

INTRO

Vediamo come è implementata in Ruby la programmazione orientata agli oggetti. Scriviamo un semplice esempio supponendo di dover programmare un videogame di guerra dove tra le altre cose sono presenti dei carri armati e dei camion per il trasporto dei soldati. Rappresentiamo questi oggetti in Ruby:

```
class CarroArmato
  def initialize
    puts "Sono un nuovo carro armato"
  end
end

class Camion
  def initialize
    puts "Sono un nuovo camion"
  end
end
```

Vediamo in ordine cosa abbiamo fatto. Innanzitutto abbiamo creato la classe usando la parola chiave `class` seguita dal nome e abbiamo terminato il codice relativo alla nostra classe con la parola chiave `end`. Abbiamo quindi creato il metodo `initialize` che viene chiamato quando istanziamo un nuovo oggetto con il metodo `new`. `initialize` è quello che in OOP si dice un **costruttore** che viene invocato ogni volta che si crea un nuovo oggetto. Alcuni linguaggi orientati agli oggetti come il C++ contemplano anche il metodo distruttore della classe che, invocato al momento del rilascio dell'istanza di un oggetto, si occupa tra le altre cose di de-allocare la memoria impegnata dall'oggetto. Nei linguaggi moderni, come Ruby, Java o C#, le attività generalmente svolte dal distruttore vengono svolte da sistemi di garbage collection, nel caso di Ruby è l'interprete a farsene carico.