

Nachdenkzettel: Collections

1. ArrayList oder LinkedList – wann nehmen Sie was?

ArrayList if you want to easily get the index of an element in the List.

LinkedList if you want to easily remove and add elements to the List.

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



- CopyOnWriteArrayList is astronomically slower than most other collections

- Most Collections have their use cases where they are slower or faster.

3. Wieso ist CopyOnWriteArrayList scheinbar so langsam?

In this collection all mutative operations, such as add or set are implemented by copying the array. This takes time and is therefore slow.

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???)

By using the synchronized collection method.

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?)

The problem with this code is that by using an iterator you should be using the iterator to remove an element from the list. Instead of list.remove() use itr.remove().

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

Get() does not remove any entries, while remove() actually does remove elements from a collection.

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?

Für die Mutigen: mal nach map/reduce googeln!

If you use Multithreading then the laptop was a good investment. If not it was wasted money. In

this case it was wasted because the code above only utilizes one core.