расшифровка

подпись подписи

26.03.2021

Подп. и дата	
Инв. и дубл.	
Взам. инв. и	
Подп. и дата	
Инв. и подл.	

2021

Утвержден

581.45338009.25777-01 90 01-ЛУ

Калькулятор стоимости строительства быстровозводимых ангаров

Техническое задание

581.45338009.25777-01 90 01-ЛУ

(Электронный)

Листов 16

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

2021

37

1. ВВЕДЕНИЕ

Калькулятор «Стоимость строительства быстровозводимых ангаров» – это программа, помогающая сформировать представление о предварительной цене ангара с определенными выбранными параметрами.

Быстровозводимые ангары – это объект, который собирается наподобие конструктора. Зачастую такие строения можно отнести к временным, что существенно упростит все согласования. Но тут всё во многом зависит от типа фундамента. Они не рассчитаны на последующие разборки, транспортирование и монтаж, а срок их службы может быть аналогичен сроку службы капитальных домов.

Для удобства использования программы «Калькулятор стоимости строительства быстровозводимых ангаров» было решено использовать две основные технологии возведения: каркасно-щитовые и каркасно-рамочные.

2. ОСНОВАНИЯ ДЛЯ РАЗРАБОТКИ

ПО разрабатывается специально для компании «Daedalus industries» в рамках курсовой работы.

3. НАЗНАЧЕНИЕ РАЗРАБОТКИ

3.1 Функциональное назначение

Программа предназначена для расчета предварительной цены быстровозводимого ангара на основании введенных пользователем параметров ангара с возможностью формирования печатной формы с записанными результатами. Посетитель сможет ознакомиться с возможными типами постройки, разновидностями арок ангара, вид панелей, а также с дополнительными настраиваемыми параметрами в самой программе.

Для администратора программа предоставляет возможность управлять учётными данными пользователя.

3.2 Эксплуатационное назначение

Программа должна эксплуатироваться на компьютерах пользователей через веб-браузер посредством клиент-серверной архитектуры.

4. ТРЕБОВАНИЯ К ПРОГРАММЕ ИЛИ ПРОГРАММНОМУ ИЗДЕЛИЮ

4.1. Требования к функциональным характеристикам

Пользователь взаимодействует с программой через интерфейс предоставляемый на сайте по интернет адресу. Первой формой, с которой встречается пользователь, является форма авторизации. Через неё пользователь вводит логин и пароль своей учётной записи или создаёт новую учётную запись. Программа проверяет тип пользователя и открывает соответствующий интерфейс.

Интерфейс пользователя предоставляет возможность:

- заполнить поля с предполагаемыми параметрами постройки ангара;
- произвести расчёт стоимости постройки данного ангара;
- сформировать печатную форму.

Интерфейс администратора предоставляет те же функции, но на нём добавляется кнопка для перехода на форму изменения учётных данных. На форме изменения учётных данных находятся поля ввода существующих логина и пароля и новых логина и пароля. Через кнопки «Удалить», «Изменить» и «Создать» соответственно предоставляется следующий функционал:

- Удаление учётной записи из списка учётных записей;
- Изменение логина и/или пароля учётной записи;
- Создание новой учётной записи.

На всех формах (кроме формы авторизации) предоставляется возможность перейти обратно на форму авторизации.

4.1.1 Требования к организации входных и выходных данных

Для авторизации пользователь вводит логин и пароль. Управлять сохранёнными учётными данными может только пользователь с правами администратора.

- Входные данные должны проверяться корректность ввода. Пользователь может вводить следующие данные для последующего расчёта:
- Высота, длина и ширина ангара (в м). Они должны быть положительными числами;
- Тип строительства ангара (предустановленный выбор в виде выпадающего списка);
- Форма ангара (предустановленный выбор в виде выпадающего списка);
- Вид панели (предустановленный выбор в виде выпадающего списка);
- Сроки строительства (в месяцах). Сроки являются положительным целым числом;
- Вид фундамента (предустановленный выбор в виде выпадающего списка);
- Тип двери (предустановленный выбор в виде выпадающего списка);
- Промокод (текстовая строка).

Выходными данными для пользователя являются итоговая цена ангара, отображающейся на экране, и печатная форма с результатами расчёта. Печатная форма кроме итоговой цены так же содержит параметры, на основе которых рассчитывалась цена.

4.1.2 Требования к временным характеристикам

После ввода параметров и начала расчётов пользователь получает итоговый результат не позднее 30 секунд. Такие же требования к процессу изменения учётных записей и выдаче печатной формы.

4.2. Требования к надежности

Так как программа будет распространяться на сайте компании «Daedalus

industries», надёжность будет зависеть от сервера компании и от исправности сети пользователя.

4.2.1 Требования к обеспечению надежного (устойчивого) функционирования программы

Пользователю, работающему с программой через веб-браузер должен быть предоставлен непрерывный доступ к веб-приложению, расположенному по определённому url-адресу. Веб-сервис не должен непредвиденно прерывать свою работу.

4.2.2 Время восстановления после отказа

При поломке серверной составляющей сайта на восстановление может уйти не более суток.

4.2.3 Отказы из-за некорректных действий оператора

В случае некорректных действий оператора достаточно обновить сайт компании.

4.3. Условия эксплуатации

Программа будет находится на удалённом сервере компании, поэтому условия эксплуатации будут относиться к серверу.

4.3.1 Климатические условия эксплуатации

Климатические условия эксплуатации, при которых должны обеспечиваться заданные характеристики, должны удовлетворять требованиям, предъявляемым к техническим средствам в части условий их эксплуатации

4.3.2 Требования к видам обслуживания

Сервер не требует специального обслуживания.

4.3.3 Требования к численности и квалификации персонала

Для корректной работы программы требуется минимум 2 человека – администратор и пользователь.

Администратор должен иметь высшее профильное образование в сфере ИТ. Пользователь должен иметь навыки использования графического интерфейса ОС.

4.4. Требования к составу и параметрам технических средств

Компьютеры администратора и пользователя должны включать в себя такие компоненты, как:

Материнскую плату, процессор с тактовой частотой не менее 1 ГГц, оперативную память с объёмом не менее 2 гб, блок питания, монитор, мышку, клавиатура и доступ к интернету.

Сервер должен включать в себя такие компоненты, как:

Материнскую плату, процессор с тактовой частотой не менее 1 ГГц, оперативную память с объёмом не менее 2 гб, блок питания, доступ к интернету.

4.5. Требования к информационной и программной совместимости

4.5.1 Требования информационным структурам на входе и выходе

Структурные единицы входной информации отсутствуют.

Структурные единицы выходной информации:

- печатная форма.

4.5.2 Требования к методам решения

Методом решения является сложение цен составных частей строительства ангара и последующее умножение на коэффициенты, зависящих от введенных пользователем параметров сроков и промокода

4.5.3 Требования к исходным кодам и языкам программирования

Исходные коды выполняются с применением языка программирования Java и языка разметки веб-страниц HTML версии 5. В качестве интегрированной среды разработки программы должна быть использована среда Eclipse IDE.

4.5.4 Требования к программным средствам, используемым программой

Системные программные средства, требуемые для работы программы, должны быть представлены лицензионной версией операционной системы Windows 10 или свободно распространяемым дистрибутивом Linux Ubuntu версии 20.4. Для доступа к веб-приложению требуется браузер (за исключением консольных и Internet Explorer).

4.5.5 Требования к защите информации и программ

Требования к защите информации и программ не предъявляются.

