ECS717 - IT Programming

Extra Lab Notes 07

14/11/2016

Arrays and ArrayList

I would start with this example of physics nobel prize winners and a way to store each one of them in a single variable….

List of Physics Nobel Prize Winners

`Person winner_a`	`firstname = "Wilhelm"` `lastname = "Roentgen"`	
`Person winner_b`	`firstname = "Hendrik"` `lastname = "Lorentz"`	
`Person winner_c`	`firstname = "Pieter"` `lastname = "Zeeman"`	
`Person winner_d`	`firstname = "Antoine"` `lastname = "Becquerel"`	
`Person winner_e`	`firstname = "Pierre"` `lastname = "Curie"`	
…	…	...

Arrays and ArrayList

But then talk about this approach being impractical when you for example want to print all of the prize winners. And that there are arrays which they already know from the args-Array of the main method

```
Person winner_a = new Person(“Wilhelm”,
  “Roentgen”);
Person winner_b = new Person(“Hendrik”,
  “Lorentz”);
Person winner_c = new Person(“Pieter”,
  “Zeeman”);
…
winner_a.printName();
winner_b.printName();
winner_c.printName();
…
```

Arrays and ArrayLis

might have gone a bit overboard with this graphic. but I wanted to show, that an array is a bit like a box with just so many slots for objects and a fixed size

Arrays are a **containers** which store a **sequence** of values or objects of the **same type**

Arrays have a **fixed size (=length N)**

Elements in the Array can be addressed by their **fixed position/index (starting with 0 to N-1)**

Arrays and ArrayLists

Arrays are a **containers** which store a **sequence** of values or objects of the **same type**

Arrays have a **fixed size (=length N)**

Elements in the Array can be addressed by their **fixed position/index (starting with 0 to N-1)**

Index 0

Arrays and ArrayLists

Arrays are a **containers** which store a **sequence** of values or objects of the **same type**

Arrays have a **fixed size (=length N)**

Elements in the Array can be addressed by their **fixed position/index (starting with 0 to N-1)**

