

# Programmierkurs Anwendungsentwicklung

## Praktikum 03

# Aufgabe 1

- Die Klasse `AtomicLong`  
(<https://docs.oracle.com/en/java/javase/11/docs/api/java.base/java/util/concurrent/atomic/AtomicLong.html>)  
implementiert nicht das Interface `Comparable`. Implementieren Sie eine Klasse `AtomicLongComparable`, die die Klasse `AtomicLong` entsprechend erweitert.
- Implementieren Sie eine Testklasse. Diese soll eine generische Methode `findeMaximum` implementieren. Die Methode bekommt eine Liste von `AtomicLongComparable` Objekten übergeben.
- Implementieren Sie folgende `main` in der Testklasse und testen Sie das Programm.

```
public static void main( String[] args ) {  
    List<AtomicLongComparable> lngLst = new ArrayList<AtomicLongComparable>();  
    lngLst.add( new AtomicLongComparable() );  
    lngLst.add( new AtomicLongComparable( 99 ) );  
    lngLst.add( new AtomicLongComparable( 42 ) );  
    System.out.println( findeMaximum ( lngLst ) );  
}
```

# Aufgabe 2

- Schreiben Sie ein Konsolenprogramm, das beliebig viele Zahlen einliest und in einer Liste speichert, z.B.

Geben Sie eine Zahl ein: 5

Eine weitere Zahl (y/n)? Y

...

- Danach gibt es ein Ratespiel, z.B.

Wie lautet die erste Zahl? 5

Richtig! Die Nächste: 4

Leider Falsch: 3 wäre richtig gewesen!

Mit try-catch



- Stellen Sie durch geeignetes Exception-Handling sicher, dass nur Zahlen bzw. y/n eingegeben werden, z.B.

Geben Sie eine Zahl ein: t

t ist keine Zahl!

Geben Sie eine Zahl ein: ...

Eine weitere Zahl (y/n)? k

'k' ist nicht 'y' oder 'n'!

Eine weitere Zahl (y/n)? ...

# Aufgabe 3 (1/2)

```
4 public class Termin {  
5     public String beschreibung;  
6     public String ort;  
7     public LocalDateTime zeit;  
8  
9     public Termin(String was, String wo, LocalDateTime wann) {  
10         beschreibung=was;  
11         ort=wo;  
12         zeit=wann;  
13     }  
14 }
```

## Aufgabe 3 (2/2)

- Implementieren Sie in einer Testklasse: Sie speichern mehrere Termine in einem Array und möchten dieses nun zeitlich sortieren, also den ersten Termin zuerst. Verwenden Sie zum Sortieren `Arrays.sort`, und übergeben Sie im zweiten Parameter einen Lambda-Ausdruck zur Implementierung der `compare`-Methode der `Comparator`-Schnittstelle.

# Aufgabe 4

- Zur Verarbeitung von Terminen aus Aufgabe 2 implementieren Sie bitte eine Methode `printTermine`, die alle Termine ausgibt, auf die ein bestimmtes Kriterium zutreffen.
- übergeben Sie an `printTermine` ein Array mit Terminen sowie einen Lambda-Ausdruck, der alle Termine auswählt, die heute geplant sind.

# Aufgabe 5

- Im folgenden Beispiel soll eine Liste von Integer-Objekten ausgegeben werden. Dabei wird die Methode `println` als Referenz an `forEach` übergeben.

```
Integer[] data = {1, 2, 3};  
var lst = Arrays.asList(data);  
lst.forEach(System.out::println);
```

Wie sieht eine äquivalente Implementierung mit einem Lambda-Ausdruck aus? Implementieren Sie ein minimalistisches Testprogramm.