**keywords**

* **as**

- pentru a face casting

println("5" as Integer)

- pentru a da un alias la un import

**import** java.lang.Math as m

**class** **GroovyAsKeyword** {

**static** **void** **main**(String[] args) {

println m.sqrt(16);

}

}

* **in**

Verifica daca vreun elemente este intr-o colectie sau array

println(5 in [1,5,6,2,1])

* **trait**

Este ca o interfata, doar ca toate metodele sunt in mod automat default

Nu permite sa punem metode protected sau package private

Permite doar public si private

Fieldurile nu mai sunt statice, ci simple fields

Un trait poate extinde alt train cu extends

O clasa poate implementa oricate trait

Daca o clasa implementeaza mai multe trait ce au o metoda identica, mereu se va la in calcul metoda de la ultima trait declarata in implements

Poate avea metode abstracte

**Numbers**

* Daca folosim def, groovy va incerca sa in tip de numar in dependenta de numarul oferit, de ex

def a = 100;

e Integer

def a = 100.0

e double

* Pe langa a face cast, putem folosi si caractere speciale pentru a specifica tipul numarului:

| **Type** | **Suffix** |
| --- | --- |
| BigInteger | G or g |
| Long | L or l |
| Integer | I or i |
| BigDecimal | G or g |
| Double | D or d |
| Float | F or f |

**Data structures**

* Groovy nu are propriile collectii de date
* [] creaza automat un ArrayList
* println([1,2,3] instanceof ArrayList)

true

* Lista poate contine si obiecte de diferite tipuri

[1,"a",true]

* Daca vrem sa cream o LinkedList, nu Array, folosim as LinkedList

[1,"a",true] as LinkedList

* Putem accesa si modifica elemente folosind [], chiar daca nu e un array, ci lista
* Putem folosi si indexi negativi, ca in JS
* def list = [1,2,3,4]  
    
  list[0] = 100  
  list [-1] = 400
* Operatorul << adauga iteme la urma
* def list = [1,2,3,4]  
    
  list << 1000;
* Putem folosi si .. ca interval:
* def list = [1,2,3,4,5]  
    
  list[1..3] = [200,300,400]
* [] nu mai creaza array, desi, putem totusi sa cream array daca vrem:

1. Definim variabila de tip []

int[] array = [1,2,3,4]  
println(array instanceof int[])

1. Folosim as

def array = [1,2,3,4] as int[]  
println(array instanceof int[])

* **Map**:

- putem crea automat Map cu [key:value]. Se creaza automat un LinkedHashMap

def map = ['key1' : 1, 'key2' : 2]

- putem acces un element fie cu get(), fie cu [key]

println(map['key1'])

- putem si adauga sau modifica asa un element:

def map = ['key1' : 1, 'key2' : 2]  
map['key1'] = 999  
map['key3'] = 3  
println(map)

* Daca avem un string dintr-un cuvant, putem omite ‘’, oricum el va fi luat ca string:

def map = [key1 : 1, key2 : 2]

* Dar nu il putem accesa asa:
* map[key1] = 999



* **Putem asa map.string:**
* map.key1 = 999  
  map.key3 = 3