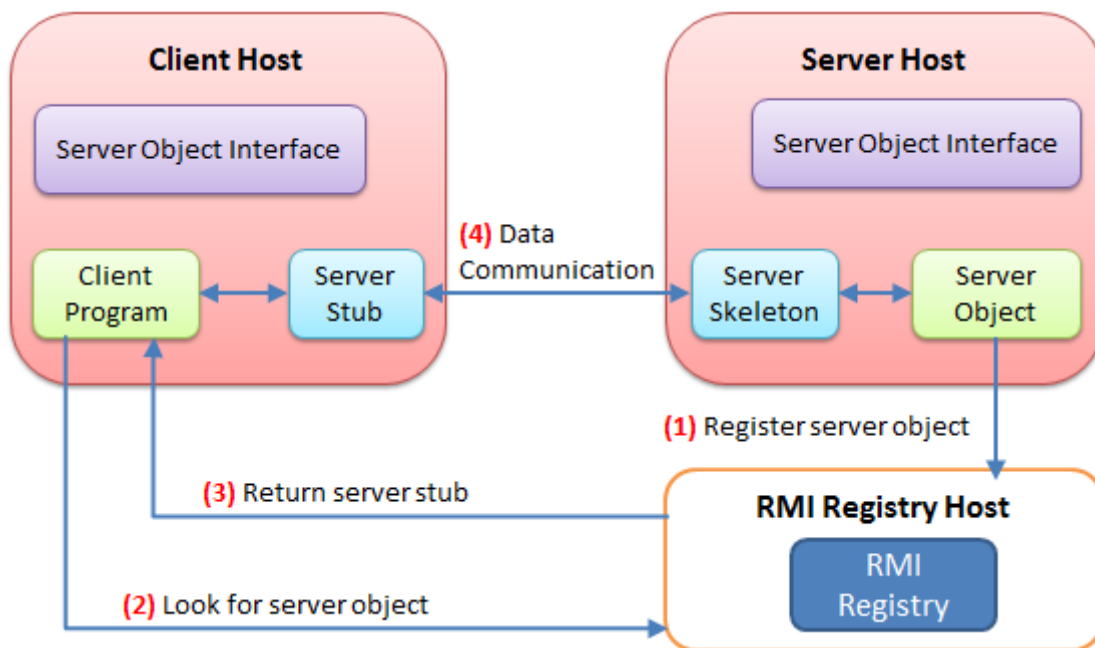


Aufgabe 1: RMI Theorie

1. Aufgabenstellung

Schau dir die Folien "RMI_tutorial" und das darin enthaltenen Codebeispiele genauer an und versuche zu verstehen wie RMI funktioniert. Zeichne eine Grafik, die alle wichtigen Schritte beschreibt und erkläre mit deinen eigenen Worten die Funktionsweise. Welche Rolle spielen dabei stub und skeleton?

Grafik



Funktionsweise

RMI steht für Remote Method Invocation und ist ein Prinzip um Methoden von einem Client aus auf einem Server auszuführen. Normalerweise würde dies über eine typische Client-Server Anwendung realisiert werden, dies ist aber relativ aufwändig zu implementieren. Eine RMI Anwendung vereinfacht den Vorgang und so ist es mit RMI möglich auf dem Client Methoden auszuführen, ohne direkt zu merken, dass die Methode eigentlich auf einem entfernten Server durchgeführt wird. Die Verbindung von Client und Server realisieren so genannte Stellvertreterobjekte. Diese existieren auf der Client- und auf der Serverseite. Die Stellvertretermethoden verdecken den tatsächlichen Übertragungsvorgang, sodass der Client demnach nicht mit einem eigenen Protokoll mit dem Server kommuniziert, sondern eine Methode aufruft, die so aussieht wie eine Servermethode. In Wirklichkeit nimmt die Stellvertretermethode nur die Parameter, verpackt sie in eine Serveranfrage und schickt sie weg.

RMI macht es also möglich, auf hohem Abstraktionsniveau zu arbeiten und entfernte Methodenaufrufe zu realisieren. Automatisch generierte Stellvertreter nehmen die Daten entgegen und übertragen sie zum Server. Nach der Antwort präsentiert der Stellvertreter das Ergebnis. Bei einem RMI Methodenaufruf ist das Zusammenspiel von Server, Registry (Namensdienst) und Client fundamental:

- Der **Server** stellt das entfernte Objekt mit der Methodenimplementierung bereit. Die Methode läuft im eigenen Adressraum, und der Server leitet eingehende Anfragen vom Netzwerk an diese Methode weiter.

- Der **Namensdienst** ist ein Anfragedienst, der Objekte und ihre Methoden mit einem eindeutigen Namen verbindet. Der Server meldet Objekte mit ihren Methoden bei diesem Namensdienst an.
- Der **Client** ist der Nutzer des Dienstes und ruft die Methode auf einem entfernten Objekt auf. Auch er hat einen eigenen Adressraum. Möchte der Client eine Methode nutzen, so fragt er beim Namensdienst an, um Zugriff zu bekommen.

Stub und Skeleton

RMI verwendet einen Standard Mechanismus um zwischen Remote Objekten zu kommunizieren: stubs und skeletons.

Ein **stub** befindet sich auf der Clientseite und verhält sich wie ein Repräsentant des Remote Objects, d.h. können alle Methoden des Remote Objects auf dem Client aufgerufen werden.

Ein **skeleton** befindet sich auf der Serverseite und ist dafür zuständig den Aufruf vom stub zum eigentlichen Remote Objekt weiter zu leiten und das Ergebnis wieder zum stub zu schicken.

2. Aufgabenstellung

Informiere dich anschließend im Internet was die VM-Parameter codebase, hostname, useCodebaseOnly und security.policy bewirken und bei welchen Szenario sie benötigt werden. Welche Vor- und Nachteile hat das verwenden eines SecurityManagers?

Erklärungen

codebase: Ein codebase kann als Quelle angesehen werden, von welcher man Klassen in eine Virtuelle Maschine laden kann.

hostname: Der Wert dieses Parameters ist die IP Adresse des Hosts.

useCodebaseOnly: Wenn true, dann ist das automatische laden von Klassen verboten außer von denen, welche in der codebase angeführt sind.

security.policy: Dieser Parameter spezifiziert die Policy Datei, welche Berechtigungen enthält um beispielsweise Remote Objekte zu laden usw.....

SecurityManager: Wird ein SecurityManager gesetzt, so überprüft er, ob für einen bestimmten Vorgang in einer RMI Anwendung (z.B. Remote Methode aufrufen) die Genehmigung erteilt worden ist. Ohne Genehmigung endet der Versuch den Vorgang auszuführen mit einer java.security.AccessControlException. Genehmigungen werden erteilt durch einen entsprechenden Eintrag in die Datei java.policy.

Vorteil: Nicht jeder Client kann Kontakt zum Server aufnehmen und dessen Dienste nutzen

Nachteil: Mehr Kommunikationsaufwand und relativ komplizierte Implementation.

3. Aufgabenstellung

Wie müssen RMI Anwendung ausgeführt werden?

RMI Anwendungen müssen ähnlich wie Client Server Anwendungen gestartet werden. Wichtig ist, dass der Server läuft und der Server auch eine Registry erzeugt hat. Der Client muss anschließend erzeugt werden.