

La Ricorsione



Prima Parte

Giselda De Vita

© Giselda De Vita - 2013

Una favola

Mia nonna mi raccontava la favola di un re cattivissimo che ogni notte uccideva una sua serva.

Una notte una serva furba, prima di essere uccisa, chiese al suo re di potergli raccontare una favola. Lui acconsentì, assicurandole che non avrebbe visto le prime luci dell'alba. La serva con voce calma si mise a raccontare :

*C'era una volta un re, seduto sul sofà che disse alla sua serva:
«Raccontami una fiaba.» E la serva incominciò:*

*C'era una volta un re, seduto sul sofà che disse alla sua serva:
«Raccontami una fiaba.» E la serva incominciò:*

*C'era una volta un re, seduto sul sofà che disse alla sua serva:
«Raccontami una fiaba.» E la serva incominciò:*

*C'era una volta un re, seduto sul sofà che disse alla sua serva:
«Raccontami una fiaba.» E la serva incominciò:*



La favola infinita

Il re cattivo si addormentò, mentre la serva continuava a sussurrare la sua favola infinita.

Quando si intravidero le prime luci del giorno, la serva ebbe la certezza che la sua vita era salva e con un sorriso concluse:

C'era una volta un re, seduto sul sofà che disse alla sua serva: «Raccontami una fiaba.» E la serva tanto raccontò che il re si addormentò!

Anche la notte successiva e un'altra notte ancora si ripeté la stessa cosa. Il re si addormentava mentre la serva raccontava la sua favola infinita. E tanto la raccontò finché non il re non la sposò.



Ricorsione

A parte il discutibile lieto fine, mia nonna, rivisitando e modificando le favole antichissime di mille e una notte, mi aveva spiegato meglio di chiunque altro il concetto di **RICORSIONE**.



Definizione

Una funzione matematica è definita *ricorsivamente* quando nella sua definizione compare un riferimento a se stessa.

La *ricorsione* consiste nella possibilità di definire una funzione in termini di se stessa.

Definizione ricorsiva

Una definizione ricorsiva consiste in due parti:

Caso base: caso sufficientemente semplice da poter essere risolto in maniera diretta senza ricorsione;

Caso induttivo/ricorsivo: non essendo direttamente risolvibile, il caso induttivo si presta ad essere risolto tramite un procedimento ricorsivo

Funzione Ricorsiva

La struttura di una funzione ricorsiva:

if (è il caso base)

risolvilo

else

usa la ricorsione su dati ridotti

Ricorsione esempio base

Una funzione che contiene al suo interno una attivazione di se stessa è detta ricorsiva.

Esempio in JAVA:

```
static void fRic(int a){
    if (a==0)
        System.out.println("fRic("+a+"): caso base");
    else {
        System.out.println("sono in fRic("+a+")");
        System.out.println("chiamata ricorsiva di fRic("+a-1+")");
        fRic(a-1);
        System.out.println ("fRic("+a+"): ho finito");
    }
}
```

Ricorsione esempio base 2

Una funzione main() che richiama la nostra fRic:

```
public static void main(String[] args) {  
    fRic(3);  
}
```

Produce il
seguente Output:

```
sono in fRic(3)  
chiamata ricorsiva di fRic(2)  
sono in fRic(2)  
chiamata ricorsiva di fRic(1)  
sono in fRic(1)  
chiamata ricorsiva di fRic(0)  
fRic(0): caso base  
fRic(1): ho finito  
fRic(2): ho finito  
fRic(3): ho finito
```

La ricorsione

- ❖ Ad ogni chiamata si semplificano/riducono i dati, così ad un certo punto si arriva ad uno dei casi di base
- ❖ Quando la funzione chiama se stessa, sospende la sua esecuzione per eseguire la nuova chiamata
- ❖ L'esecuzione riprende quando la chiamata interna a se stessa termina
- ❖ La sequenza di chiamate ricorsive termina quando quella più interna incontra uno dei casi di base
- ❖ Ogni chiamata alloca sullo stack, in stack frame diversi, nuove istanze dei parametri e delle variabili locali

Esempio di definizione ricorsiva

La funzione che definisce l'elevamento di un numero x ad una potenza non negativa n è un esempio di funzione ricorsiva:

$$x^n = \begin{cases} 1 & \text{se } n = 0 \text{ (caso base)} \\ x \cdot x^{n-1} & \text{se } n > 0 \text{ (caso ricorsivo)} \end{cases}$$

Proviamo a implementare questa funzione in Java e analizziamo come avviene un'invocazione ricorsiva.