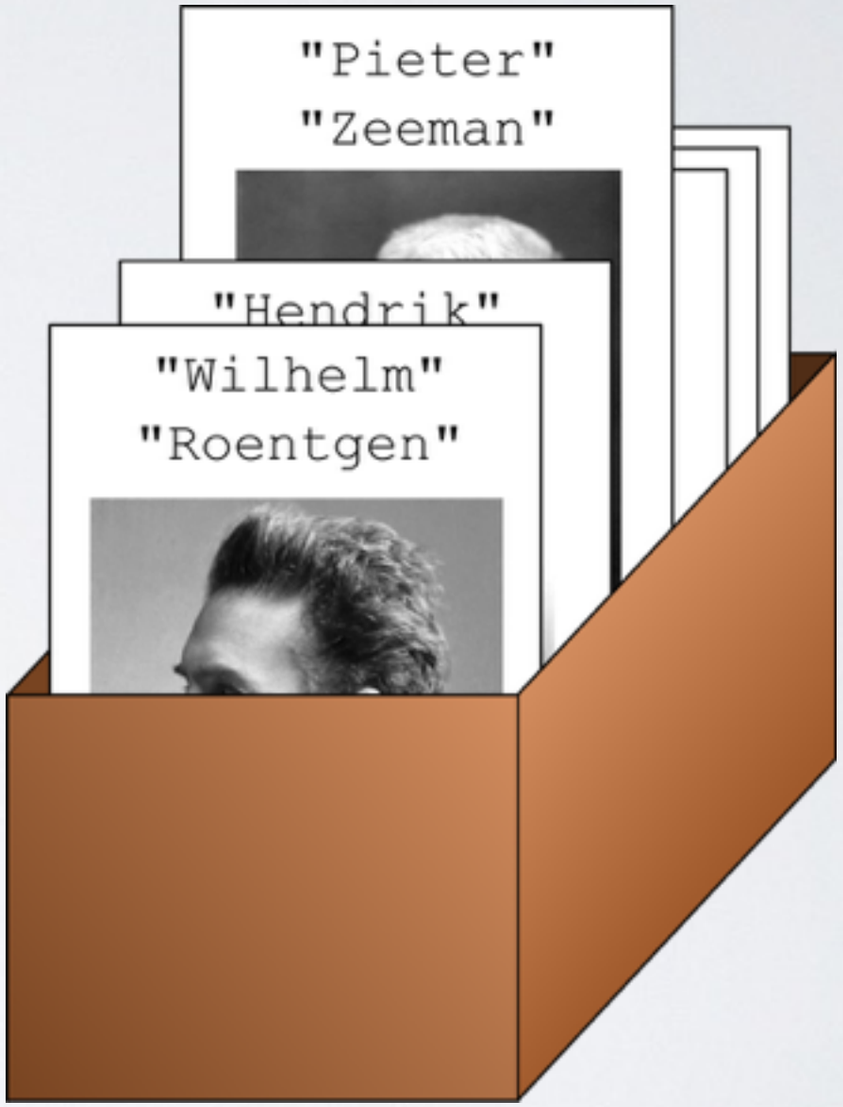
Index 1

Arrays and ArrayLists

Arrays are a **containers** which store a **sequence** of values or objects of the **same type**

Arrays have a **fixed size (=length N)**

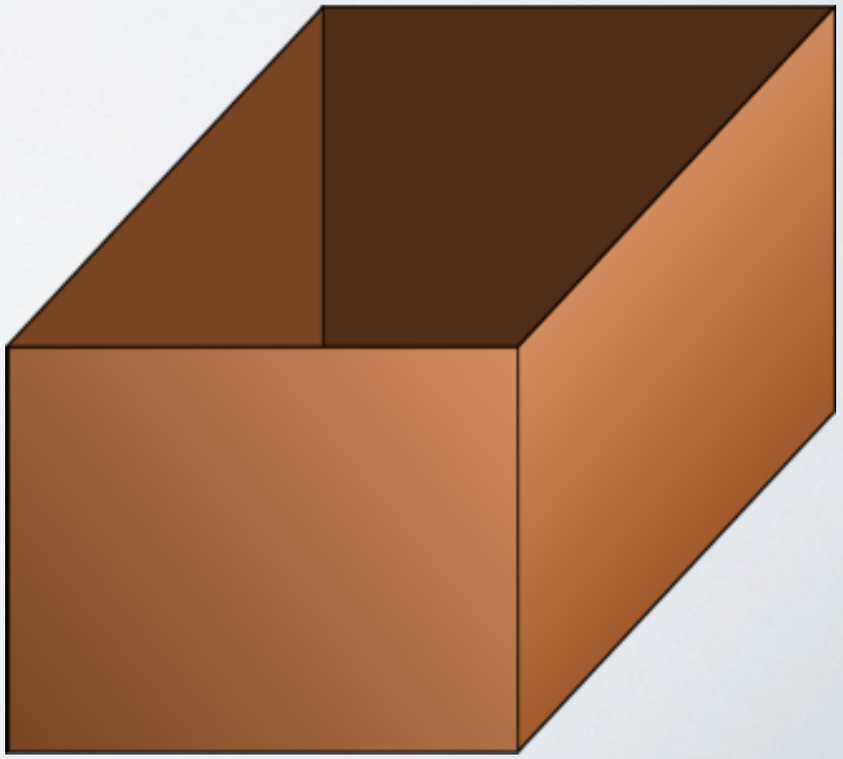
Elements in the Array can be addressed by their **fixed position/index (starting with 0 to N-1)**