4.6. Требования к маркировке и упаковке

Программа поставляется в виде jar-файла, поэтому не требует установки.

Программа будет упакована на компакт диске (CD) и вместе с ней прилагаться небольшая документация по использованию. Маркировка программного изделия будет иметь название продукта, тема разработки, кем разработана и год выпуска.

4.7. Требования к транспортированию и хранению

Требования не предъявляются

5. ТРЕБОВАНИЯ К ПРОГРАММНОЙ ДОКУМЕНТАЦИИ

Предварительный состав программной документации включает в себя:

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

6. ТЕХНИКО-ЭКОНОМИЧЕСКИЕ ПОКАЗАТЕЛИ

Калькулятор «Стоимость строительства быстровозводимых ангаров» позволит любому посетителю сайта ознакомиться с видами ангаров и их стоимостью. Это располагает его к использованию услуг фирмы и упрощает составление первоначального заказа. Программный продукт (ПП), предоставляемый компанией «Daedalus industries», в сравнение с аналогами имеет более интуитивно понятный интерфейс, простоту в использовании и имеет большое количество учтённых параметров.

7. СТАДИИ И ЭТАПЫ РАЗРАБОТКИ

Ключевые этапы разработки представлены в таблице 1.

Таблица 1 – План-график выполнения работы

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
Раздел 3. Настройка среды разработки для операционных систем семейств Windows и Linux	10	10%	31 неделя
Раздел 4. Настройка среды разработки для подключения к системе контроля версий	7	7%	32 неделя
Раздел 5. Реализация исходного кода по зонам ответственности	23	23%	34 неделя
Раздел 6. Сборка и тестирование программного продукта	8	8%	35 неделя
Раздел 7. Настройка программной среды для развертывания и запуска программного продукта	10	10%	36 неделя
Раздел 8. Руководство пользователя программного продукта	10	10%	37 неделя
Защита	0,3	0,3%	38 неделя

8. ПОРЯДОК КОНТРОЛЯ И ПРИЕМКИ

Процедура защиты курсовой работы предполагает следующие этапы:

1. Настройка среды Eclipse в нескольких операционных системах разных семейств.
2. Клонирование репозитория GitHub, извлечение рабочей копии и выполнение основных команд.
3. Работа с сервисом Travis CI.
4. Выполнить развертывание и запуск программного продукта.
5. Знание своей зоны ответственности.

Обоснование использования определённых инструментов при разработке программного продукта

При разработке программного продукта будет использоваться следующий набор инструментов:

- Eclipse IDE – интегрированная среда разработки используется в связи с тем, что это свободно распространяемое ПО, которое может использоваться в коммерческих целях. Так же эта среда программирования имеет качественные модули для разработки приложений на языке Java и поддержку остальных требуемых для разработки инструментов.
- Maven – фреймворк для автоматизации сборки проектов используется для уменьшения затрачиваемого времени на сборку проектов. Он является свободно распространяемым ПО.
- JUnit – библиотека для модульного тестирования. Используется для автоматизации проверки приложения и уменьшение затрат при тестировании. Библиотека выбрана в связи с тем, что она стандартом для тестирований Java приложений.
- Система контроля версий git используется для обеспечения совместной работы разработчиков. Данная система выбрана в связи с тем, что это децентрализованная система контроля версий. Так же эта система удобна из-за того, что она поставляется в качестве модуля вместе со средой Eclipse.

Приложения

ПРИЛОЖЕНИЕ П-2

Руководство пользователя

ФБГОУ ВО УФИМСКИЙ ГОСУДАРСТВЕННЫЙ АВИАЦИОННЫЙ
ТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ

СОГЛАСОВАНО

Доцент кафедры АСУ ФГОУ
УГАТУ
Казанцев А.В.

Личная Расшифровка
подпись подписи

27.03.2021

УТВЕРЖДАЮ

Студент группы ПИ-223
ФИРТ ФГОУ УГАТУ,
модератор Панаев С.И.

Личная Расшифровка
подпись подписи

26.03.2021

Калькулятор стоимости строительства быстровозводимых ангаров

Руководство пользователя

ЛИСТ УТВЕРЖДЕНИЯ

581.45338009.25777-01 90 01-ЛУ

(Электронный)

Листов 10

СОГЛАСОВАНО

Доцент кафедры АСУ ФГОУ
УГАТУ
Казанцев А.В.

Личная Расшифровка
подпись подписи

27.03.2021

УТВЕРЖДАЮ

Студент группы ПИ-223
ФИРТ ФГОУ УГАТУ,
модератор Панаев С.И.

Личная Расшифровка
подпись подписи

26.03.2021

Подп. и дата	
Инф. и подп.	
Взам. инф. и	
Подп. и дата	
Инф. и подп.	

2021

Утвержден

581.45338009.25777-01 90 01-ЛУ

Калькулятор стоимости строительства быстровозводимых ангаров

Руководство пользователя

581.45338009.25777-01 90 01-ЛУ

(Электронный)

Листов 10

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

2021

Аннотация

Данный документ является руководством пользователя для программы «Калькулятор стоимости строительства быстровозводимых ангаров». Документ разработан в рамках курсовой работы по теме: «Калькулятор стоимости строительства быстровозводимых ангаров».

Содержание

1. Назначение программы.....	5
1.1 Сведения о назначении программы.....	5
1.2 Функциональное назначение.....	5
1.3 Эксплуатационное назначение.....	5
2. Условия выполнения программы.....	6
3. Работа с программой.....	7
3.1 Возможности оператора.....	7
3.2 Возможности администратора.....	9
3.3 Сбои и ошибки.....	10

1. Назначение программы.

1.1 Сведения о назначении программы

Калькулятор «Стоимость строительства быстровозводимых ангаров» – это программа, помогающая сформировать представление о предварительной цене ангара с определенными выбранными параметрами.

1.2 Функциональное назначение

Программа предназначена для расчета предварительной цены быстровозводимого ангара на основании введенных пользователем параметров ангара с возможностью формирования печатной формы с записанными результатами. Посетитель сможет ознакомиться с возможными типами постройки, разновидностями арок ангара, вид панелей, а также с дополнительными настраиваемыми параметрами в самой программе.

Для администратора программа предоставляет возможность управлять учётными данными пользователя.

1.3 Эксплуатационное назначение

Программа должна эксплуатироваться на компьютерах пользователей через веб-браузер посредством клиент-серверной архитектуры.

2. Условия выполнения программы.

Компьютеры администратора и пользователя должны включать в себя такие компоненты, как:

- Материнскую плату
- процессор с тактовой частотой не менее 1 ГГц
- оперативную память с объёмом не менее 2 гб
- блок питания
- монитор
- мышку
- клавиатура
- доступ к интернету
- браузер.

3. Выполнение программы.

3.1 Возможности оператора.

Программа будет доступна по адресу через интернет с использованием браузера. После ввода данного адреса, перед оператором возникнет окно с авторизацией, как показано на рисунке 1.