Implementazione

Prevedere sempre un main() che effettua la prima chiamata alla funzione ricorsiva.

```
public static void main(String[] args) {  
    int y = potenza(4,3);  
}  
  
static int potenza(int x, int n){  
    if (n==0)  
        return 1;  
    else  
        return x*potenza(x,n-1);  
}
```

Analizziamo l'implementazione

```
public static void main(String[] args) {  
  
    int y = potenza(4,3);  
  
}  
  
static int potenza(int x, int n){  
    if (n==0)  
        return 1;  
    else  
        return x*potenza(x,n-1);  
}
```

Caso base

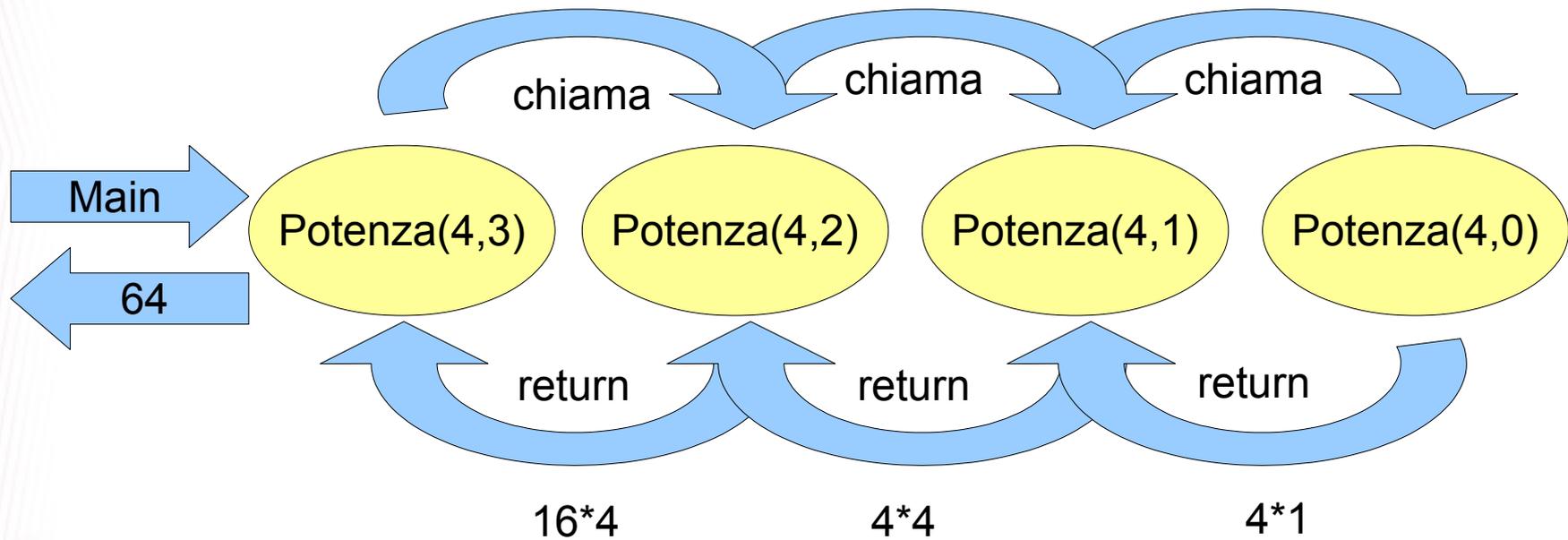
Semplificazione
del problema

Esempio

La chiamata a *potenza* ($x, n-1$) chiede alla funzione *potenza* di risolvere un problema più semplice di quello iniziale, il valore dell'esponente più basso, ma è sempre lo stesso problema.

La funzione continua a chiamare se stessa fino a raggiungere il caso di base che sa risolvere immediatamente.

