|  |  |  |
| --- | --- | --- |
|  |  |  |
| МИНОБРНАУКИ РОССИИ | | |
| Федеральное государственное бюджетное образовательное учреждение  высшего образования  **«МИРЭА – Российский технологический университет»**  **РТУ МИРЭА** | | |

Институт Информационных технологий

Кафедра Инструментального и прикладного программного обеспечения

**ПРАКТИЧЕСКАЯ РАБОТА №2**

по дисциплине «Архитектура клиент-серверных приложений»

**Тема практической работы: Концепция удаленного вызова метода (RMI) в JAVA**

**Студент группы** ИКБО-10-19 Магин Константин Андреевич

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

(подпись студента)

**Руководитель практической работы** преподаватель Степанов В. С.

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

(подпись руководителя)

Работа представлена «\_\_\_»\_\_\_\_\_\_\_\_\_\_\_\_ 2021 г.

Допущен к работе «\_\_\_» \_\_\_\_\_\_\_\_\_\_\_\_ 2021 г.

Москва 2021

**Цель:**

Знакомство с механизмами удаленного вызова процедур в JAVA.

**Задание:**

Используя информацию из описания данной практической работы, необходимо реализовать удалённый метод решения квадратных уравнений общего вида ax2 + bx + c = 0. При этом, условие уравнения передавать на сервер, а клиентская часть должна получать результат в виде объектов пользовательского класса. Клиент и сервер должны работать на одном хосте.

**Ход работы:**

В ходе выполнения практической работы были созданы файлы для клиента и сервера, а также реализован логика работы программыс помощью имплементации интерфейса funcReturn.

**Исходный код программы:**

funcReturn.java

package com.company.pr2;  
  
import java.rmi.Remote;  
import java.rmi.RemoteException;  
  
public interface funcReturn extends Remote {  
 String printResult(double a, double b, double c) throws RemoteException;  
}

funcImpl.java

package com.company.pr2;  
  
import java.rmi.RemoteException;  
  
public class funcImpl implements funcReturn{  
 @Override  
 public String printResult(double a, double b, double c) throws RemoteException {  
 double D = b \* b - 4 \* a \* c;  
 if (D > 0) {  
 double x1, x2;  
 x1 = (-b - Math.*sqrt*(D)) / (2 \* a);  
 x2 = (-b + Math.*sqrt*(D)) / (2 \* a);  
 return "Корни уравнения: x1 = " + x1 + ", x2 = " + x2;  
 }  
 else if (D == 0) {  
 double x;  
 x = -b / (2 \* a);  
 return "Уравнение имеет единственный корень: x = " + x;  
 }  
 else {  
 return "Уравнение не имеет действительных корней!";  
 }  
 }  
}

Client.java

package com.company.pr2;  
  
import java.rmi.registry.LocateRegistry;  
import java.rmi.registry.Registry;  
  
public class Client {  
 private Client(){}  
 public static void main(String[] args){  
 try{  
 Registry registry = LocateRegistry.*getRegistry*(2732);  
  
 funcReturn stub = (funcReturn) registry.lookup("func");  
  
 System.*out*.println(stub.printResult(1, -8 ,15));  
 } catch (Exception e) {  
 e.printStackTrace();  
 }  
 }  
}

Server.java

package com.company.pr2;  
  
import java.rmi.registry.LocateRegistry;  
import java.rmi.registry.Registry;  
import java.rmi.server.UnicastRemoteObject;  
  
public class Server {  
 public Server(){}  
 public static void main(String[] args){  
 try{  
 funcImpl obj = new funcImpl();  
  
 funcReturn stub = (funcReturn) UnicastRemoteObject.*exportObject*(obj, 0);  
  
 Registry registry = LocateRegistry.*createRegistry*(2732);  
  
 registry.bind("func", obj);  
 System.*err*.println("Server ready");  
 } catch (Exception e) {  
 System.*err*.println("Server exception: " + e.toString());  
 e.printStackTrace();  
 }  
  
 }  
}

**Вывод программы:**

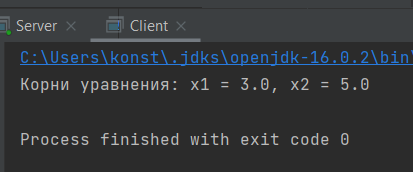


Рисунок 1 – запуск клиента. Получение решенного уравнения от сервера.

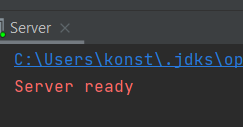


Рисунок 2 – готовность запущенного сервера

**Вывод:**

В результате выполнения практической работы были получены знания и навыки работы с механизмами удаленного вызова процедур в JAVA.