

Laufzeittypinformation

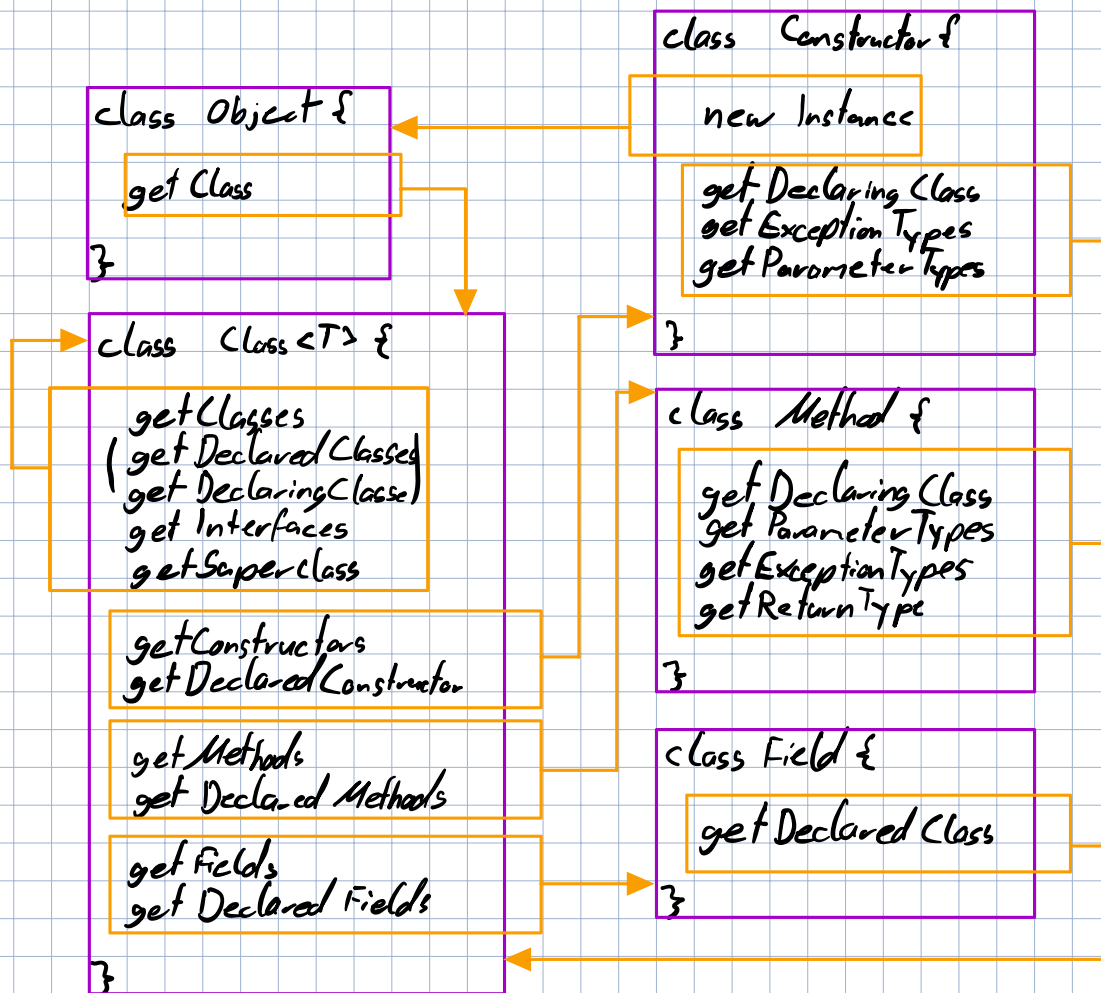
Programatischer Zugriff auf Typinformationen

- über Klassen, Konstruktoren, Methoden & Felder
- zur Laufzeit
- man kann auf alle Datentypen zugreifen, nicht aber auf den ausführbaren Code

Mechanismus

- Untersuchen von Objekten
- Erzeugen neuer Instanzen
- Aufruf von Methoden
- Lesen und Schreiben von Feldern
- Arbeiten mit Annotationen

Paket java.lang.reflect



Unterschied bei Zugriff auf Member:

- get xxx ... nur public, aber auch die geerbten
- getDeclaredxxx ... alle in dieser Klasse deklarierten

JVM erzeugt für jede Klasse ein eindeutiges Objekt vom Typ `Class<T>`

Typparameter T ist wiederum der Typ des Objekts, d.h.

- Class-Objekt zur Klasse Person ist vom Typ `Class<Person>`
 - class-Objekt zur Klasse String ist vom Typ `Class<String>`
- ```
Class<Person> personClass = Person.class;
```

Zugriff auf Class-Objekt

- mit `class` (siehe oben)
- mit `obj.getClass()`

Beachte: Generischer Typ nicht erlaubt

```
Class<?> personClass = person1.getClass();
```

aber Typcast möglich

```
Class<Person> personClass =
(Class<Person>) person1.getClass();
```

- mit `Class.forName` mit String des vollqualifizierten Namens:

```
Class<?> stringClass =
Class.forName("java.lang.String");
```

! Kein generischer Typ, weil nach Kompilieren  $\Rightarrow$  Objects! *Matthias*