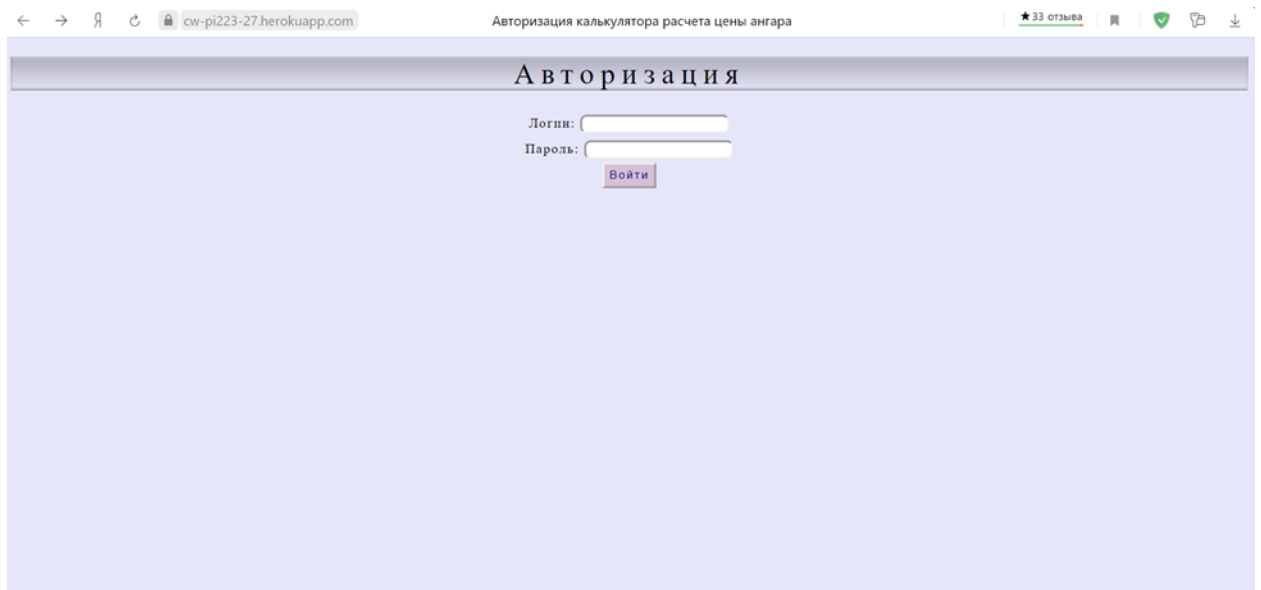


Рисунок 1 – окно авторизации

Завершив авторизацию, оператор сможет воспользоваться функционалом данной программы, как показано на рисунке 2.



Рисунок 2 – окно с калькулятором

Для работы с калькулятором, пользователю понадобится ввести числа в такие поля, как «Длина», «Ширина», «Высота», «Выберите сроки(мес.)». Затем выбрать тип строительства, форма ангара, тип панелей, фундамент и типа двери из представленных в данной программе. Так же при наличии промокода можно ввести его для получения скидки. После ввода и выбора всех параметров нужно нажать кнопку «вычислить». Программа вычислит итоговую сумму по введённым в неё данным, как показано на рисунке 3.



Рисунок 3 – сумма постройки

Для сохранения данных и результата в pdf формате надо нажать кнопку «Скачать PDF-файл». В результате получится pdf-файл, как показано на рисунке 4.

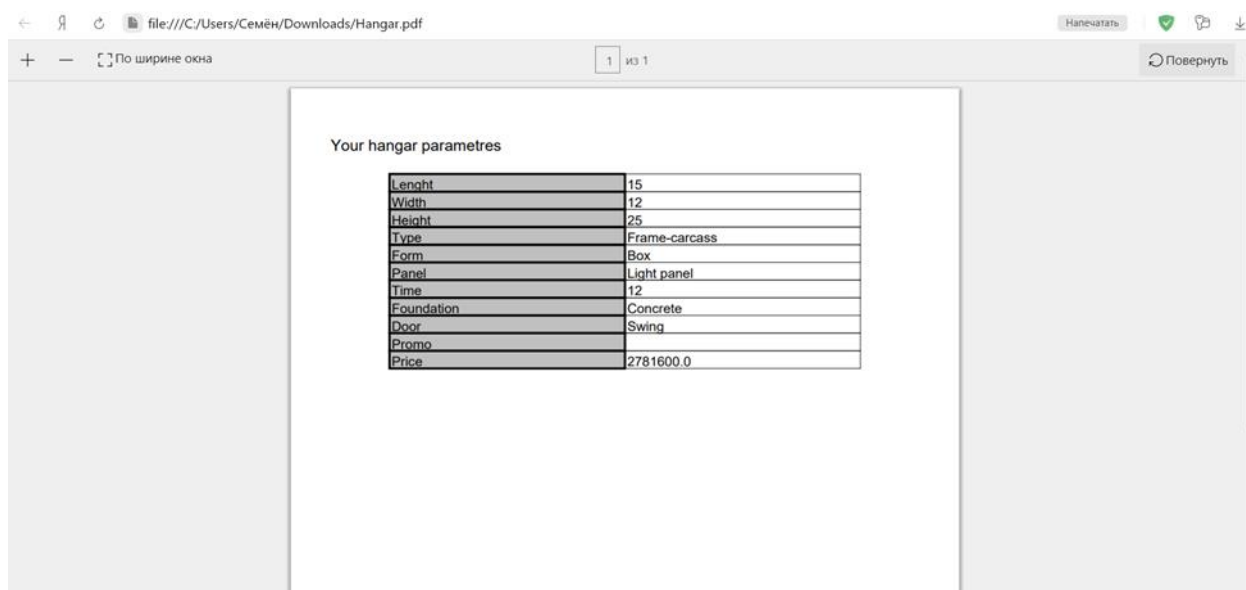


Рисунок 4 – сохранённый pdf файл

На сайте можно авторизироваться через учётную запись администрации. При авторизации под логином администрации, оператору будут доступны новые возможности такие, как показано на рисунке 5.

