МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ

РОССИЙСКОЙ ФЕДЕРАЦИИ

ФГБОУ ВО

«Пермский государственный национальный исследовательский университет»

Механико-математический факультет

Кафедра информационной безопасности и систем связи

**Отчёт**

по лабораторной работе №3 «Разработка распределенного приложения, использующего технологию веб-служб»

по дисциплине «Технологии разработки распределенных приложений»

Работу выполнил Проверил

студент гр. КМБ-16 доцент кафедры

Якин Никита прикладной математики

Николаевич и информатики,

20.11.2020 к.ф.-м.н., доц.

Деменев Алексей

Геннадьевич

Пермь, 2020 г

**Содержание**

[**Введение** 3](#_Toc54955317)

[**Основная часть** 4](#_Toc54955318)

[**Постановка задачи** 4](#_Toc54955319)

[**Выбор предметной области и инструментов** 4](#_Toc54955320)

[**Сценарии использования** 5](#_Toc54955321)

[**Описание приложения** 6](#_Toc54955322)

[**Заключение** 9](#_Toc54955323)

[**Литература** 10](#_Toc54955324)

# **Введение**

**Цель:** изучение возможностей технологии веб-служб для создания распределенных приложений.

**Формируемые компетенции:** способность применять на практике теоретические основы и общие принципы разработки распределенных систем; способность использовать на практике стандарты сетевого взаимодействия компонент распределенной системы.

Необходимо реализовать веб-сервис и приложение, его использующее

1. Веб-сервис и приложение, его использующее, должны быть разработаны на разных объектно-ориентированных языках программирования.
2. Веб-сервис должен принимать параметры и передавать приложению результат.
3. В качестве параметров должны передаваться объекты классов, написанных самостоятельно, т.е. не должны передаваться строки, числа или другие простейшие типы.

# **Основная часть**

## **1.1 Постановка задачи**

Спроектировать, реализовать, протестировать и оценить приложение в соответствии с таблицей №1.

***Таблица 1. Распределение баллов за выполнение работы (***Максимальное количество баллов — 6)***:***

|  |  |
| --- | --- |
| **Требование к заданию** | **Максимальное количество баллов** |
| Приложение, написанное студентом, работает в сети Интернет без сбоев. | 2 |
| Веб-сервис и приложение, его использующее, написаны на разных языках программирования. | 1 |
| Веб-сервис принимает от приложения исходные данные и возвращает результат. | 1 |
| В качестве параметров веб-сервис принимает объекты классов. | 2 |

## **1.2 Выбор предметной области и инструментов**

Для разработки веб-сервиса был выбран язык программирования Python 3.7 [1]. В качестве IDE использовался PyCharm Community Edition [2]. PyCharm Community Edition является бесплатной версией, обладающей усеченным набором возможностей [3].

Веб-сервис реализуется с помощью «gRPC» [4][5] - это высокопроизводительный фреймворк для удаленного вызова процедур, разработанный компанией Google. В качестве языка описания веб-сервиса используется «Protocol Buffers (Protobuf)» - язык описания интерфейсов (IDL) [6], который подразумевает определение сервиса через указание методов, которые могут быть вызваны удалённо, с их параметрами и типами принимаемых и возвращаемых данных. Это осуществляется с помощью *протофайла* - текстового файла с расширением *.proto*. В этом файле описываются «сообщения» – объекты, содержащие набор полей, представляющих собой пары {тип данных – имя}. Также в *протофайле* описываются сервисы, каждый из которых содержит набор методов с описанием типа принимаемых и возвращаемых данных.

