

Die Aufgaben können (müssen aber nicht) in Gruppen von bis zu drei Leuten bearbeitet werden. Abgabe bis **spätestens Mittwoch 28.04.2010** per Mail mit dem Betreff **PMSS2010 HA2 <Matrikelnummer>** (z. B. „PMSS2010 HA2 12345678“) an **pm@cs.uni-kassel.de**. Werden die Aufgaben in einer Gruppe bearbeitet, genügt **eine** Abgabe per Mail mit den Namen aller Gruppenmitglieder. Diese Hausaufgabe gibt 10 + 2 Punkte.

Als Vorbereitung laden Sie die Eclipse Java Entwicklungsumgebung herunter und richten sie ein (wie im Vorlesungsvideo gezeigt). Machen Sie sich mit der Umgebung und ihrer Bedienung vertraut.

Aufgabe 1 - Fragenteil (5P)

1. Beschreiben Sie das Test-First-Vorgehen. Welche Vorteile bietet es? (3P)
2. Generieren Sie in Eclipse Getter- und Setter-Methoden für folgende Klasse (braucht nicht abgegeben zu werden):

```
class Student {  
    String name;  
    int matrikelnr;  
}
```

Beschreiben die den Vorteil von Getter- und Setter-Zugriffsmethoden für Membervariablen und Felder von Klassen an diesem Beispiel! (2P)

Aufgabe 2 - Implementierungsteil (5P + 2P)

Es ist ein Geldautomat zu programmieren, welcher mit einem Bankkonto zusammenarbeitet. Für die Implementierung des Kontos kann (muss aber nicht) die vorgestellte Version aus der Übung als Ausgangspunkt benutzt werden. Der Kunde initiiert eine Abhebung durch Angabe der PIN und des gewünschten Betrages. Der Geldautomat prüft sowohl die PIN, als auch den Kontostand des Kunden sowie den Bargeldbestand im Automaten. Sind diese Prüfungen erfolgreich, wird das Geld ausbezahlt und das Konto entsprechend belastet.

1. Geldautomat Implementieren und Testen (5P)

Implementieren Sie den Geldautomat und das Bankkonto sowie dazugehörige JUnit-Tests! Schreiben Sie mindestens vier verschiedene Tests (Erfolgreiche Auszahlung und Fehlerfälle)!

2. Zusatzaufgabe (2P)

Berücksichtigen Sie die im Automat vorhandenen Scheingrößen: Sind z.B. nur 20€-Scheine und größer im Automat vorhanden, ist das Abheben von z.B. 30€ nicht möglich. Implementieren Sie diese Funktionalität mit einem zugehörigen Test!

Hinweise: Ihr Programm braucht nur ein Konto (ein Kontostand, eine PIN) verwalten können! Zum Abheben reicht ein Methodenaufruf (Parameter: pin, betrag)! Verwenden sie für Fehlerfälle Exceptions! Trennen Sie den JUnit-Test von der restlichen Implementierung (separate Klasse)!

Abzugeben ist das aus Eclipse exportierte Projekt (als Email-Anhang, wie in der Übung gezeigt)!