



Learning Object-Oriented Programming and Design with TDD

About super

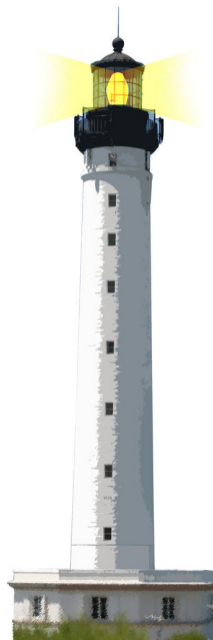
S. Ducasse

<http://stephane.ducasse.free.fr>



<http://www.pharo.org>

Extra



Super well which one...

Java is confusing at wish:

- super in accessing super same named superclass fields (Duh! Ugly)
 - **Syntactically:** `super.x`
- super in method to invoke overridden super method (super is the receiver see Lecture)
 - **Syntactically:** `super.method();`
- super in constructor
 - **Syntactically:** `super(x)`
 - first line represents the superclass and looks for constructor function based on argument
 - in any for the other line, represents the newly created object but changes lookup and in plain method.
- `static <T> void copy(List<? super T> dest, List<? extends T> src)`
 - to say any super type of T.
 - so readable



Super : accessing super same named superclass fields

```
class Vehicle {
    int maxSpeed = 120;
}

class Car extends Vehicle {
    int maxSpeed = 180;

    void display() {
        System.out.println("Maximum Speed: " + super.maxSpeed);
    }
}

class Test {
    public static void main(String[] args) {
        Car small = new Car();
        small.display();
    }
}
```

super in constructors

- means look in superclass and invoke the constructor with the same signature as the super call

```
public Meal {  
    public Meal(String flavour) {  
        this.flavour = flavour;  
    }  
}
```

```
public Crisps(String flavour, int quantity) {  
    super(flavour); // pass flavour to the super class constructor  
    this.quantity = quantity;  
}
```

super in constructor: II

```
public Meal {  
    public void setFlavour(String flavour) {  
        this.flavour = flavour;  
    }  
}
```

```
public Crisps(String flavour, int quantity) {  
    this.quantity = quantity;  
    super.setFlavour(flavour);  
}
```

Clearly not good style. But super here is not the superclass but the instance.



Subclass may want to access hidden superclass

```
public class Box extends Rectangle {  
    ...  
    public double getArea() {  
        return (super.getArea() + height * length + width * height) * 2;  
    }  
}
```

- `super.getArea()` executes the method `rectangle.getArea` on the box instance



Conclusion

Be precise :)



A course by Stéphane Ducasse
<http://stephane.ducasse.free.fr>

Reusing some parts of the Pharo Mocc by

Damien Cassou, Stéphane Ducasse, Luc Fabresse
<http://mocc.pharo.org>



Except where otherwise noted, this work is licensed under CC BY-NC-ND 3.0 France
<https://creativecommons.org/licenses/by-nc-nd/3.0/fr/>