Был сформирован следующий «протофайл»:

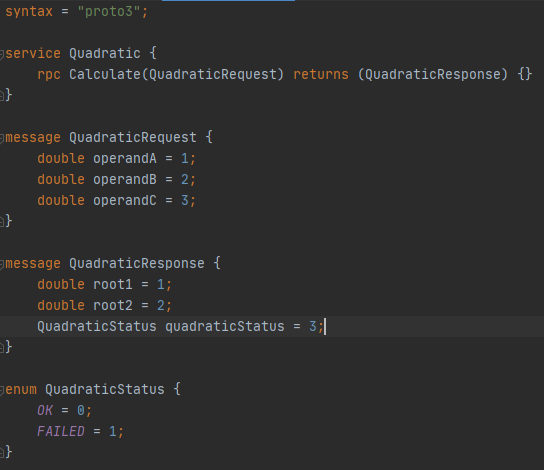


Рис. 1. Структура протофайла

Для реализации клиентского приложения, использующего веб-сервис, был выбран С# [7]. В качестве IDE использовался Visual Studio 2019 community[8]. Community версия имеет пробный период 90 дней, после которого для продления использования следует просто авторизоваться в Visual Studio с использованием аккаунта Microsoft.

Для взаимодействия с файлом инициализации в C# использовалась open-source библиотека ini-parser [9].

## **1.3 Сценарии использования**

Приложение может использоваться в следующих сценариях.

1. **Вычисление корней квадратного уравнения**

**Шаги выполнения:**

* + - 1. Администратор сервера заполняет файл connection.ini
      2. Администратор запускает сервер
      3. Клиент заполняет файл connection.ini
      4. Клиент запускает клиентское приложение
      5. Клиент вводит коэффициенты a, b и с.

**Ожидаемый результат:** Клиентское приложение получает ответ от веб-сервиса.

**Способ тестирования.**

Для тестирования пройдём по шагам и проверим полученный результат с Wolfram Alpha.

От веб-сервиса был получен следующий ответ (рис. 2):

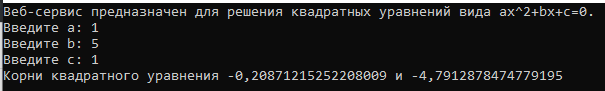


Рис. 2. Результат работы клиентского приложения

Результат полученный от Wolfram Alpha (рис. 3-5):

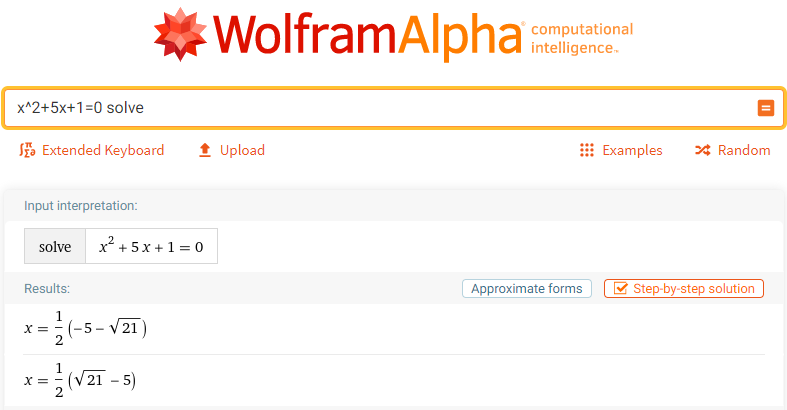


Рис. 3. Результат решения квадратного уравнения

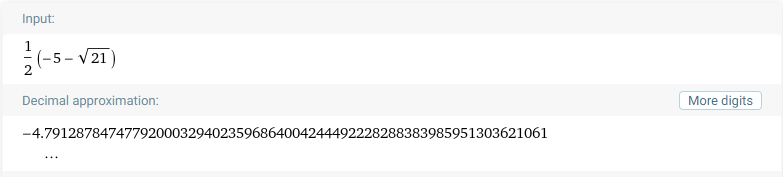


Рис. 4. Результат решения квадратного уравнения 2

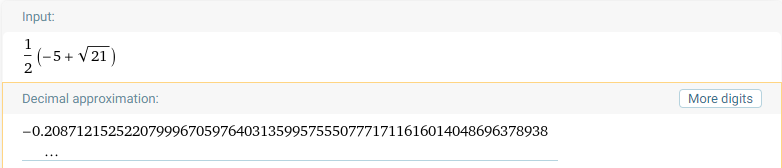


Рис. 5. Результат решения квадратного уравнения 3

## **1.4 Описание приложения**

Решение лабораторной работы разделено на два приложения: сервер, написанный на Python и клиентское-приложение, написанное на C#, это сделано для выполнения второго требования лабораторной работы «Веб-сервис и приложение, его использующее, написаны на разных языках программирования». Исходный код приложений размещён на GitHub: <https://github.com/DrakeTHPS/TRRP3>.

