|  |  |  |
| --- | --- | --- |
|  |  |  |
| МИНОБРНАУКИ РОССИИ | | |
| Федеральное государственное бюджетное образовательное учреждение  высшего образования  **«МИРЭА – Российский технологический университет»**  **РТУ МИРЭА** | | |

**Институт информационных технологий (ИТ)**

**Кафедра инструментального и прикладного программного обеспечения (ИиППО)**

**КУРСОВАЯ РАБОТА**

по дисциплине: Шаблоны программных платформ языка Джава

по профилю: Разработка программных продуктов и проектирование информационных систем

направления профессиональной подготовки: 09.03.04 «Программная инженерия»

Тема: «Онлайн-калькулятор с микросервисом для облачных вычислений на основе Spring»

Студент: Остранков Константин Дмитриевич

Студент: Егоров Дмитрий Романович

Группа: ИКБО-03-19

Работа представлена к защите\_\_\_\_\_\_\_\_\_\_(дата)\_\_\_\_\_\_\_\_\_/ 25.05.2021 /

Руководитель: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ст. преподаватель Зорина Н. В.

Работа допущена к защите\_\_\_\_\_\_\_\_\_\_\_(дата)\_\_\_\_\_\_\_\_\_/Зорина Н. В./

Оценка по итогам защиты: \_\_\_\_\_\_\_\_\_\_

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

|  |  |  |
| --- | --- | --- |
|  |  |  |
| МИНОБРНАУКИ РОССИИ | | |
| Федеральное государственное бюджетное образовательное учреждение  высшего образования  **«МИРЭА – Российскийтехнологическийуниверситет»**  **РТУ МИРЭА** | | |

**Институт информационных технологий (ИТ)**

**Кафедра инструментального и прикладного программного обеспечения (ИиППО)**

**ЗАДАНИЕ**

**на выполнение курсовой работы**

по дисциплине: Шаблоны программных платформ языка Джава

по профилю: Разработка программных продуктов и проектирование информационных систем

направления профессиональной подготовки: Программная инженерия (09.03.04)

Студент: Остранков Константин Дмитриевич

Группа: ИКБО-03-19

Срок представления к защите: 20.05.2021

Руководитель: старший преподаватель Зорина Наталья Валентиновна

**Тема:** Онлайн-калькулятор с микросервисом для облачных вычислений на основе Spring

**Исходные данные:** индивидуальное задание на разработку; документация по Spring Framework и JEE, документация по языку Java (версия не ниже 8); инструменты и технологии: JDK (версия не ниже 8), создание Spring MVC web-приложений, SpringORM, Maven, gitHub, IntelliJIDEA. Нормативный документ: инструкция по организации и проведению курсового проектирования СМКО МИРЭА 7.5.1/04.И.05-18.

**Перечень вопросов, подлежащих разработке, и обязательного графического материала:** 1. Провести анализ предметной области и формирование основных требований к приложению. 2. Обосновать выбор средств ведения разработки. 3. Разработать приложение с использование фреймворка Spring и выбранной технологии, и инструментария.4.Провести тестирование приложения. 5. Оформить пояснительную записку по курсовой работе 6. Провести анализ текста на антиплагиат 7. Создать презентацию по выполненной курсовой работе.

Руководителем проведён инструктаж по технике безопасности, противопожарной технике и правилам внутреннего распорядка.

Зав. кафедрой ИиППО: \_\_\_\_\_\_\_\_\_\_\_/Р. Г. Болбаков/, «\_\_\_\_\_»\_\_\_\_\_\_\_\_\_\_\_\_2021 г.

Задание на КР выдал: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_/ Н.В.Зорина/, «\_\_\_\_\_»\_\_\_\_\_\_\_\_\_\_\_\_2021 г.

Задание на КР получил: \_\_\_\_\_\_\_\_\_\_\_/К. Д. Остранков /, «\_\_\_\_\_»\_\_\_\_\_\_\_\_\_\_\_\_2021 г.

|  |  |  |
| --- | --- | --- |
|  |  |  |
| МИНОБРНАУКИ РОССИИ | | |
| Федеральное государственное бюджетное образовательное учреждение  высшего образования  **«МИРЭА – Российский технологический университет»**  **РТУ МИРЭА** | | |

**Институт информационных технологий (ИТ)**

**Кафедра инструментального и прикладного программного обеспечения (ИиППО)**

**ЗАДАНИЕ**

**на выполнение курсовой работы**

по дисциплине: Шаблоны программных платформ языка Джава

по профилю: Разработка программных продуктов и проектирование информационных систем

направления профессиональной подготовки: Программная инженерия (09.03.04)

Студент: Егоров Дмитрий Романович

Группа: ИКБО-03-19

Срок представления к защите: 20.05.2021

Руководитель: старший преподаватель Зорина Наталья Валентиновна

**Тема:** Онлайн-калькулятор с микросервисом для облачных вычислений на основе Spring

**Исходные данные:** индивидуальное задание на разработку; документация по Spring Framework и JEE, документация по языку Java (версия не ниже 8); инструменты и технологии: JDK (версия не ниже 8), создание Spring MVC web-приложений, SpringORM, Maven, gitHub, IntelliJIDEA. Нормативный документ: инструкция по организации и проведению курсового проектирования СМКО МИРЭА 7.5.1/04.И.05-18.

**Перечень вопросов, подлежащих разработке, и обязательного графического материала:** 1. Провести анализ предметной области и формирование основных требований к приложению, 2. Обосновать выбор средств ведения разработки. 3. Разработать приложение с использование фреймворка Spring и выбранной технологии, и инструментария.4.Провести тестирование приложения. 5. Оформить пояснительную записку по курсовой работе 6. Провести анализ текста на антиплагиат 7. Создать презентацию по выполненной курсовой работе.

Руководителем проведён инструктаж по технике безопасности, противопожарной технике и правилам внутреннего распорядка.

Зав. кафедрой ИиППО: \_\_\_\_\_\_\_\_\_\_\_/Р. Г. Болбаков/, «\_\_\_\_\_»\_\_\_\_\_\_\_\_\_\_\_\_2021 г.

Задание на КР выдал: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_/ Н.В.Зорина/, «\_\_\_\_\_»\_\_\_\_\_\_\_\_\_\_\_\_2021 г.

Задание на КР получил: \_\_\_\_\_\_\_\_\_\_\_/К. Д. Остранков /, «\_\_\_\_\_»\_\_\_\_\_\_\_\_\_\_\_\_2021 г.

УДК 004.4

Остранков К.Д., Егоров Д.Р.«Онлайн-калькулятор с микросервисом для облачных вычислений на основе Spring» / **Курсовая работа** по дисциплине «Шаблоны программных платформ на языке Джава» профиля «Разработка программных продуктов и проектирование информационных систем» направления профессиональной подготовки бакалавриата 09.03.04 «Программная инженерия» (4-ый семестр) / руководитель старший преподаватель Н.В. Зорина / кафедра ИППО Института ИТ МИРЭА.

Целью работы является создание микросервиса для облачных вычислений на основе Spring MVC и Angular.

В рамках работы осуществлены краткий анализ аналогов веб-приложения для облачных вычислений, анализ методов вычислений, составлен список требований для данного приложения

Ostrankov K.D., Egorov D.R. "Online calculator with a microservice for cloud computing based on Spring" / Course work in the discipline "Patterns of software platforms in Java language" of the profile "Software development and design of information systems" of the direction of professional training for bachelor's degree 09.03.04 "Software engineering" (4- semester) / head senior teacher N.V. Zorina / Department of IOPS, Institute of IT MIREA.

The aim of the work is to create a microservice for cloud computing based on Spring MVC and Angular. As part of the work, a brief analysis of analogs of a web application for cloud computing, an analysis of computational methods was carried out, a list of requirements for this application was compiled

М. МИРЭА. Ин-т ИТ. Каф. ИиППО. 2021 г. Иванов Д.М.

