

# Arrays

Topic of Software Systems (TCS module 2)

Lecturer: Marieke Huisman



# ARRAYS

---

- Arrays are a frequently used way to implement lists
- In Module 1 (Algorithms pearl) you learned to use arrays in Python
- Java arrays have the same purpose and are similar to Python arrays (not exactly the same!)

# ARRAYS IN JAVA

---

- An array is a data structure to store elements of the same type
- Arrays have a fixed length
- Array elements are stored and accessed in sequence
- Examples: array of int, float, char, String, Account, Room, another array
- Array needs to be declared and created before being used

# ARRAY DECLARATION AND CREATION

---

## Declaration

- Defines an **identifier** to refer to an array

`int[] values; String[] args; Room[] rooms;`

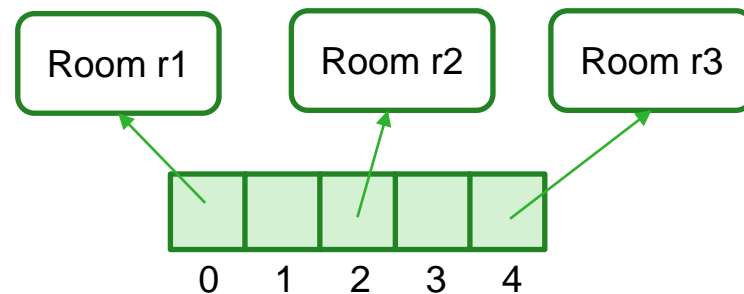
## Creation

- Allocates **memory positions** for the elements of the array

`int[] values = new int[50];`

`String[] args = new String[10];`

`Room[] rooms = new Room[5];`



# ARRAY USAGE

---

- Read values from an array

```
int i = values[3]; // copy 4th value of values to i
```

```
Room r = rooms[2]; // copy 3rd reference of rooms to r
```

- Modify values from an array

```
values[3] = i; // copy i to 4th position of values
```

```
rooms[2] = r; // copy reference r to 3rd position of rooms
```

# ARRAY ITERATION

---

‘Classic’ pattern: **iterate** over the elements of an array

```
for (int i = 0; i < values.length; i++) {  
    values[i] = i + 1;  
}
```

**Array initialisation**: array elements have to be **created** to be used

```
for (int i = 0; i < rooms.length; i++) {  
    rooms[i] = new Room(101+i);  
}
```

# ARRAYS AS METHOD PARAMETERS

## EXAMPLE

---

```
public static int sumValues(int[] values){  
    int sum = 0;  
    for(int i = 0; i < values.length; i++){  
        sum += values[i];  
    }  
    return sum;  
}
```

Array is passed as  
parameter

Array values are used

# MULTIDIMENSIONAL ARRAYS

```
class MultiDim {
```

```
    private int [][] m;
```

Declaration

```
    public MultiDim(int x, int y) {
```

```
        m = new int [x][y];
```

Initialisation

```
        for (int i = 0; i < x; i++) {
```

```
            for (int j = 0; j < y; j++) {
```

```
                m[i][j] = 0;
```

Assignment to an element

```
            }}
```

```
    public void increaseRow(int x) {
```

```
        for (int i = 0; i < m[x].length; i++) {
```

Length of inner array

```
            m[x][i] = m[x][i] + 1;
```

Reading an element

```
        }}
```



# ARRAY LIMITATIONS

WHAT DO YOU HAVE TO DO IF YOU WANT TO...

---

- Remove an element of an arbitrary position of an array without creating an empty space?
  - Shift all elements to the beginning of the array one-by-one!
- Add an extra element to the array (list) when all positions are already taken?
  - Create new (bigger array), copy all elements and add new element
- You have to do all the manipulations by hand!!!