



Java

Thinking in Objects

Key vocab:

- Class
- Object
 - Instance
- Attribute
- Value
 - Field
 - Variable
- Method



Java

Thinking in Objects

One of the hardest parts of getting started in programming is the process of visualising and breaking down a problem...

There are two basic approaches...



Java

Thinking in Objects

...Procedural

Where you look at what needs to be done in what order. (Languages like C work like this)

...Object Oriented

Where you look at what components make up the system, what we know about them, and *then* at what each of them do. (Java is this sort!)



Java

Thinking in Objects

The example the courseware using is a product ordering system. A **procedural approach** might look something like...

Create Catalogue

Create products

Create Customers

Add products to order

Bill customer

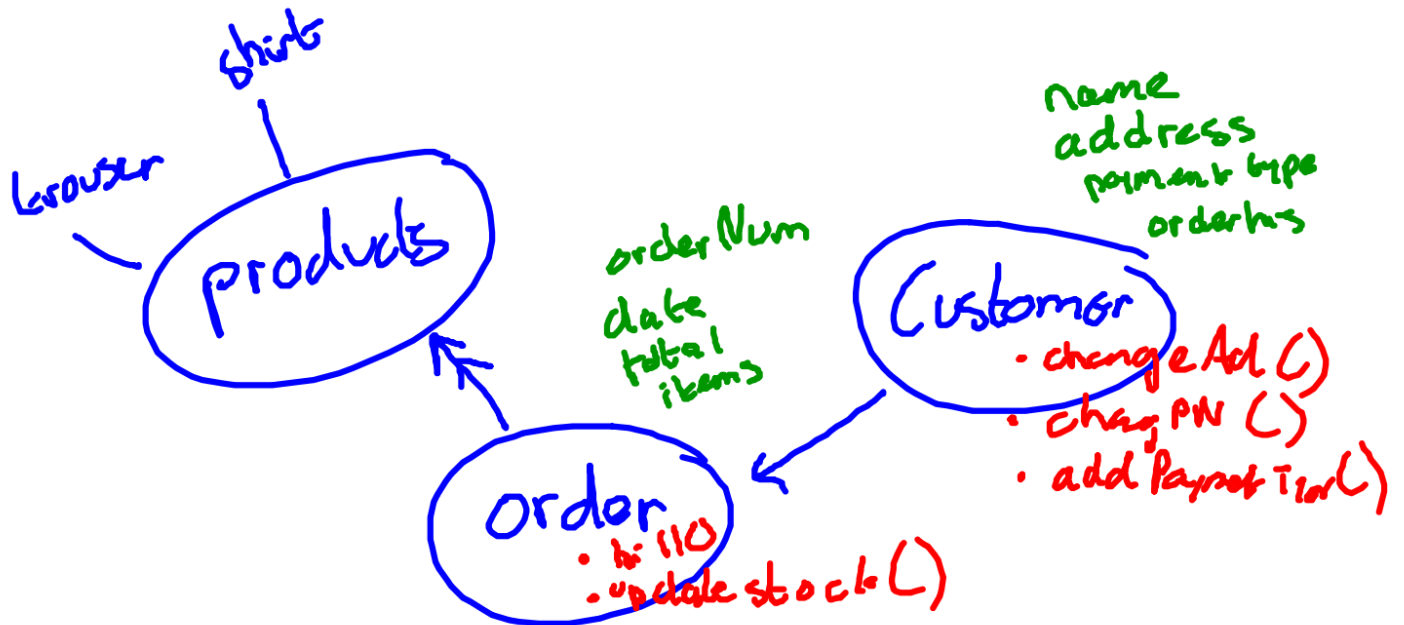
Update stock levels



Java

Thinking in Objects

What might an **object oriented approach** look like?





Java

Thinking in Objects

A **class** is the basic building block of Java.

"Shirt" might be a class for example. It defines everything we know about shirts (Attributes) and everything we can do with them (Methods) in our system.

Attributes

size [s, m, l]

colour []

price

material

stocklevel

Methods

discount()

getStockLevel()

addToOrder()



Thinking in Objects

An **object** is a particular **instance** of a **class** - for example this shirt here...



