

Gruppe Pawlowsky, Sochovsky

Server/Client Mastermind

VSDB 4BHITS 29.01.2012

Pawlowsky, Sochovsky

Inhalt

Aufgabenstellung.....	2
Start der Applikation	2
Designüberlegung.....	3
Aufwandsabschätzung.....	5
Arbeitsteilung	5
Arbeitsdurchführung	5
Resultate/Niederlagen	6
Test	6

Aufgabenstellung

MasterMind: Server-Client Applikation

Entwickeln Sie ein MasterMind-Spiel mit Hilfe von Java-Sockets als Server-Client Applikation. Das MasterMind-Spiel wird mit 6 Farben gespielt, wobei die Farben als Zahlen (1 bis 6) dargestellt werden. Zu erraten sind 4 Positionen. Am Beginn wird der Server mit den 4 Positionen initialisiert, d.h. die 4 Farben (=Zahlen, Konstruktor) werden für das Spiel festgelegt. Beispiel: 5331

Der Client stellt nun eine Verbindung zum Server her und der Benutzer kann über eine einfache Benutzereingabe seinen Tipp abgeben (Bsp: 1344). Der Tipp wird an den Server übertragen und dort ausgewertet. Das Resultat besteht aus den Zeichen S (schwarz) oder W (weiss). S (schwarz) für die richtige Farbe an der richtigen Position. W für die richtige Farbe an der falschen Position. Für die restlichen falschen Farben wird nichts übertragen. In diesem Beispiel wäre die Antwort: SW.

Startpunkt (TCPMMindCom.java, TCPMMindClient.java): [Link](#)

Abgabe:

- o jar-Datei (inklusive Quelldateien)
- o Abgabedokument beschreibt, wie man die Applikation startet (für Server und Client)
- o Demonstration der Applikation

Gruppenarbeit (2er Teams)

Benotungskriterien

- o 8 Punkte: Spiel ist nur teilweise implementiert, aber Kommunikation Client - Server - Client funktioniert
- o 12 Punkte: Spiel ist vollständig implementiert
- o 16 Punkte: Spiel ist vollständig implementiert und Client kann nach Spielende ein Spiel neu starten (dh. Server besetzt die Positionen neu)

