HOOFDSTUK 16

Comparable – Comparator

```
java.lang
public interface Comparable<T>
 {
     int compareTo(T o);
 }
```

```
public class Student
{
    private String naam;
    private int studnr;
    //...
}
```

HoGent

```
public class Student implements Comparable<Student>
{
    private String naam;
    private int studnr;
    //...
    public int compareTo(Student andere)
    {
        return this.studnr - andere.studnr;
    }

    @Override
    public int hashCode()
    {
        return studnr;
    }
```

```
    @Override
    public boolean equals(Object obj)
    {
        if (this == obj)
            return true;
        if (obj == null)
            return false;
        if (getClass() != obj.getClass())
            return false;
        Student andere=(Student) obj;
            return this.studnr == andere.studnr;
    }

}
```


```
java.util
public interface Comparator<T>
 {
    int compare(T o1, T o2);
 }
```

```
class Vergelijknaam implements Comparator<Student>
{
      public int compare(Student s1,Student s2)
      {
        int result = s1.getNaam().compareTo(s2.getNaam());
        if(result == 0)
            result = s1.compareTo(s2);
        return result;
      }
}
```


Java 7: Klasse Objects

```
public class SomeClass
{
    private String dataone;
    private int datatwo;

    @Override
    public int hashCode()
    {
        return Objects.hash(dataone, datatwo);
    }
```

```
@Override
public boolean equals(Object obj) {
        if (this == obj)
            return true;
        if (obj == null)
            return false;
        if (getClass() != obj.getClass())
            return false;
        final SomeClass other = (SomeClass) obj;
        if (!Objects.equals(this.dataone, other.dataone)) {
            return false;
        }
        if (this.datatwo != other.datatwo) {
            return false;
        }
        return true;
    }
}
```