|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 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, кандидат ФМН  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.

**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 |

**Выбор предметной области**

Предметная область: обработка простейших арифметических вычислений на стороне сервера.

**Ход выполнения работы**

Приложение написано на языках Java 11 (серверная часть), Golang (клиентская часть). Среды разработки приложения: IntelliJ IDEA, GoLand[1].

В качестве фреймворка для вызова удалённых процедур был выбран gRPC[3]

В качестве протокола сериализации был выбран Protocol Buffers[5].

В качестве утилиты для генерации кода использовалась утилита *protoc*[4]

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

**Сценарии использования приложения**

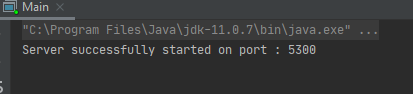
1. Выполнение умножения на удалённом сервере. Предусловия: у пользователей соответствующие адресы и порты в файлах «default.properties», также запущено серверное приложение.
   1. Пользователь – клиент запускает приложение
   2. Вводит данные для операции
   3. Выбирает операцию SUM
   4. Получает ответ от сервера с результатом операции, либо с ошибкой с указанием причины/кода ошибки
2. Выполнение умножения 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 SUB
   4. Получает ответ от сервера с результатом операции, либо с ошибкой с указанием причины/кода ошибки
3. Выполнение умножения 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 MUL
   4. Получает ответ от сервера с результатом операции, либо с ошибкой с указанием причины/кода ошибки
4. Выполнение умножения 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 DIV
   4. Получает ответ от сервера с результатом операции, либо с ошибкой с указанием причины/кода ошибки

**Описание приложения**

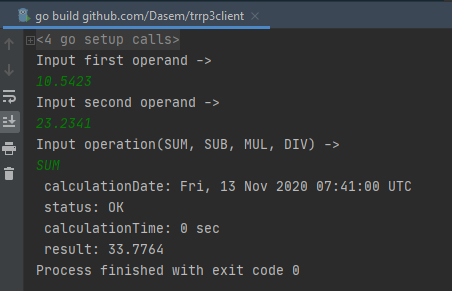
С помощью приложения есть возможность вызова арифметических операций на удалённом сервере с помощью вызова удалённых процедур и получением результата расчёта. Описание структуры сообщения находится в файле 

**Приёмочное тестирование**

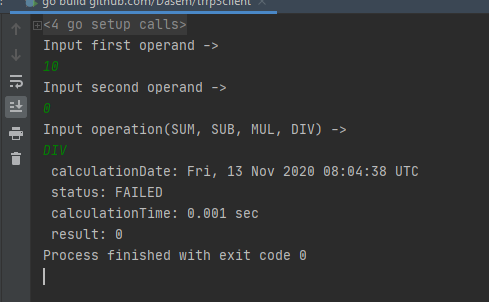
Запуск сервера вызова удалённых процедур



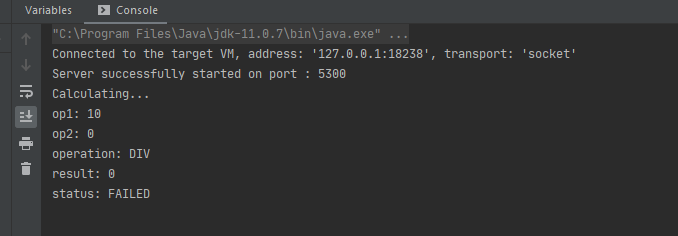
Запуск клиента и вызов операции сложения



Запуск клиента и вызов операции деления на ноль



Логи сервера



**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.

**Программное обеспечение, использовавшееся при выполнении работы, а также лицензии**

1. GoLand, ссылка для получения бесплатной лицензии студентами и преподавателями: <https://www.jetbrains.com/shop/eform/students>, лицензия: <https://www.jetbrains.com/company/privacy.html>
2. IntelliJ IDEA, ссылка для получения бесплатной лицензии студентами и преподавателями: <https://www.jetbrains.com/shop/eform/students>, лицензия: <https://www.jetbrains.com/company/privacy.html>
3. gRPC лицензия (Apache): <https://github.com/grpc/grpc/blob/master/LICENSE>
4. protoc скачать: <https://github.com/protocolbuffers/protobuf>, лицензия: <https://github.com/protocolbuffers/protobuf/blob/master/LICENSE>
5. ProtocolBuffers лицензия: <https://github.com/protocolbuffers/protobuf/blob/master/LICENSE>