**ОГЛАВЛЕНИЕ**

[ВВЕДЕНИЕ 7](#_Toc72861541)

[1. СБОР И АНАЛИЗ ТРЕБОВАНИЙ К ПРОГРАММНОМУ ПРОДУКТУ 8](#_Toc72861542)

[2. РАЗРАБОТКА ВЕБ-СЕРВИСА 12](#_Toc72861543)

[3. ТЕСТИРОВАНИЕ ПРОГРАММНОГО ПРОДУКТА 20](#_Toc72861544)

[3.1 Контейнеризация приложения 20](#_Toc72861545)

[3.2 Проверка работоспособности приложения 22](#_Toc72861546)

[3.3 Проверка работоспособности приложения 24](#_Toc72861547)

[ЗАКЛЮЧЕНИЕ 25](#_Toc72861548)

[СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ 26](#_Toc72861549)

[ПРИЛОЖЕНИЯ 28](#_Toc72861550)

**СПИСОК ИСПОЛЬЗОВАННЫХ СОКРАЩЕНИЙ**

* СSS - каскадные таблицы стилей (Cascading Style Sheets)
* HTML - язык гипертекстовой разметки (HyperText Markup Language)
* ORM – Object-Relational Mapping
* MVC – *Model-View-Controller*
* ПО — Программное обеспечение;
* UML — Unified Modeling Language (унифицированный язык моделирования);
* IEEE — Institute of Electrical and Electronics Engineers (Институт инженеров электротехники и электроники);

# ВВЕДЕНИЕ

Счет был придуман в древнем мире в основном для торговли. И когда нужно было посчитать 5 голов скота, даже 50 - вычисления были относительно простыми. Однако, развивалась государственность, технологии, торговые отношения становились сложнее, появлялись сферы науки, в которых также требовался счет. Все это вело к тому, что вычисления стали сложнее, а записывать все на бумаге, тем более держать в голове, стало также труднее. Потому люди придумали для себя специальные устройства для проведения математических операций - абаки, счеты, арифмометры. Уже эти вычислительные устройства можно назвать калькуляторами. С течением времени и появлением электровакуумных ламп появились первые электронные калькуляторы, перешедшие затем на полупроводниковые транзисторы.

К нашему времени калькуляторы уже стали частью повседневной жизни. Рассчитать стоимость продуктовой корзины, посчитать процент по кредиту, распределить зарплату по месяцу - со всем этим может помочь калькулятор, который находится по умолчанию на каждом компьютере или телефоне.

Принимая это во внимание, мы решили разработать свою версию калькулятора по двум причинам. Во-первых, мы хотели создать свою версию калькулятора, в которую можно будет легко добавлять свои функции для быстрой работы. А во-вторых, такое “классическое” приложение с относительно простым базовым функционалом как калькулятор является хорошим вариантом для изучения и применения новых технологий и фреймворков. Это было особенно важно для нас, как для начинающих программистов, ведь ценность специалиста в сфере IT частично оценивается по его владению новейшими технологиями разработки, языками программирования и современными фреймворками.

Основная часть курсовой работы состоит из следующих разделов:

1. анализ предметной области
2. разработка программного продукта
3. определение стадий, этапов и сроков программы и документации;

# СБОР И АНАЛИЗ ТРЕБОВАНИЙ К ПРОГРАММНОМУ ПРОДУКТУ

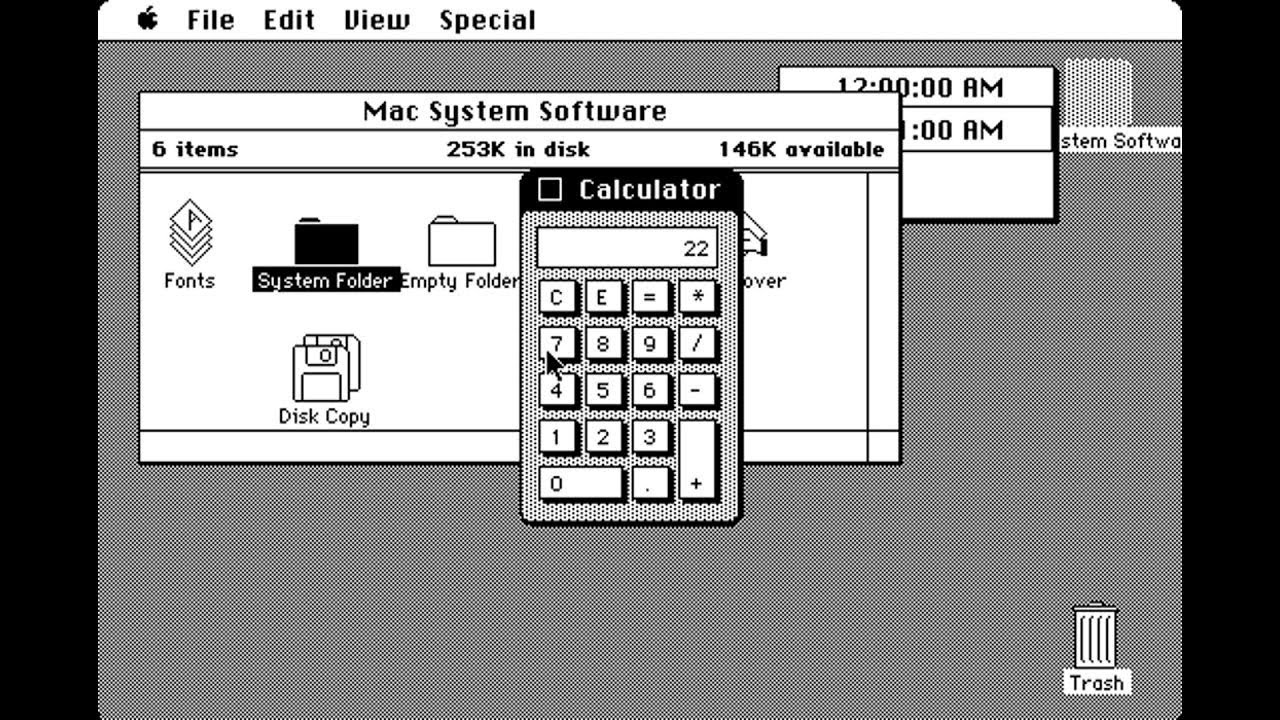
Для составления адекватных требований к итоговому продукту следует обратиться к истории разработки программных калькуляторов. Нас не столь интересует история механических калькуляторов, сколько электронные девайсы. Как таковыми, калькуляторами первоначально, с 1940-х годов, назывались устройства, из которых потом произошли компьютеры: огромные машины, направленные на расчет сложных вычислений. И только потом, в начале 60-х годов 20 века появились отдельные устройства для вычислений - так называемые программируемые калькуляторы. С развитием и распространением персональных компьютеров, в качестве части стандартного программного обеспечения, входящего в состав поставки операционной системы на них появились аналоги калькуляторов, начиная с Macintosh System 1 и Windows 1.0. В них были только самые простые функции: сложение, вычитание, деление и умножение (рис.1.1). 

