

# Übungsblatt 4

Abgabe: 04.05.2009

---

## Aufgabe 1 Eine unheimliche Begegnung

Anfang der 60er Jahre erzählte ein US-amerikanisches Ehepaar über ihre Entführung durch Außerirdische nachdem sie ein Unidentifiziertes Flug-Objekt (UFO) am Himmel gesichtet hatten. Mittlerweile behaupten Millionen von Menschen Opfer von außerirdischen Entführungen gewesen zu sein.

Bei einem Spaziergang im Wald entdeckte ein Schüler einen kuriosen Gegenstand. Nach ein paar Untersuchungen wurde er auf einen kleinen Schriftzug *Ujo.class* auf dem Gegenstand aufmerksam und entschied sich den Gegenstand aufzubewahren. Im Verlauf der Jahre bekam der Schüler sein Abitur und ging zur Uni, um dort Informatik zu studieren.

Im Sommersemester 2009 besucht unser Student die Vorlesung “Praktische Informatik 2” und ihm wird klar, dass er mithilfe der **Reflection-API** seinen Gegenstand (*Java-Objekt*) in einer Java-Laufzeitumgebung genauer untersuchen kann.

Er entwirft daher eine Klasse **Spion.java**, die zusammen mit dem *Ujo* in eine Java-Laufzeitumgebung (JVM) eingesperrt werden soll. Sein Spion besteht lediglich aus einer **Main**-Methode und benutzt die Funktionalität aus dem Package **java.lang.reflect**. Von der JVM aus soll der Spion Auskünfte über das Unbekannte Java-Objekt (UJO) liefern. Die *Ujo.class*-Datei liegt auf der PI 2 Webseite “<http://www.informatik.uni-bremen.de/agbs/lehre/ss09/pi2/#zettel>” zum Herunterladen.

### Aufgabe 1.1 Woraus besteht das *Ujo* ? (40 %)

Der Spion soll mit dem *Ujo*-Objekt durch den Befehl “`java Spion Ujo`” aktiviert werden und soll herausfinden von welcher Basisklasse *superclass* die *Ujo*-Klasse abgeleitet ist.

Zusätzlich soll der Spion Informationen über die Beschaffenheit des *Ujo* preisgeben. Dies umschließt alle *Feldvariablen*, *Konstruktoren* und *Methoden* der **eigenen** *Ujo*-Klasse. Dabei sollen für jede der *Feldvariablen* die Modifikatoren (*public*, *private*, *protected*, *final*, *static*, ...), der Typ und Name mithilfe der Ausgabe-Routine `System.out.println()` ausgegeben werden.

Bei *Konstruktoren* und *Methoden* sollen zusätzlich zu Modifikatoren, ggf. der Rückgabentyp (*int*, *String*, *boolean*, ...), der Name und ggf. Typen der formalen Parameter jeweils ausgegeben werden. Der Name der *Konstruktoren* kann dabei weggelassen werden.

Falls ein *Konstruktor* oder eine *Methode* eine Ausnahme (*Exception*) auswirft, soll der entsprechende Name der *Exception* und die Liste der dazugehörigen Ausnahmehmethoden in einer übersichtlichen Art mit ausgegeben werden.

Hinweise:

Benutzt dafür die Methoden **getDeclaredFields**, **getDeclaredConstructors** und **getDeclaredMethods** aus dem *java.lang.reflect*-Package. Ausnahmenmethoden können einfach mit **toString()** ausgegeben werden. Der Name der Modifikatoren kann mit der Klassenmethode *toString()* der **Modifier**-Klasse folgendermaßen aufgerufen werden: **Modifier.toString( ... )**.

**Aufgabe 1.2 Wie verhält es sich ? (60 %)**

Um das Verhalten des *Ujo* genauer beobachten zu können, werden unterschiedliche *Ujo*-Objekte mithilfe jedes einzelnen Konstruktors erzeugt und daraufhin alle möglichen Methoden aufgerufen. Die statischen Methoden werden dabei nur einmal aufgerufen.

Falls der aufgerufene Konstruktor oder die Methode eine Exception auswirft, sollt ihr diese innerhalb einer *try-catch*-Anweisung auffangen, eine angepasste Fehlermeldung und die verfügbare Ausnahme mithilfe der Methode **getTargetException()** ausgeben.

Gegebenenfalls soll der Rückgabewert jedes einzelnen Methodenaufrufs mithilfe der Methode **toString()** ausgegeben werden.

Hinweise:

Die Klassen **Class**, **Method**, **Field** und **Modifier** werfen allerlei Ausnahmen aus, die aufgefangen werden müssen. Gebt zu jeder Ausnahme eine passende Fehlermeldung und den verfügbaren Text mit **getMessage()** aus. Diese Klassen sind in der Java-API-Dokumentation "<http://java.sun.com/javase/6/docs/api/>" beschrieben. Zur Vereinfachung sollt ihr annehmen, dass das *Ujo* keinerlei Arrays enthält und dass es sich dabei um ein konkretes Java-Objekt handelt.