

## Laufzeit

Programme müssen häufig große Datenmengen bearbeiten. Dabei kommt es darauf an, wieviel Zeit sie für die Berechnungen benötigen. Laut einer Studie von IBM vor einigen Jahren wird z.B. 25% der Rechenzeit aller Computer der Welt darauf verwendet, Daten zu sortieren. Da lohnt es sich, schnelle Verfahren zu entwickeln und so die Kosten von Hardware, Strom usw. zu reduzieren.

Der Begriff „Laufzeit“ bezeichnet die Dauer, die ein Programm (in unserem Fall eine Java-Methode) benötigt, um eine Aufgabe zu erledigen.

## Messung der Laufzeit

Die Laufzeit einer Java-Methode lässt sich sehr einfach messen. Bevor die Methode aufgerufen wird, hält man die aktuelle Zeit fest. Dann wird die Methode ausgeführt. Nachdem die Methode beendet ist, wird wieder die aktuelle Zeit festgehalten. Die Differenz zwischen Anfangs- und Endzeit ist die Zeit, die die Methode benötigt hat.



Um die Laufzeitmessung etwas zu vereinfachen kannst du die Klasse „Stoppuhr“ verwenden. Die Methode `gestoppteZeit()` gibt die Laufzeit **in Sekunden** an.

Beispiel:

```
Stoppuhr uhr = new Stoppuhr();
uhr.start();
langwierigeMethode();
uhr.stop();
double laufzeit = uhr.gestoppteZeit();
```

## Aufgabe 1

Im ausgeteilten Material findest du eine Java-Klasse mit den drei Sortierverfahren (Selection Sort, BubbleSort, Insertion Sort), sowie die Klasse Stoppuhr.

Bearbeite die **main-Methode**:

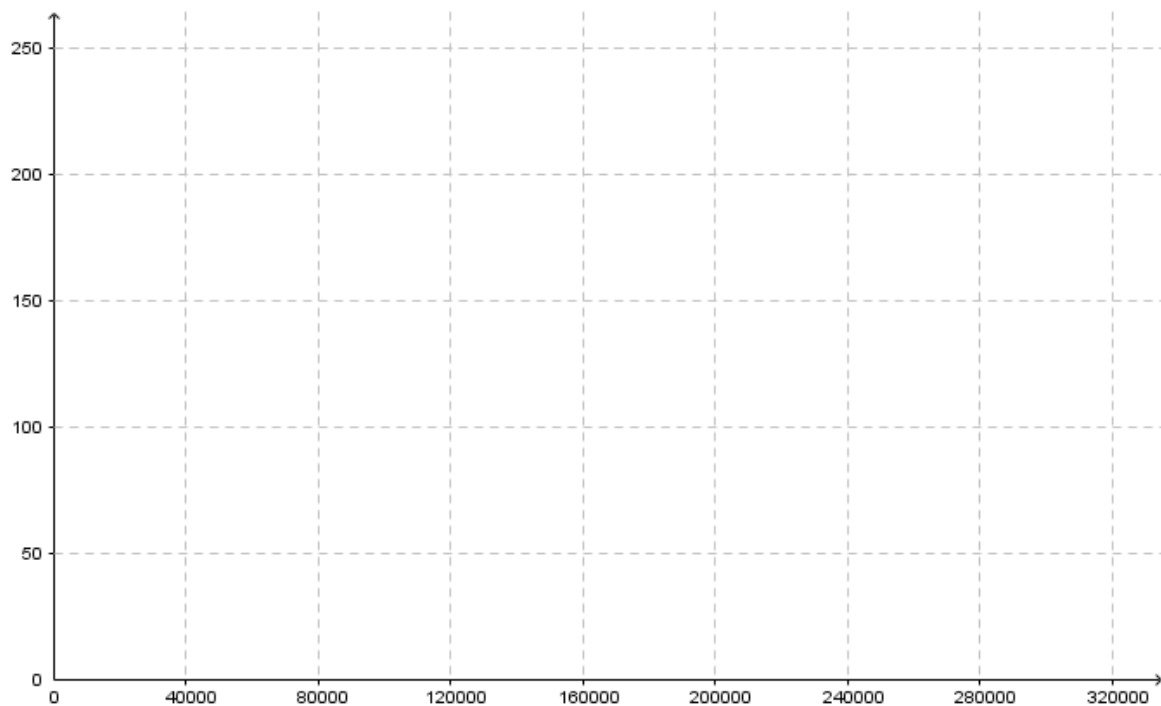
Rufe die drei Sortierverfahren auf, miss jeweils die dafür benötigte Zeit und gib diese auf der Konsole aus. Das Array wird vor jedem Sortieren mit Zufallszahlen gefüllt (Achtung, die Zeit zum Füllen wird nicht mitgezählt).

Lass das Programm mehrmals laufen: mit 5.000 Elementen, 10.000, 20.000 usw.

Notiere die benötigten Zeiten in der Tabelle auf Seite 2.

Anschließend zeichne die Ergebnisse in das Koordinatensystem.

Länge der Liste	Selection Sort	Bubble Sort	Insertion Sort
5.000			
10.000			
20.000			
40.000			
80.000			
160.000			
320.000			
nur schätzen: 640.000			
nur schätzen: 1.280.000			



## Aufgabe 2

Betrachte das Wachstum der Laufzeit des **Selection Sort** von 80.000 auf 160.000 und von 160.000 auf 320.000 Einträge. Um welchen Faktor verlängert sich die Laufzeit jeweils?

Betrachte genauso das Wachstum für den Bubble Sort und den InsertionSort.

Vervollständige die folgenden Sätze:

Wenn man die Länge der Liste verdoppelt, wächst die Laufzeit um ...

Wenn man die Länge der Liste vervierfacht, wächst die Laufzeit um ...

Wenn man die Länge der Liste verzehnfacht, wächst die Laufzeit um ...