Рис. 1.1 - снимок экрана Macintosh System 1 с открытым калькулятором (на переднем плане

С тех пор ПО-калькуляторы претерпели существенные изменения. Добавились более различные более сложные функции и шаблоны, различные вариации (“программист”, в котором доступны вычисления в системах счисления, отличных от десятичной, инженерный с дополнительными функциями и прочие). Внутри приложений появились функции построения графиков. (рис. 1.2)

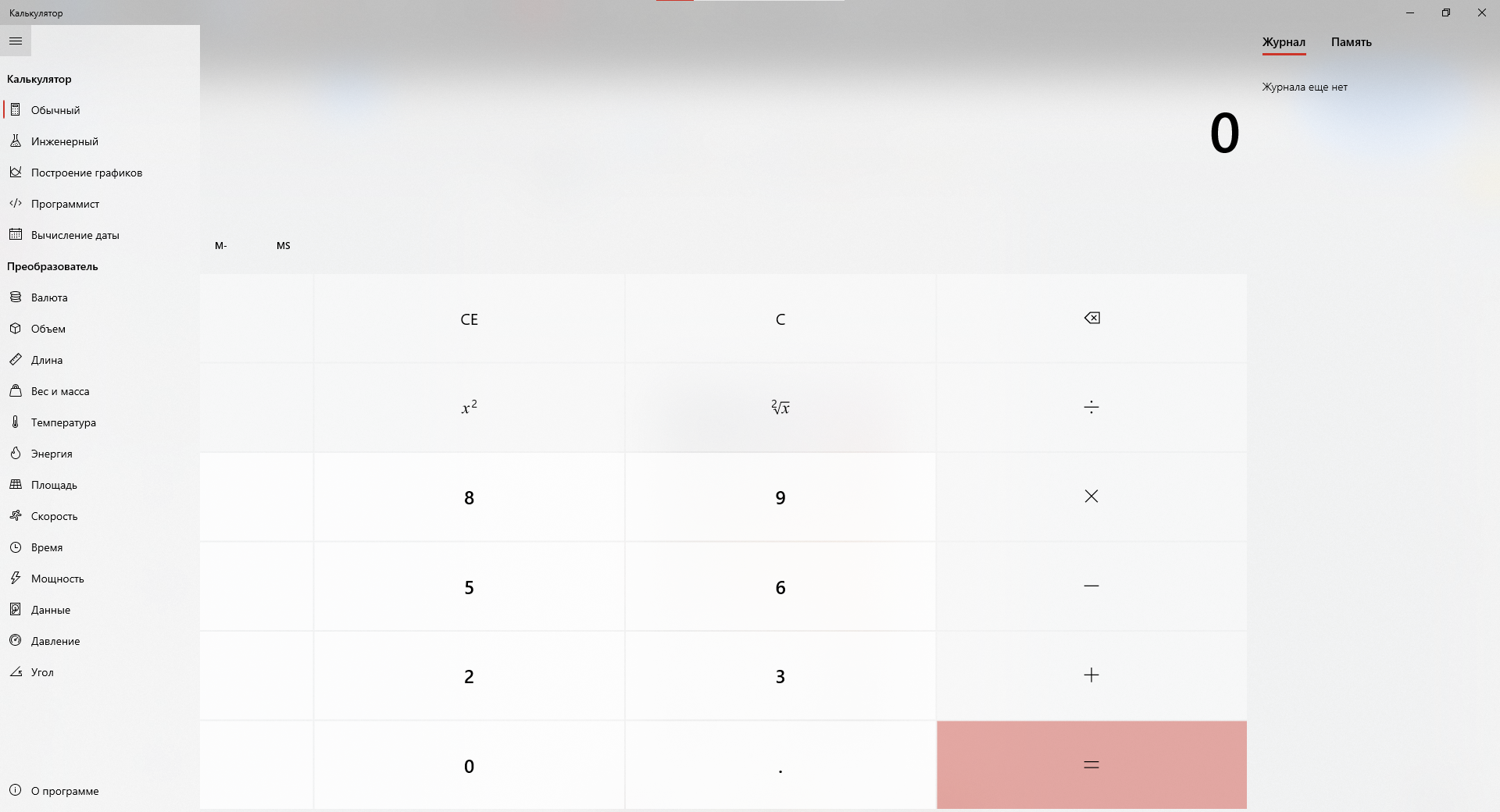


Рис. 1.2 - стандартный калькулятор, поставляемый с Windows 10. На рисунке видно различные варианты калькулятора, а также инструменты для расчета различных физических величин

Однако, целью подобного ПО является предоставление пользователю относительно простых функций, тех, что могут ему понадобиться при каждодневных несложных расчетах. Также существует специализированные приложения для более сложных многоступенчатых вычислений. В целях формирования требований для нашего проекта обратимся к нескольким популярным онлайн-проектам.

Symbolab - сайт, который позиционирует себя не только как калькулятор, но и средство изучения различных разделов математики. Для этого на сайте есть страницы с формулами и правилами по разделам, а также во время вычислений выдаются промежуточные результаты с комментариями по используемым формулам. (рис. 1.3)

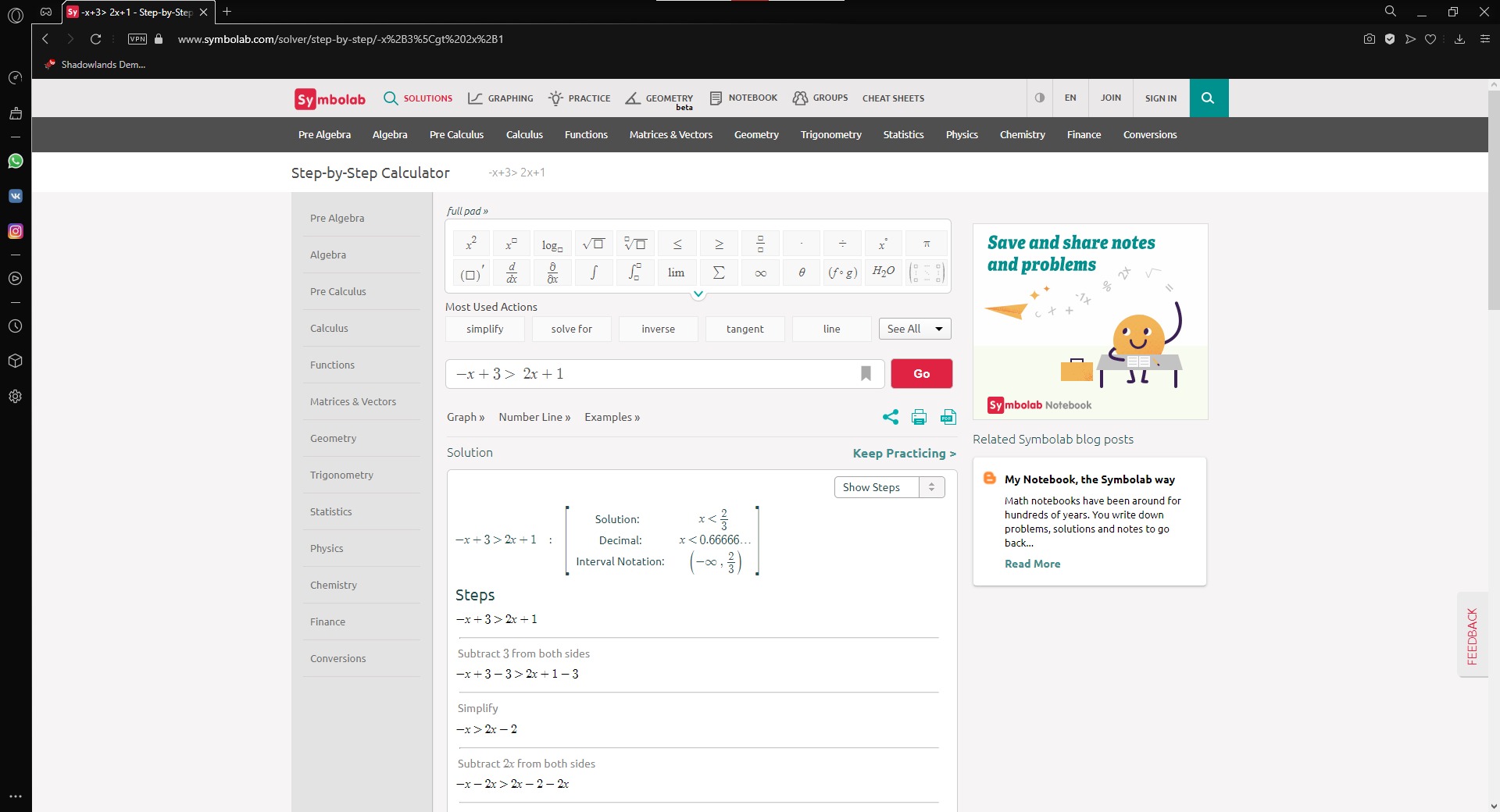


Рис. 1.3 - скриншот сайта Symbolab.com. На рисунке видно пошаговое решение введенной задачи. Также в меню навигации видно упомянутые страницы с дополнительными функциями (построение графиков, “шпаргалки”)

