SOFTWARE ENGINEERING
UNIVERSITÄT KASSEL

UNIKASSEL
VERSITÄT

Design Patterns SS 2015
Hausaufgabe 10

Die Aufgaben müssen von jedem Teilnehmer einzeln bearbeitet und abgegeben werden. Die Abgabe muss **bis spätestens Montag 29.06.2014 um 23:59 Uhr** über unser Hausaufgabenverwaltungssystem **https://se.cs.uni-kassel.de/hms/** erfolgen (nicht per Email). Die Abgabe ist nur als einzelne *.zip Datei möglich. Daher müssen alle für eine Abgabe relevanten Daten zu einem solchen Archiv kombiniert werden.

Hinweise zur Abgabe:

- Die Hausaufgabe ist in Form von **einem** exportierten Eclipse Projekten abzugeben. Mit Hilfe der Eclipse Export Funktion ist es möglich mehrere Projekte **in einer zip-Datei** zu exportieren. Ist ein Projekt nicht korrekt exportiert, kann es bei der Korrektur nicht berücksichtigt werden (es bietet sich also an, den Import des exportierten Projektes auszuprobieren).

  WICHTIG: Benennen Sie ihre Projekte für diese Abgabe nach folgendem Schema:

  DPSS15_HA<a>_<b>_<Matrikelnummer>,

  wobei <a> für die aktuelle Hausaufgabe und <b> für die Aufgabennummer steht.
  Beispiel für Aufgabe 1:
  DPSS15_HA3_1_12345678.
- Libraries müssen in einen lib oder im Projekt eingefügt werden oder als weitere Projekt mit dem Namen DPSS15_HA<a>_libs_<Matrikelnummer>. Es darf keine Abhängigkeit zu anderen Projekten bestehen oder zu externen Libraries.

Allgemeines:
Orientiert euch für die Lösung der Aufgaben an den zugehörigen Übungen und Vorlesungen:
http://seblog.cs.uni-kassel.de/category/ss15/designpatternss15/

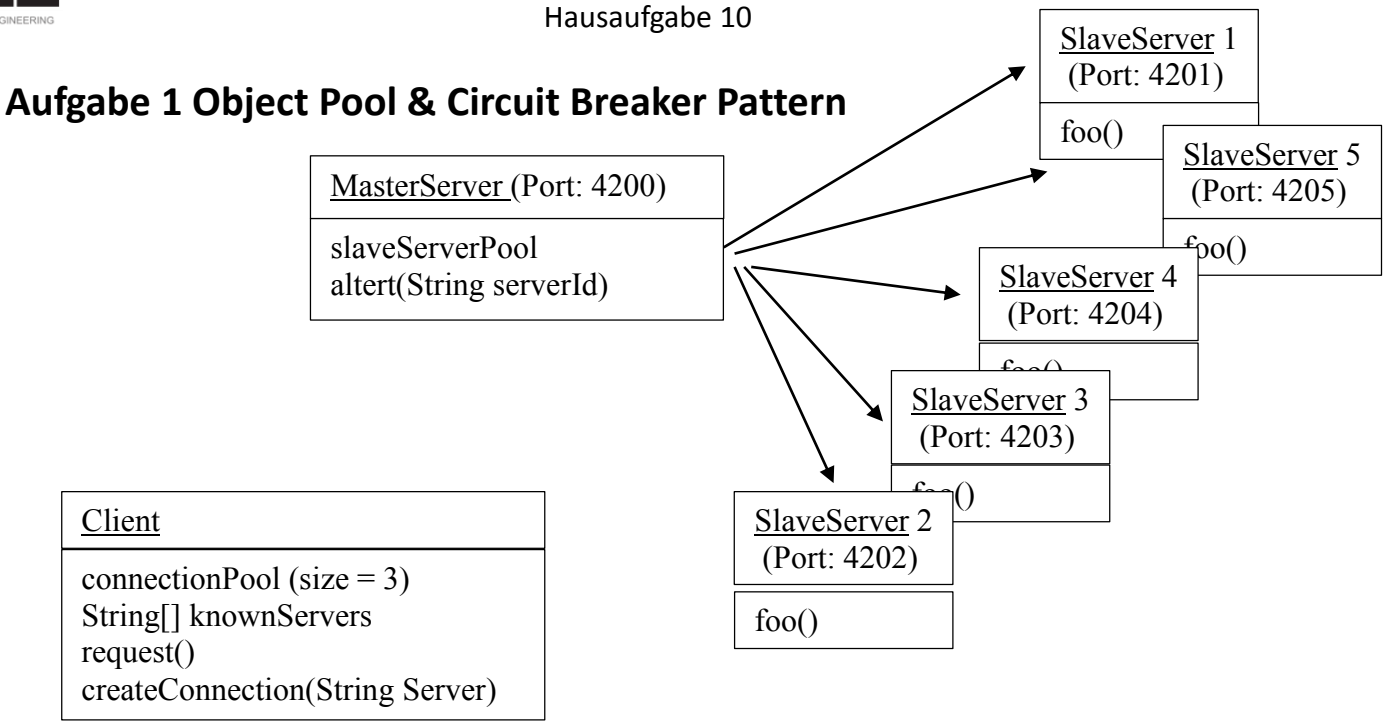
Zusammensetzung der Note:
Es wird n Hausaufgaben geben, von denen **n-2 Sinnvoll** bearbeitet werden müssen.
Sinnvoll bedeutet mindestens 50% der Punkte.
Am Ende ergibt sich die Note aus dem Durchschnitt n-2 besten Abgaben.
Zusätzlich wird es zwei Zusatzaufgaben geben mit denen fehlenden Abgaben ausgeglichen werden können.

