

The Optional class

Dealing with null

- Typical situation:

```
String s=someMethodThatReturnsAString();  
if(s!=null) {  
    s=s.toUpperCase();  
}
```

...

```
// later in the code:
```

```
if(s!=null) {  
    s=s+"/";  
}
```

Class Optional in package java.util

- The class Optional stores objects for you and tests if they are null before performing an operation.
- If the object was null, nothing happens

```
Optional<String> s=Optional.of(someMethodThatReturnsAString());  
s = s.map( String::toUpperCase );
```

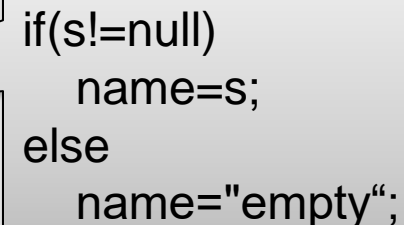
...

// later in the code:

```
s = s.map( (t)-> t+"/" );
```

- Of course, at the end you would like to have the object stored in the Optional:

```
String name = s.orElse("empty");
```



```
if(s!=null)  
    name=s;  
else  
    name="empty";
```