Wolfram Alpha - крайне популярный сайт для сложных вычислений. Он представляет из себя базу знаний и набор вычислительных алгоритмов. Сайт отличается тем, что он буквально отвечает на вопросы любого характера, в том числе и на сложные математические вычисления, поддерживая вывод множества параметров для сложных результатов вычислений. (рис. 1.4)

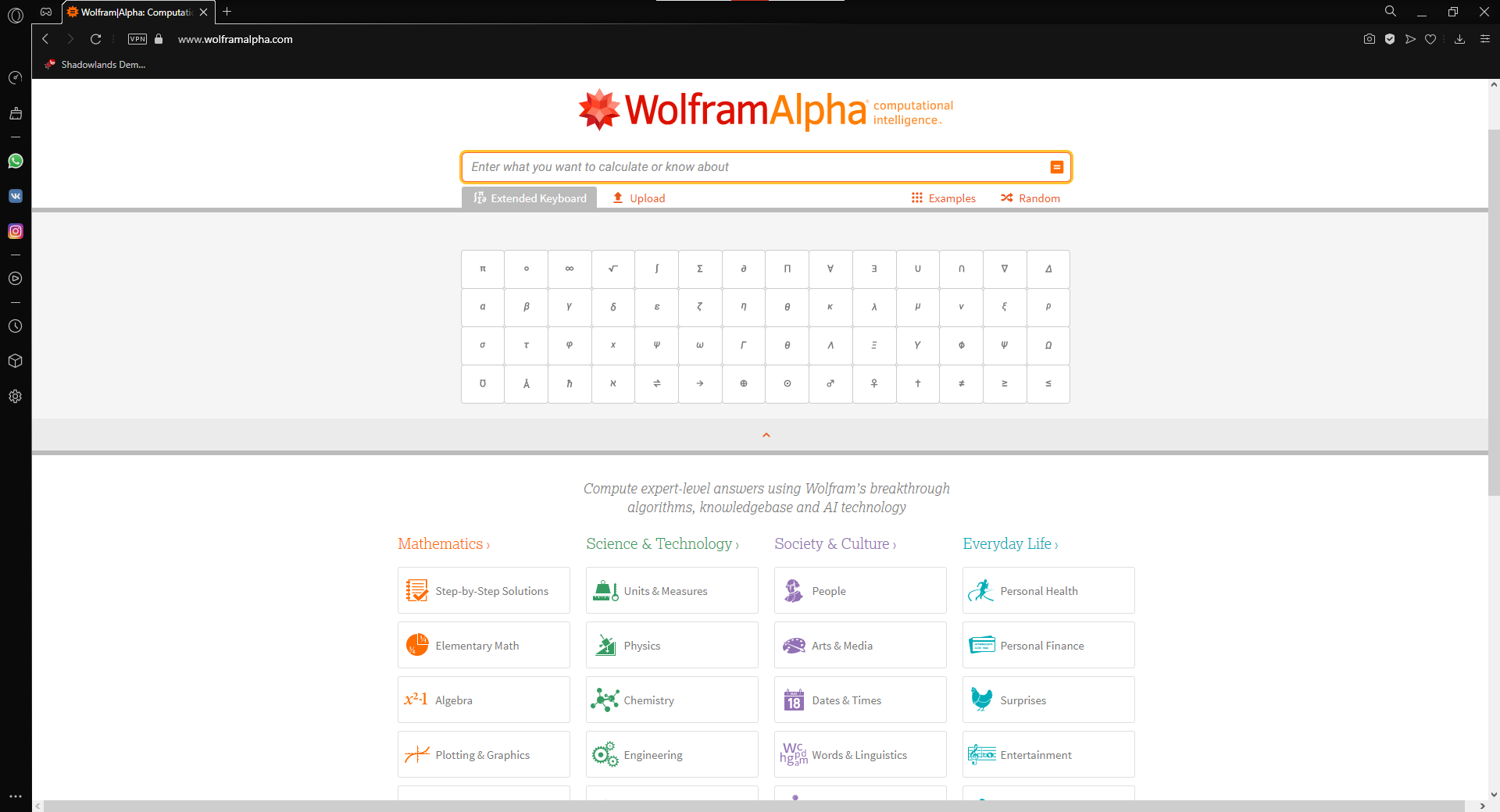


Рис. 1.4 - домашняя страница Wolfram Alpha с включенной клавиатурой дополнительных символ для ввода математических задач.

**Вывод к разделу 1:**

На основе этих сайтов и некоторых собственных пожеланий мы сформировали следующие требования к проекту:

1. Получившийся продукт должен реализовывать хотя бы базовый функционал калькулятора (простейшие вычисления) и иметь понятный интерфейс.
2. Серверная часть проекта должна быть податливой и легкой для изменения, чтобы у разработчика имелась возможность простого добавления новых функций.
3. В целях самосовершенствования и развития, в сайте, помимо указанных в задании компонентов, должно использоваться большое количество новых для нас, разработчиков, технологий и фреймворков, дабы разобраться в их работе и структуре на работающем примере.

# РАЗРАБОТКА ВЕБ-СЕРВИСА

Первым шагом в разработке нашего программного продукта был выбор фреймворков и библиотек, так как писать собственный сервис для создания веб-сервера занимает много времени и не входит в требования курсовой работы. Мы остановились на связке String Boot и Apache Tomcat, так как именно с этим вариантом мы ознакомились, прослушав курс по предмету “Шаблоны программных платформ языка Джава” и выполнив ряд практических заданий. Первоначально, структура проекта представляла из себя серверную часть, написанную на Java, при запуске сервера в браузере открывалась веб-страница, созданная с помощью HTML, CSS и Javascript, которая служила интерфейсом для ввода и отправки информации. Это был самый базовый вариант, с простейшим функционалом и оформлением.

Однако, нашей целью было использовать больше новых для нас технологий, поэтому, для создания того, что в итоге стало полноценной клиентской частью нашего приложения мы обратились к фреймворку Angular. Он был разработан Google на основе TypeScript (языке программирования, созданном Microsoft на основе Javascript для расширения его возможностей) и предлагает компонентный подход к разработке веб-приложений. Проект состоит из множества компонентов, связанных между собой. Это позволяет разрабатывать модульный и гибкий проект и не повторяться в написании разметки и скриптовой части. Каждый компонент состоит из трех файлов - .css файла со стилями, .ts-файла со скриптовой частью, написанной на TypeScript и .html-файла с разметкой страницы. (рис. 2.1)

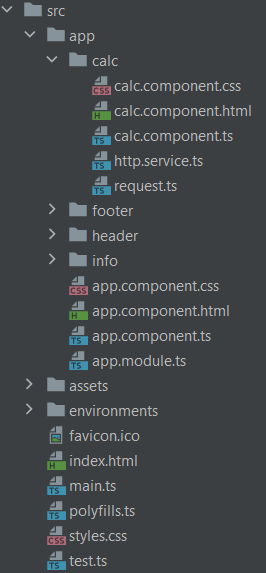


Рис. 2.1. - структура Angular-проекта в IDE WebStorm. На рисунке видны компоненты в папке app, где также располагается “основной” компонент, содержимое которого показывается на странице. Также ниже располагаются файлы для index.html и дополнительные ресурсы.

