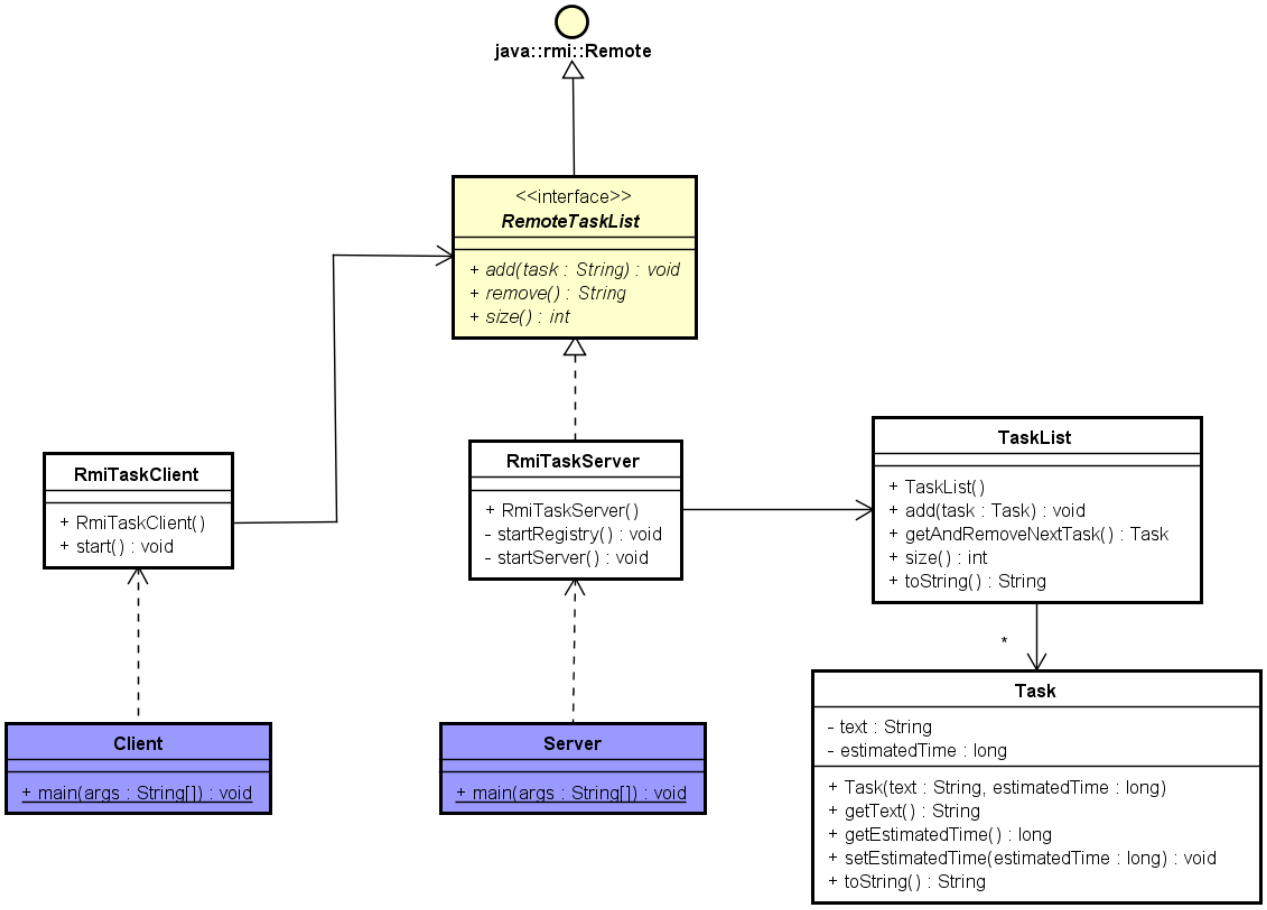
Exercise 08.01

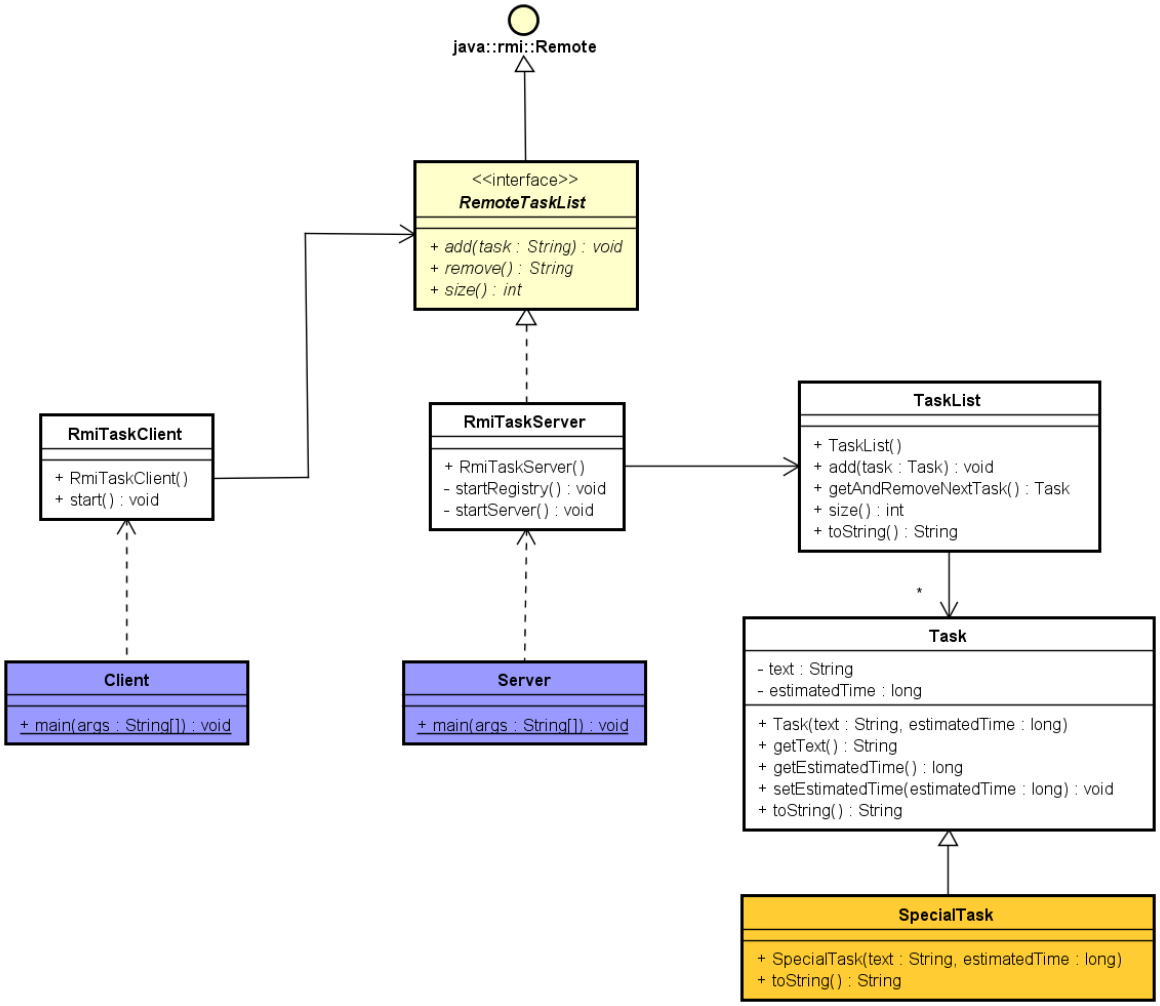
Implement an RMI system where the server has a task list (almost similar to the socket version from exercise 06.03) and clients can add a task, get the next (first) task and get the size of the task list. Consider how to use/send Task objects in RMI.



You have to divide into client and server part, i.e. two projects in Eclipse/IntelliJ (server and client part). The server part contains everything except RmiTaskClient and Client and the client part only contains Client, RmiTaskClient and RmiTaskList (the interface).

Exercise 08.02

The purpose for this exercise is use 1) dynamic download and 2) security managers and run it from the command prompt or a batch file.



Implement the system above (change the previous exercise) such that you

a) Divide classes into two projects in Eclipse/IntelliJ. The client project cannot have the file `SpecialTask` (this file is only known by the server until runtime). The code for this class is given below

b) Change the implementation to method remove in class `RmiTaskServer` such that you return a `SpecialTask` object (instead of a `Task` object). Do not change the interface.

c) Change the two classes Client and Server (the main method's) such that you define a security manager (see the code below)

d) Compile all classes (either from a command prompt or by running the main method from Eclipse and accepting that the program throws an exception and then terminate it)

