

Übung 4

interfaces und Exceptions

1. Erzeugen Sie ein **interface**, das einen Nachrichtenaustausch zwischen zwei Objekten mit Hilfe einer Funktion

```
void senden(String nachricht, Object empfaenger)
```

und

```
String empfangen(Object sender)
```

realisieren soll.

2. Implementieren Sie zwei Testobjekte (evtl. verschiedener selbstdefinierter Klassen), die das in der vorigen Aufgabe erstellte **interface** implementieren, und sich gegenseitig Nachrichten verschicken, indem aus dem jeweils anderen Objekt die komplette Methode aufgerufen wird. D.h. beim **senden ()** einer Nachricht wird **empfangen ()** aus dem anderen Objekt aufgerufen.¹
3. Implementieren Sie ein Testprogramm, um das Senden und Empfangen von Nachrichten anhand der in der vorigen Aufgabe erstellten Testobjekte beobachten zu können.
Tipp: Mit den in jedem Java-Objekt vorhandenen Methoden **this.getClass().getName ()** und **this.hashCode ()** welche auch automatisch aufgerufen werden, wenn ein Objekt an **System.out.println ()** übergeben wird, können Informationen über Klasse und Objekt für **System.out.println ()** geholt werden, um die beteiligten Objekte zu identifizieren.
4. Erzeugen Sie eine Java-Klasse **Modulo** mit einer statischen Methode/Funktion

```
int modulo(int dividend, int divisor),
```

die den Teilungsrest bei der ganzzahligen Division $\frac{\text{dividend}}{\text{divisor}}$ an den Aufrufer zurückgeben soll. Dabei soll der Fall, dass bei der Berechnung ein **ArithmeticException**-Fehler (weil beispielsweise der **divisor** gleich 0 ist) auftritt, abgefangen werden und durch eine entsprechende Fehlermeldung sowie die Rückgabe des **dividend** an den Aufrufer quittiert werden.
5. Erzeugen Sie eine Java-Klasse **Alarm**, die im Konstruktor eine (beliebige) Exception auslöst, wenn beim Erzeugen eines Objektes dieser Klasse der String "**ein**" übergeben wird.

¹Sie werden ggf. in **Software Engineering** ein **Design Pattern** bzw. ein Java-Interface namens **Observer** kennenlernen, das solche Nachrichtensysteme stark vereinfacht.