TypeScript-файлы имеют особую структуру, отличную от Javascript, код скриптов нельзя вставить в html, но можно описывать глобальные переменные, запускать функции при запуске страницы и прочая. (рис. 2.2)

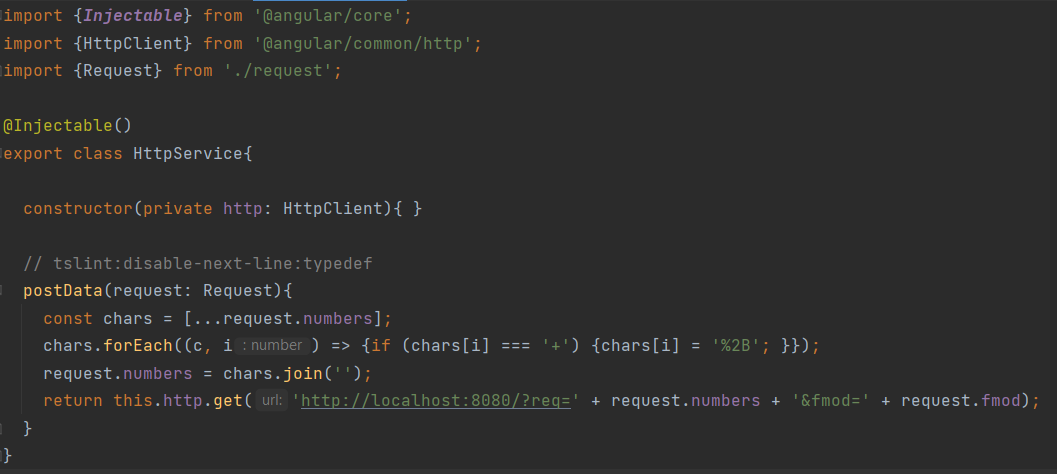


Рис. 2.2 - структура TypeScript-файла. В начале идет список импортов (из стандартных библиотек Angular), затем описывается класс, его поля и методы. Крайне похоже на Java, в чем заключается дополнительное удобство.

Также для создания фронтенда мы применили фреймворк Bootstrap - находящийся в свободном доступе инструмент, включающий набор HTML- и CSS-шаблонов для оформления сайтов, конкретно - кнопок, навигации, форм и прочего. Он был выбран для создания простого дизайна, чтобы не разрабатывать его самому. Из Bootstrap нами были взяты панель навигации и общие инструменты разметки страниц для организации адаптивности веб-дизайна. (рис. 2.3)



Рис. 2.3. - теги Bootstrap пишутся так же, как и стандартные теги HTML, позволяя добавлять определенную стилизацию, описанную в шаблонах фреймворка.

Для бэкенда нашего приложения мы использовали язык Java, фреймворк Spring, и систему сборки Maven.

При помощи Spring разворачиваем контроллер, который слушает порт, и принимает запросы с фронтенда. (рис. 2.4)

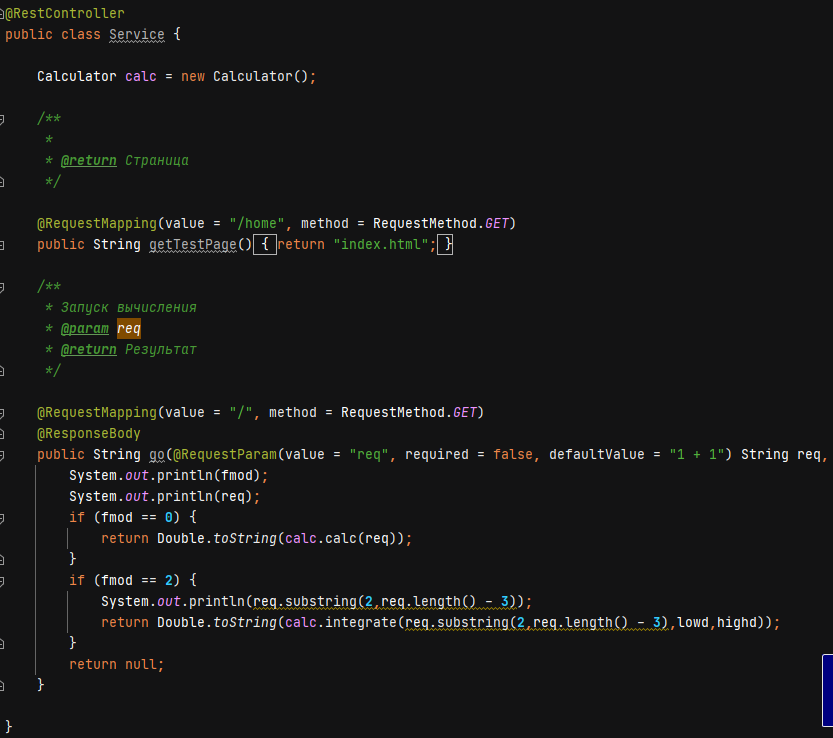


Рисунок 2.4 – Основная часть контроллера приложения

Одним из важнейших элементов является заголовки CORS. CORS — это механизм, который дает контролировать доступ к данным на веб странице по сети. Необходим особый набор хедеров для безопасной связи.

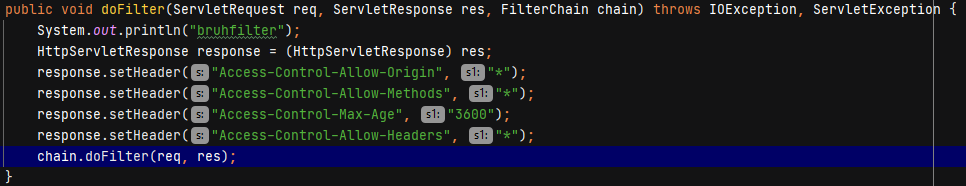
Спецификация CORS включена в WHATWG's Fetch Living Standard. Эта спецификация описывает, как CORS в настоящее время реализуется в браузерах. Более ранняя спецификация была опубликована в качестве рекомендации W3C. 

Рисунок 2.5 – Установка допустимых хедеров

Список представленных хедеров(Рис. 2.5):

* Access-Control-Allow-Origin: указывает, какие домены могут обращаться к ресурсам сайта.
* Access-Control-Allow-Methods: определяет, какие HTTP-запросы (GET, PUT, DELETE и т. д.) могут быть использованы для доступа к ресурсам.
* Access-Control-Max-Age: указывает время жизни предзапроса (также он называется "предполетным") доступности того или иного метода, после которого должен быть выполнен новый запрос на тот или иной метод.
* Access-Control-Allow-Headers: определяет список допустимых хедеров в запросе

Также был установлен дополнительный Web-фильтр для некоторых типов запросов(Рис. 2.6).



Рисунок 2.6 – Установка дополнительного фильтра

В логической части калькулятора был реализован его стандартный функционал, такой как базовые арифметические операции. Так же была реализована тригонометрия и интегрирование.

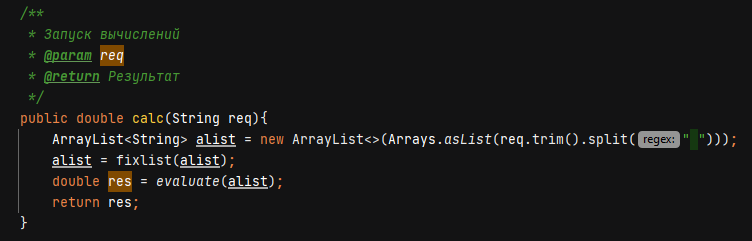
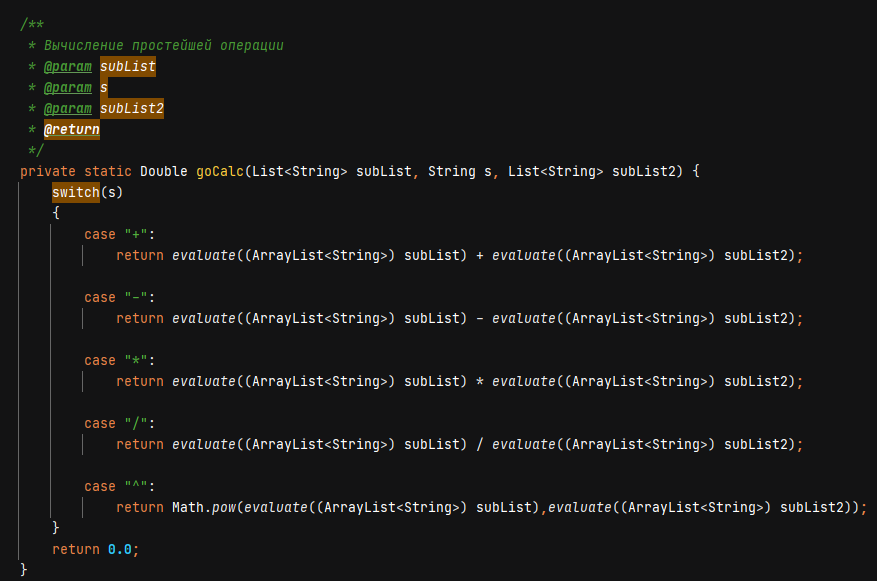


Рисунок 2.7 – Разбиение переданной строки на список действий и вычисление

Калькулятор получает на вход данные с фронтенда, и разбивает строку на последовательность действий в правильном порядке(Рис. 2.7), и выполняет арифметические операции, согласно правилам.

