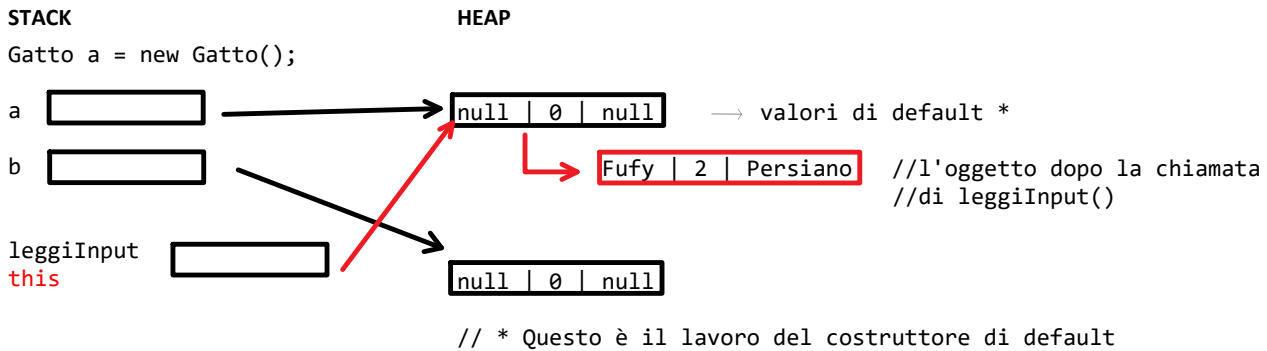


Lezione 6-03

lunedì 6 marzo 2023 11:24

```
public void leggiInput(){
// metodo dinamico (non c'è la parola chiave static e c'è il parametro this)
    System.out.println(" nome = ");
    this.nome = tastiera.nextLine();
    System.out.println(" razza = ");
    this.razza = tastiera.nextLine();
    System.out.println("anni = ");
    this.anni = tastiera.nextLine();
}
```



```
Gatto a = new Gatto();
Gatto b = new Gatto();
```

a.leggiInput(); // il metodo leggiInput modifica i campi dell'oggetto Gatto puntato da a

Il parametro `this` prende come puntatore il puntatore con cui il metodo è stato chiamato (in questo caso `this` prenderà come puntatore, il puntatore di a)

Altro esempio

```
public int getEtaInAnniUmani(){
    if(anni <= 2)
        return anni*11;
    else
        return 22 + (anni-2)*5;
}
```

L'obiettivo di questo metodo è quello di restituire gli anni del gatto in anni umani.

In questo esempio di codice, il parametro `this` non è esplicitato però c'è (è un modo per scrivere le cose più velocemente). Ci sono dei casi in cui bisogna specificare, nel caso in cui si può avere ambiguità con i nomi dei campi.

Metodo toString()

```
public String toString(){
    return "nome = " + nome + "\nrazza = " + razza + "\nanni = " + anni;
    // il parametro this è implicito
}
```

```
Gatto tramot = new Gatto()
System.out.println(tramot); //la chiamata del metodo toString() è implicito
```

Il metodo `toString()` è un nome notevole che ricorre in tutte le librerie Java
Con il codice così com'è stampa questo:

```
nome = null
razza = null
anni = 0
```

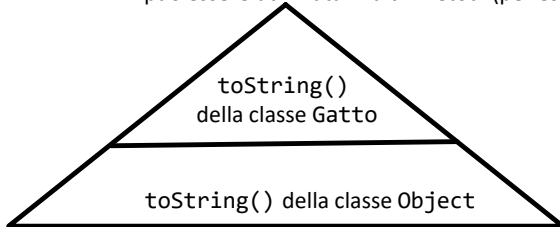
ma senza aver definito il metodo `toString()`?

```
Gatto@3fee733d
```

Stampa il nome del tipo dell'oggetto e il numero dell'indirizzo dell'oggetto Gatto. Questo perché il metodo `toString()` del tipo `Object` primitivo dato da Java stampa il nome dell'oggetto e il suo indirizzo (questo per la gerarchia degli oggetti di Java). Se il metodo `toString()` è stato definito dall'oggetto, la priorità la prende il metodo `toString()` dell'oggetto Gatto.

Invece il metodo `toString()` viene chiamato direttamente dal metodo `println()`. Questo spiega il motivo per cui non c'è il bisogno di chiamare esplicitamente il metodo `toString()`

Importante: il metodo `toString()` non stampa l'oggetto ma trasforma l'oggetto in una stringa e questa stringa può essere utilizzata in altri metodi (per esempio nel metodo `println()`)



In pochissime parole `toString()` della classe Gatto batte `toString()` della classe `Object`

Metodi `get()` e `set()`

```
class Specie {  
    private String nome;  
    private int popolazione;  
    private double tassoCrescita;  
}
```

Per ricevere e modificare i campi privati di un oggetto si usano rispettivamente i metodi `get()` e `set()` creati direttamente dal programmatore dentro la classe `Specie`

```
public void setSpecie(String n, int p, double t){  
    nome = n;  
    if(p<0)  
        System.out.println("Valori negativi popolazione non accettati");  
    else popolazione = p;  
    tassoCrescita = t;  
}
```

```
public String getNome(){return nome}  
public int getPopolazione(){return popolazione}  
public String getTassoCrescita(){return tassoCrescita}
```