

ZÁPADOČESKÁ UNIVERZITA V PLZNI

**Fakulta aplikovaných věd**

**Katedra informatiky a výpočetní techniky**

# **KIV/PSI**

*Protokol RMI  
(aplikace pro server a klient)*

Autor: Antonín NEUMANN, A14N0139P

Akademický rok: 2014/2015

# 1 Zadání

Sestavte program pro vzdálené volání metody (RMI) v Javě, který bude jako parametry přenášet netriviální datové struktury. Vlastní úlohu pro výpočet si zvolte dle vlastního uvážení (např. pro zadané vrcholy trojúhelníka vypočtete jeho obvod, obsah a polohu těžiště).

## 2 Úvod

Pro řešení této semestrální práce jsem se rozhodl implementovat výpočet obvodu a obsahu zadaného trojúhelníka a dále server poskytuje metodu vracující aktuální datum a čas na serveru (tato sloužila hlavně předně pro testování vzdáleného volání metod).

## 3 Implementace

Implementace se dělá na serverovou část, která poskytuje metody pro vzdálené volání, a na klientskou část, která tyto metody volá.

Společnou část tvoří třída `Triangle.java` reprezentující objekt trojúhelníka, který je tvořen velikostí svých stran. Aby bylo možné trojúhelník přenášet jako parametr metod, musí tato třída implementovat rozhraní `java.io.Serializable` a `java.rmi.Remote`.

Dále je společné rozhraní popisující metody, které server poskytuje jako službu.

### 3.1 Server

Serveru implementuje rozhraní `IServerRmi`, které dědí od `java.rmi.Remote`, a sám dědí od `java.rmi.server.UnicastRemoteObject` což je základní předpoklad pro komunikaci pomocí protokolu RMI.

Server své služby musí pomocí `java.rmi.registry.Registry` zaregistrovat na určité adrese a portu, kde je později klientské programy naleznou a budou moci volat.

Server poskytuje tyto metody:

- `java.util.Date getDate()` – vrátí aktuální datum na serveru
- `double getObsah(Triangle t)` – ze zadaného trojúhelníku vypočítá jeho obsah
- `int getObvod(Triangle t)` – ze zadaného trojúhelníku vypočítá jeho obvod

### 3.2 Klient

Klient se pouze pomocí `java.rmi.registry.Registry` připojí na adresu, na které běží server a postupně ze serveru zavolá všechny metody, které poskytuje.

## 4 Uživatelská dokumentace

### 4.1 Překlad a spuštění na Windows

Pro přeložení zdrojových kódů je stačí spustit dávkový soubor `compile.cmd`.

Pro spuštění serveru po úspěšném překladu stačí spustit dávkový soubor `run_server.cmd`, tím se server spustí a začne naslouchat na zadané adrese a portu.

Pro spuštění klienta spustíme soubor `run_client.cmd`, klient se pokusí připojit na stejnou adresu a port na které by měl server již naslouchat. Pokud se mu to podaří zavolá se serveru vzdálené metody.

### 4.2 Překlad a spuštění na Linuxu

### 4.3 Ruční spuštění

Server i klient přijímají až 2 nepovinné parametry, první parametr je adresa (za lomítkem na localní adrese „localhost/PARAM\_ADDR“) a druhým je port. Parametry lze vynechávat zprava:

```
java Server rmi-server 8025
```

```
java ClientRmi rmi-server (pouzije se vychozi port 8025)
```

## 5 Závěr

Práce splňuje zadání a zjednodušeně ukazuje jak lze v jazyce Java pracovat se vzdáleným voláním metod, které mají jako parametr vlastní datový objekt (tedy netriviální datovou strukturu).

Při realizace reálného serveru poskytujícího vzdálené metody na protokolu RMI vhodné vytvořit rozhraní i pro datové objekty přenášené jako parametr. Tím by se zjednodušila správa a implementace těchto objektů v případě několika nezávislých klientů spravovaných někým jiným než správcem serverové aplikace.