

Nachdenkzettel: Collections

1. ArrayList oder LinkedList – wann nehmen Sie was?

Answer:

- **ArrayList** implements a list through an array that is dynamically resized
- -> use, if you want to access certain elements often/quickly
- **LinkedList** implements a list through a double-concatenated list
- -> use, when you want to add, change or delete elements quickly/more often

2. Interpretieren Sie die Benchmarkdaten von: <http://java.dzone.com/articles/java-collection-performance>. Fällt etwas auf?

Answer:

- collections are slower in data processing (e.g. copyonwriteArray)



3. Wieso ist CopyOnWriteArrayList scheinbar so langsam?

Answer: A copy of the ArrayList is created and automatically synchronized for each update operation at a specific point. This can affect performance.

4. Wie erzeugen Sie eine thread-safe Collection (die sicher bei Nebenläufigkeit ist) (WAS?? die ArrayLists, Linkedlists, Maps etc. sind NICHT sicher bei multithreading??? Wer macht denn so einen Mist???)

Answer: make the collection immutable

5. Achtung Falle!

```
List<Integer> list = new ArrayList<Integer>;
```

```
Iterator<Integer> itr = list.iterator();
while(itr.hasNext()) {

    int i = itr.next();
    if (i > 5) { // filter all ints bigger than 5
        list.remove();
    }
}
```

Falls es nicht klickt: einfach ausprobieren...

Macht das Verhalten von Java hier Sinn?

Gibt es etwas ähnliches bei Datenbanken? (Stichwort: Cursor. Ist der ähnlich zu Iterator?)

Answer: The list.remove()-Statement removes at the given index and not the value itself given to the method.

6. Nochmal Achtung Falle: What is the difference between get() and remove() with respect to Garbage Collection?

Answer: remove() makes a certain value liable for garbage collection
get() references the value

7. Ihr neuer Laptop hat jetzt 8 cores! Ihr Code für die Verarbeitung der Elemente einer Collection sieht so aus:

```
Iterator<Integer> itr = list.iterator();
while(itr.hasNext()) {
    int i = itr.next();
    //do something with i...
}
```

War der Laptop eine gute Investition?

Answer: yes, it can be use to run complex code

Für die Mutigen: mal nach map/reduce googeln!