МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ

РОССИЙСКОЙ ФЕДЕРАЦИИ

ФГБОУ ВО

«Пермский государственный национальный исследовательский университет»

**Отчёт**

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

по дисциплине «Технологии разработки распределенных приложений»

Работу выполнил Проверил

студент гр. ФИТ-2 доцент кафедры

Габов Егор математического обеспечения

Владимирович вычислительных систем,

22.12.2020 к.ф.-м.н., доц.

Деменев Алексей

Геннадьевич

Пермь, 2020 г

**Содержание**

[**Введение** 3](#_Toc59527799)

[**1.** **Основная часть** 4](#_Toc59527800)

[**1.1 Постановка задачи** 4](#_Toc59527801)

[**1.2 Выбор предметной области и инструментов** 4](#_Toc59527802)

[**1.3Сценарии использования программы-клиента** 6](#_Toc59527803)

[**1.4 Тестирование** 7](#_Toc59527804)

[**Заключение** 10](#_Toc59527805)

[**Литература** 11](#_Toc59527806)

# **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 серверного и клиентского приложения был выбран язык программирования Python 3.8 [1][2]. В качестве IDE использовался PyCharm Community Edition [3]. PyCharm Community Edition является бесплатной версией, обладающей усеченным набором возможностей [4].

Веб-сервис реализуется с помощью «gRPC» [5][6] - 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.

В качестве утилиты для генерации кода использовалась утилита *protoc* [7]. Поставляется вместе с Protocol Buffers.

Был сформирован следующий «протофайл»:

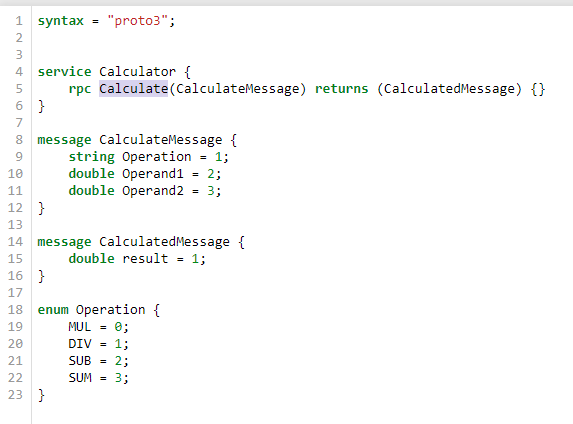


Рис. 1. Структура протофайла

Таким образом, были спроектированы два объекта:

* «CalculateMessage» с двумя числовыми полями (double Operand1 и double Operand2) – для первого и второго операндов – и одним текстовым (string Operation) – для наименования операции. Будет формироваться клиентским приложением и посылаться серверу.
* «CalculatedMessage» с одним числовым полем (double result) – для результата выполнения арифметической операции. Будет формироваться сервером и отправляться клиенту для передачи результата выполнения запроса.

Также в файле описан сервис «Calculator», содержащий единственный метод «Calculate», принимающий на вход объект типа «CalculateMessage» и возвращающий в качестве результата объект типа «CalculatedMessage».

## **Сценарии использования программы-клиента**

Приложение может использоваться в следующих сценариях.

1. **Сложение целых чисел**

**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 первое целое число.
      6. Клиент вводит второе целое число.

**Ожидаемый результат:** Клиентское приложение получает ответ от веб-сервиса, соответствующий результату сложения введенных чисел.

1. **Вычитание целых чисел**

**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 первое целое число.
      6. Клиент вводит второе целое число.

**Ожидаемый результат:** Клиентское приложение получает ответ от веб-сервиса, соответствующий результату вычитания введенных чисел.

1. **Умножение целых чисел**

**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 первое целое число.
      6. Клиент вводит второе целое число.

**Ожидаемый результат:** Клиентское приложение получает ответ от веб-сервиса, соответствующий результату умножения введенных чисел.

1. **Целочисленное деление**

**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 первое целое число.
      6. Клиент вводит второе целое число.

**Ожидаемый результат:** Клиентское приложение получает ответ от веб-сервиса, соответствующий результату целочисленного деления введенных чисел.

## **1.4 Тестирование**

При разработке с помощью инструмента gRPC на основе файла serv\_proto.proto было получено два автогенерируемых класса – serv\_proto\_pb2.py и serv\_proto\_pb2\_grpc.py – для работы с спроектированным сервисом.

