|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РФ | | | | | | | | |
|  | |  |  | | |  |  | |
| Пермский государственный национальный  исследовательский университет | | | | | | | | |
|  |  | |  | | |  | |  |
|  | ОТЧЕТ  по лабораторной работе №3  «Разработка распределенного приложения, использующего технологию веб-служб»  по дисциплине  «Технологии разработки распределенных приложений» | | | | | | |  |
|  |  | | |  |  | | |  |
|  | Работу выполнил студент гр. КМБ-16  Тарарков Андрей Викторович  16.10.2020 | | |  | Проверил  доцент кафедры прикладной математики и информатики,  к. ф.-м.н., доц.  Деменев Алексей Геннадьевич | | |  |
|  |  | | |  |  | | |  |
| Пермь, 2020 г. | | | | | | | | |

# 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.

Каждый студент выполняет индивидуальное задание.

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.

# Постановка задачи

Требуется спроектировать, реализовать, протестировать и оценить распределённое приложение согласно таблице №1

Максимальное количество баллов, которые студент может получить за выполнение работы равно шести. Распределение баллов за выполнение работы представлено в следующей таблице:

Таблица 1. Требования к выполнению работы

|  |  |
| --- | --- |
| **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 |

Предметная область: обработка простейших арифметических вычислений на стороне сервера.

# Основная часть

## Выбор используемых инструментов

Приложение написано на языках Java 11 (серверная часть), Golang (клиентская часть). Среды разработки приложения: IntelliJ IDEA[2], GoLand[1]. Среды разработки доступны по свободной лицензии для студентов и преподавателей в соответствии с документом TOOLBOX SUBSCRIPTION AGREEMENT FOR EDUCATION[6].

В качестве фреймворка для вызова удалённых процедур был выбран gRPC[3]. gRPC – фреймворк с открытым исходным кодом, разработанный компанией Google, доступный для бесплатного использования. gRPC также является веб сервисом, так как использует язык описания интерфейсов IDL[7], по определению веб сервисов из Web Services Glossary[8].

В качестве протокола сериализации был выбран Protocol Buffers[5]. Протокол с открытым исходным кодом также размещён в публичном репозиторие и доступен для скачивания.

В качестве утилиты для генерации кода использовалась утилита *protoc*[4]. Поставляется вместе с Protocol Buffers.

Исходный код приложения сохранён в GitHub репозиториях <https://github.com/Dasem/trrp3client> , <https://github.com/Dasem/trrp3server> (сервер и клиент соответственно).

## Сценарии использования приложения

1. Выполнение операции на удалённом сервере. Предусловия: у пользователей соответствующие адресы и порты в файлах «default.properties», также запущено серверное приложение.
   1. Пользователь – клиент запускает приложение.
   2. Вводит данные для операции.
   3. Выбирает операцию.
   4. Получает ответ от сервера с результатом операции, либо с ошибкой с указанием причины/кода ошибки.
2. Корректность работы в сети Интернет без сбоев. Предусловия: в файле конфигурации указаны отличные от стандартных порт/адрес.
   1. Пользователь – клиент запускает приложение.
   2. Вводит данные для операции.
   3. Выбирает операцию.
   4. Получает ответ от сервера с результатом операции, либо с ошибкой с указанием причины/кода ошибки.

## Описание приложения

С помощью приложения есть возможность вызова арифметических операций на удалённом сервере с помощью вызова удалённых процедур и получением результата расчёта. Описание структуры сообщения находится в файле *calculator.proto*.

## Приёмочное тестирование

### Соответствие критерию «Веб-сервис принимает от приложения исходные данные и возвращает результат»

* Установим параметры приложений для сервера:

port=5300

И клиента:

grpc\_server\_ip=127.0.0.1

grpc\_server\_port=5300

* В целях проверки корректности выполнения критерия необходимо провести запуск сервера вызова удалённых процедур (рис. 1).

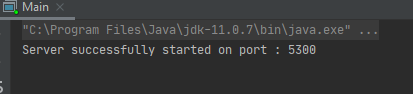


Рисунок 1

* Запуск клиента и вызов операции сложения. Передаваемые параметры: операнд 1 = 10.5423, операнд 2 = 23.2341, операция = SUM. Результат 33.7764 (рис. 2).

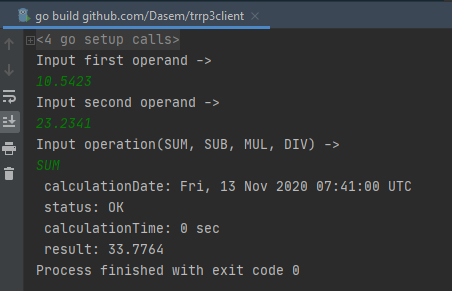


Рисунок 2

* Запуск клиента и вызов операции деления на ноль (рис.3).

