**Design Patterns**

Daniel Dimitrijevic

Version1.0(13.02.2013)

Inhaltsverzeichniss:

Contents

[Aufgabenstellung 3](#_Toc350516609)

[**Abgabe** 3](#_Toc350516610)

[**Benotungskriterien** 3](#_Toc350516611)

[Ideen 5](#_Toc350516612)

[UML 5](#_Toc350516613)

[Schätzung 5](#_Toc350516614)

[Beispiel 5](#_Toc350516615)

[Problemfälle 5](#_Toc350516616)

[Quellen 6](#_Toc350516617)

Aufgabenstellung

Es soll basierend auf der [Vorlage](http://elearning.tgm.ac.at/mod/resource/view.php?id=11069) eine einfache GUI-Chat-Applikation implementiert werden. Beachten Sie dabei, dass das MVC-Pattern erfüllt werden muss. Das fertig gestellte Produkt soll als ausführbares Jar-File abgegeben werden. Beachten Sie dabei, dass entsprechende Parameter gesetzt werden müssen. Alle notwendigen Zusatzfiles (Sources, Libraries und Dokumentation) müssen ins Jar-File gepackt werden. Verwenden Sie Ant zum Deployen und Builden!

**Abgabe**

* Quellcode + Modellierung + JavaDoc/Kommentare
* Demonstration der Applikation + Protokoll

Gruppenarbeit (2er Teams)

**Benotungskriterien**

* 8 Punkte: Chatten von 2 Teilnehmern + 1 Zusatzfunktion + Ant build.xml  
  [Bsp. Benutzer kann seine Anzeigefarbe wählen]
* 16 Punkte: Chatten von mehreren Teilnehmern  
  [gleichzeitig, mit einer Server-Instanz]  
  [GUI in SWT implementieren]
* 24 Punkte: Chatten von mehreren Teilnehmern  
  [gleichzeitig, ohne eine Server-Instanz - keine Broadcast-Lösung!]

Exeptionhandling

MVC

Patterns

Parameter

JavaDoc!

