

Παρουσίαση Java



Νικόλας Κοξένογλου

Java



Java



Java

- Τι γλώσσα είναι ;
- Γιατί αξίζει να την ξέρει κανείς;
- Τι διαφορές έχει με τη C;
- Γιατί Java και όχι Python;
- Γιατί δεν παίζω Skyrim;
- Γιατί είμαι εδώ;

Πώς Γράφω Java

- Γενικές απαιτήσεις είναι να έχετε το JDK, Java Development kit, και ένα υπολογιστή ονς.
- Για Windows και Apple OS:
<http://www.oracle.com/technetwork/articles/javase/index-jsp-138363.html>
- Για Linux γράψτε στο terminal:
- `$ sudo apt-get install openjdk-*-jdk (* = 7, πιο μελλοντικά 8)`
- Ένα text editor που σας βολεύει (π.χ. Vim, Kate και Notepad++) ή ένα IDE όπως το Eclipse.
- Για compile γράφουμε το όνομα του αρχείου στο terminal με javac στην αρχή π.χ. `$ javac Start.java`
- Για εκτέλεση του προγράμματος γράφουμε java και το class αρχείο που εμφανίστηκε το οποίο περιέχει την main συνάρτηση π.χ. `$ java Start`

Καλές Πρακτικές

- Ισχύουν οι πλείστες γενικές πρακτικές από τη C όσο αναφορά :
 - Καλή δομή.
 - Global variables.
 - Σχόλια.
 - Χτίσιμο βήμα βήμα.
- Χρησιμοποιείτε αφηρημένες κλάσεις για γενικό κώδικα έτσι ώστε να είναι πιο εύκολο να γράψετε objects και methods τα οποία κάνουν κάτι πιο συγκεκριμένο.
- Και αλλά πολλά:
<https://sites.google.com/site/sureshdevang/the-do-s-and-don-ts-of-java-programming>

Μια μικρή σύγκριση

C – Hello.c

```
#include<stdio.h>
int main(int argc, char *argv[]){

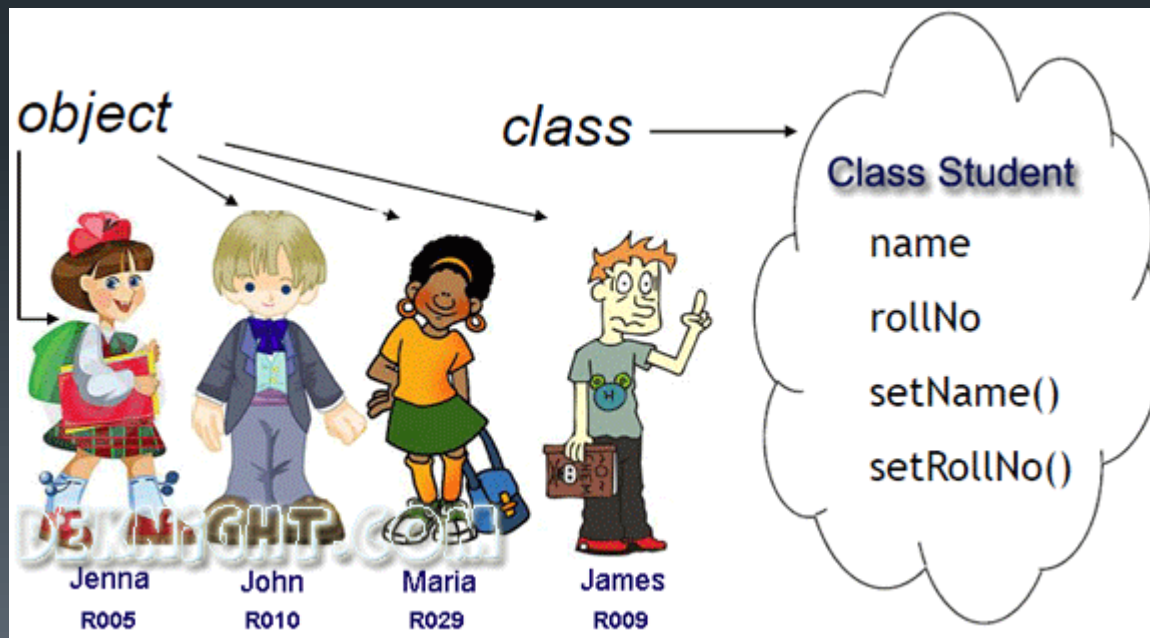
    printf("Hello World\n");
    return(0);
}
```

Java – Hello.java

```
public class hello{
    public static void main(String args[]) {
        //"out" belongs to class "System" and is of type "PrintStream"
        System.out.println("Hello World");
    }
}
```

Java

- Java είναι μια αντικειμενοστραφής βασισμένη σε κλάσεις γλώσσα.
- Οι κλάσεις αποτελούν γενικά χαρακτηριστικά τα οποία μπορούν να έχουν τα objects.
- Τα objects μπορούν να εκτελέσουν διάφορα methods όπως εκτύπωση στην οθόνη, εκτέλεση GUI και πολλές άλλες.