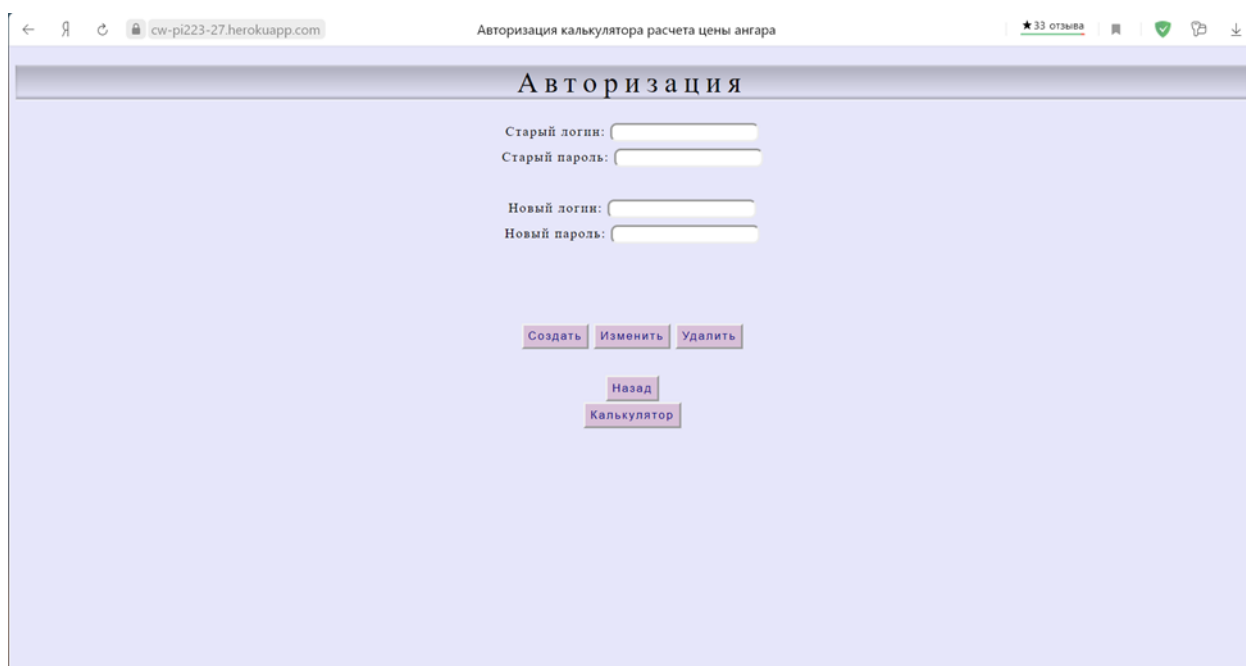


Рисунок 5 – редактирование логинов и паролей пользователей

3.2 Возможности администратора

Администратору доступен весь функционал оператора, а также создание новых пользователей и редактирование сохранённых аккаунтов. Для создания нового аккаунта нужно ввести логин и пароль в поля «Новый логин» и «Новый пароль». Для изменения уже имеющегося пользователя, нужно ввести данные этого пользователя в «Старый логин» и «Старый пароль», затем ввести новые данные в поля «Новый логин» и «Новый пароль». Для удаления нужно ввести его данные в поля «Старый логин» и «Старый пароль». При неудачных попытках выполнить действия администратором, сайт сообщит об ошибке, как показано на рисунке 6.

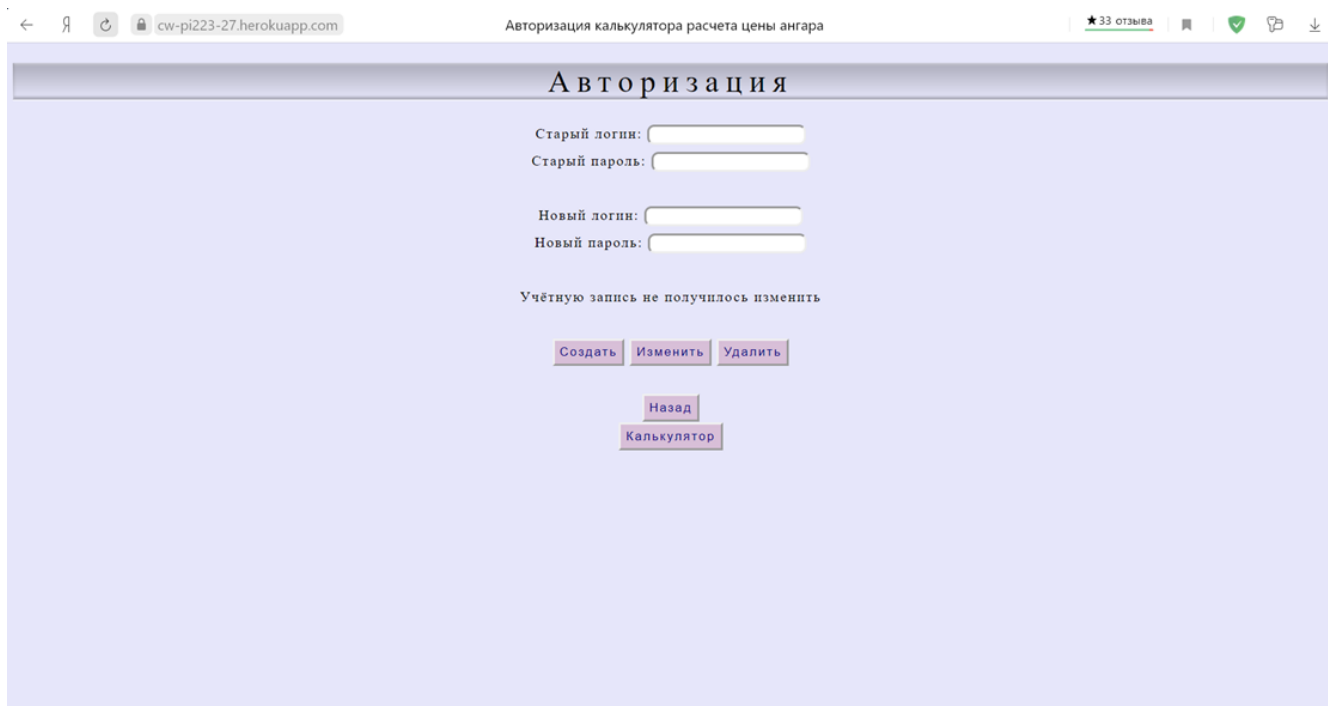


Рисунок 6 – ошибка действий администратора

3.3 Сбои и ошибки

При неисправности сайта достаточно обновить страницу или вернуться на предыдущую.

Приложения

ПРИЛОЖЕНИЕ П-3

Программный код Unit-теста №1 – TypeHangarTest

```
package calcTest;

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

public class TypeHangarTest {

    @Test
    public void testHangar() {

        calculator.Hangar hangarBox = new calculator.BoxHangar(5, 5, 5);
        calculator.Hangar hangarTent = new calculator.TentHangar(4, 5, 5);
        calculator.Hangar hangarArc = new calculator.ArcHangar(1, 1, 1);
        boolean rightArea = hangarBox.getHangarArea() == 125.0 &&
hangarTent.getHangarArea() == 25.0 && hangarArc.getHangarArea() == 3*Math.PI/4;
        boolean rightDoorArea = hangarBox.getHangarDoorArea() == 16.25 &&
hangarTent.getHangarDoorArea() == 12.4 && hangarArc.getHangarDoorArea() == 0.6;
        assertTrue(rightArea&&rightDoorArea);

    }

}
```

Приложения

ПРИЛОЖЕНИЕ П-4

Программный код Unit-теста №2 – HangarTest

```
package calcTest;

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

public class hangarTest {
    @Test
    public void test() {

        calculator.Hangar h = new calculator.Hangar(5, 9, 6) { };
        assertTrue(h.getHangarFoundationArea()      =      =      54      &&
h.getHangarDoorArea() == 29.25);

    }

}
```

Приложения

ПРИЛОЖЕНИЕ П-5

Программный код Unit-теста №3 – PdfWriterTest

```
package calcTest;

import static org.junit.Assert.*;
import java.io.File;
import org.junit.Test;
import filework.PDFWriter;

public class PdfWriterTest {
    @Test
    public void testPdf() {

        PDFWriter PDF = new PDFWriter();
        String[] outputForFile= new String[] {"1", "1", "1", "hi", "hi", "hi",
"time", "fond", "door", "promo", "price, he-he"};
        PDF.write(outputForFile);
        assertTrue(new File(new File(new
File(PDFWriter.class.getProtectionDomain().getCodeSource().getLocation().getPath()).
getParent()).getParent()+"/Hangar.pdf").exists());
    }
}
```