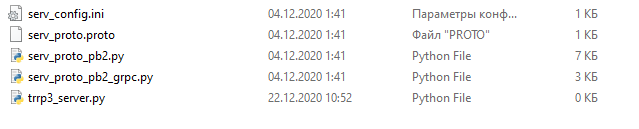


Рис. 2. Автогенерируемые классы serv\_proto\_pb2.py и serv\_proto\_pb2\_grpc.py

Для настройки сетевых параметров применяются конфигурационные файлы «serv\_config.ini» и «cl\_config.ini» для сервера и клиента соответственно.

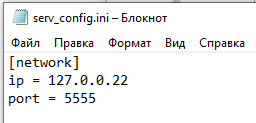


Рис. 2. Содержимое файла serv\_config.ini

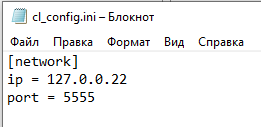


Рис. 3. Содержимое файла cl\_config.ini

Запустим сервер:



Рис. 4. Запуск сервера

Протестируем выполнение каждого сценария:

Сценарий №1

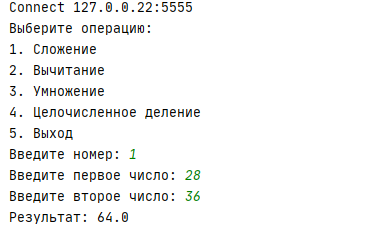


Рис. 5. Выполнение сценария №1

Результат: От сервера получен ответ, соответствующий результату сложения введенных чисел. Тест пройден успешно.

Сценарий №2

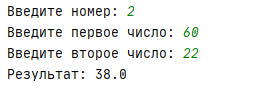


Рис. 6. Выполнение сценария №2

Результат: От сервера получен ответ, соответствующий результату вычитания введенных чисел. Тест пройден успешно.

Сценарий №3

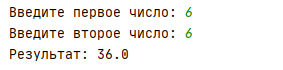


Рис. 7. Выполнение сценария №3

Результат: От сервера получен ответ, соответствующий результату умножения введенных чисел. Тест пройден успешно.

Сценарий №4

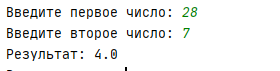


Рис. 8. Выполнение сценария №4

Результат: От сервера получен ответ, соответствующий результату деления введенных чисел. Тест пройден успешно.

Запустим приложение с измененными параметрами сети. Наблюдаем, что работоспособность сохраняется.



Рис. 9. Запуск сервера с измененными параметрами сети

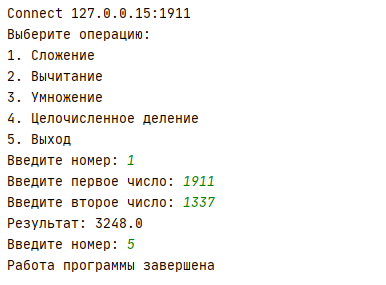


Рис. 10. Работа клиента с измененными параметрами сети

# **Заключение**

В ходе работы достигнута цель и выполнены следующие требования:

* Требование «Приложение, написанное студентом, работает в сети Интернет без сбоев» выполнено полностью. Ожидаемое количество баллов: 2.
* Требование «Веб-сервис и приложение, его использующее, написаны на разных языках программирования» не выполнено. Ожидаемое количество баллов: 0.
* Требование «Веб-сервис принимает от приложения исходные данные и возвращает результат» выполнено полностью. Ожидаемое количество баллов: 1.
* Требование «В качестве параметров веб-сервис принимает объекты классов» выполнено полностью. Ожидаемое количество баллов: 2.

Итого: 5 баллов

# **Литература**

* + 1. Лицензия Python 3.8, <https://docs.python.org/3.7/license.html> (Дата обращения 21.12.2020)
    2. Python 3.8.0. Ссылка наскачивание URL: <https://www.python.org/downloads/release/python-380/> (дата обращения: 21.12.2020)
    3. PyCharm Community Edition. Распространяется под лицензией Apache 2, URL: <https://www.jetbrains.com/pycharm/features/editions_comparison_matrix.html> (Дата обращения 21.12.2020)
    4. PyCharm Community Edition. Ссылка на скачивание. URL: <https://www.jetbrains.com/pycharm/download/#section=windows> (Дата обращения 21.12.2020)
    5. Ссылка на FAQ по gRPC. URL: <https://grpc.io/faq/> (дата обращения: 21.12.2020)
    6. Ссылка на лицензию gRPC (Apache 2.0). URL: <https://github.com/grpc/grpc/blob/master/LICENSE> (дата обращения: 21.12.2020
    7. protocolbuffers/protobuf. URL: <https://github.com/protocolbuffers/protobuf> (дата обращения: 21.12.2020)
    8. Спецификация IDL. URL: <https://www.omg.org/spec/IDL/4.2/> (дата обращения: 21.12.2020)