Λίγα για τα Objects

Dog.java

```
public class Dog{  
  
    String breed;  
    //Constructor gives attributes to object  
    Dog(String b){  
        breed = b; //pass attribute to local string  
    }  
    void printName(){  
        System.out.println(breed);  
    }  
    //Method like a function but in an object  
    void bark(){  
        System.out.println("Woof Woof Woof");  
    }  
    void dig(){  
        System.out.println("*I am digging*");  
    }  
}
```

Λίγα για τα Objects

Start.java

```
import java.util.Scanner; //Just like #include, Object is not in the same default package or directory
public class Start{
    //When you compile the compiler looks for main function
    public static void main(String args[]) {
        Scanner input = new Scanner(System.in);
        String action;

        //Variable deceleration like int
        Dog Hound; //Object is in the same default package or directory
        Dog Mounten;

        //Creating new object
        Hound = new Dog("Sendor"); // Passing attribute to constructor
        Mounten = new Dog("Gregor");

        Hound.printName();
        Mounten.printName();

        System.out.print("Do you want to Bark or Dig ");
        action = input.nextLine();
        if (action.equals("Dig")){
            //Use the dig method
            Hound.dig();
        }else if(action.equals("Bark")){
            //Use the bark method
            Hound.bark();
        }
    }
}
```

Λίγα για Classes

animal.java

```
public class animal{  
  
    //Animal>Dog>Breed  
    //Common attributes> Specific> More specific  
    void IsOnEarth(){  
        //Creating class animal so we remove repetition  
        //with generic object attributes  
        System.out.println("Yep it does on earth");  
    }  
}
```

Dog2.java

```
public class Dog2 extends animal{  
    //extend dog class with animal attributes  
    //Class subclass extends superclass  
  
    String breed;  
    //Constructor gives attributes to object  
    Dog2(String b){  
        breed = b; //pass attribute to local string  
    }  
  
    void IsOnEarth(){  
        super.IsOnEarth();  
    }  
    void printName(){  
        System.out.println(breed);  
    }  
    //Method like a function bun in an object  
    void bark(){  
        System.out.println("Woof Woof Woof");  
    }  
  
    void dig(){  
        System.out.println("*I am digging*");  
    }  
}
```

Λίγα για Classes

Start2.java

```
import java.util.Scanner; //Just like #include, Object is not in the same default package or directory
public class Start2{

    //When you compile the compiler looks for main function
    public static void main(String args[]) {

        //Variable deceleration like int
        Dog2 Hound; //Object is in the same default package or directory
        Dog2 MounTen;

        //Creating new object
        Hound = new Dog2("Sendor"); // Passing attribute to constructor
        MounTen = new Dog2("Gregor");

        Hound.IsOnEarth();
    }
}
```

Λίγα για Interfaces

CharacterInterface.java

```
public interface CharacterInterface{  
    //interface allows multiple programmers to work on the same project  
    //interface indicates the behaviour of objects  
  
    //generic methods to be implemented by other programmers  
    //with indicated return types  
  
    String getHit();  
    int damage(int curHealth, int damage);  
}
```

Stannis.java

```
//Extends just adds methods with implement you must implement everything  
public class Stanis implements CharacterInterface{  
    //object specific behaviour  
    public String getHit(){  
        String StanisGotHit = "Don't hit the King";  
        return StanisGotHit;  
    }  
    public int damage(int curHealth, int damage){  
        int newHealth = curHealth - damage;  
        if (newHealth > 0 ){  
            return newHealth;  
        }else{  
            return 0;  
        }  
    }  
}
```

Λίγα για Interfaces

Davos.java

```
//Extends just adds methods with implement you must implement everything
public class Davos implements CharacterInterface{

    //object specific behaviour
    public String getHit(){
        String DavosGotHit = "The onion knight is Strong";
        return DavosGotHit;
    }
    public int damage(int curHealth, int damage){
        int newHealth = curHealth - damage;
        if (newHealth > 0 ){
            return newHealth;
        }else{
            return 0;
        }
    }
}
```

Λίγα για Interfaces

Start3.java

```
public class Start3 {  
  
    public static void main(String args[]){  
        //Setting new objects  
        Stanis S = new Stanis();  
        Davos D = new Davos();  
  
        int Health;  
        String Words;  
        //using the objects methods  
        Words = S.getHit();  
        Health = S.damage(200, 50);  
        System.out.println(Words + " " + Health);  
  
        Words = D.getHit();  
        Health = D.damage(300, 100);  
        System.out.println(Words + " " + Health);  
  
    }  
}
```

The End

Και τώρα Workshop!



```
if (not calm) {  
    keep calm}  
  
else {  
    code on}
```