

Forum:[Hack Forums](#)

Thread: [\[TUT\] - Java 2 \(Classes, methods and Objects\)](#)

terkitos - May 14, 2009, 3:11 am

Hi, I'm here of new and now I'm gonna try to explain you what's about classes and objects, what are and how are used for.

Classes, one class is the code between "class" and the last "}" (for example):

```
Code class LOL {  
    CODE HERE  
}
```

In the example there is a class named LOL. There's no more.

Inside the classes, there are methods, so a .jar file it's like this:

```
Code -JAR FILE-----|  
|           |  
| -CLASS FILE-----| | | |
||         | |  
|| -METHODS -----| | |  
||   CODE   | | |  
||         | | |  
||         | | |  
|_____|| |
```

(What hard have been to do the boxes Â¬Â¬)

That's how it's organized a .jar file (I think it's clean :D)

Method's syntax is like that :

```
Code (public/private) (static) (void/int/String/OBJECT) NAME (ARGUMENTS) {  
    CODE  
}
```

So let's write a little example of one class with three methods that they print something :D

```
Code class Example {  
    public void method1() {  
        System.out.println("Hi, I'm in the first method!");  
    }  
    public void method2() {  
        System.out.println("Hi, I'm in the second method!");  
    }  
    //And now let's try the INT return  
    public int method3() {  
        return 3; //returns the number 3  
    }  
    //All codes start from the MAIN method, so lets make one to do this working :D  
    public static void main(String[] args) {
```

```
//String[] args is a variable that contains the arguments line, splitted by spaces so... put it but
don't use it until you know what it is ^^
method1();    //Calls method1 and print it
method2();    //Calls method2 and print it
int number = method3(); //Set the var "number" with the return of method3
System.out.println("The last method is: "+number); //Method3 printer :P
}
}
Easy!, no?
```

Now, as you know what is a method and what's a class, let's explain what's an object, this is hard when you see this kind of programming first time hehe

You have to think an object like as an special kind of VARIABLE, like INT, DOUBLE... and so on... but it has special "parameters" an object can contain one or more variables to set the options.

Think that there is a CLASS named CAR, so a car has doors, tyres, seats, etc... then, the number of each, would be the "options" that I said before.

Let's make an example of a class named CAR

```
Code class Car {
    int tyres, doors, seats; //initializing variables to set a value later.
    Car (int tyres, int doors, int seats) {
        this.tyres = tyres;
        this.doors = doors;
        this.seats = seats;
    }
}
```

This would be an example, now let's explain it :D

First is to initialize variables and there is a method named Car, this is a BUILDER to set the variables with some value, that they are given when you create the object. Now let's complete the example with an entire example of printing some values of any object. (I'll use the car's example)

```
Code class Car {
    protected int tyres, doors, seats; //initializing variables to set a value later. and protected for not
to be changed out this method
    Car (int tyres, int doors, int seats) {
        this.tyres = tyres;
        this.doors = doors;
        this.seats = seats;
    }
    public int getTyres() { return tyres; } //get the number of tyres, doors or seats
    public int getDoors() { return doors; }
    public int getSeats() { return seats; }
    public int setTyres(int tyres) { this.tyres = tyres; } //change the number of tyres
}

class Main {
    public static void main(String[] args) {
        Car ferrari = new Car(4,5,7); //first, number of tyres, second, doors and third seats
        //now let's gonna print the number of each
```

```
System.out.println("Tyres: "+ferrari.getTyres());
System.out.println("Doors: "+ferrari.getDoors());
System.out.println("Seats: "+ferrari.getSeats());
ferrari.setTyres(100); //change the number of tyres
System.out.println("Tyres after change: "+ferrari.getTyres()); //see that the tyres has been
changed
    }
}
END!!!!!!
```

I'm tired of writing, and thinking what to write

I hope you have understood something I have just written and tell me the mistakes I have had, I'll edit asap.

I'll continue doing tutorials! (First was JAVA LOOPS)

Cobra Strike - May 14, 2009, 3:37 am

Wow nice tutorial. Great job.

terkitos - May 14, 2009, 8:28 am

Thank you man :D

<short :(>

TheHackz - May 17, 2009, 1:39 pm

thats a really good tut dude

andie12serador - May 27, 2011, 1:31 pm

LOL at the last Code. I thought you'll gonna explain it more deeper at that part because its the main part of the explanations that to be done but you just said, IM TIRED. hahah

Anyways, Nice tuts, it was very easy to understand.