Die Zusatzaufgaben werden deutlich umfangreicher als normale Hausaufgaben sein und dienen nicht zur Verbesserung der Note.
Zusatzaufgaben müssen mit **80%** bestanden werden, um wie folgt im Durchschnitt der n-2 besten Hausaufgaben berücksichtigt zu werden:
Wenn bei einer Zusatzaufgabe **100%** der Punkte erreicht wurden, wird sie wie eine **40%** Hausaufgabe bei der Berechnung berücksichtigt. Bei **90%** wird sie wie eine **30%** Aufgabe bewertet, bei **80%** wird sie wie eine **20%** Aufgabe bewertet.

Die Veranstaltung gilt als nicht bestanden, wenn mehr als **zwei Hausaufgaben nicht abgegeben** wurden oder der Mittelwert der zur Benotung herangezogenen Abgaben **weniger als 50%** beträgt.

Fragen bitte an hahn@uni-kassel.de

Aufgabe 1 Object Pool & Circuit Breaker Pattern



Baut mithilfe des Object Pools und des Circuit Breaker Pattern eine Client/Server Anwendung. Es soll einen MasterServer geben, welcher in einem SlaveServerPool 5 SlaveServer in je einem eigenen Thread startet und überwacht.

Ein SlaveServer öffnet je einen ServerSocket. Anfragen werden sofort mit einer Zufallszahl beantwortet. Mit einer Wahrscheinlichkeit von ~5% soll ein Thread.sleep(1000*60) aufgerufen werden. Der SlaveServer dann sozusagen überlastet.

Ein Client soll in einem ConnectionPool je drei Verbindungen zu SlaveServern aufrechterhalten. Wenn eine Verbindung unterbrochen wurde, soll sofort eine neue Verbindung zu einem anderen SlaveServer aufgebaut werden. Das aufbauen einer Verbindung über die createConnection-Methode soll mithilfe eines Thread.sleep(1000 * 5) künstlich verlangsamt werden und asynchron ausgeführt werden. (createConnection ist also teuer, weswegen wir den connectionPool brauchen)

Die request-Methode soll, mithilfe des Circuit Breaker Pattern, eine Anfrage an einen zufälligen SlaveServer aus dem ConnectionPool stellen. Sollte dieser nicht binnen 20ms reagieren, wird der Server aus dem ConnectionPool entfernt (die Verbindung geschlossen) und die altert-Methode des MasterServers (per Socket) aufgerufen. Die Anfrage soll dann wiederholt werden. Wenn es zurzeit keine Connections im Pool gibt (werden gerade neu aufgebaut), soll eine böse Fehlermeldung auf der Konsole erscheinen ☺

Die altert-Methode des MasterServers merkt sich die eingehenden ServerIds. Sollte ein SlaveServer 3 mal negativ auffallen, wird der entsprechende Thread beendet und ein neuer SlaveServer gestartet (have you tried turning it off and on again)

- Verwendet ordentliche Konsolen-Ausgaben um das System zu Testen.
- Testet die Anwendung indem ihr die Ausfallwahrscheinlichkeit eurer SlaveServer erhöht.

Fragen bitte an hahn@uni-kassel.de