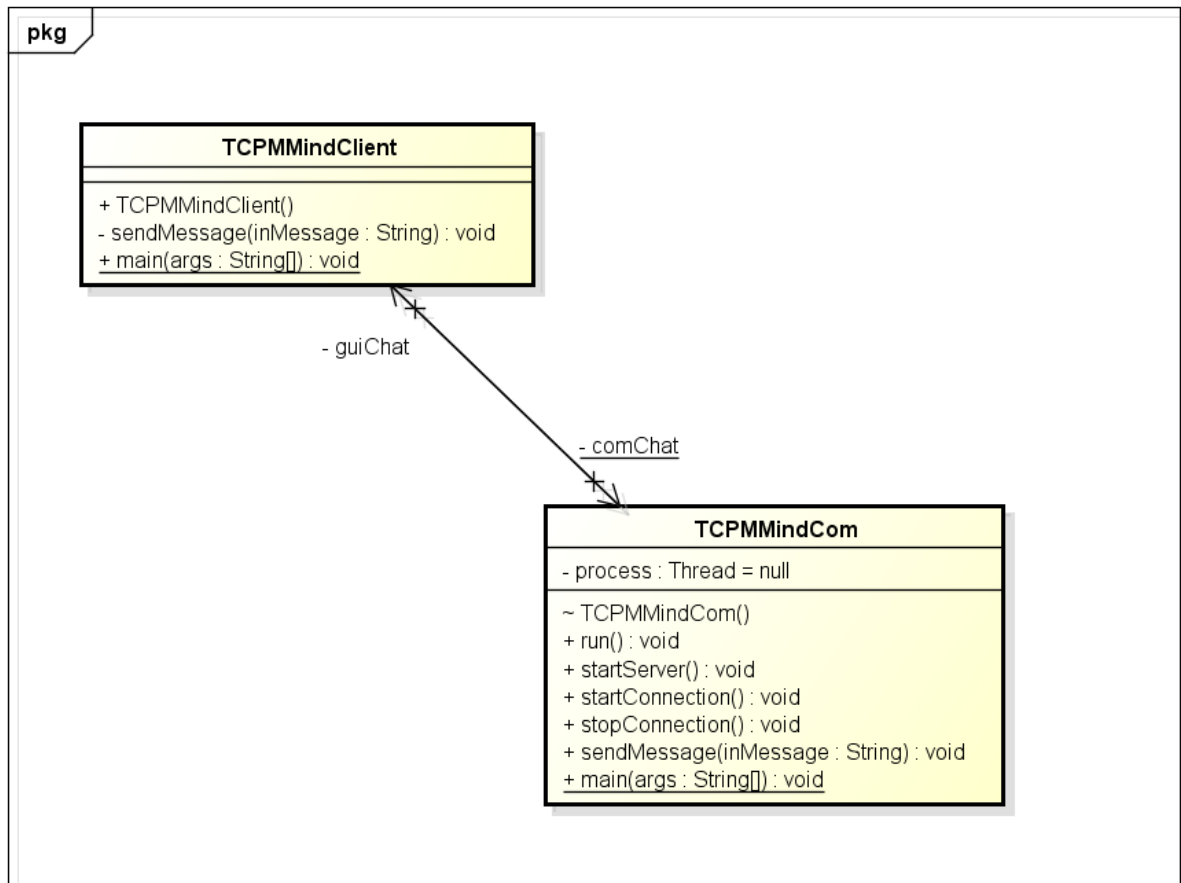
Start der Applikation

Um die Applikation zu starten, muss man zuerst den Server starten und einen freien Port 1234 haben. Hierbei werden noch keine Konsolenargumente benötigt. Anschließend muss im selben Netzwerk ein Client gestartet werden, bei diesem muss man also Konsolenargument die IP-Adresse des Servers übergeben. Der Client muss ebenfalls den Port 1234 frei haben.

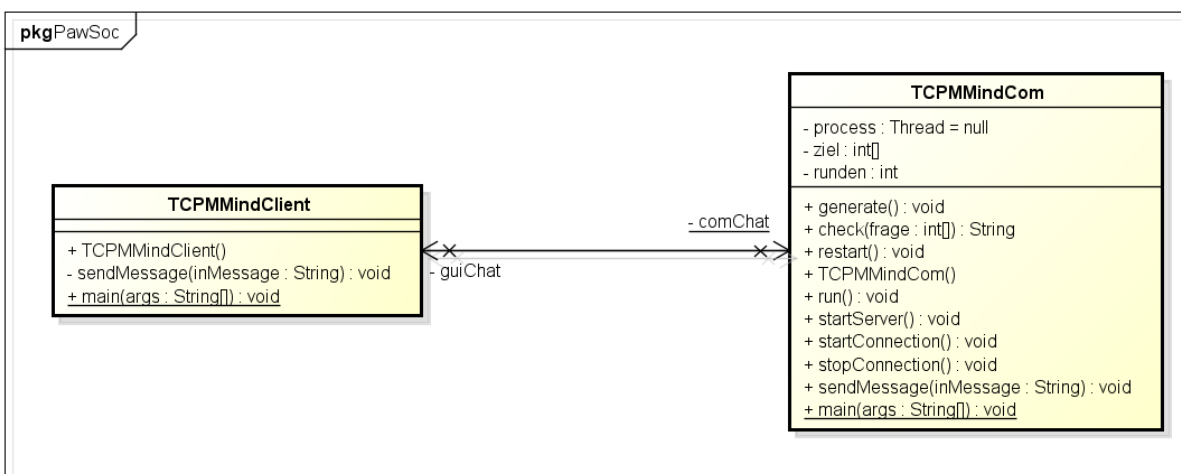
Dann ist die Applikation auch schon gestartet und man kann beginnen mit dem Client Nachrichten an den Server zu schicken und dieser wertet diese dann aus und gibt eine Antwort zurück.

