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

Институт информационных технологий

КАФЕДРА ИНСТРУМЕНТАЛЬНОГО И ПРИКЛАДНОГО
ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ (ИиППО)

Практическая работа №1
«Концепция удаленного вызова метода(RMI) в Java»

По дисциплине: «Архитектура клиент-серверных приложений»

Выполнил студент группы ИКБО-10-19 Дараган Ф.А.

Принял преподаватель Степанов П.В.

Практическая работы выполнена «___»_____2021 г.
(подпись студента)

«Зачтено» «___»_____2021 г.
(подпись руководителя)

Москва 2021

Оглавление

Практическая работа № 2 Концепция удаленного вызова метода(RMI) в Java

Цель работы

Ознакомится с системой удаленного вызова методов в языке 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.

Выполнение практической работы

На листингах 1, 2 показаны вспомогательные классы для программы.

Листинг 1. Класс Комплексного числа

```
package Pr2;

import java.io.Serializable;

public class Complex implements Serializable {
    private double re;
    private double im;

    Complex(double re, double im) {
        this.re = re;
        this.im = im;
    }

    Complex(double re) {
        this(re, 0);
    }

    @Override
    public String toString() {
        if (im == 0)
            return "" + re;
        else if (re == 0)
            return im + "i";
        else
```

```
            return re + (im > 0 ? " + " : " - ") + Math.abs(im) + "i"
;
    }
}
```

Листинг 2. Класс для хранения ответа

```
package Pr2;

import java.io.Serializable;

public class Answer implements Serializable {
    private Complex[] roots;

    Answer(Complex r1, Complex r2) {
        this.roots = new Complex[]{ r1, r2 };
    }

    Answer(Complex r) {
        this.roots = new Complex[]{ r };
    }

    Answer() {
        this.roots = new Complex[0];
    }

    @Override
    public String toString() {
        switch (roots.length) {
            case 0:
                return "No answer";
            case 1:
                return "Single answer: " + roots[0];
            case 2:
                return roots[0] + ", " + roots[1];
        }

        return "Something wrong";
    }
}
```

На листингах 3, 4 показан класс для решения уравнения и интерфейс класса для решения уравнения.

Листинг 3. Интерфейс класса для решения уравнения

```
package Pr2;

import java.rmi.Remote;
import java.rmi.RemoteException;
```

```
public interface Equation extends Remote {

    Answer solve(double a, double b, double c) throws RemoteException
;

}
```

Листинг 4. Класс для решения уравнения

```
package Pr2;

import java.rmi.RemoteException;

public class EquationImpl implements Equation {

    @Override
    public Answer solve(double a, double b, double c) throws RemoteEx
ception {

        double d = b * b - 4 * a * c;

        if (d > 0) {
            return new Answer(new Complex( (-b + Math.sqrt(d)) / (2 *
a) ),
                              new Complex( (-b - Math.sqrt(d)) / (2 *
a) ));
        } else if (d == 0) {
            return new Answer(new Complex(-b / (2 * a)));
        } else {
            return new Answer(new Complex( -b / (2 * a), Math.sqrt(-
d) / (2 * a)),
                              new Complex( -b / (2 * a), -Math.sqrt(-
d) / (2 * a)));
        }
    }

}
```

На листинге 5 представлен класс для сервера.

Листинг 5. Класс Сервера для удаленного вызова метода

```
package Pr2;

import java.rmi.AlreadyBoundException;
import java.rmi.Remote;
import java.rmi.RemoteException;
import java.rmi.registry.LocateRegistry;
import java.rmi.registry.Registry;
import java.rmi.server.UnicastRemoteObject;

public class Server {
```

```
    public static final String name = "serverPr2";
    public static final int port = 8000;

    public static void main(String[] args) throws RemoteException, Al
readyBoundException, InterruptedException {

        final Equation server = new EquationImpl();

        final Registry registry = LocateRegistry.createRegistry(Serve
r.port);

        Remote stub = UnicastRemoteObject.exportObject(server, 0);
        registry.bind(Server.name, stub);

        Thread.sleep(Integer.MAX_VALUE);

    }

}
```

На листинге 6 представлен класс клиента, в котором происходит обращение к серверу для вызова удаленного метода и получения результата.

Листинг 6. Класс Клиента для удаленного вызова метода

```
package Pr2;

import java.rmi.NotBoundException;
import java.rmi.RemoteException;
import java.rmi.registry.LocateRegistry;
import java.rmi.registry.Registry;

public class Client {
    public static void main(String[] args) throws RemoteException, No
tBoundException {
        final Registry registry = LocateRegistry.getRegistry(Server.p
ort);

        Equation calculator = (Equation) registry.lookup(Server.name)
;

        Answer answer = calculator.solve(1,2,5);
        System.out.println();
        System.out.println(answer.toString());
    }
}
```

Для работы приложения необходимо сначала запустить сервер и лишь потом клиент. Результат выполнения программы представлен на рисунках 1 и 2.

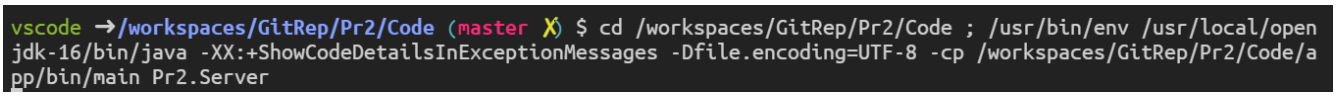

Рис. 1. Скриншот запуска сервера

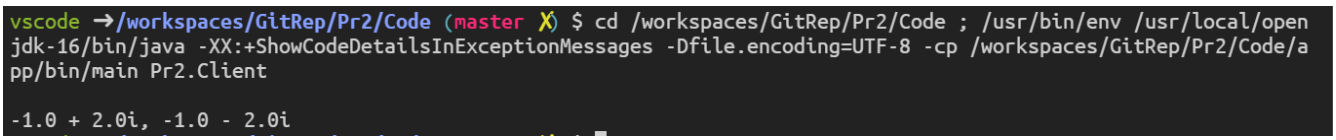

Рис. 2. Скриншот запуска клиента и результат выполнения

Выводы по работе

В ходе выполнения работы мы научились использовать систему удаленного вызова метода в Java, передающую пользовательский тип данных на клиент, в результате выполнения метода.

Используемая литература

1. Вязовик, Н. А. Программирование на Java : учебное пособие / Н. А. Вязовик. — 2-е изд. — Москва : ИНТУИТ, 2016. — 603 с. — Текст : электронный // Лань : электронно-библиотечная система. — URL: https://e.lanbook.com/book/100405 (дата обращения: 13.09.2021). — Режим доступа: для авториз. пользователей.
2. Наир, В. Предметно-ориентированное проектирование в Enterprise Java : руководство / В. Наир ; перевод с английского А. В. Снастина. — Москва : ДМК Пресс, 2020. — 306 с. — ISBN 978-5-97060-872-2. — Текст : электронный // Лань : электронно-библиотечная система. — URL: https://e.lanbook.com/book/179503 (дата обращения: 13.09.2021). — Режим доступа: для авториз. пользователей.
3. Васильев, А. Н. Самоучитель Java с примерами и программами : учебное пособие / А. Н. Васильев. — 4-е, изд. — Санкт-Петербург : Наука и Техника, 2017. — 368 с. — ISBN 978-5-94387-745-2. — Текст : электронный // Лань : электронно-библиотечная система. — URL: https://e.lanbook.com/book/101548 (дата обращения: 13.09.2021). — Режим доступа: для авториз. пользователей.