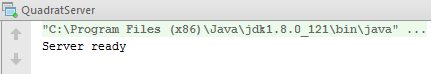
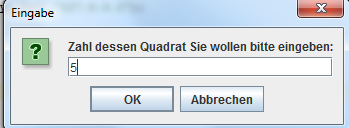
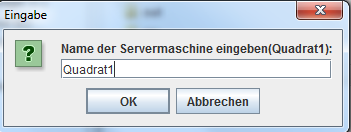
|  |  |
| --- | --- |
| **Package** | **Description** |
| [**java.rmi**](https://docs.oracle.com/javase/7/docs/api/java/rmi/class-use/Remote.html#java.rmi) | Provides the RMI package. |
| [**java.rmi.activation**](https://docs.oracle.com/javase/7/docs/api/java/rmi/class-use/Remote.html#java.rmi.activation) | Provides support for RMI Object Activation. |
| [**java.rmi.dgc**](https://docs.oracle.com/javase/7/docs/api/java/rmi/class-use/Remote.html#java.rmi.dgc) | Provides classes and interface for RMI distributed garbage-collection (DGC). |
| [**java.rmi.registry**](https://docs.oracle.com/javase/7/docs/api/java/rmi/class-use/Remote.html#java.rmi.registry) | Provides a class and two interfaces for the RMI registry. |
| [**java.rmi.server**](https://docs.oracle.com/javase/7/docs/api/java/rmi/class-use/Remote.html#java.rmi.server) | Provides classes and interfaces for supporting the server side of RMI. |
| [**javax.management.remote.rmi**](https://docs.oracle.com/javase/7/docs/api/java/rmi/class-use/Remote.html#javax.management.remote.rmi) | The RMI connector is a connector for the JMX Remote API that uses RMI to transmit client requests to a remote MBean server. |
| [**javax.rmi**](https://docs.oracle.com/javase/7/docs/api/java/rmi/class-use/Remote.html#javax.rmi) | Contains user APIs for RMI-IIOP. |
| [**javax.rmi.CORBA**](https://docs.oracle.com/javase/7/docs/api/java/rmi/class-use/Remote.html#javax.rmi.CORBA) | Contains portability APIs for RMI-IIOP. |
| [**org.omg.stub.java.rmi**](https://docs.oracle.com/javase/7/docs/api/java/rmi/class-use/Remote.html#org.omg.stub.java.rmi) | Contains RMI-IIOP Stubs for the Remote types that occur in the java.rmi package. |

Das java.rmi.Remote beschreibt die Funktionen, die auf dem Server zur Verfügung stehen, und definiert damit das Verhalten des entfernten Objekts, ohne dieses zu implementieren.

b) Wenn man eine generische Methode, die Tasks lösen kann, implementiert ist es möglich verschiedene Datentypen beim Methodenaufruf zu übergeben und auch diese als Rückgabewert zu erhalten. Eine generische Methode kann jeden Datentyp entgegennehmen falls keine Schranken gesetzt sind. Mit Schranken kann man die Eigenschaften des Datentyps vorschreiben. Z.B. Number and Comparable. Somit muss der übergebene Datentyp eine Nummer und vergleichbar sein. Dies ist ein sehr großer Vorteil denn somit muss man nicht mehr jeweils eigene Methoden, die das Gleich machen, für eigene Datentypen schreiben.

c) Falls richtig implementier und bei Verwendung verschiedener Ports ist es möglich auch mehrere Anfragen simultan durchzuführen. Mein Server ist derzeitig noch nicht in der Lage simultane Anfragen durchzuführen.

Dokumentation:

1. Server wird gestartet
2. Eingeben der Zahl für die Berechnung
3. Eingeben des Namen der Servermaschine
4. Ergebnis wir auf der Konsole ausgegeben