Приложения

ПРИЛОЖЕНИЕ П-6

Программный код – AuthBaseController

```
package authorization;

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

//Класс для работы с "базой" паролей
public class AuthBaseController {
    public static File fileBase = new File(new
File(AuthBaseController.class.getProtectionDomain().getCodeSource().getLocation().g
etPath())+"/AuthBase.txt"); //путь до "базы"

    //В файле каждая строка это новая учётная запись. Логин и пароль
разделенны символом ";"

    //Функция для проверки того, есть ли данная учётная запись в базе
    public static boolean checkAuth(String login, String password) {
        BufferedReader reader;
        try {
            String line;
            reader = new BufferedReader(new FileReader(fileBase));
            while((line = reader.readLine())!=null) {
                String[] tempArr= line.split(";");
                if (tempArr[0].equals(login) &&
tempArr[1].equals(password)) {
                    reader.close();
```

```

        return true;
    }
}

reader.close();
} catch (IOException e) {
    e.printStackTrace();
}

return false;
}

//Функция добавления учётной записи
public static boolean addAuth(String login, String password) {
    if (!isCorrectAuth(login, password)) {
        return false;
    }
    BufferedWriter writer;
    try {
        boolean authInBase=checkAuth(login, password);
        writer = new BufferedWriter(new FileWriter(fileBase,true));
        if (authInBase == false) {
            writer.newLine();
            writer.append(login+";" +password);
        }
        writer.close();
        return !authInBase;
    } catch (IOException e) {
        // TODO Auto-generated catch block
        e.printStackTrace();
    }
    return false;
}

```

```

    }

    //Функция изменения учётной записи
    public static boolean changeAuth(String login, String password, String
newLogin, String newPassword) {
        if (!(isCorrectAuth(login, password)&&isCorrectAuth(newLogin,
newPassword))) {

            return false;
        }
        boolean isCorrectChange = false;
        BufferedReader reader;
        BufferedWriter writer;
        try {
            String line;
            String buffer = "";
            reader = new BufferedReader(new FileReader(fileBase));
            while((line = reader.readLine())!=null) {
                String[] tempArr= line.split(";");
                if            (tempArr[0].equals(login)            &&
tempArr[1].equals(password)) {
                    buffer+=newLogin+";"+newPassword+"\n";
                    isCorrectChange = true;
                } else {
                    buffer+=line+"\n";
                }
            }
            reader.close();
            if (isCorrectChange) {
                writer = new BufferedWriter(new FileWriter(fileBase));
                writer.write(buffer.substring(0, buffer.length()-1));
                writer.close();
            }
        }
    }
}

```

```

        }
    } catch (IOException e) {
        e.printStackTrace();
    }
    return isCorrectChange;
}

//Функция удаления учётной записи
public static boolean deleteAuth(String login, String password) {
    if (!isCorrectAuth(login, password)) {
        return false;
    }
    boolean isCorrectChange = false;
    BufferedReader reader;
    BufferedWriter writer;
    try {
        String line;
        String buffer = "";
        reader = new BufferedReader(new FileReader(fileBase));
        while((line = reader.readLine())!=null) {
            String[] tempArr= line.split(";");
            if (tempArr[0].equals(login) &&
tempArr[1].equals(password)) {
                isCorrectChange = true;
            } else {
                buffer+=line+"\n";
            }
        }
        reader.close();
        if (isCorrectChange) {
            writer = new BufferedWriter(new FileWriter(fileBase));

```

```

        writer.write(buffer.substring(0, buffer.length()-1));
        writer.close();
    }
} catch (IOException e) {
    e.printStackTrace();
}
return isCorrectChange;
}

//Проверка корректности логина и пароля
public static boolean isCorrectAuth(String login, String password) {
    boolean isCorrect =
!(login==""||password==""||login.contains(";")||password.contains(";"));
    if (isAdminAuth(login, password)) {
        isCorrect = false;
    }
    return isCorrect;
}

public static boolean isAdminAuth(String login, String password) {
    boolean isAdmin=false;

    BufferedReader reader;
    String[] adminAuth;
    try {
        reader = new BufferedReader(new FileReader(fileBase));
        adminAuth= reader.readLine().split(";");
        reader.close();
        if
(login.equals(adminAuth[0])&&password.equals(adminAuth[1])) {
            isAdmin = true;
        }
    }
}

```

```
        } catch (IOException e) {  
            e.printStackTrace();  
        }  
        return isAdmin;  
    }  
}
```

Приложения

ПРИЛОЖЕНИЕ П-7

Программный код – AuthManager

```
package authorization;

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

//Сервлет для обработки изменения в учётных записях
@WebServlet(name="AuthManager", urlPatterns="/JavaAuthManager")

public class AuthManager extends HttpServlet {

    //Полученные с страницы данные

    private static String inputLogin;

    private static String inputPass;

    private static String inputNewLogin;

    private static String inputNewPass;

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

        inputLogin = request.getParameter("login");

        inputPass = request.getParameter("password");

        inputNewLogin = request.getParameter("newLogin");

        inputNewPass = request.getParameter("newPassword");
```

//Выполнение определённой функции в зависимости от нажатой кнопки, а так же вывод сообщения об успехе.

```
if (request.getParameter("create")!=null) { //Функция добавления
    if (AuthBaseController.addAuth(inputNewLogin, inputNewPass)) {
        request.setAttribute("labelText", "Учётная запись была
удачна добавлена");
    } else {
        request.setAttribute("labelText", "Учётную запись не
получилось добавить");
    }
    request.getRequestDispatcher("/Edit.jsp").forward(request,
response);
} else if (request.getParameter("change")!=null) { //Функция изменения
    if (AuthBaseController.changeAuth(inputLogin, inputPass,
inputNewLogin, inputNewLogin)) {
        request.setAttribute("labelText", "Учётная запись была
успешно изменена");
    } else {
        request.setAttribute("labelText", "Учётную запись не
получилось изменить");
    }
    request.getRequestDispatcher("/Edit.jsp").forward(request,
response);
} else if (request.getParameter("del")!=null) { //Функция удаления
    if (AuthBaseController.deleteAuth(inputLogin, inputPass)) {
        request.setAttribute("labelText", "Учётная запись была
успешно удалена");
    } else {
```



```
        request.setAttribute("labelText", "Учётную запись не  
получилось удалить");  
    }  
    request.getRequestDispatcher("/Edit.jsp").forward(request,  
response);  
}  
}  
  
}
```

Приложения

ПРИЛОЖЕНИЕ П-8

Программный код – Authorization

```
package authorization;

import java.io.IOException;
import javax.servlet.ServletException;
import javax.servlet.annotation.WebServlet;
import javax.servlet.http.HttpServlet;
import javax.servlet.http.HttpServletRequest;
import javax.servlet.http.HttpServletResponse;
//Сервлет для авторизации
@WebServlet(name="Auth", urlPatterns="/JavaAuth")
public class Authorization extends HttpServlet {
    //Полученные с страницы данные
    private static String inputLogin;
    private static String inputPass;
    protected void doPost(HttpServletRequest request, HttpServletResponse
response) throws ServletException, IOException {
        inputLogin = request.getParameter("login");
        inputPass = request.getParameter("password");

        request.getSession().setAttribute("AuthCorrect", "false");
        request.getSession().setAttribute("AuthAdminCorrect", "false");
        //Проверяем введённые данные
        if (AuthBaseController.checkAuth(inputLogin, inputPass)) {
            request.getSession().setAttribute("AuthCorrect", "true");

            //Проверяем введённые данные, если они админские
```

```
        if (AuthBaseController.isAdminAuth(inputLogin,inputPass)) {  
            request.getSession().setAttribute("AuthAdminCorrect",  
"true");  
  
            request.getRequestDispatcher("/Edit.jsp").forward(request, response);  
        } else {  
  
            request.getRequestDispatcher("/Form.jsp").forward(request, response);  
        }  
  
        } else {  
  
            request.getRequestDispatcher("/Authorization.jsp").forward(request, response);  
        };  
    }  
}
```