Index 2

Arrays and ArrayLists

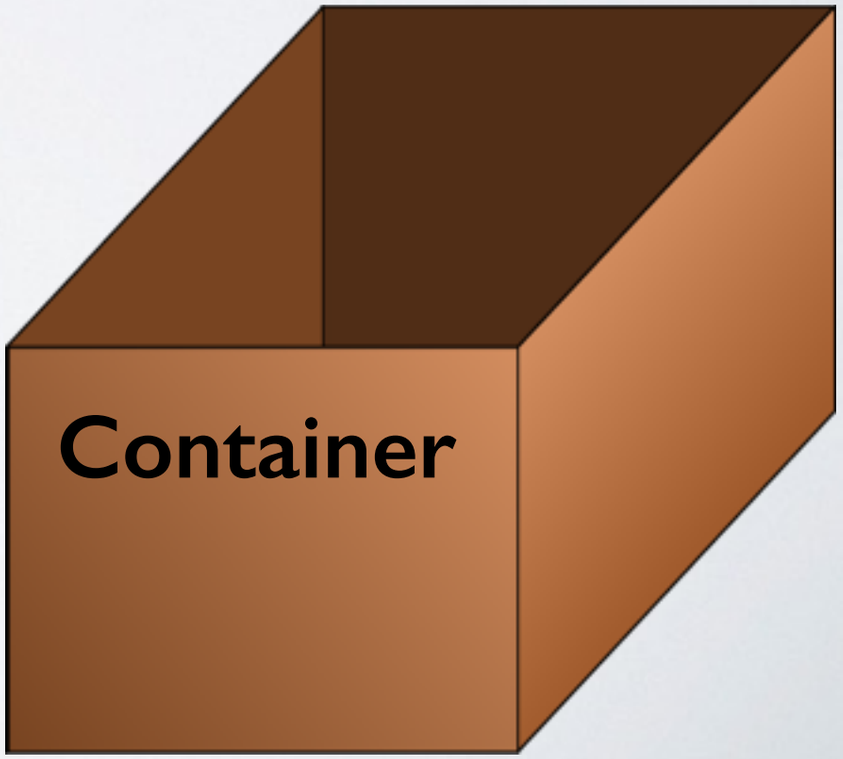
```
// creating the empty container
Person[] winners = new Person[201];
```



Arrays and ArrayLists

```
// creating the empty container
Person[] winners = new Person[201];
```

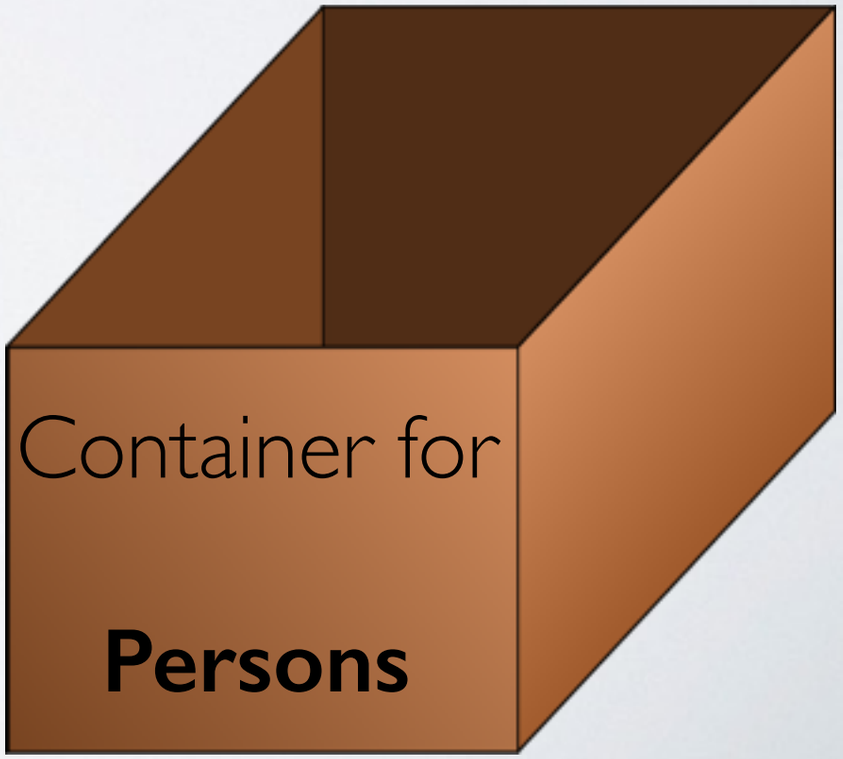
- *Arrays are a **containers** which store a **sequence** of values or objects of the **same type***



Arrays and ArrayLists

```
// creating the empty container
Person[] winners = new Person[201];
```