e) Navigate to the bin folder (the folder with class files) for the server and in a command prompt or bat file execute this:

```
java -Djava.security.policy=all.policy Server
pause
```

with `all.policy` given as a text file with the content:

```
grant {
   permission java.security.AllPermission;
};
```

f) Navigate to the bin folder (the folder with class files) for the client, copy the file `all.policy` to this folder and in a command prompt or bat file execute this:

```
java -Djava.rmi.server.codebase=http://ict-engineering.dk/class/
     -Djava.security.policy=all.policy Client
pause
```

SpecialTask.java

```
public class SpecialTask extends Task
{
   public SpecialTask(String text, long estimatedTime)
   {
      super(text, estimatedTime);
   }

   public String toString()
   {
      return "SpecialTask: " + super.toString();
   }
}
```

Server.java

```
public class Server
{
   public static void main(String[] args) throws Exception
   {
      if (System.getSecurityManager() == null)
      {
         System.setSecurityManager(new SecurityManager());
      }
      RemoteTaskList server = new RmiTaskServer();
   }
}
```

Client.java

```
import java.rmi.RemoteException;
public class Client
{
   public static void main(String[] args) throws RemoteException
   {
      if (System.getSecurityManager() == null)
      {
         System.setSecurityManager(new SecurityManager());
      }
      RmiTaskClient client = new RmiTaskClient();
      client.start();
   }
}
```