Приложения

ПРИЛОЖЕНИЕ П-9

Программный код – ArcHangar

```
package calculator;

public class ArcHangar extends Hangar {
    public ArcHangar(double _height, double _width, double _length) {
        super(_height, _width, _length);
    }
    @Override
    public double getHangarDoorArea() {
        return width*height*0.6;
    }
    @Override
    public double getHangarArea() {
        return Math.PI*(width/2)*((width/2)+length);
    }
}
```

Программный код – BoxHangar

```
package calculator;

public class BoxHangar extends Hangar {
    public BoxHangar(double _height, double _width, double _length) {
        super(_height, _width, _length);
    }

    @Override
    public double getHangarArea() {
        return (2*(width*height+length*height)+length*width);
    }
}
```

Программный код – Calculator

```
package calculator;

import java.io.IOException;
import java.util.HashMap;
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 filework.PDFWriter;

@WebServlet(name="Calc", urlPatterns="/JavaCalc")
public class Calculator extends HttpServlet {
    public static String[] outputForFile;
    protected void doPost(HttpServletRequest request, HttpServletResponse
response) throws ServletException, IOException {
        request.setCharacterEncoding("UTF-8");
        RequestCalc Calc = RequestCalc.fromRequestParameters(request);
        Calc.CalculateHangarCost(request);
        //Создаём Pdf файл
        PDFWriter PDF = new PDFWriter();
        PDF.write(outputForFile);
        request.getRequestDispatcher("/Results.jsp").forward(request,
response);
    }
    private static class RequestCalc {
```

```

private final HashMap<String, Double> promoMap; //Хеш-меп с
промокодами и соответствующими коэффициентами скидки

private final int[] ArrDoor; //Цены дверей
private final double[] ArrTime; //Цены, в зависимости от времени
private final double[] ArrType; //Цены от типа ангара
private final int[] ArrFoundation; //Цены от типа фундамента
private final int[] ArrPanel; //Цены от вида панелей
private final String inputDoor;
private final String inputTime;
private final String inputx;
private final String inputy;
private final String inputz;
private final String inputType;
private final String inputFoundation;
private final String inputFormHangar;
private final String inputPromo;
private final String inputPanel;

//Массивы для перевода выбранного пункта листа в текст
String[] typeToString = {"Каркасно-рамочные", "Каркасно-
щитовой"};

String[] formToString = {"Арочный", "Прямостенный",
"Шатровый"};

String[] panelToString = {"Облегчённая сендвич-панель",
"Сендвич-панель", "Утеплённая сендвич-панель"};

String[] foundationToString = {"Отсутствие", "Каменный",
"Железобетонный", "Бетонный"};

String[] doorToString = {"Распашные", "Подъёмно-секционные",
"Шторные", "Рулонные"};

String[] typeToStringEn = {"Frame-carcass", "Frame-board"};
String[] formToStringEn = {"Arc", "Box", "Tent"};

```

```

String[] panelToStringEn = {"Light panel", "Panel", "Heavy panel"};
String[] foundationToStringEn = {"None", "Stone", "Reinforced
concrete", "Concrete"};

String[] doorToStringEn = {"Swing", "Lifting-sectional", "Curtain",
"Roll-up"};

private RequestCalc (String door, String time,String x,String y,String
z,String type,String foundation,String formHangar,String promo,String panel) {

    this.inputDoor = door;
    this.inputTime = time;
    this.inputx = x;
    this.inputy = y;
    this.inputz = z;
    this.inputType = type;
    this.inputFoundation = foundation;
    this.inputFormHangar = formHangar;
    this.inputPromo = promo;
    this.inputPanel = panel;
    promoMap = new HashMap<String, Double>();
    ArrDoor = new int[]{6900, 9000, 8400, 4000};
    ArrTime = new double[]{0.5, 0.75, 1, 2.5};
    ArrType = new double[]{1.15,1};
    ArrFoundation = new int[]{0,650,3000,1400};
    ArrPanel = new int[]{1200,1600,2500};
    promoMap.put("Romashka", 0.9);
    promoMap.put("sudo getZachet", 0.5);
    promoMap.put("Kaz", 0.85);
    promoMap.put("Demin", 3.0);
    promoMap.put("1", 0.999);
}

```

public

static

RequestCalc


```

fromRequestParameters(HttpServletRequest request) {
    return new RequestCalc(
        request.getParameter("door"),
        request.getParameter("time"),
        request.getParameter("x"),
        request.getParameter("y"),
        request.getParameter("z"),
        request.getParameter("type"),
        request.getParameter("foundation"),
        request.getParameter("formHangar"),
        request.getParameter("promo"),
        request.getParameter("panel"));
}

public void CalculateHangarCost(HttpServletRequest request) {
    int door;
    int time;
    double x;
    double y;
    double z;
    int type;
    int foundation;
    int formHangar;
    String promo = inputPromo;
    int panel;
    //Пытаемся прочитать вход как числа
    try {
        door=Integer.parseInt(inputDoor);
        time=Integer.parseInt(inputTime);
        x=Double.parseDouble(inputx);
        y=Double.parseDouble(inputy);

```

```

        z=Double.parseDouble(inputz);
        type=Integer.parseInt(inputType);
        foundation=Integer.parseInt(inputFoundation);
        formHangar=Integer.parseInt(inputFormHangar);
        panel=Integer.parseInt(inputPanel);
    }
    catch (NumberFormatException e) {
        door=0;
        time=0;
        x=0;
        y=0;
        z=0;
        type=0;
        foundation=0;
        formHangar=0;
        panel=0;
    }

    request.setAttribute("doorResult", doorToString[door]);
    request.setAttribute("timeResult", inputTime);
    request.setAttribute("xResult", inputx);
    request.setAttribute("yResult", inputy);
    request.setAttribute("zResult", inputz);
    request.setAttribute("typeResult", typeToString[type]);
    request.setAttribute("foundationResult",
foundationToString[foundation]);

    request.setAttribute("formHangarResult",formToString[formHangar]);
    request.setAttribute("promoResult",inputPromo);
    request.setAttribute("panelResult",panelToString[panel]);
    double price = 0;

    //определяем нужный тип ангара

```

```

        Hangar hangar;
        switch(formHangar) {
        case(0):
            hangar = new ArcHangar(z,y,x);
            break;
        case(1):
            hangar = new BoxHangar(z,y,x);
            break;
        case(2):
            hangar = new TentHangar(z,y,x);
            break;
        default:
            hangar = new ArcHangar(z,y,x);
            break;
        }
        price =
(hangar.getHangarArea()*ArrPanel[panel]*ArrType[type]+hangar.getHangarFoundatio
nArea()*ArrFoundation[foundation]+hangar.getHangarDoorArea()*ArrDoor[door])*get
Time(time)*getPromo(promo);

        outputForFile= new String[] {inputx, inputy, inputz,
typeToStringEn[type], formToStringEn[formHangar], panelToStringEn[panel],
inputTime, foundationToStringEn[foundation], doorToStringEn[door], inputPromo,
Double.toString(Math.floor(price/100)*100)}; //заполняем значения для вывода в
файл

        request.setAttribute("price",
Double.toString(Math.floor(price/100)*100)); //Выводим округлённую итоговую
цену ангара

    }

    //функция возвращает коэффициент в зависимости от времени
постройки

```

```

private double getTime(int time) {
    if (time<2) {
        return ArrTime[3];
    } else if (time>=2&&time<=6) {
        return ArrTime[2];
    } else if (time>6&&time<=12) {
        return ArrTime[1];
    } else {
        return ArrTime[0];
    }
}

