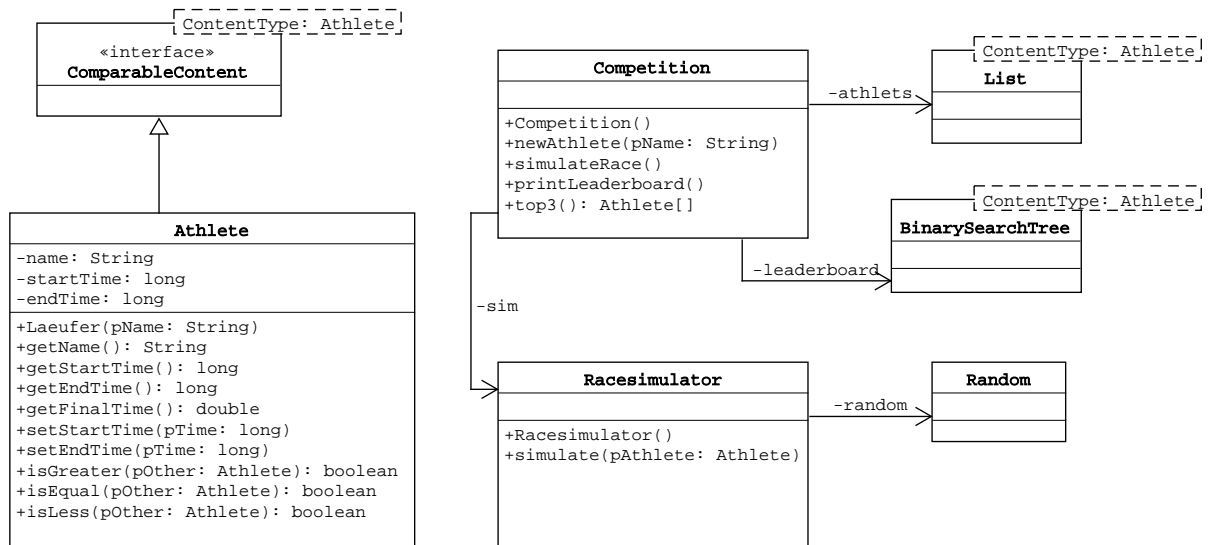


## Nichtlineare Datenstrukturen Langlaufsimulator



- BESCHREIBE die Beziehungen zwischen den einzelnen Klassen.
- IMPLEMENTIERE die Langlaufsimulation anhand der Modellierung und der Klassendokumentation.
- ENTSCHEIDE, ob es sinnvoll wäre, die Rangliste (leaderboard) als lineare Liste zu modellieren, anstatt eines binären Suchbaums. Nimm dabei insbesondere Bezug zur Umsetzung der Methode `printLeaderboard()`.

### Dokumentation der Klasse Athlete

Die Klasse `Athlete` speichert die Renndaten eines Läufers. Sie dient hauptsächlich der Datenhaltung. Die Zeiten werden als Zeitstempel gespeichert. Ein Zeitstempel ist eine Zahl, die die vergangenen Millisekunden seit dem 1. Januar 1970 angibt. Mit `System.currentTimeMillis()` kann der aktuelle Zeitstempel abgerufen werden.

```
public long getStartTime()
```

Gibt den Zeitstempel der Startzeit des Läufers zurück.

```
public long getEndTime()
```

Gibt den Zeitstempel der Endzeit des Läufers zurück.

```
public void setStartTime(long pTime)
```

Setzt die Startzeit des Läufers auf den angegebenen Zeitstempel.

```
public void setFinalTime(long pTime)
```

Setzt die Endzeit des Läufers auf den angegebenen Zeitstempel.

```
public double getEndTime()
```

Berechnet die Zeit des Läufers als Differenz aus der Start- und Endzeit auf Sekunden gerundet.

```
public boolean isGreater(Athlete pOther)
```

Gibt `true` zurück, wenn die Zeit des Läufers größer (also langsamer) als die des Läufers `pOther` ist.

```
public boolean isEqual(Athlete pOther)
```

Gibt `true` zurück, wenn die Zeit des Läufers gleich der des Läufers `pOther` ist.

```
public boolean isLess(Athlete pOther)
```

Gibt `true` zurück, wenn die Zeit des Läufers kleiner (also schneller) als die des Läufers `pOther` ist.

### Dokumentation der Klasse `Competition`

Die Klasse `Competition` stellt einen Langlaufwettbewerb dar. Sie ist die Hauptklasse für den Simulator.

```
public void newAthlete(String pName)
```

Erstellt einen neuen Läufer und speichert ihn in der Liste der Teilnehmer

```
public void simulateRace()
```

Simuliert ein Langlaufrennen, indem für jeden Teilnehmer mittels des `Racingsimulators` eine Zeit für das Rennen generiert wird. Steht eine Zeit fest, wird der Läufer in die Rangliste eingefügt.

```
public void printLeaderboard()
```

Gibt die aktuelle Rangliste des Rennens auf der Kommandozeile aus. Sortiert vom schnellsten zum langsamsten Läufer.

```
public Athlete[] top3()
```

Gibt ein Array mit den drei besten Läufern des Rennens zurück. Wenn das Rennen noch nicht vollständig simuliert wurde, wird ein leeres Array zurückgegeben.

### Dokumentation der Klasse `Racesimulator`

Die Klasse `Racesimulator` erzeugt simulierte Zeiten für Läufer. Sie benutzt dafür ein Objekt der Klasse `java.util.Random`.

```
public void simulate(Athlete pAthlete)
```

Erzeugt eine Start und eine Endzeit für den Läufer `pAthlete`.