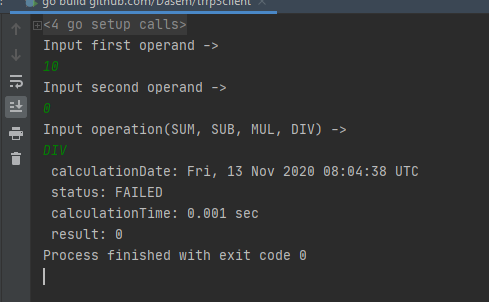


Рисунок 3

* Реакция сервера (рис. 4) на указанный на рисунке 3 вызов процедуры – ошибка (status: FAILED), так как происходит деление на 0. Таким образом демонстрируется получение исходных данных от клиента и возвращение результата.

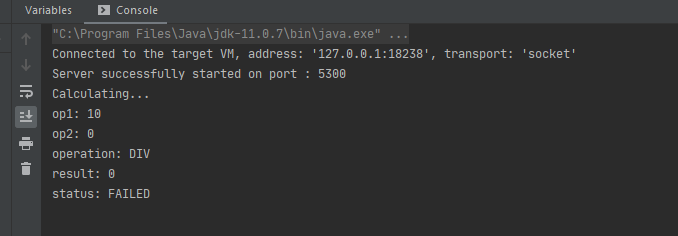


Рисунок 4

### Соответствие критерию «Веб-сервис и приложение, его использующее, написаны на разных языках программирования»

Использование различных языков программирования возможно проверить путём анализатора исходного кода на GitHub прикреплённых репозиториев (рис. 5, рис. 6).

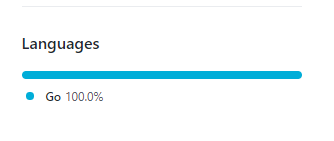


Рисунок 5 (https://github.com/Dasem/trrp3client)

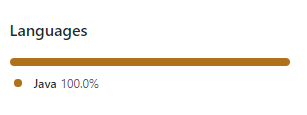


Рисунок 6 (https://github.com/Dasem/trrp3server)

### Соответствие критерию «В качестве параметров веб-сервис принимает объекты классов»

Как было сказано в описании приложения, приложение при передаче данных оперирует объектами классов, описанных в файле *calculator.proto*. Конкретные объекты (CalculateMessage, CalculatedMessage), использующиеся при передаче файлов продемонстрированы на рисунке 7



Рисунок 7

### Соответствие критерию «Приложение, написанное студентом, работает в сети Интернет без сбоев»

Изменим параметры приложения, использующиеся в пункте 1.4.1:

Сервер:

port=5301

Клиент:

grpc\_server\_ip= 192.168.0.115

grpc\_server\_port=5301

И запустим приложение по аналогу с пунктом 1.4.1 (рис.8):

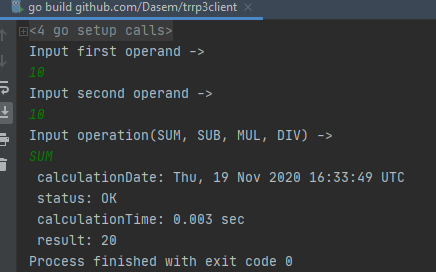


Рисунок 8

Поведение корректно с изменёнными данными сети.

# Заключение

В работы достигнута цель и выполнены следующие требования:

Приложение, написанное студентом, работает в сети Интернет без сбоев. 2 балла, так как имеются конфигурационные файлы на сторонах клиента и сервера.

Веб-сервис и приложение, его использующее, написаны на разных языках программирования. 1 балл (Golang, Java 11).

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

В качестве параметров веб-сервис принимает объекты классов. 2 балла. Объекты классов описаны в файле *calculator.proto,* реализации для конкретных языков автогенерируются.

Требования выполнены полностью (6/6 баллов), корректность выполненных операций доказана приёмочными тестами.

# Перечень используемых источников

1. Ссылка для скачивания. url: <https://www.jetbrains.com/go/> (дата обращения: 19.11.2020)
2. Ссылка для скачивания. url: <https://www.jetbrains.com/idea/> (дата обращения: 19.11.2020)
3. Лицензия. url: <https://github.com/grpc/grpc/blob/master/LICENSE> (дата обращения: 19.11.2020)
4. Ссылка для скачивания. url: <https://github.com/protocolbuffers/protobuf> (дата обращения: 19.11.2020)
5. Лицензия: url: <https://github.com/protocolbuffers/protobuf/blob/master/LICENSE> (дата обращения: 19.11.2020)
6. Лицензия. url: <https://www.jetbrains.com/student/license_educational.html> (дата обращения: 19.11.2020)
7. Глоссарий. url: <https://www.w3.org/TR/2004/NOTE-ws-gloss-20040211/#webservice> (дата обращения: 19.11.2020)
8. Спецификация. url: <https://www.omg.org/spec/IDL/4.2/> (дата обращения: 19.11.2020)