



Die Aufgaben müssen einzeln bearbeitet und abgegeben werden. Die Abgabe muss **bis spätestens Montag, den 27.05.2013 um 23:59 Uhr** über unser Hausaufgabenabgabesystem <http://seblog.cs.uni-kassel.de/dpss13/> erfolgen. Die Abgabe ist nur als einzelne *.zip oder *.jar-Datei möglich. Daher müssen alle für eine Abgabe relevanten Daten zu einem solchen Archiv kombiniert werden.

Hinweise zur Abgabe:

- Die Hausaufgabe als exportiertes Eclipse Projekt (*.zip, nicht den gesamten Workspace) abgeben. Das kann mit Hilfe der Eclipse Export Funktion durchgeführt werden. Ist das 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 ihr Projekt für diese Abgabe nach folgendem Schema:

DPSS13_HA<a>_<Matrikelnummer>,

wobei <a> für die aktuelle Hausaufgabe steht.

Beispiel:

DPSS13_HA5_12345678.

Allgemeines

Orientieren Sie sich für die Lösung der Aufgaben an den zugehörigen Übungen und Vorlesungen: <http://seblog.cs.uni-kassel.de/category/currentterm/design-patterns/>

Die Benotung beruht auf den Hausaufgaben.

Hierfür werden die gesamten Hausaufgaben minus zwei vom Studenten abgegebenen Hausaufgaben addiert, die mit mehr als 50 % bewertet wurden. Nicht abgegebene Hausaufgaben oder Betrugsversuche bekommen 0 Punkte.

Die Prüfung gilt als nicht bestanden, wenn alle Hausaufgabenpunkte weniger als **50 %** ergeben oder der Student mehr als **zwei Hausaufgaben nicht abgibt**.



Aufgabe 1 (Template-Pattern) (3P)

- Hierfür soll eine Telefonanlage mit verschiedenen Telefonarten nach dem Template-Pattern programmiert werden.
- Hierfür werden folgende Elemente benötigt: Phone, TelephoneSystem, DeskTelephone, MobilePhone
- Ein Telefon soll mindestens folgende Methoden enthalten: pickUp, isOn, isBusy und dial mit der Rufnummer als Parameter
- Die Telefonanlage soll folgende Methode enthalten, die die zu wählende Rufnummer in einer internen Variable speichert: call(String value)
- Der Ablauf der Methode dial ist folgender:
 - isOn,
 - isBusy,
 - pickUp,
 - call
- Das System soll in einem Test, anhand der in der Telefonanlage gespeicherten Rufnummer, verifiziert werden.



Aufgabe 2 (Decorator-Pattern) (3P)

- In dieser Aufgabe soll ein Feuerwehrsysteem entwickelt werden
- Hierfür werden folgende Elemente benötigt: Action, Extinguish, RequestBackup, Protect, FireFighterTeam und MissionReport
- Die Aktionen sollen folgende Funktionalität haben: doAction(MissionReport report), welche im übergebenen MissionReport alle Aktionen als Text speichert
- Die Aktion Verstärkung holen (RequestBackup) ist eine Erweiterung von Absichern (Protect). Es wird erst die Verstärkung gerufen und weiter abgesichert
- Das System soll in einem Test, in dem alle Situationen vorkommen, verifiziert werden



Aufgabe 3 (Proxy-Pattern) (6P)

- Es soll das Spiel „Settlers of Catan“ nach dem Proxy-Pattern umgesetzt werden
- Hierfür werden zwei Applikationen benötigt
 - Ein Server, der z.B. auf dem Port 13337 hört
 - Ein Client, der sich bei Bedarf mit dem Server verbinden kann
- Weiterhin werden folgende Elemente benötigt: Game, Map, Road, Settlement, Player
- Der Client soll eine Textausgabe haben, welche das aktuelle Spielfeld anzeigt
- Das Pattern soll in Map für das Erzeugen von Road und Settlement realisiert werden:
 - Die Klasse Map beinhaltet zwei Methoden: createRoad() und createSettlement()
 - Die create-Methoden schicken eine Nachricht per TCP/IP an den Server, dieser erzeugt daraufhin das jeweilige Objekt und hängt es an seine Game-Instanz. Nachfolgend schickt es der Server als JSON zurück an den Client, welcher das Objekt dann an den Player hängt
- Für das Verpacken/Auspacken von Objekten in/aus JSON kann SDMLib verwendet werden
- Das System soll in einem Modelltest verifiziert werden



Aufgabe 4 (Adapter-Pattern) (3P)

- Setzt eine eigene Idee nach dem Adapter-Pattern um
- Es dürfen keine Beispiele aus dieser Veranstaltung oder von Quellen, wie z.B. Wikipedia verwendet werden
- Das System soll anhand eines sinnvollen Tests verifiziert werden