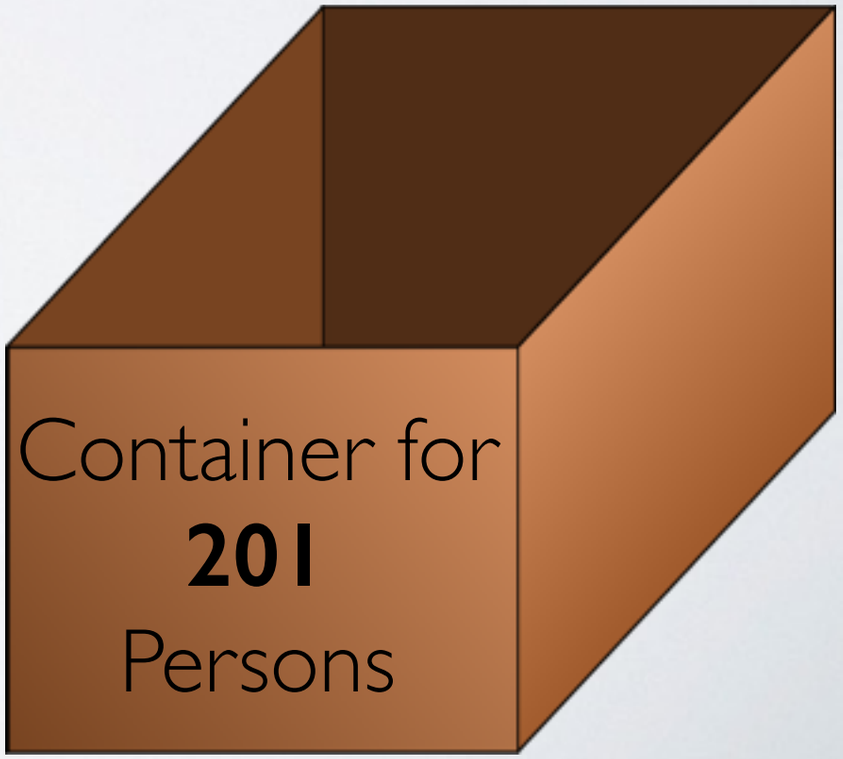
- *Arrays are a **containers** which store a **sequence** of values or objects of the **same type***



Arrays and ArrayLists

```
// creating the empty container
Person[] winners = new Person[201];
```

- *Arrays have a* ***fixed size (=length)***



Arrays and ArrayLists

Elements in the Array can be addressed by their ***index (starting with 0)***

e.g.

```
winners[0] // the first element in the
array
```

Arrays and ArrayLists

Elements in the Array can be addressed by their ***index (starting with 0)***

```
// creating the empty container
Person[] winners = new Person[201];

System.out.println("1st is " + winners[0]);
```

What is the output?

Arrays and ArrayLists

Elements in the Array can be addressed by their ***index (starting with 0)***

```
// creating the empty container
Person[] winners = new Person[201];

System.out.println("1st is " + winners[0]);
```

What is the output?

```
> 1st is null
```

Arrays and ArrayLists

```
// creating the empty container
Person[] winners = new Person[201];

// putting the first element in the Array
winners[0] = new Person(“Wilhelm”,
“Roentgen”);
```

Arrays and ArrayLists

The whole process:

```
Person[] winners = new Person[201];

winners[0] = new Person(“Wilhelm”,
“Roentgen”);

winners[1] = new Person(“Hendrik”,
“Lorentz”);

winners[2] = new Person(“Pieter”, “Zeeman”);
…
```

Arrays and ArrayLists

We want to print all the names:

```
winners[0].printName();
winners[1].printName();
winners[2].printName();
winners[3].printName();
```

Arrays and ArrayLists

We want to print all the names:

```
winners[0].printName();
winners[1].printName();
winners[2].printName();
winners[3].printName();
```

Can you think of a better way?

Arrays and ArrayLists

Arrays and Loops go well together:

```
for(int i = 0; i < winners.length; i++){
   winners[i].printName();
}
```

Arrays and ArrayLists

Arrays and ArrayLists

ArrayLists are a **containers** which store a **sequence** of values or objects of the **same type**

ArrayLists have a **variable size and no size restriction**

Elements in the Array can be addressed by their **position/ index (starting with 0), but it can change**

