|  |  |  |
| --- | --- | --- |
|  |  |  |
| 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:** Многопоточные приложения, как основа

распределённых клиент-серверных вычислений

**Студент группы** ИКБО-01-19 Кузин Д.О.

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

(подпись студента)

**Руководитель практической работы**  Китанин С.С.

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

(подпись руководителя)

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

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

Москва 2021

# Задание на практическую работу

Использую 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

Содержимое файлов программы представлено на листингах 1-3=4.

Листинг 1. Содержимое файла CalculateService.java

public interface CalculateService extends Remote {  
 Map<String, String> calculate(List<Double> values) throws RemoteException;  
}

Листинг 2. Содержимое файла CalculateServiceImpl.java

public class CalculateServiceImpl implements CalculateService {  
  
 private final String x = "x";  
 private final String x1 = "x1";  
 private final String x2 = "x2";  
 private final String err = "error";  
  
 @Override  
 public Map<String, String> calculate(List<Double> values) throws RemoteException {  
 Map<String, String> resultMap = new HashMap<>();  
 double a = values.get(0);  
 double b = values.get(1);  
 double c = values.get(2);  
  
 double D = Math.*pow*(b, 2) - 4 \* a \* c;  
 if(D < 0) {  
 resultMap.put(err, "Дискриминант меньше нуля");  
 return resultMap;  
 }  
 else if(D == 0) {  
 resultMap.put(x, String.*valueOf*((-b / (2 \* a))));  
 return resultMap;  
 }  
 else {  
 resultMap.put(x1, String.*valueOf*((-b - Math.*sqrt*(D)) / (2 \* a)));  
 resultMap.put(x2, String.*valueOf*((-b + Math.*sqrt*(D)) / (2 \* a)));  
 return resultMap;  
 }  
 }  
  
}

Листинг 3. Содержимое файла ServerMain.java

public class ServerMain {  
 private static final String *UNIQUE\_BIND\_NAME* = "server.calculator";  
  
 public static void main(String[] args) throws RemoteException, AlreadyBoundException, InterruptedException {  
 CalculateService calculateService = new CalculateServiceImpl();  
  
 Registry registry = LocateRegistry.*createRegistry*(2727);  
  
 Remote stub = UnicastRemoteObject.*exportObject*(calculateService, 0);  
 registry.bind(*UNIQUE\_BIND\_NAME*, stub);  
 Thread.*sleep*(Integer.*MAX\_VALUE*);  
 }  
}

Листинг 4. Содержимое файла ClientMain.java

public class ClientMain {  
 private static final Scanner *scanner* = new Scanner(System.*in*);  
 private static final String *UNIQUE\_BIND\_NAME* = "server.calculator";  
  
 public static void main(String[] args) throws RemoteException, NotBoundException {  
 Registry registry = LocateRegistry.*getRegistry*(2727);  
 CalculateService calculateService = (CalculateService) registry.lookup(*UNIQUE\_BIND\_NAME*);  
  
 List<Double> values = new ArrayList<>();  
 System.*out*.println("Дано выражение ax^2 + bx + c = 0");  
 System.*out*.print("a = ");  
 values.add(*scanner*.nextDouble());  
 System.*out*.print("b = ");  
 values.add(*scanner*.nextDouble());  
 System.*out*.print("c = ");  
 values.add(*scanner*.nextDouble());  
  
 Map<String, String> calculationResult = calculateService.calculate(values);  
  
 for(Map.Entry<String, String> entry: calculationResult.entrySet()) {  
 System.*out*.println(entry.getKey() + " = " + entry.getValue());  
 }  
 }  
}

Результаты работы программы представлены на рисунке 1.

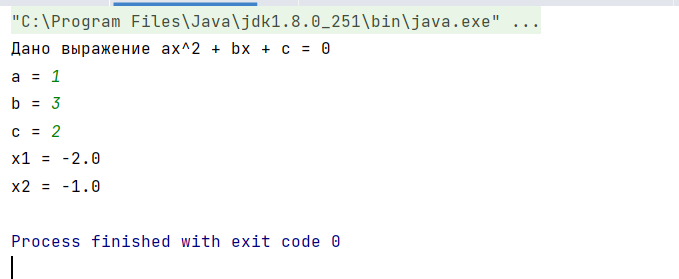


Рисунок – Результат работы программы

# Вывод

В результате выполнения работы были получены знания и практический опыт работы с механизмами удаленного вызова процедур в Java.