Risultato



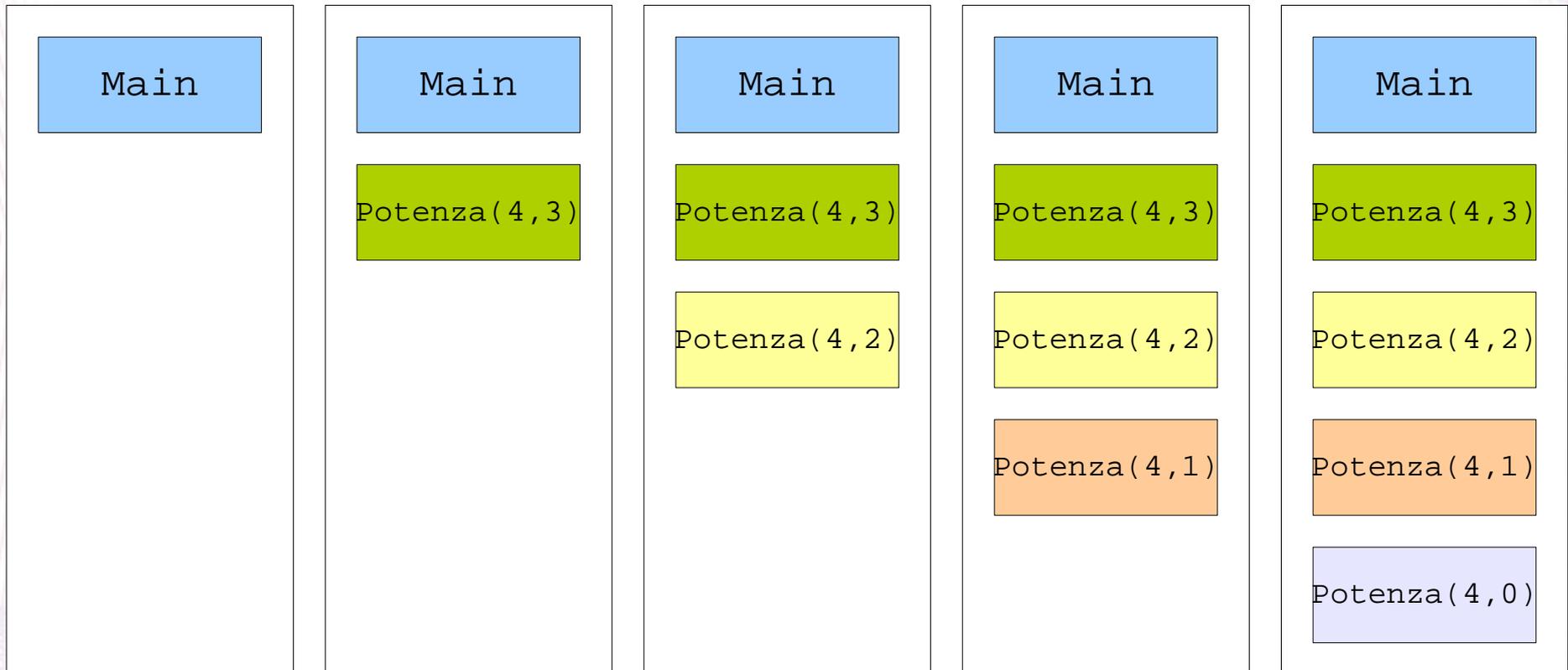
In memoria

Il main
chiama
Potenza(4,3)

Potenza(4,3)
chiama
Potenza(4,2)

Potenza(4,2)
chiama
Potenza(4,1)

Potenza(4,1)
chiama
Potenza(4,0)