Arrays and ArrayLists

ArrayLists allow to insert new elements

at the last position:

```
add(E e)
```

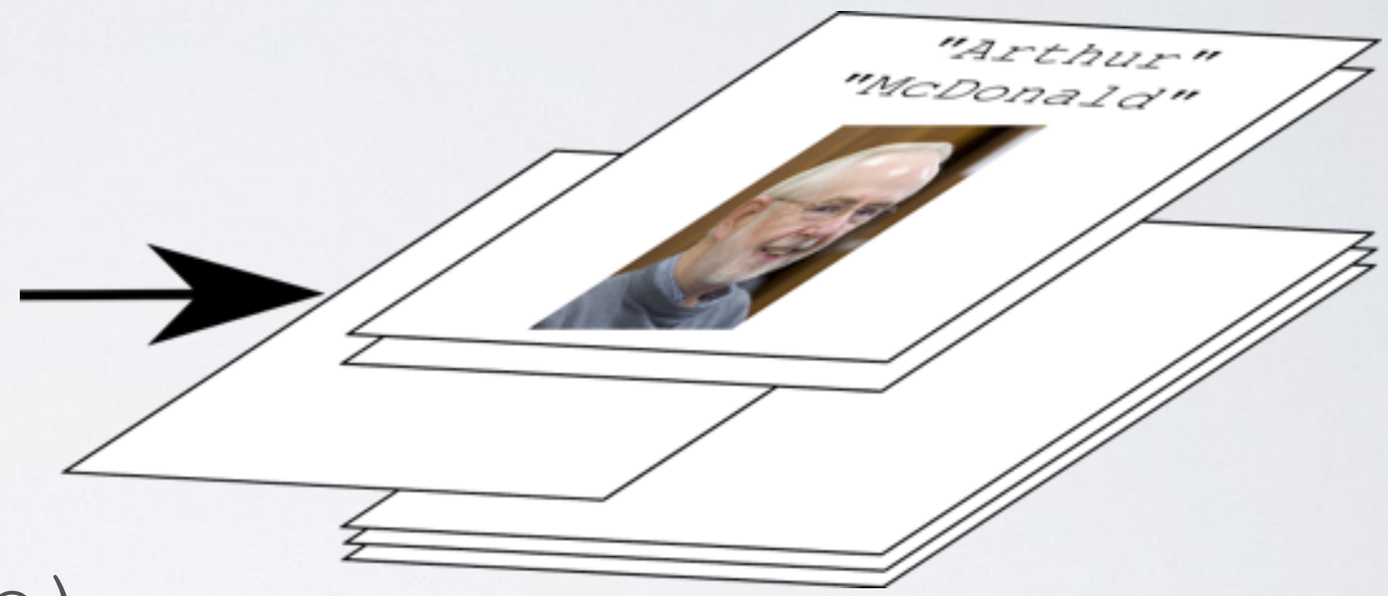
in the middle:

```
add(int index, E e)
```

at the first position:

```
add(0, E e)
```

(E is the type of the objects in the ArrayList)



Arrays and ArrayLists

`boolean add(E e)`	Appends an element to the list and the size of the list increases
`void add(int index, E e)`	Inserts an element at the index position; the indices of the following elements are changed
`E remove(int index)`	removes the object at the index from the list and returns it
`E get(int index)`	returns the element at the index position
`E set(int index, E e)`	Change the element at an index position
`int size()`	returns the length of the ArrayList

more information in the Java Docs:
http://docs.oracle.com/javase/7/docs/api/java/util/ArrayList.html

Arrays and ArrayLists

```
import java.util.ArrayList; // first line in file
```

Arrays and ArrayLists

```
import java.util.ArrayList; // first line in file

…

// creating the empty ArrayList
ArrayList<Person> winners = new ArrayList<Person>();
```

Arrays and ArrayLists

```
import java.util.ArrayList; // first line in file

…

// creating the empty ArrayList
ArrayList<Person> winners = new ArrayList<Person>();

// first element in the list
winners.add(new Person(“Wilhelm”, “Roentgen”));
// second element in the list
winners.add(new Person(“Pieter”, “Zeeman”));
// inserted as second element in the list
winners.add(1, new Person(“Hendrik”, “Lorentz”));
```

Arrays and ArrayLists

ArrayList go well with loops, too:

```
for(int i = 0; i < winners.size(); i++){
   winners.get(i).printName();
}
```

Questions?

QM+ discussion forum