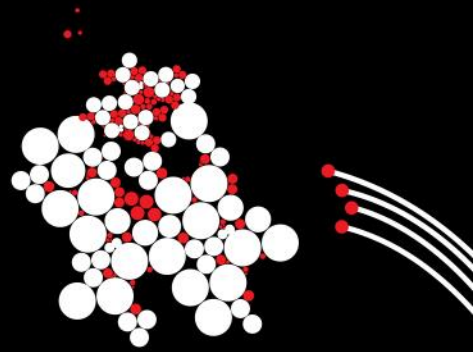
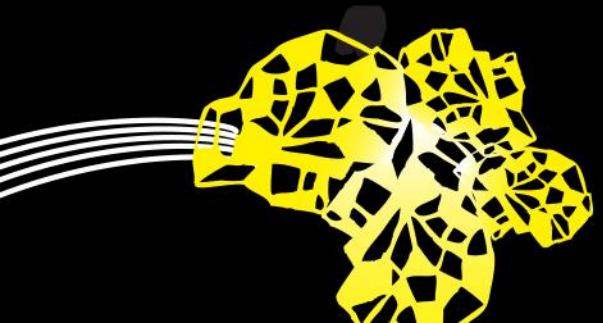
UNIVERSITY OF TWENTE.



List Implementation with Array

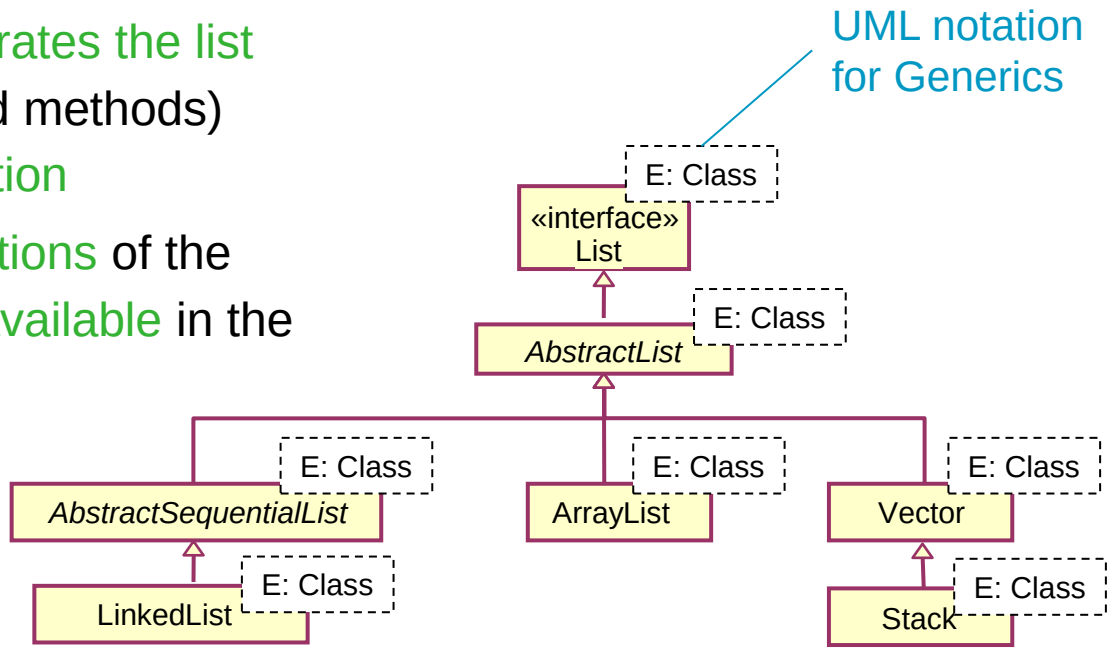
Topic of Software Systems (TCS module 2)

Lecturer: Faizan Ahmed



List<E> IMPLEMENTATIONS

- List<E> interface separates the list concept (and related methods) from its implementation
- Various implementations of the List<E> interface are available in the java.utils package



List<E> IMPLEMENTATION WITH ARRAY

BASED ON ARRAYLIST<E>

```
public class SimpleArrayList<E> implements List<E> {
    public static final int INITIAL_SIZE = 10;
    private Object[] list;
    private int size;

    public SimpleArrayList() {
        this.list = new Object[INITIAL_SIZE];
        this.size = 0;
    }
```

list.length == *INITIAL_SIZE*

size == 4

List<E> IMPLEMENTATION WITH ARRAY

QUERIES

```
public int size() {
    return this.size;
}

public boolean isEmpty() {
    return size == 0;
}

public boolean contains(Object o) {
    boolean result = false;
    for (int i = 0; i < this.size && !result; i++) {
        if (list[i].equals(o))
            result = true;
    }
    return result;
}
```

```
public E get(int index) {
    if (index < 0 || index >= size)
        throw new ArrayIndexOutOfBoundsException();
    return (E) this.list[index];
}
```