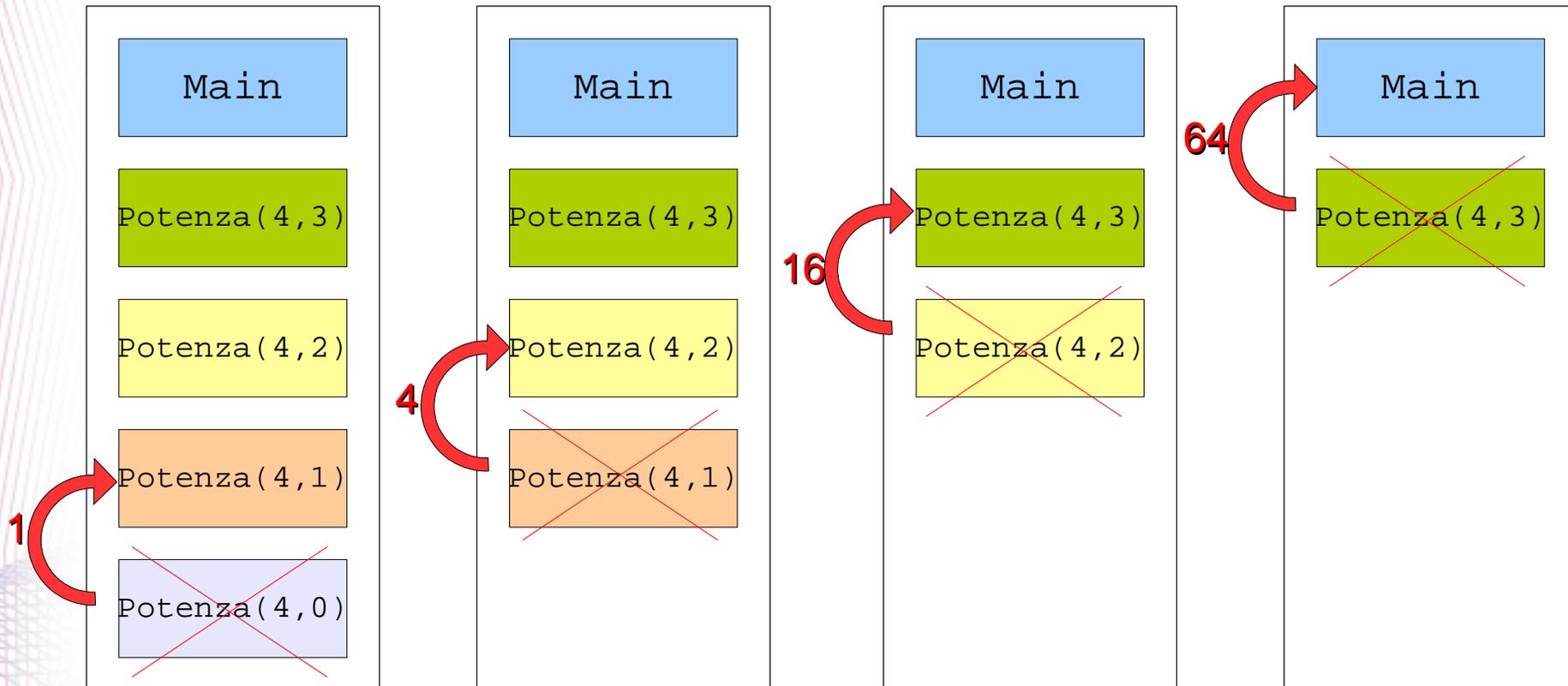
In memoria

Potenza(4,0)
termina e
restituisce il valore
1

Potenza(4,1)
termina e
restituisce il valore
 $4 * 1 = 4$

Potenza(4,2)
termina e
restituisce il valore
 $4 * 4 = 16$

Potenza(4,3)
termina e
restituisce il valore
 $4 * 16 = 64$



Un altro esempio

Per capire ancora meglio l'ordine delle chiamate ricorsive e il rispettivo risultato, vediamo l'esempio della funzione *inverti una stringa*.

In questo caso, la definizione della funzione 'inverti una stringa' non è nativamente ricorsiva, ma sfrutteremo le chiamate ricorsive per incrementare la posizione e per poi far leggere la stringa a ritroso.

Inverti Stringa

```
public static void main(String[] args) throws IOException {
    System.out.print("\nInserisci stringa: ");

    BufferedReader in = new BufferedReader(new InputStreamReader(System.in));
    String s;
    if ((s = in.readLine()) != null && s.length() != 0)
    {
        String invertita = inverti(s,0);
        System.out.println("String invertita: " + invertita);
    }
}
```

Il metodo main leggerà l'input dallo Standard Input e chiamerà la funzione *inverti*(stringa, 0)

Inverti Stringa

```
private static String inverti(String s, int pos ) {  
    if(pos < s.length())  
    {  
        return inverti(s, pos+1) + s.charAt(pos);  
    }  
    else  
        return "";  
}
```

La funzione ricorsiva *inverti* chiama se stessa, avanzando la posizione del carattere, finché non si arriva alla fine della stringa (posizione == s.length()) che rappresenta il caso base e restituisce una stringa vuota.

Successivamente, risolvendo le chiamate ricorsive, viene prodotto in output la Stringa invertita.

Output:

```
Inserisci stringa: prova funzione Ricorsiva!  
String invertita: !avisrociR enoiznuf avorp
```

Stringa Palindroma

Una sequenza è detta palindroma se letta da destra a sinistra è identica a quella letta da sinistra a destra (es. Radar, Anna).
E così viene anche definita una frase palindroma, salvo gli spazi:

Amo Roma, Aromi di mora

La definizione ricorsiva:

Caso base: Una stringa con 0 o 1 carattere è **palindroma**

Una stringa è palindroma se:

- ◆ il primo carattere è uguale all'ultimo
- ◆ la stringa fatta di tutti i caratteri meno il primo e l'ultimo è **palidroma**

Implementazione

Il main() che chiama il metodo isPalindroma

```
BufferedReader in = new BufferedReader(new InputStreamReader(System.in));
String s;
if ((s = in.readLine()) != null && s.length() != 0)
{
s = s.replace(" ", "");
if(isPalindroma(s, 0, s.length()-1))
    System.out.println(s + " è palindroma!");
else
    System.out.println(s + " non è palindroma!");
}
```

Implementazione

```
private static boolean isPalindroma(String s, int ch_inizio, int ch_fine ) {  
    if(ch_inizio >= ch_fine)  
        return true;  
    else if(s.charAt(ch_inizio) != s.charAt(ch_fine))  
        return false;  
    else  
        return isPalindroma(s, ch_inizio+1, ch_fine -1) ;  
}
```

Caso base
Analizzo 0 o 1
carattere

Interruzione della
Ricorsione.

Chiamata
ricorsiva