Рисунок 2.7 – Функция простейшего вычисления

Калькулятор последовательно выполняет все операции при помощи взаимной рекурсии. (Рис. 2.7, Рис. 2.8)

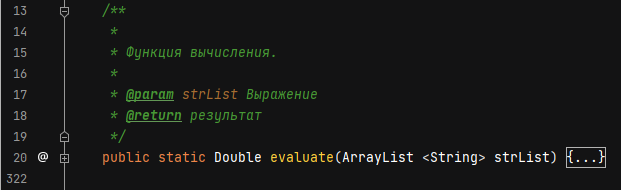


Рисунок 2.8 – Функция сложного вычисления

Рекурсия состоит из 2-х функций, сложного вычисления, которая ответственна за порядок действий и за все унарные арифметические операции, такие как тригонометрия или взятие по модулю, и простейшего вычисления, которая отвечает за все бинарные операции.

Благодаря рекурсии, достигается высокий уровень масштабирования и хорошая вычислительная сложность. А также возможность расширения системы.

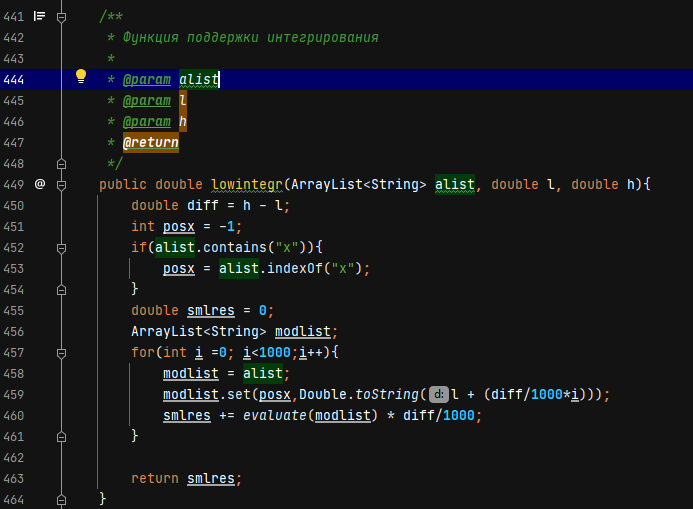


Рисунок 2.9 – Функция интегрирования

Похоже работает функция интегрирования(Рис. 2.9). Приложение способно считать определенные интегралы при помощи геометрического смысла интеграла. Происходит разбиение области интегрирования на малые куски, и подсчет их площади, с последующим сложением.

Основная часть приложения показана на UML-диаграммы(Рис. 2.10).

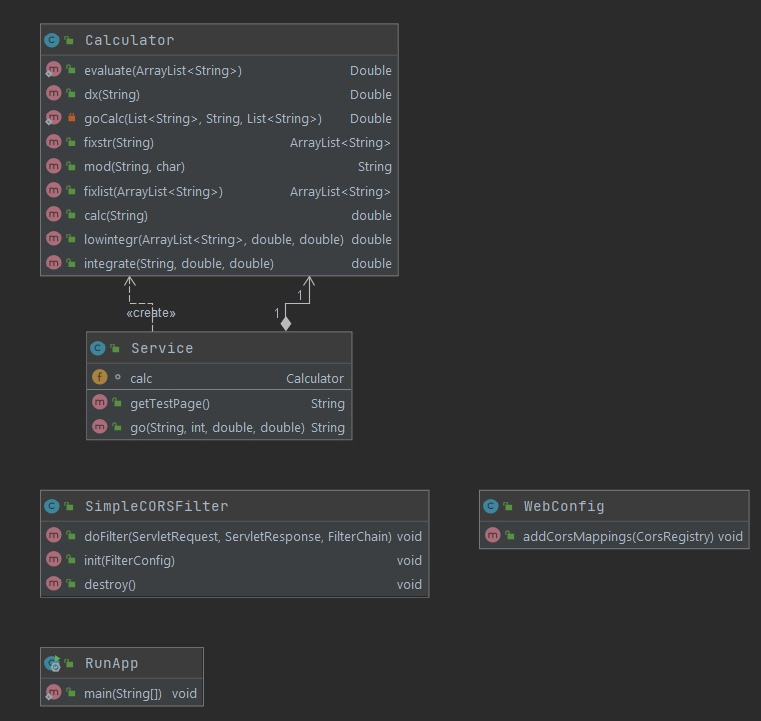


Рисунок 2.9 – UML-диаграмма

**Вывод раздела 2:**

В результате анализа требований к приложению были сделаны план реализации проекта.

На основе требований к калькуляторам, которые были изучены в разделе 1, мы поставили себе серию задач. На основе списка этих задач, было реализовано и описано программное приложение.

# ТЕСТИРОВАНИЕ ПРОГРАММНОГО ПРОДУКТА

### 3.1 Контейнеризация приложения

Обе части приложения была проведена контейнеризация при помощи Docker.