Designüberlegung

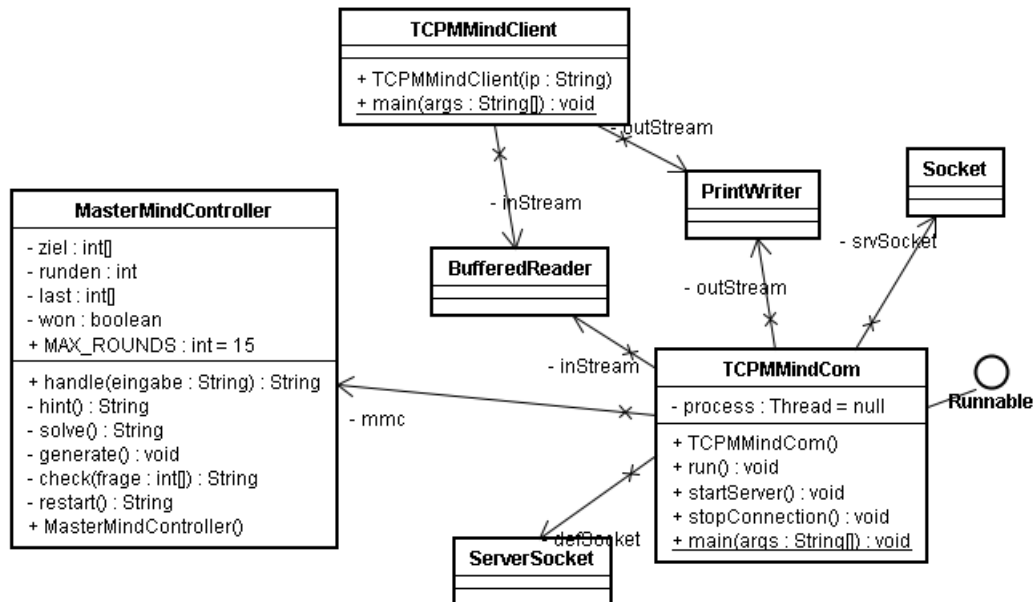
Vom Lehrer wurde eine bereits vorhandene Server/Client Kommunikation gegeben.
Diese hat folgendermaßen ausgesehen.



Mit dieser Designüberlegung haben wird dann versucht die Mastermind Logik einzubauen:



Jedoch war nach längeren Überlegungen die Trennung der Logik und des Servers einfacher und erweiterbarer:



Damit man die Server/Client Verbindung auch mit zukünftigen Aufgaben einsetzen kann haben wir uns entschlossen beim Verarbeiten der Eingabe vom Benutzer nur dann "" zurückzugeben, wenn etwas falsch gelaufen ist. Alle restlichen Rückgaben können vom Server direkt an den Benutzer zurückgegeben werden. Außerdem haben wir den Client und den Server innerhalb der Klassen komplett getrennt, sodass die Kommunikation nur noch über einen PrintWriter und einen BufferedReader und die dazugehörigen Sockets funktioniert.

Aufwandsabschätzung

Client/Server Verbindung mit nachfolgenden Eingaben implementiert: 3h

Mastermind Logik vollständig implementiert: 3h

Umgang mit Benutzereingaben + Synopsis: 2h

Tests: 2h

Neustarten implementiert: 0,5h

Arbeitsteilung

Pawlowsky:

Client/Server Verbindung mit nachfolgenden Eingaben implementiert

Umgang mit Benutzereingaben + Synopsis

Tests

Sochovsky:

Mastermind Logik vollständig implementiert

Umgang mit Benutzereingaben + Synopsis

Tests

Arbeitsdurchführung

Sochovsky:

Freitag 20.01.2012: Mastermind Logik teilweise implementiert 2h

Sonntag 22.01.2012: Mastermind Logik + Neustart 90% implementiert 2h

Sonntag 29.01.2012: Mastermind check Methode verbessert + Tests 3h

Gesamt: 7h

Pawlowsky:

Donnerstag 26.01.2012: Client-Server Kommunikations-Design komplett überarbeitet 3h

Sonntag 29.01.2012: Feinschliffe der Nachrichtenübergabe + Synopsis 4h

Gesamt: 7h

Resultate/Niederlagen

Sochovsky:

Check-Methode funktioniert nicht wenn mehrere Ziffern gleichzeitig richtige Position sind aber auch nicht: BSP Lösung 1122 check 1111 Ergebnis SSWW

Lösung:

Statt einem einfachen boolean Array das sinnbildlich einen Eintrag in Ziel sperrt, wird nun der Eintrag in frage und ziel gesperrt

Pawlowsky:

Anfangs war bei der Serverkommunikation der Client durch Methoden von einem Server abhängig und dadurch war eine Netzwerkfunktionalität nicht unabhängig gegeben.

Lösung:

Die Funktionalitäten von Server und Client komplett trennen wie oben in der Designüberlegung angegeben.

Test

Test der Funktionalität der check Methode

```
Willkommen zu Client-Server-Mastermind
Geben sie nun 4 Zahlen je zwischen 1 und 6 ein, um einen Tipp abzugeben
Mit 'restart' koennen sie den Server und somit das Spiel neu starten und
mit 'hint und 'solve' kann man sich einen Tipp oder die Loesung geben lassen.
Viel Spass beim Spielen!
1111
S
1122
S
1133
SSW
solve
Loesung: 1343
```

Wie man sieht wird die Eingabe nicht doppelt W und S gezählt sondern es stimmt eine 1 und eine 3 und zusätzlich ist ein 3er halbrichtig