

Tagesprogramm

Map

Eigene Objekte

Queue versus Map

typisch für **Queue** und **Deque**:

jeder Eintrag wird genau einmal verwendet

kein direkter Zugriff auf „bestimmten“ Eintrag

typisch für **Array** und **Map**:

jeder Eintrag wird wiederholt (oder nie) verwendet

man muss „bestimmten“ Eintrag adressieren

Index in Array ist immer vom Typ `int`

Map ähnelt Array, aber beliebiger Referenztyp als „**Key**“

Umgang mit Map

```
private static void useMap(Map<String, String> map) {  
    for (int i = 1; i <= 100; i++) {  
        map.put("String " + i, "AssociatedString" + i);  
    }  
  
    for (int j = 3; j <= 100; j += 3) {  
        System.out.println(map.get("String " + j));  
    }  
}
```

einige Methoden von Map:

```
V put(K key, V value); // assoziiert key mit value  
                        // Ergebnis ist alter Wert (oder null)  
  
V get(K key);          // liest den mit key assoziierten Wert  
  
boolean containsKey(Object key); // kommt key vor?
```

Aufgabe: LIFO, FIFO, Array oder Map?

Finden Sie in Gruppen zu 2 bis 3 Personen eine Antwort auf folgende Frage:

Wann ist welche dieser Datenstrukturen vorteilhaft?

Zeit: 3 Minuten

Klasse

Klasse ist **Vorlage** für neue Objekte

```
PairOfStrings p = new PairOfStrings();
```

jedes Objekt hat eigene **Objektvariablen**

```
PairOfStrings q = new PairOfStrings();  
p.x = "a";    q.x = "b";    System.out.println(p.x + q.x); → ab  
p = q;    System.out.println(p.x + q.x); → bb
```

direkte Zugriffe auf Objektvariablen von außen verpönt,
stattdessen Zugriffe über Methoden des Objekts

```
p.set("a", "b");    System.out.println(p.get1());
```

public ermöglicht Zugriffe von außen,
private verbietet Zugriffe von außen

Aufgabe: Referenzen

Führen Sie sich vor Augen, wie die Referenzen in diesen Beispielen nach jedem einzelnen Schritt genau aussehen (mittels „Tafeln“):

```
PairOfStrings p = new PairOfStrings();  
PairOfStrings q = new PairOfStrings();  
  
p.set("a", "b");  
q.set("c", "d");  
System.out.println(p.get1()+q.get1()+q.get2()+p.get2());  
  
p.set(p.get2(), q.get1());  
q = p;  
System.out.println(p.get1()+q.get1()+q.get2()+p.get2());
```