

Die Aufgaben müssen von jedem Teilnehmer einzeln bearbeitet und abgegeben werden. Die Abgabe muss **bis spätestens Montag 25.05.2014 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 drei Projekte **in eine 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>__<Matrikelnummer>,</p></div><div data-bbox="167 372 829 389" data-label="Text">

wobei <a> für die aktuelle Hausaufgabe und 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 Proxy Pattern

Implementiert für Bomberman einen Netzwerkstack mithilfe des Proxy Pattern, welcher einen Multiplayermodus für zwei Spieler ermöglicht. Verwendet die GUI und das Spiel aus Hausaufgabe 5 (Abgabefrist wurde hier verlängert)

- Erweitert das Klassenmodell aus HA 5 (SDMLib, xcore, emf, UML Lab, ...) (das Datenmodell soll nicht per Hand geschrieben sein) um die folgende Klassen:
 - `abstract BombermanPlayer`
 - `LocalBombermanPlayer`
 - `RemoteBombermanPlayer`
 - `abstract Bomb`
 - `LocalBomb`
 - `RemoteBomb`
- Beim Starten des Spiels soll der eigene Port und die IP und der Port des gegnerischen Spielers angegeben werden. `BombermanGame.startGame(int localPort, String remoteIP, int remotePort)`
- In der Methode `BombermanGame.tick()` soll jede Sekunde die Anzeige des Spiels synchronisiert werden (mit dem `RemotePlayer`, `RemoteBombs`, usw.)
- Für das Verpacken/Auspacken von Objekten in/aus JSON kann SDMLib verwendet werden (muss aber nicht)
- Schreibt einen Test, welcher zwei Clients startet, die gegeneinander spielen
 - Verwendet sinnvolle Assertions