```

//функция возвращает размер скидки, если верно введён

промокод.

```

public double getPromo(String promo) {
    if (promoMap.containsKey(promo)) {
        return promoMap.get(promo);
    } else {
        return 1;
    }
}
}
}
}

```

Программный код – Hangar

```
package calculator;

public abstract class Hangar {
    double height;
    double width;
    double length;
    public Hangar(double x_height, double y_width, double z_length) {
        height=x_height;
        width=y_width;
        length=z_length;
    }
    public double getHangarArea() {
        return 2*(width*height+length*height+length*width);
    }
    public double getHangarDoorArea() {
        return width*height*0.65;
    }
    public double getHangarFoundationArea() {
        return length*width;
    }
}
```

Программный код – TentHangar

```
package calculator;

public class TentHangar extends Hangar {
    public TentHangar(double _height, double _width, double _length) {
        super(_height, _width, _length);
    }
    @Override
    public double getHangarDoorArea() {
        return width*height*0.62;
    }
    @Override
    public double getHangarArea() {
        return
2*(22/30*width*height+length*Math.sqrt(Math.pow(0.1*width,2)+Math.pow(2/3*height,2)) + length*Math.sqrt(Math.pow(1/3*height,2)+Math.pow(0.4*width,2)) );
    }
}
```

Программный код – PDFWriter

```
package filework;

import java.io.*;
import com.itextpdf.text.BaseColor;
import com.itextpdf.text.Document;
import com.itextpdf.text.DocumentException;
import com.itextpdf.text.Font;
import com.itextpdf.text.Paragraph;
import com.itextpdf.text.Phrase;
import com.itextpdf.text.pdf.BaseFont;
import com.itextpdf.text.pdf.PdfPCell;
import com.itextpdf.text.pdf.PdfPTable;
import com.itextpdf.text.pdf.PdfWriter;

public class PDFWriter implements WriterInFile {
    public PDFWriter() {
    }

    public void write(String[] forWrite) {
        Document document = new Document(); //Создаём документ
        try {
            PdfWriter.getInstance(document, new FileOutputStream(new
File(new
File(PDFWriter.class.getProtectionDomain().getCodeSource().getLocation().getPath()).
getParent()).getParent()+"/Hangar.pdf"));
        } catch (FileNotFoundException | DocumentException e) {
            e.printStackTrace();
        }
    }
}
```

```

    }
    document.open();
    BaseFont times = null;//Создаём шрифт
    try {
        times = BaseFont.createFont("/fonts/times.ttf", "UTF-8",
BaseFont.EMBEDDED);
    } catch (DocumentException | IOException e) {
        e.printStackTrace();
    }
    //Заполняем текстом
    String string_pdf = "Your hangar parametres";
    Paragraph paragraph = new Paragraph();
    paragraph.add(new Paragraph(string_pdf, new Font(times,14)));
    try {
        document.add(paragraph);
    } catch (DocumentException e1) {
        e1.printStackTrace();
    }
    paragraph.clear();
    String string_pdf3 = " ";
    paragraph.add(new Paragraph(string_pdf3, new Font(times,14)));
    try {
        document.add(paragraph);
    } catch (DocumentException e1) {
        e1.printStackTrace();
    }
    //Создаём таблицу с 2 столбцами
    PdfPTable table = new PdfPTable(2);
    addTableCells(table, forWrite);
    try {

```



```

        document.add(table);
    } catch (DocumentException e) {
        e.printStackTrace();
    }
    document.close();
}
//Заполнение таблицы
private void addTableCells(PdfPTable table, String[] forWrite) {
    //1 столбец заполняет названиями, 2 столбец заполняется
значением
    String[] columnTitles= {"Lenght", "Width", "Height", "Type", "Form",
"Panel", "Time", "Foundation", "Door", "Promo", "Price"};
    for (int i=0;i<forWrite.length;i++) {
        PdfPCell header = new PdfPCell();
        header.setBackgroundColor(BaseColor.LIGHT_GRAY);
        header.setBorderWidth(2);
        header.setPhrase(new Phrase(columnTitles[i]));
        table.addCell(header);
        table.addCell(forWrite[i]);
    }
}
}

```

Программный код – WriterInFile

```
package filework;
```

```
public interface WriterInFile {  
    public void write(String[] forWrite);  
}
```

Приложения

ПРИЛОЖЕНИЕ П-16

Программный код – Authorization

```
<% @ page language="java" contentType="text/html; charset=UTF-8"
    pageEncoding="UTF-8"%>
<!DOCTYPE html>
<html>
<head>
<meta charset="UTF-8">
<link rel="stylesheet" href="css/style.css">
<title>Авторизация калькулятора расчета цены ангара</title>
</head>
<body>
<h1 class=titl>Авторизация</h1>
    <form                action="${pageContext.request.contextPath}/JavaAuth"
method="post" class=lab>
    <center>
        <label for="first">Логин:</label>
        <input        class=box        type="text"        name="login"        id="login"
value="${login}"><br>
        <label for="second">Пароль: </label>
        <input  class=box  type="text"  name="password"  id="password"
value="${password}"><br>
        <input class=btn type="submit" name="input" value="Войти">
    </center>
    </form>
</body>
</html>
```

Приложения

ПРИЛОЖЕНИЕ П-17

Программный код – Edit

```
<% @ page language="java" contentType="text/html; charset=UTF-8"
    pageEncoding="UTF-8"%>
    <%if (request.getSession().getAttribute("AuthAdminCorrect")!="true") {
        request.getRequestDispatcher("/Authorization.jsp").forward(request,
response);} %>
    <!DOCTYPE html>
    <html>
    <head>
    <meta charset="UTF-8">
    <link rel="stylesheet" href="css/style.css">
    <title>Авторизация калькулятора расчета цены ангара</title>
    </head>
    <body>
    <h1 class=titl>Авторизация</h1>
        <form        action="{pageContext.request.contextPath}/JavaAuthManager"
method="post" class=lab>
        <center>
            <label for="first">Старый логин:</label>
            <input        class=box        type="text"        name="login"        id="login"
value="{login}"><br>
            <label for="second">Старый пароль: </label>
            <input        class=box        type="text"        name="password"        id="password"
value="{password}"><br><br>
            <label for="first">Новый логин:</label>
            <input        class=box        type="text"        name="newLogin"        id="newLogin"
value="{newLogin}"><br>
```

```

<label for="second">Новый пароль: </label>
<input class=box type="text" name="newPassword" id="newPassword"
value="\${newPassword}"><br><br>
<strong> </strong> \${labelText}<br><br>
<input class=btn type="submit" name="create" value="Создать">
<input class=btn type="submit" name="change" value="Изменить">
<input class=btn type="submit" name="del" value="Удалить"><br><br>
</form>
<form action="\${pageContext.request.contextPath}/Authorization.jsp">
<input class=btn type="submit" name="sign" value="Назад">
</form>
<form action="\${pageContext.request.contextPath}/Form.jsp">
<input class=btn type="submit" name="sign" value="Калькулятор">
</form>
</center>
</body>
</html>

