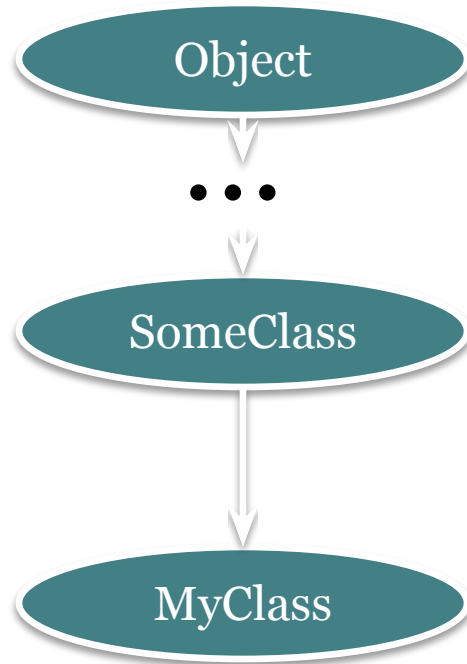
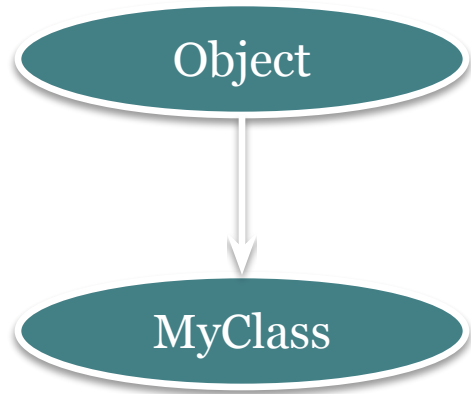


Class Object

A decorative horizontal line consisting of a thick teal bar on top, followed by a thin white bar, and then several thin, parallel teal and white lines extending to the right.

Общий родитель для всех классов



ОСНОВНЫЕ МЕТОДЫ

- toString();
- equals(Object o);
- hashCode();
- clone();
- finalize();
- notify();
- notifyAll();
- wait();
- getClass();

Обёртки для примитивов

```
Integer integer = 10;
```

```
Integer integer = Integer.valueOf(10);
```

```
String str = "";
```

```
Long; Character; Short; Float; Double; Boolean
```

```
BigInteger bi = BigInteger.valueOf(100);
```

```
BigInteger bi = new BigInteger(100);
```

Приведение типов

- явное

```
Object o = "";
```

```
String str = (String) o;
```

- неявное

```
int i = 10;
```

```
long n = i;
```

- object instanceof Class

final

- field
- method
- class

Generics

без:

```
List list = new ArrayList();  
list.add("string");  
String str = (String) list.get(0);
```

C:

```
List<String> list = new ArrayList<String>();  
//List<String> list = new ArrayList<>(); c 1.7
```

```
list.add("string");  
String str = list.get(0);
```

Не типизированный класс

```
public class Box {  
  
    private Object value;  
  
    public Object getValue() {  
        return value;  
    }  
  
    public void setValue(Object value) {  
        this.value = value;  
    }  
}
```


Типизированный класс

```
public class Box<T> {  
  
    private T value;  
  
    public T getValue() {  
        return value;  
    }  
  
    public void setValue(T value) {  
        this.value = value;  
    }  
}
```

Конкретизация

```
<T extends Comparable<T>>
```

```
T value1, value2;
```

```
value1.compareTo(value2);
```

Только на этапе компиляции!!!

Д/з

- Реализовать интерфейс `List<T>`
- Наследоваться от своей реализации и написать `SortedList<T extends Comparable<T>>`, переопределить метод `add` и `contains` (на бинарный поиск)