Attributes	Values
desc	Filson Antique Tin Cloth Shirt - Long Sleeve (For Men)
price	£ 47.62
rating	4
colours	4
sizes	S, XL, 2XL, 3XL
quantity	1

Item #6173A
Filson Antique Tin Cloth Shirt - Long Sleeve (For Men)
GBP 47.62 Compare at GBP 68.49 Save 30%
★★★★☆ 6 reviews Write a review
Colors Available: [Red] [Dark Blue] [Olive] [Grey] [Tan]
Sizes Available: [S] [XL] [2XL] [3XL]
Quantity: [-] 1 [+]
Add to Cart Add to Wish List

↓
add to Cart (1)

↓
add to Wish List



Java

Thinking in Objects

What other **classes** might make up our ordering system?



Java

Thinking in Objects

What other **classes** might make up our ordering system?

Shirt

```
productNum  
size  
colours[]  
sleeveLength  
price
```

```
addToCart()  
addToWishlist()  
compare()
```

Customer

Order



Java

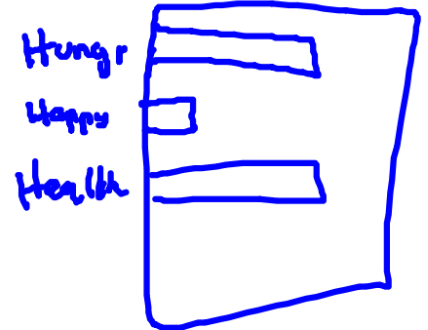
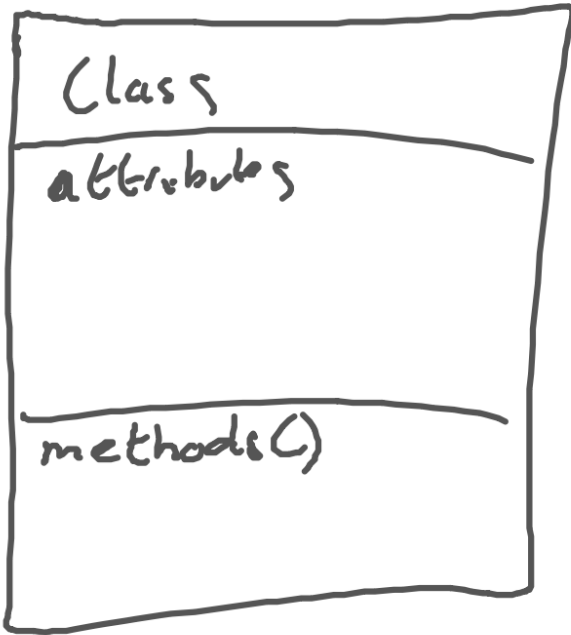
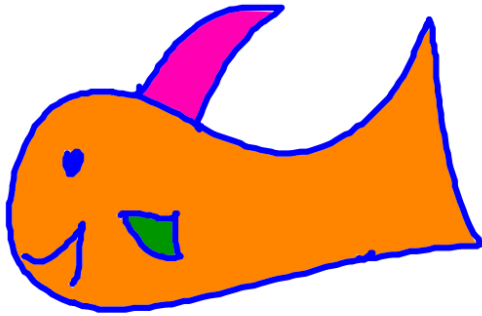
Thinking in Objects

Have a go at doing the same breakdown - Class, Attribute, Methods for some of the following:

House

Whale

Student





Thinking in Objects

Classes can be hierarchical. You can save yourself time and troubleshooting by planning carefully.

Shirt
<code>productNum</code> <code>size</code> <code>colours[]</code> <code>sleeveLength</code> <code>price</code>
<code>addToCart()</code> <code>addToWishlist()</code> <code>compare()</code>