```

Программный код – Results

```
<% @ page language="java" contentType="text/html; charset=UTF-8"
    pageEncoding="UTF-8"%>
    <%if (request.getSession().getAttribute("AuthCorrect")!="true") {
        request.getRequestDispatcher("/Authorization.jsp").forward(request,
response);} %>
<!DOCTYPE html>
<html>
<head>
<meta charset="UTF-8">
<link rel="stylesheet" href="css/style.css">
<title>Результат вычисления стоимости ангара</title>
</head>
<body>
<h1 class=titl>Результат вычисления стоимости ангара</h1>
<h2>Ваши введенные данные:</h2>
<p class=lab><strong>Длина:</strong> ${xResult}<br>
<strong>Ширина:</strong> ${yResult}<br>
<strong>Высота:</strong> ${zResult}<br>
<strong>Тип ангара:</strong> ${typeResult}<br>
<strong>Форма ангара:</strong> ${formHangarResult}<br>
<strong>Тип панели:</strong> ${panelResult}<br>
<strong>Сроки(мес.):</strong> ${timeResult}<br>
<strong>Тип фундамента:</strong> ${foundationResult}<br>
<strong>Тип двери:</strong> ${doorResult}<br>
<strong>Промокод:</strong> ${promoResult}<br>
<strong>Итоговая сумма: </strong> ${price}</p>
```

```
<a class=link href="/Hangar.pdf"> Открыть PDF-файл</a><br><br>
<a class=link href="/Hangar.pdf" download> Скачать PDF-файл</a><br><br>
<form action="${pageContext.request.contextPath}/Form.jsp">
  <input class=btn type="submit" name="sign" value="Назад">
</form>
</body>
</html>
```

Приложения

ПРИЛОЖЕНИЕ П-19

Программный код – Form

```
<% @ page language="java" contentType="text/html; charset=UTF-8"
    pageEncoding="UTF-8"%>
<%if (request.getSession().getAttribute("AuthCorrect")!="true") {
    request.getRequestDispatcher("/Authorization.jsp").forward(request,
response);} %>
<!DOCTYPE html>
<html>
<head>
<meta charset="UTF-8">
<link rel="stylesheet" href="css/style.css">
<title>Калькулятор расчеты цены ангара</title>
</head>
<body>
<h1 class=titl>Калькулятор расчета цены ангара</h1>
    <form action="\${pageContext.request.contextPath}/JavaCalc" method="post"
class=lab>
        <label for="first">Длина:</label>
        <input class=box type="text" name="x" id="x" value="\${x}">
        <label for="second">Ширина: </label>
        <input class=box type="text" name="y" id="y" value="\${y}">
        <label for="third">Высота: </label>
        <input class=box type="text" name="z" id="z" value="\${z}"><br>
        <label for="fourth">Выберите тип строительства ангара: </label>
        <select name="type" class=case>
            <option value="0" >Каркасно-рамочные</option>
            <option value="1" >Каркасно-щитовой</option>
```



```

        </select><br>
<label for="fifth">Выберите форму ангара: </label>
<select name="formHangar" class="case">
    <option value="0" >Арочный </option>
    <option value="1" >Прямостенный</option>
    <option value="2" >Шатровый</option>
</select><br>
<label for="sixth">Выберите тип панели: </label>
<select name="panel" class="case">
    <option value="0" >Облегчённая сендвич-
панель</option>
    <option value="1" >Сендвич-панель</option>
    <option value="2" >Утепленная сендвич-
панель</option>
</select><br>
<label for="seventh">Выберите сроки(мес.): </label>
<input class="box" type="text" name="time" id="time"
value="{time}"><br>
<label for="eighth">Выберите фундамент: </label>
<select name="foundation" class="case">
    <option value="0" >Отсутствие</option>
    <option value="1" >Каменный</option>
    <option value="2" >Железобетонный</option>
    <option value="3" >Бетонный</option>
</select><br>
<label for="ninth">Выберите тип двери: </label>
<select name="door" class="case">
    <option value="0" >Распашные</option>
    <option value="1" >Подъёмно-секционные</option>
    <option value="2" >Шторные</option>

```

```

        <option value="3" >Рулонные</option>
    </select><br>
    <label for="tenth">Промокод:</label>
    <input      class=box      type="text"      name="promo"      id="promo"
value="\${promo}"><br>
    <input class=btn type="submit" name="sign" value="Вычислить">
</form>
<form action="\${pageContext.request.contextPath}/Authorization.jsp">
<input class=btn type="submit" name="sign" value="Назад">
</form>
</body>
</html>

```

Список использованной литературы

- 1 Зуева А.В. Быстровозводимые здания и модульное строительство. [Электронный ресурс] – Режим доступа: <https://moluch.ru/archive/107/25643/>, свободный. – (дата обращения: 18.04.21)
- 2 Федеральный закон от 07.02.1992 №2300-1 "О защите прав потребителей" // URL: http://www.consultant.ru/document/cons_doc_LAW_305/ (дата обращения: 24.04.21)
- 3 ГОСТ 12.0.004 – 2015. Межгосударственный стандарт. Система стандартов безопасности труда. Организация обучения безопасности труда. Общие положения [Текст] – введ. 2016 – 03 – 01 – М.: Издательство стандартов, 2019 – 66с. – (Система стандартов по информации, библиотечному и издательскому делу).
- 4 ГОСТ 12.1.005-88. «Межгосударственный стандарт. Система стандартов безопасности труда. Общие санитарно-гигиенические требования к воздуху рабочей зоны» [Текст] – введ. 1989 – 01 – 01 – М.: Стандартинформ, 2008 – 66с. – (Система стандартов по информации, библиотечному и издательскому делу).
- 5 Федеральный закон от 21.12.1994 №69-ФЗ "О пожарной безопасности" // URL: http://www.consultant.ru/document/cons_doc_LAW_5438/ (дата обращения: 02.05.21)
- 6 Абстрактные классы в Java на конкретных примерах. URL: <https://javarush.ru/groups/posts/1973-abstraktniye-klassy-v-java-na-konkretnykh-primera-kh> (дата обращения: 15.05.21)
- 7 The Eclipse Foundation, 2020. URL: <https://www.eclipse.org/> (дата обращения: 23.05.21)
- 8 Для чего в Java нужны интерфейсы. URL: <https://javarush.ru/groups/posts/1981-dlja-chego-v-java-nuzhny-interfeysy> (дата обращения: 16.05.21)
- 9 Что такое JSP? Разбираемся с возможностями на практике. URL:

<https://javarush.ru/groups/posts/2655-что-такое-jsp-разбираемся-s-возможностями-na-praktike> (18.05.21)

10 Учимся верстать: что такое CSS. URL: https://skillbox.ru/media/code/что_такое_css/ (дата обращения: 22.05.21)

11 Основы CSS. Так что же такое CSS? URL: https://developer.mozilla.org/ru/docs/Learn/Getting_started_with_the_web/CSS_basics (дата обращения: 23.05.21)

12 Heroku. URL: <https://www.heroku.com/> (дата обращения: 20.05.21)

13 Travis CI. URL: <https://travis-ci.org/> (дата обращения: 21.05.21)

14. ГОСТ 19.201-78. «Единая система программной продукции (ЕСПД). ТЗ. Требования к содержанию и оформлению» [Текст] – введ. 1980 – 01 – 01 – М.: Стандартинформ, 2010 – 6с. – (Система стандартов по информации, библиотечному и издательскому делу).

15 ГОСТ 19.401-78. «Единая система программной продукции (ЕСПД). Текст программы. Требования к содержанию и оформлению» [Текст] – введ. 1980 – 01 – 01 – М.: Стандартинформ, 2010 – 6с. – (Система стандартов по информации, библиотечному и издательскому делу).