Приложение сервер считывает из конфигурационного файла connection.ini (рис. 6) данные для запуска сервера. После успешного считывания данных запускается сервер на порте, указанном в конфигурационном файле, и ждёт вызова процедуры от клиента.

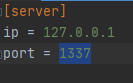


Рис. 6. Файл connection.ini

В случае получения от вызова процедуры от клиента, серверное приложение получает объект, описанный в proto файле как QuadraticRequest (риc. 1). На основании коэффициентов, полученных из объекта QuadraticRequest вычисляется дискриминант. Если дискриминант меньше нуля, то клиенту отправляется объект QuadraticResponse со статусом FAILED. Если же дискриминант больше либо равняется нулю, то вычисляются корни квадратного уравнения и отправляются вместе со статусом OK.

Клиентское приложение считывает из конфигурационного файла connection.ini (рис. 6) данные для обращения к серверу. После успешного считывания создаётся соединение с сервисом.

У пользователя запрашиваются коэффициенты квадратного уравнения, корни которого он хочет получить, затем формируется объект QuadraticRequest и отправляется веб-сервису с ожиданием ответа. После получения ответа проверяется статус, если статус QuadraticResponse равняется OK, достаются полученные корни и выводятся в консоль (рис. 2). Если статус FAILED, то в консоль выводится сообщение, что уравнение не имеет вещественных корней (рис. 7).

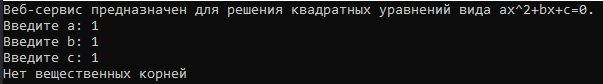


Рис. 7. Пример квадратного уравнения без вещественных корней

# **Заключение**

В ходе работы достигнута цель и выполнены следующие требования:

Приложение, написанное студентом, работает в сети Интернет без сбоев. (Наличие конфигурационного файла connection.ini) 2 балла.

Веб-сервис и приложение, его использующее, написаны на разных языках программирования. (Сервер написан на Python, клиентское приложение написано на C#) 1 балл.

Веб-сервис принимает от приложения исходные данные и возвращает результат. 1 балл.

В качестве параметров веб-сервис принимает объекты классов. (В протофайле описаны объекты, отправляемые и принимаемые клиентом и сервисом (рис. 1)) 2 балл.

Итого: 6 баллов

# **Литература**

* + 1. Лицензия Python 3.7, <https://docs.python.org/3.7/license.html> (Дата обращения 19.11.2020)
    2. PyCharm Community Edition. Распространяется под лицензией Apache 2, URL: <https://www.jetbrains.com/pycharm/features/editions_comparison_matrix.html> (Дата обращения 19.11.2020)
    3. PyCharm Community Edition. Ссылка на скачивание. URL: <https://www.jetbrains.com/pycharm/download/#section=windows> (Дата обращения 19.11.2020)
    4. Ссылка на FAQ по gRPC. URL: <https://grpc.io/faq/> (дата обращения: 19.11.2020)
    5. Ссылка на лицензию gRPC (Apache 2.0). URL: <https://github.com/grpc/grpc/blob/master/LICENSE> (дата обращения: 19.11.2020
    6. Спецификация IDL. URL: <https://www.omg.org/spec/IDL/4.2/> (дата обращения: 19.11.2020)
    7. .Net open-source платформа без лицензирования <https://dotnet.microsoft.com/platform/free>
    8. УСЛОВИЯ ЛИЦЕНЗИИ НА ПРОГРАММНОЕ ОБЕСПЕЧЕНИЕ MICROSOFT MICROSOFT VISUAL STUDIO COMMUNITY 2019 <https://visualstudio.microsoft.com/ru/license-terms/mlt031819/>
    9. Ini-parser. Open-source библиотека распространяется по лицензии MIT. URL: <https://github.com/rickyah/ini-parser/blob/development/LICENSE>