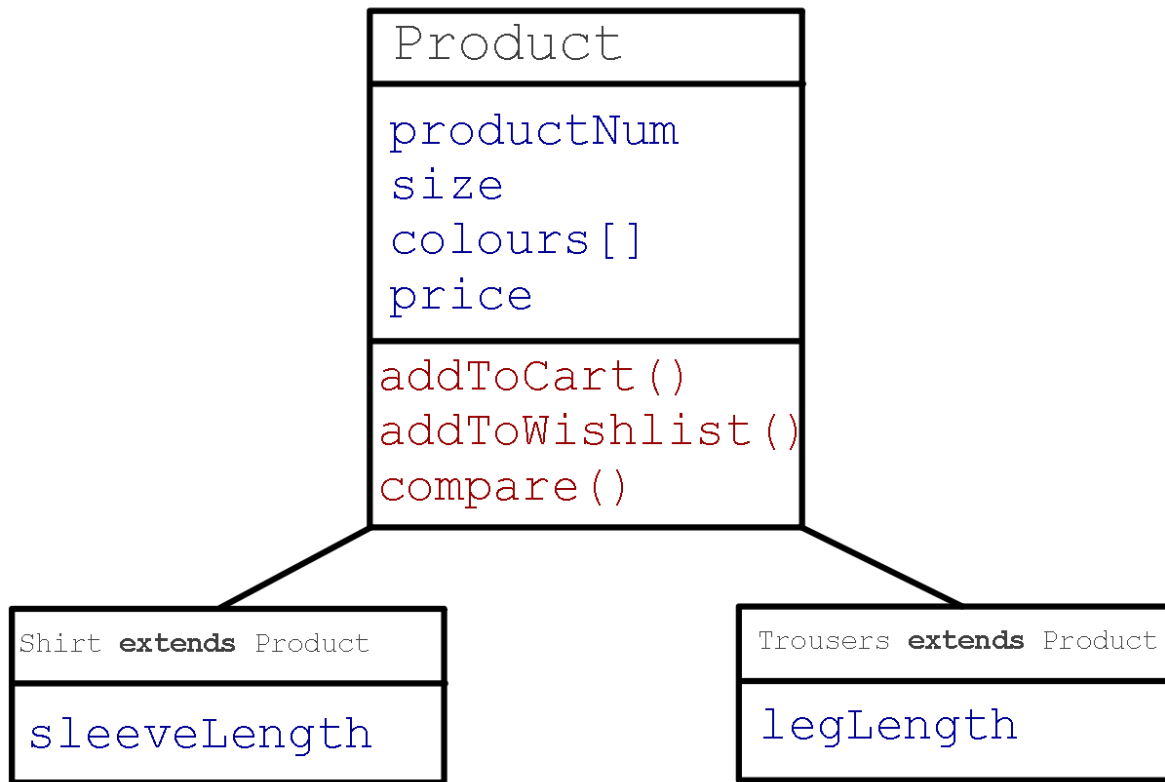
Trousers
<code>productNum</code> <code>size</code> <code>colours[]</code> <code>legLength</code> <code>price</code>
<code>addToCart()</code> <code>addToWishlist()</code> <code>compare()</code>

Can you see the duplication?



Java

Thinking in Objects



Greenfoot Working with Objects Java

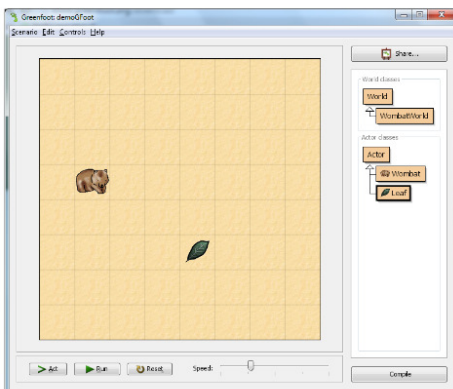
Download Greenfoot
version: 2.3.0

Requirements
Greenfoot requires the *Java 6* or *Java 7 JDK*. Note that you need the version marked *JDK*; not *JRE*, *Netbeans* or *Java EE*.

[Download JDK](#)

Installation Instructions can be found [here](#).

Windows	Mac OS X	Mac OS X
For XP, Vista, 7, 8	Needs at least 10.5 (Leopard)	Includes JDK 7 Needs at least 10.7.3 (Lion)
		
Ubuntu	Pure Java	Stand Alone
Supports Debian	For any OS	Includes BlueJ and JDK, runs on Windows from a USB stick
		



```
Leaf
Class Edit Tools Options
Compile Undo Cut Copy Paste Find... Close Source Code
import greenfoot.*; // (World, Actor, GreenfootImage, and Greenfoot)

/**
 * Leaf - a class for representing leaves.
 *
 * @author Michael Kolling
 * @version 1.0.1
 */
public class Leaf extends Actor
{
    public Leaf()
    {
    }
}
```

