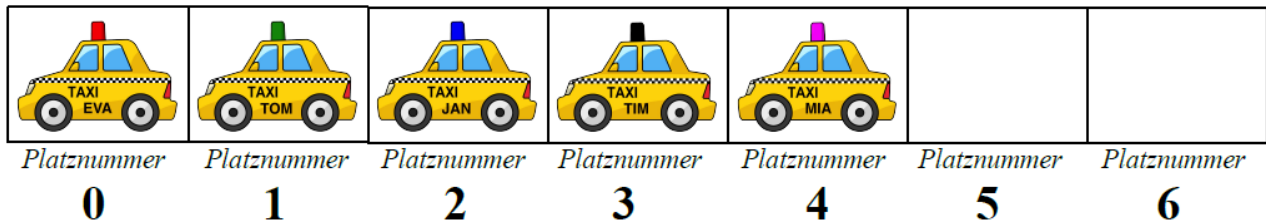


Arrays – Felder

Gelegentlich braucht man in einem Programm sehr viele **Attribute desselben Datentyps**. In solchen Fällen deklariert und initialisiert man nicht alle gleichartigen Objekte einzeln, sondern fasst sie in einem Array zusammen.

Die Datenstruktur **Array** speichert (beliebig) viele Objekte desselben Datentyps. Man kann Arrays mit einem Taxistand vor dem Bahnhof vergleichen:



- Es ist immer eine **festgelegte Größe** an Plätzen vorhanden (hier: 7 Stück).
- Auf einem Taxistand dürfen nur Taxis stehen, nichts anderes. → Ein Array verwaltet auch nur Objekte **eines gemeinsamen Datentyps**.
- Es müssen nicht alle Plätze benutzt werden, einige können auch leer sein.
- Mehr Objekte als Felder vorhanden sind können nicht gespeichert werden.
- Die Plätze werden – **beginnend mit Null** – durchnummeriert.

Programmierung in Java

Arrays kennzeichnet man im Code, indem man eckige Klammern hinter den Datentyp bei der **Deklaration** der Attribute schreibt:

```
int[] punktestaende;  
String[] namensliste;  
RECHTECK[] mauern;
```

Ein Array ist wiederum ein **eigenständiges Objekt** und muss neu erzeugt werden. Beim **Initialisieren** eines Arrays musst du eine Größe festlegen und nochmals den Datentyp angeben.

```
punktestaende = new int[10];  
namensliste = new String[10];  
mauern = new RECHTECK[15];
```

Die Arrays sind nach dem Erzeugen allerdings noch leer und speichern noch nichts!

Man kann jeden Platz im Array einzeln ansprechen und darin Werte/Objekte speichern.

Platz 0 im Array
wird mit 5 belegt.

```
punktestaende[0] = 5;  
namensliste[0] = "Sepp";  
mauern[0] = new RECHTECK();
```

Beachte, dass bei Arrays von Objekten jedes einzelne Objekt erst noch erzeugt werden muss.

Man mit den gespeicherten Inhalten (primitive Datentypen als auch Referenzobjekten) arbeiten, indem man sie über **ihren Platz im Array** anspricht:

```
punktestaende[0] = punktestaende[0] + 10;  
mauern[0].setzeGroesse(10,10);  
mauern[4].setzeFarbe("rot");
```

Aufgabe 1:

Schreibe eine neue Klasse *HIGHSCORE* in einem Projekt *Array*, mit den beiden Array-Attributen *punktestaende* und *namensliste*.

- a) Initialisiere die beiden Arrays im Konstruktor mit der Länge 10. Setze im Konstruktor bereits die beiden ersten Spielernamen auf „Sepp“ und „Susi“ und die beiden ersten Punktestände auf 12000 und 7500.
- b) Schreibe eine Methode `nenneRang(int platznummer) : String`, die zurück gibt, welcher Spieler mit wie vielen Punkten auf der entsprechenden Platznummer im Array gespeichert ist (**Tipp: Strings und ints können mit + verbunden werden**).
- c) Schreibe eine Methode `setzeRang(int platznummer, String name, int punkte) : void` die einen neuen Spieler und seine Punkte an einer bestimmten Platznummer einträgt. Überprüfe die Funktion dieser Methode durch das Nutzen der anderen.

Aufgabe am Projekt Moorhuhn:

Öffne dein Projekt *Moorhuhn*. Nutze für die fünf Hühner ein passendes Array der Länge 5. Überarbeite dein Projekt, sodass das Spiel wieder fehlerfrei spielbar ist.