Class Loader

- Delegationsmodell
- Es gibt 3 Stueck, sind fuer verschiedene Bereiche zustaendig
  1. Bootstrap - Standard JDK Klassen, im jre/lib/rt.jar (Java.lang), Kernkomponenten, zB : String.classClassLoader
  2. Extensions Class Loader jre/lib/ext
  3. System Class Loader: Eigene Klassen, im Class Path
- Jedes Class Loader ist ein Objekt

- Class Loader erstellt von jeder Klasse ein Klassenobjekt

1. Bootstrap:
   ArrayList.class.getClassLoader(). Gibt den zurueck, den die Klasse geladen hat.
2. SCL: Katze.class.getClassLoader(). Class Loader fragt zuerst Eltern ClassLoader ob Klasse geladen wurde

- Klasse identifiziert sich nicht nur durch vollen Namen, sondern auch durch den Class Loader, der ihn geladen hat. Gleiche waeren dann aber inkompatibel

Beispiel:

```
MyClassLoader extends ClassLoader
public Class <?> findClass(String qualifiedClassName, byte[] byteCodeDerKlasse) {
return defineClass(qualifiedClassName, byteCode, 0, byteCode.length);
}
Katze katze = new Katze
Wird vom SCL geladen
-> Bytecode der Katze holen. Dafuer extra einlesen, geht sonst nicht

BufferedInputStream bis = new BufferedInputStream(new
FileInputStream{"bin/exkurs_ClassLoader/Katze.class"));
ByteArrayOutputStream baos = new ByteArrayOutputStream();
(Kompilierte Klassen sind im Bin Verzeichnis)
(Mit OutputStream schreiben wir ins Array)

int zahl;
while((zahl = bis.read()) != -1) {
baos.write(zahl);
}
byte[] bytecode = baos.toByteArray();
Sysoutprint(Arrays.toString(bytecode));

MyClassLoader myClassLoader = new MyClassLoader();
Damit lade ich das Class Objekt nochmal
Class<?> katzeClass = MyClassLoader.findClass("exkurs_ClassLoader.Katze", bytecode)
(Voller Name der Katzenclasse)

Sysoutprint("Zweiter ClassLoader von Katze " + katzeClass.getClassLoader());
--> Nun Klasse zum zweiten Mal von MyClassLoader geladen

Object katze2 = katzeClass.newInstance();
try {
katze = (Katze) katze2;
} catch(Class Cast Exception) {
Sysout.err(e.getMessage());

---> AAAAABER selbst ClassLoader erstellen und Klasse laden laut Dozent keinen Sinn
```

Wann Klassen geladen werden

- Nicht: Bei Deklaration, null
- Sondern: Wenn benutzt, also eine Instanz erstellt wird, oder auf ein statisches Feld oder Block zugegriffen wird.
  Und evtl. vorhandene Elternklasse wird dann zuerst geladen!!!
- In dem Moment wo Klasse geladen werden, werden alle statischen Felder und Bloecke initialisiert.
  Auch Konstanten, da diese statisch sind.
- Nur Interfaces, die von der Klasse implementiert sind, werden nicht initialisiert.
  Bei innere Klasse - auch Toplevel Klasse
- Ausnahme, wenn Kindklasse ein statisches Feld vom Elternteil abruft, dann wird NUR die Elternklasse geladen.
  Denn Grundlage: Statische Attribute werden niemals vererbt, Kindklassen koennen aber ohne Qualifizierung darauf zugreifen

Reflection
Man liest Objekte ueber das Klassenobjekt aus und kann sie so manipulieren.
Alles was Static ist, gehoert zum Klassenobjekt. (siehe auch Pruefungsaufgaben)
Auslesen scheint auch bei private gesetzten Feldern zu gehen. Aber kein Manipulieren moeglich!

Mit Class Objekten kann man was machen, wenn man ueber die Klassenobjekte auf Klasse zugreift
Damit werden auch Klassen ausgelesen, siehe Java Beans, Getter/Setter werden ausgelesen (passiert also ueber Reflection). Oracle: Wenn man keine Ahnung hat, soll man Finger von lassen

In Bibliothek, JavaPaket-> Paket namens reflect. Felder wie Field, Constructor, Method.class, etc.
Sind die Komponenten einer Klasse

```
Class<Person> person = Person.class;
Field[] fields = person.getDeclaredFields();
for (int i... field.length..) {
Sysoutprint(i fields[i].getName()); (Name des Attributs ausgeben lassen)
}
--> 0: firstName, 1 lastName, 2 age

Person p = new Person("Max", "MUstermann", 30);
sysoutprint("Max Alter: " + fields[2].getInt(p));
--> Alter 30

An Bibliotheksklassen koennten wir nicht fummeln, nur an unseren eigenen
fields[2].setInt(p, 100);
fields[0].set(p, "Neuer Name");
sysoutprint(p.age);
--> Alter 100

Normalerweise kann man keine private Felder manipulieren. Aber mit Flag setzen:
fields[1].setAccessible(true);
schon. Sonst wurde Sicherheitsmechanismus greifen und Fehlermeldung erscheinen.

Sicherheitsaspekt? Seine Pakete kann man irgendwie fuer Reflection oeffnen oder nicht.
```