**Autoboxing**

* Groovy tot foloseste datele primitive, de ex

def a = 10

e o primitiva, dar el face mereu boxing in wrapper class cand e necesar. De ex:

def a = 10.toString()

El va face automat conversia din int in Integer

class Foo {

static int i

}

assert Foo.class.getDeclaredField('i').type == int.class

assert Foo.i.class != int.class && Foo.i.class == Integer.class

**Classes**

* Putem avea intr-un fisier mai multe clase publice.
* Mereu oricum se creaza o clasa cu main() identica cu numele la fisier
* Iata de ce, nu putem crea si o clasa cu acelasi nume ca a fisierului si script in fisier:
* class MyClass{  
   static void main(String[] args) {  
   println("Hello")  
   }  
  }  
  println("what")



* **Inner class pot accesa membrii privati din clasa parinte**

class Outer {

private String privateStr

def callInnerMethod() {

new Inner().methodA()

}

class Inner {

def methodA() {

println "${privateStr}."

}

}

}

**Constructor**

* Putem crea obiecte folosind un constructor definit prin 3 metode:

1. Folosind new
2. Folosind un array si as
3. Folosind doar un array

Cand folosim un array, se vor lua elementele sale si se vor pune automat in constructor in ordinea lor

Test obj1 = new Test(1,2);  
Test obj2 = **[**1,2**]** as Test;  
Test obj3 = **[**1,2**]**

toate 3 sunt identice

**Map Constructor**

class Test{  
 int a;  
 int b;  
  
}  
Test obj1 = new Test(a:1,b:2);

* Grooovy, daca nu avem niciun constructor, pe langa cel default, mai creaza unul ce ia un Map<> unde key sunt numele la variabile si value valorile lor si le atribuite
* Dcaa definim noi vreun constructor, macar si default, map constructor nu mai e generat
* Putem si asa face, e tot aceeasi:
* Test obj1 = new Test([a:1,b:2]);

**Method**

* Nu putem trimite parametrii cu nume la o metoda:
* def method(a,b,c){  
    
  }  
  method(a:100,b:5,c:2);



* **Daca primul sau ultimul parametru e un map, putem evita []**
* def method(Map map, a){  
    
  }  
  method(a:100,b:5,c:2, 100);

**e echivalent cu**

method([a:100,b:5,c:2], 100)

* **Metodele suporta default value pentru parametrii:**

def method(Map map, a=100){  
  
}

* In groovym ... sau [] ca ultim argument in metoda e tot aceeasi:

def method(int... arg){  
 return arg;  
}  
println(method(1,2,3,4))

e similar cu

def method(int[] arg){  
 return arg;  
}  
println(method(1,2,3,4))

* In caz de supraincarcare, metoda cu un parametru e aleasa:
* def method(int[] arg){  
   return "array";  
  }  
  def method(int arg){  
   return "single parameter"  
  }  
  println(method(1))

single parameter