Министерство образования Республики Беларусь

Учреждение образования

БЕЛОРУССКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ ИНФОРМАТИКИ

И РАДИОЭЛЕКТРОНИКИ

Распределенные информационные системы

Отчёт

по лабораторной работе №1

«Технология Java TM Remote Method Invocation

(Java RMI)»

Выполнила: студент гр. 714301

Макаренко А.Д

Проверил: Кунцевич А.А.

Минск 2020

*Цель:* освоить технологию RMI, выполнить удаленный вызов методов.

Вариант 4. Клиент удаленно вызывает метод сервера, который может быть запущен на удаленном компьютере. На клиенте вводится строка через пробелы. Сервер возвращает эту строку склеенной клиенту, который выводит ее на экран.

Ход работы

1. Удалённый интерфейс

*public* *interface* Hello *extends* *Remote* {

*String* sayHello() *throws* *RemoteException*;

*String* removeSpaces(*String* *string*) *throws* *RemoteException*;

}

1. Реализация сервера

*public* *class* Server *implements* *Hello* {

*public* Server() {}

*public* *String* sayHello() {

return "Hello, world!";

}

*public* *String* removeSpaces(*String* *string*) {

return string.replaceAll("\\s+","");

}

*public* *static* *void* main(*String* *args*[]) {

try {

*Server* obj = new Server();

*Hello* stub = (Hello) UnicastRemoteObject.exportObject(obj, 0);

*// Bind the remote object's stub in the registry*

*Registry* registry = LocateRegistry.getRegistry();

registry.bind("Hello", stub);

System.err.println("Server ready");

} catch (*Exception* *e*) {

System.err.println("Server exception: " + e.toString());

e.printStackTrace();

}

}

}

1. Реализация клиента

*public* *class* Client {

*private* Client() {}

*public* *static* *void* main(*String*[] *args*) {

*String* host = (args.length < 1) ? null : args[0];

try {

*Registry* registry = LocateRegistry.getRegistry(host);

*Hello* stub = (Hello) registry.lookup("Hello");

*// String response = stub.sayHello();*

*Scanner* in = new Scanner(System.in);

System.out.print("Input a string: ");

*String* string = in.nextLine();

in.close();

*String* response = stub.removeSpaces(string);

System.out.println("response: " + response);

} catch (*Exception* *e*) {

System.err.println("Client exception: " + e.toString());

e.printStackTrace();

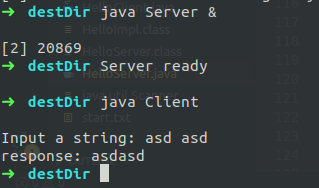
}

}

}

}

1. Компиляция и запуск; работа клиента



*Выводы:* В ходе данной работы получилось освоить технологию RMI, выполнить удаленный вызов методов.