Сборка каждого образа была произведена при помощи Dockerfile

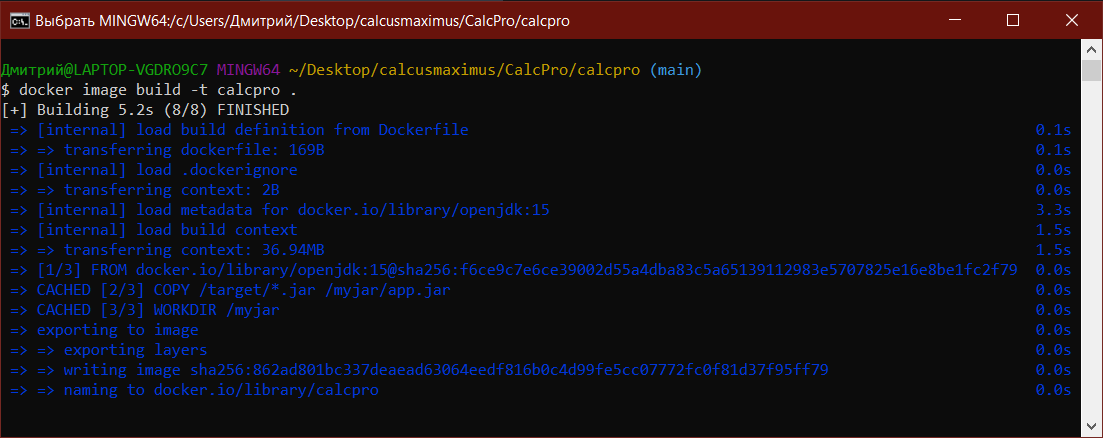


Рисунок 3.1.1 – Сборка образа бэкенда

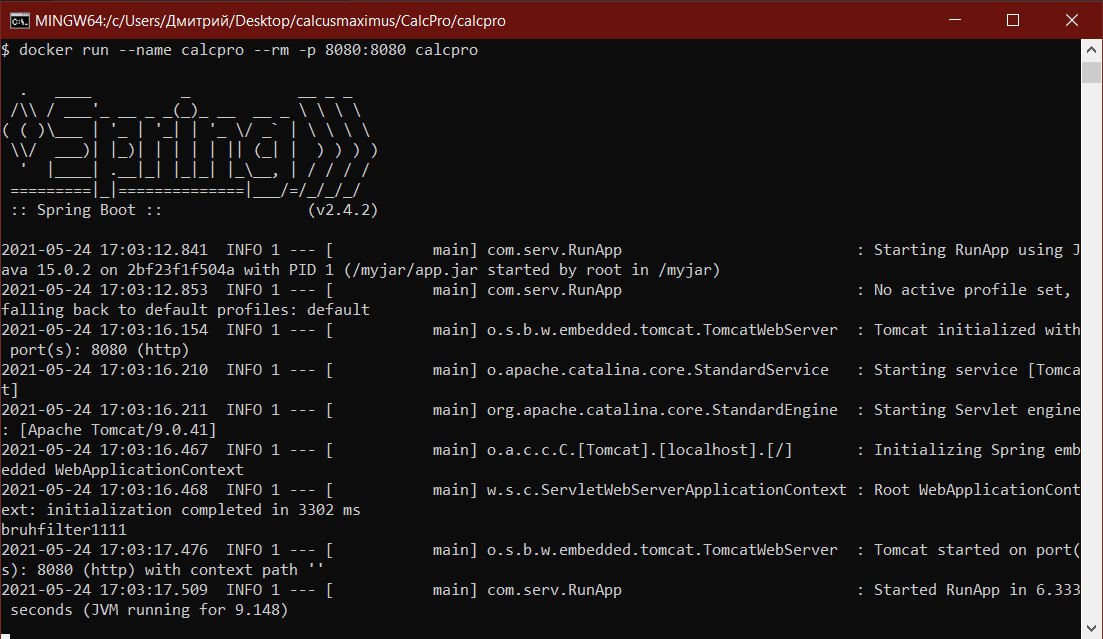


Рисунок 3.1.2 – Запуск образа бэкенда

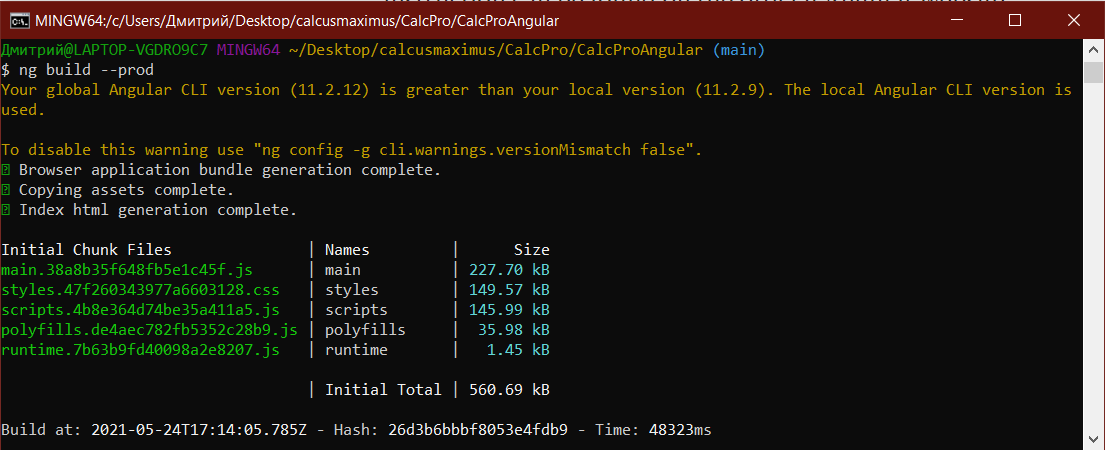


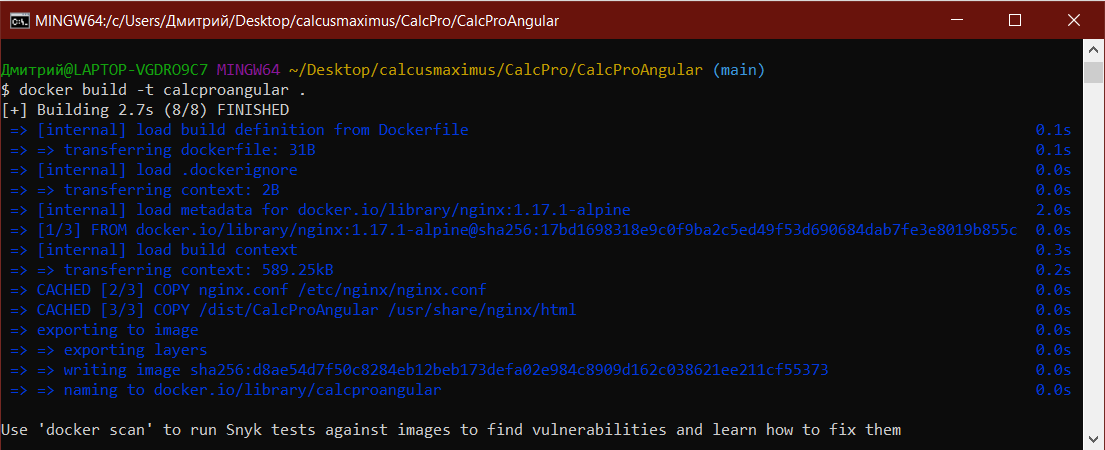
Рисунок 3.1.3 – Сборка образа фронтенда

Рисунок 3.1.4 – Сборка образа фронтенда(часть 2)

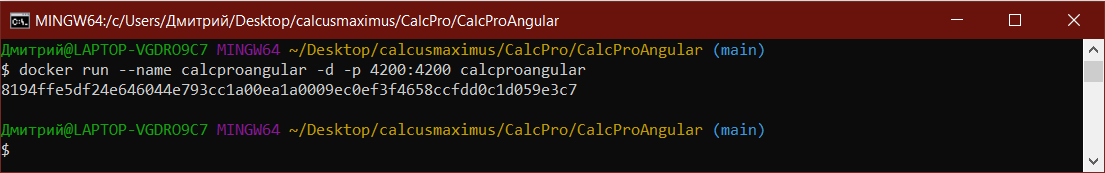


Рисунок 3.1.5 – Запуск образа фронтенда

### 3.2 Проверка работоспособности приложения

Для проверки работоспособности приложения мы использовали браузер Mozilla Firefox версии 88.0.1.

При тестировании мы разворачивали приложение на localhost:4200.



Рисунок 3.2.1 – Вписывание выражения

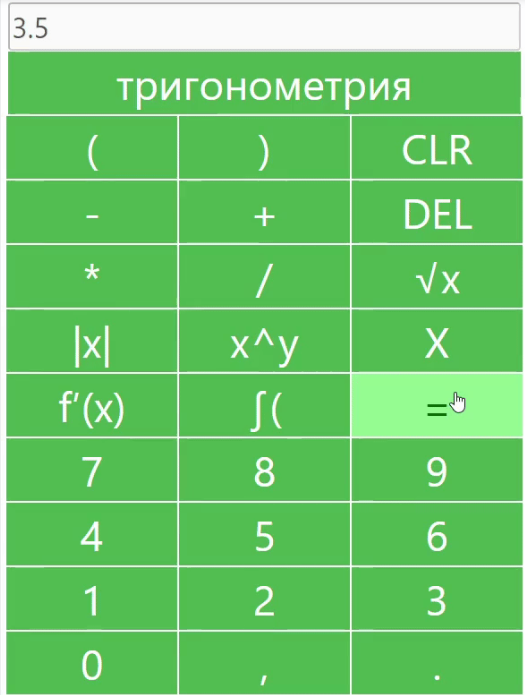


Рисунок 3.2.2 – Результат

### 3.3 Юнит-тестирование

Было произведено юнит тестирование приложение при помощи библиотеки Mockito(Рис.3.2.2).

