Чернівецький національний університет імені Юрія Федьковича

Навчально-науковий інститут фізико-технічних та комп’ютерних наук

Кафедра Математичних проблем управління і кібернетики

Лабораторна робота № 4

“Тема: СТВОРЕННЯ КАНАЛУ ЗВ'ЯЗКУ. RPC Client-Server ”

з дисципліни

“Програмування комп’ютерних мереж”

Виконав:

ст. гр. 541 Гришко Владислав Валерійович

Прийняв:

Доцент Лазорик В. В.

Чернівці – 2024

**Хід виконання роботи**

Код Серверу:

import java.io.File;  
import java.io.FileInputStream;  
import java.rmi.RemoteException;  
import java.rmi.registry.LocateRegistry;  
import java.rmi.registry.Registry;  
import java.rmi.server.UnicastRemoteObject;  
  
public class FileServer extends UnicastRemoteObject implements FileService {  
  
 private static final int PORT = 1099;  
 private static final String FILE\_NAME = " Звіт.docx";  
  
 protected FileServer() throws RemoteException {  
 super();  
 }  
  
 *@Override* public byte[] getFileChunk(String *fileName*, int *offset*, int *chunkSize*) throws RemoteException {  
 try (FileInputStream fis = new FileInputStream(new File(*fileName*))) {  
 fis.skip(*offset*);  
 byte[] buffer = new byte[*chunkSize*];  
 int bytesRead = fis.read(buffer, 0, *chunkSize*);  
 if (bytesRead < *chunkSize*) {  
 byte[] actualBytes = new byte[bytesRead];  
 System.**arraycopy**(buffer, 0, actualBytes, 0, bytesRead);  
 return actualBytes;  
 }  
 return buffer;  
 } catch (Exception *e*) {  
 System.err.println("Помилка читання файлу: " + *fileName*);  
 *e*.printStackTrace();  
 throw new RemoteException("Помилка читання файлу: " + *fileName*, *e*);  
 }  
 }

*@Override* public long getFileSize(String *fileName*) throws RemoteException {  
 File file = new File(*fileName*);  
 if (file.exists() && file.isFile()) {  
 return file.length();  
 } else {  
 System.err.println("Файл не знайдено: " + *fileName*);  
 throw new RemoteException("Файл не знайдено: " + *fileName*);  
 }  
 }  
  
 *@Override* public String getFileName() throws RemoteException {  
 System.out.println("Клієнт підключився до сервера.");  
 return FILE\_NAME;  
 }  
  
 public static void main(String[] *args*) {  
 try {  
 System.out.println("Запускаємо сервер...");  
  
 FileServer server = new FileServer();  
 Registry registry = LocateRegistry.**createRegistry**(PORT);  
 registry.rebind("FileService", server);  
  
 System.out.println("Сервер запущений на порту " + PORT);  
 System.out.println("Очікування підключення клієнтів...");  
  
 } catch (Exception *e*) {  
 System.err.println("Помилка запуску сервера:");  
 *e*.printStackTrace();  
 }  
 }  
}

Код клієнта:

import java.io.FileOutputStream;  
import java.rmi.registry.LocateRegistry;  
import java.rmi.registry.Registry;  
  
public class FileClient {  
 public static void main(String[] *args*) {  
 try {  
 System.out.println("Підключення до сервера...");  
 Registry registry = LocateRegistry.**getRegistry**("localhost", 1099);  
 FileService fileService = (FileService) registry.lookup("FileService");  
 System.out.println("Клієнт успішно підключився до сервера.");  
  
 String fileName = fileService.getFileName();  
 System.out.println("Отримано назву файлу: " + fileName);  
  
 long fileSize = fileService.getFileSize(fileName);  
 System.out.println("Розмір файлу: " + fileSize + " байтів");  
  
 try (FileOutputStream fos = new FileOutputStream(fileName)) {  
 int chunkSize = 1024;  
 for (int offset = 0; offset < fileSize; offset += chunkSize) {  
 byte[] chunk = fileService.getFileChunk(fileName, offset, chunkSize);  
 fos.write(chunk);  
 System.out.println("Отримано блок: " + chunk.length + " байтів");  
 }  
 }  
 System.out.println("Файл успішно завантажено як " + fileName);  
  
 } catch (Exception *e*) {  
 System.err.println("Помилка клієнта:");  
 *e*.printStackTrace();  
 }  
 }  
}

Результат:

Передача файлу Звіт.docx

1. Запускаємо сервер на порту 1099:

Зображення, що містить знімок екрана, текст, програмне забезпечення, Мультимедійне програмне забезпечення

Автоматично згенерований опис

1. Запускаємо клієнта:

Зображення, що містить знімок екрана, текст, Мультимедійне програмне забезпечення, програмне забезпечення

Автоматично згенерований опис

1. Клієнт підключається до сервера:

Клієнт:

Зображення, що містить текст, знімок екрана, Мультимедійне програмне забезпечення, програмне забезпечення

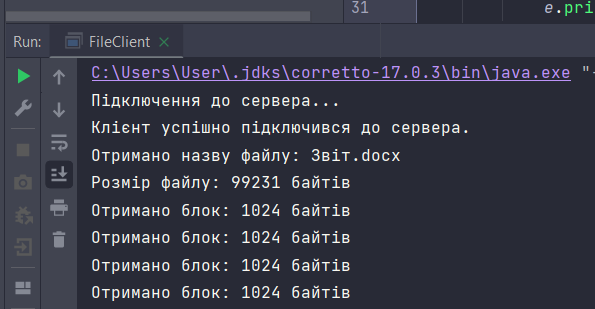
Автоматично згенерований опис

Сервер:

Зображення, що містить текст, знімок екрана, програмне забезпечення, Мультимедійне програмне забезпечення

Автоматично згенерований опис

1. Клієнт отримав назву файла і зберіг

Зображення, що містить текст, знімок екрана, програмне забезпечення, Мультимедійне програмне забезпечення

Автоматично згенерований опис