List<E> IMPLEMENTATION WITH ARRAY

COMMANDS

```
public boolean add(E e) {
    if (this.size == this.list.length) {
        Object[] newList = new Object[this.list.length * 2];
        for (int i = 0; i < size; i++)
            newList[i] = this.list[i];
        list = newList;
    }
    this.list[this.size++] = e;
    return true;
}
```

allocate twice the space (ArrayList uses a more complex algorithm)

copy all elements to new array (could be done with System.arraycopy())

COPYING LISTS

ALIAS

```
List<Room> rl1 = new ArrayList<Room>();
… // Insert elements in the list
List <Room> rl2 = rl1;
```

- Consequence
  - Changes in rl2 are reflected in rl1
- Example
  - Command rl2.set(2,r1) also changes the third element of rl1!

COPYING LISTS

CLONING (SHALLOW COPY)

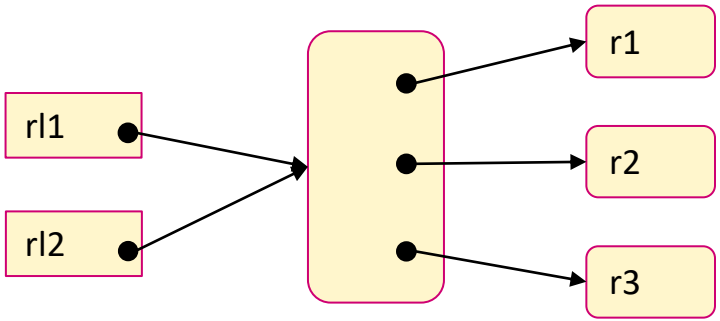
```
List<Room> rl1 = new ArrayList<Room>();
… // Insert elements in the list
List<Room> rl3 = (List<Room>) ((ArrayList<Room>) rooms).clone();
```

- Consequence
  - Changes in rl3 are not reflected in rl1
- Example
  - Command rl3.set(2,r1) does not change the third element of rl1!

COPYING LISTS

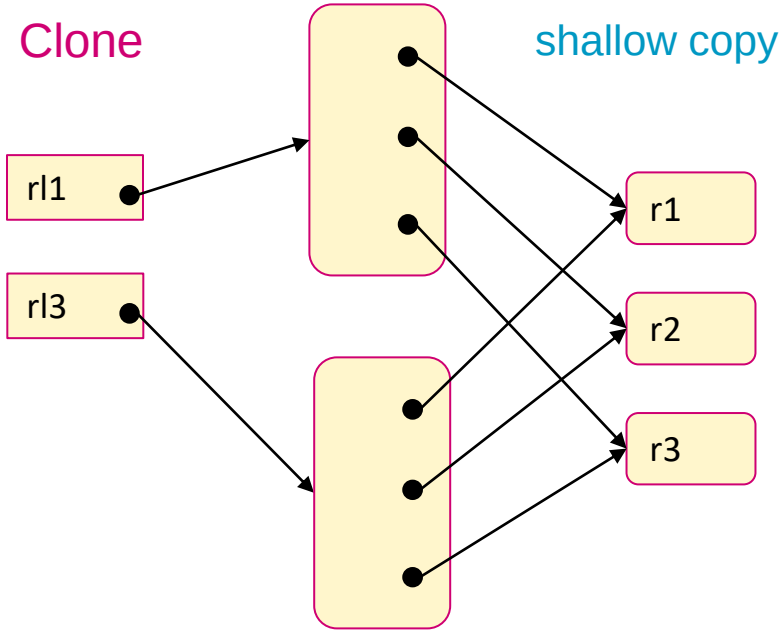
ALIAS VERSUS CLONING

Alias



Clone

shallow copy



COPYING LISTS

CLONING EXAMPLE

```
public Object clone() {
    SimpleArrayList<E> v = new SimpleArrayList<E>();
    for (int i = 0; i < this.size; i++)
        v.add((E) this.list[i]);
    return v;
}
```

cloneable object!

create new list

copy all elements to new list