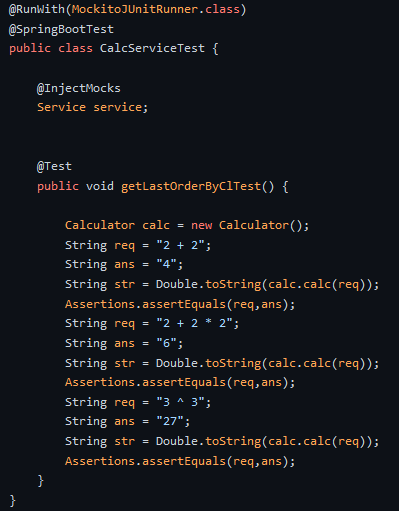


Рисунок 3.3.1 – Один из вариантов теста

Был создан объект Calculator и протестирована

### 3.4 Генерация документации

Для генерации документации приложения используется плагин Javadoc. При помощи него мы генерируем из прикрепленных к методам и классам комментариев сайт с удобным представлением документации

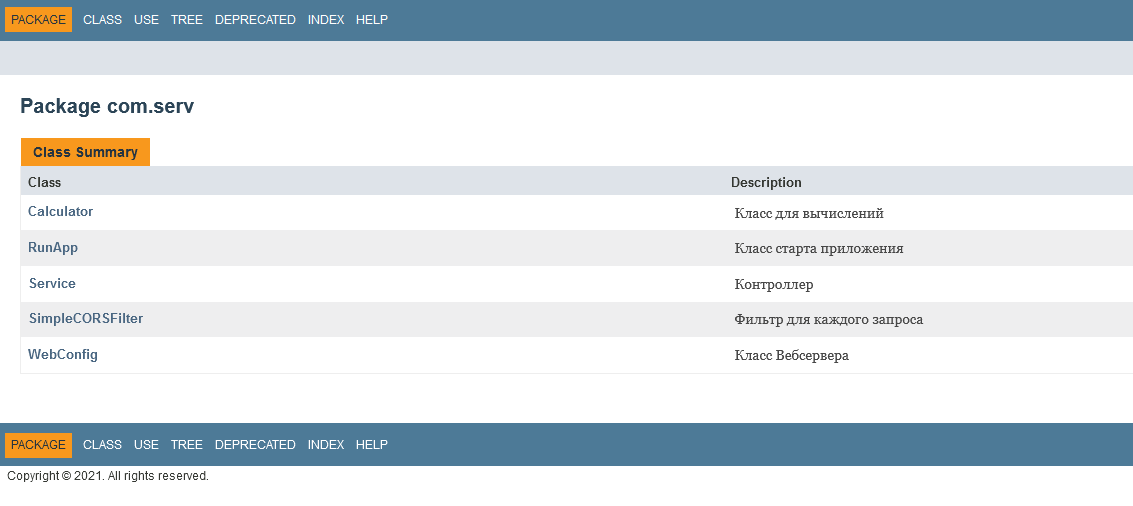


Рисунок 3.4.1 – Заглавная страница документации

**Выводы к разделу 3:**

В данном разделе был продемонстрирован пример развертывания приложения, тестирование приложения при помощи браузера и модульного тестирования и генерация документации

# ЗАКЛЮЧЕНИЕ

В процессе выполнения данной курсовой работы была проведена разработка веб-приложения «Онлайн-калькулятор с микросервисом для облачных вычислений на основе Spring». Спроектированное приложение дает студентам возможность дистанционно загружать свои работы, а преподавателям, в свою очередь, – смотреть их и оценивать.

Был проведён анализ предметной области с выявлением базовых функций для подобного класса приложений, а также определены требования, которые должны быть реализованы.

Был протестированы требования заказчика, изучены техники тестирования и выбраны те их них, которые позволяют провести наиболее качественное и быстрое тестирование веб-системы.

Был составлен план разработки приложения и развертывание образа Dockerдля демонстрации в виртуальной среде.

В процессе разработки курсовой работы были приобретены следующие компетенции:

ПК-1 - способен выполнять разработку и интеграцию программных модулей и компонент, верификацию выпусков программных продуктов информационных систем

ПК-1.1 - Знать: методы и средства сборки модулей и компонент программного обеспечения при создании информационных систем

ПК-1.12 - Уметь: применять методы и средства создания программных интерфейсов информационных систем

ПК-1.14 - Владеть: разработкой процедур сборки модулей и компонент программного обеспечения при внедрении информационных систем

# СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ

1. Докука, О. Практика реактивного программирования в Spring 5 / О. Докука, И. Лозинский. — Москва : ДМК Пресс, 2019. — 508 с. — ISBN 978-5-97060-747-3. — Текст : электронный // Лань : электронно-библиотечная система. — URL: https://e.lanbook.com/book/131708 (дата обращения: 23.05.2021)Фримен А. Профессиональный ASP.NET Core MVC. — Apress, 2016. — 608 с. [Дата обращения: 11.05.21];
2. Сейерс, Э. Х. Docker на практике / Э. Х. Сейерс, А. Милл ; перевод с английского Д. А. Беликов. — Москва : ДМК Пресс, 2020. — 516 с. — ISBN 978-5-97060-772-5. — Текст : электронный // Лань : электронно-библиотечная система. — URL: https://e.lanbook.com/book/131719 [Дата обращения: 20.05.21];
3. Кочер, П. С. Микросервисы и контейнеры Docker : руководство / П. С. Кочер ; перевод с английского А. Н. Киселева. — Москва : ДМК Пресс, 2019. — 240 с. — ISBN 978-5-97060-739-8. — Текст : электронный // Лань : электронно-библиотечная система. — URL: https://e.lanbook.com/book/123710 [Дата обращения: 19.05.21];
4. Вязовик, Н. А. Программирование на Java : учебное пособие / Н. А. Вязовик. — 2-е изд. — Москва : ИНТУИТ, 2016. — 603 с. — Текст : электронный // Лань : электронно-библиотечная система. — URL: https://e.lanbook.com/book/100405 [Дата обращения: 05.05.21];
5. Маркелов, А. А. Введение в технологию контейнеров и Kubernetes / А. А. Маркелов. — Москва : ДМК Пресс, 2019. — 194 с. — ISBN 978-5-97060-775-6. — Текст : электронный // Лань : электронно-библиотечная система. — URL: <https://e.lanbook.com/book/131702> [Дата обращения: 05.05.21];
6. Дилеман, П. Изучаем Angular 2 / П. Дилеман ; под редакцией А. Н. Киселева ; перевод с английского Р. Н. Рагимова. — Москва : ДМК Пресс, 2017. — 354 с. — ISBN 978-5-97060-461-8. — Текст : электронный // Лань : электронно-библиотечная система. — URL: https://e.lanbook.com/book/100906 [Дата обращения: 02.05.21].
7. Пономарев, С. Н. Angular на примерах. Создаем weв - приложения с нуля : руководство / С. Н. Пономарев. — Санкт-Петербург : Наука и Техника, 2019. — 288 с. — ISBN 978-5-94387-785-8. — Текст : электронный // Лань : электронно-библиотечная система. — URL: https://e.lanbook.com/book/139153 [Дата обращения: 16.05.21];
8. Коузен, К. Современный Java: рецепты программирования / К. Коузен. — Москва : ДМК Пресс, 2018. — 275 с. — ISBN 978-5-97060-134-1. — Текст : электронный // Лань : электронно-библиотечная система. — URL: https://e.lanbook.com/book/116121 [Дата обращения 20.05.21];
9. Мухамедзянов, Р. Р. JAVA. Серверные приложения / Р. Р. Мухамедзянов. — Москва : СОЛОН-Пресс, 2007. — 336 с. — ISBN 5-93455-134-5 . — Текст : электронный // Лань : электронно-библиотечная система. — URL: https://e.lanbook.com/book/13633 [Дата обращения 20.05.21];
10. Моуэт, Э. Использование Docker / Э. Моуэт ; научный редактор А. А. Маркелов ; перевод с английского А. В. Снастина. — Москва : ДМК Пресс, 2017. — 354 с. — ISBN 978-5-97060-426-7. — Текст : электронный // Лань : электронно-библиотечная система. — URL: https://e.lanbook.com/book/93576 [Дата обращения 13.05.21].

# ПРИЛОЖЕНИЯ

Исходный код данного веб-приложения, включающий в себя все файлы, является слишком объемным, прикреплять его в качестве приложения неуместно, можно ознакомиться с ним перейдя по ссылке: <https://github.com/TheRealOkone/CalcPro> .