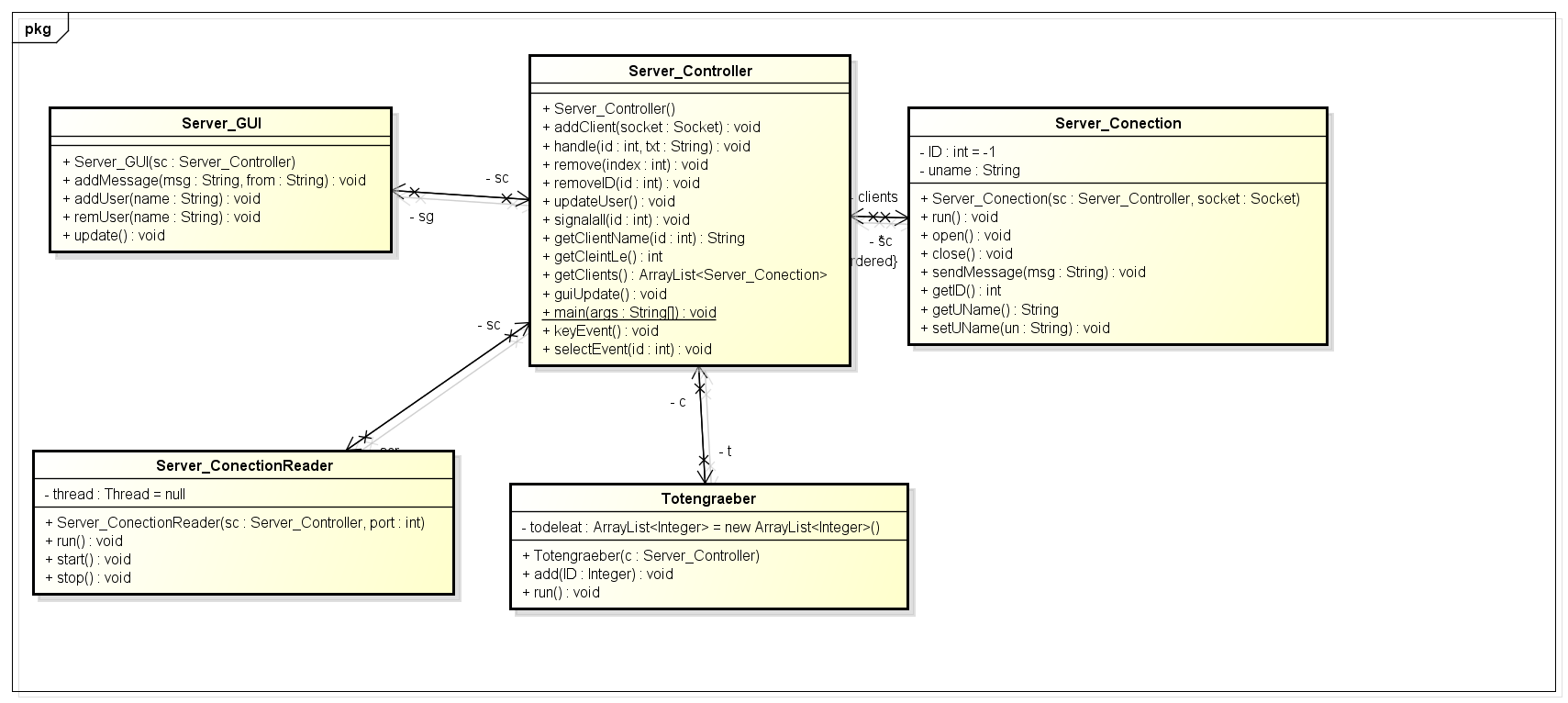
Sauberes Beenden

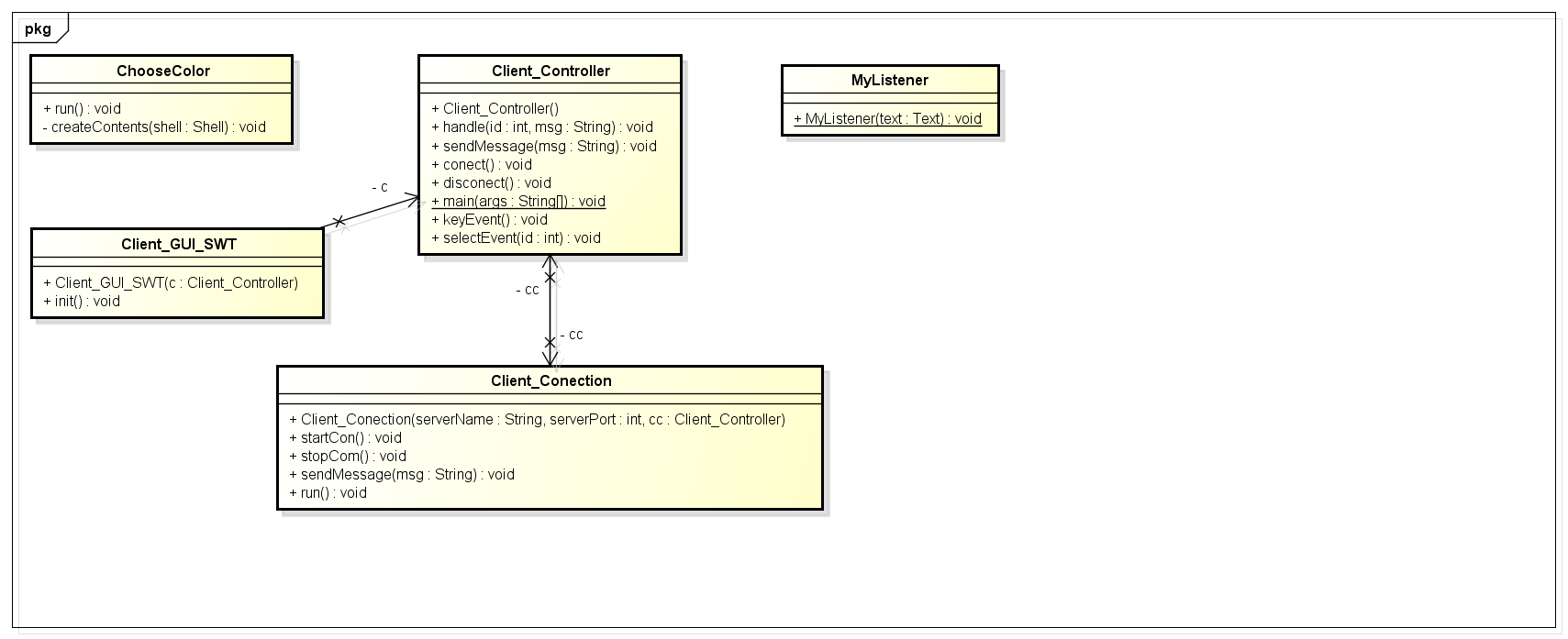
Ideen

Server verwaltet Verbindungen und schickt an alle Clients die Nachrichten

Client schickt die Nachricht an den Server und erhaltet die Nachricht selber.

UML

**Server**

**Client:**

Schätzung

Geschätzt Wirklich

Backhausen: 25h 20

Dimitrijevic: 25h 10

Beispiel

Problemfälle

Quellen

Java API

SWT API