Azzolini Riccardo 2018-12-14

Organizzazione della memoria e ricorsione

1 Invocazione di un metodo

1. **Passaggio per valore dei parametri**: al momento della chiamata, i parametri formali vengono allocati e inizializzati con i *valori* degli argomenti corrispondenti.
2. **Esecuzione**.
3. **Restituzione del risultato** e **rientro** al codice chiamante, indicati dall'istruzione `return`.
4. **Distruzione dell'ambiente di esecuzione**: viene rilasciata la memoria utilizzata per l'esecuzione, (cioè per le variabili locali e i parametri formali).

2 Memoria usata dalla JVM

Memoria statica: utilizzata per i campi statici delle classi.

- Viene allocata quando le classi sono caricate per l'esecuzione.
- La quantità necessaria per una classe può essere stabilita a priori esaminando esclusivamente il testo di tale classe.

Stack: contiene i dati usati dai singoli metodi che vengono man mano eseguiti.

- La sua struttura evolve dinamicamente durante l'esecuzione, in base alle chiamate di metodi.

Heap: memorizza gli oggetti creati dinamicamente durante l'esecuzione.

3 Gestione dell'heap

Gli oggetti vengono creati dinamicamente, richiamando i costruttori delle classi attraverso le espressioni `new`.

L'organizzazione dell'heap è mantenuta dal **garbage collector**, che:

- recupera la memoria occupata da oggetti non più referenziati (e quindi inaccessibili), in modo che possa essere riutilizzata
- riorganizza l'heap per rimediare ai problemi di allocazione dovuti alla frammentazione

4 Stack e record di attivazione

Lo stack è una *pila* (struttura LIFO) di **record di attivazione**.

I record di attivazione sono aree di memoria locale che contengono dati relativi a ciascun metodo attivato. Ogni record corrisponde a un'invocazione di un metodo, e contiene:

- informazioni di controllo, per
  - ricevere i risultati dei metodi richiamati
  - effettuare correttamente il rientro dai metodi richiamati
- dati del metodo
  - variabili locali
  - parametri formali, inizializzati con i valori degli argomenti forniti nella chiamata

5 Ricorsione

Un'entità è detta **ricorsiva** se è *definita in termini di se stessa.*

In particolare, un metodo si dice ricorsivo se richiama se stesso.

5.1 Esempio

Si può utilizzare un metodo ricorsivo per calcolare il fattoriale di un numero:

$$n! = \begin{cases} 1 & \text{se } n = 0 \\ n \cdot (n-1)! & \text{altrimenti} \end{cases}$$

```
public static int fattoriale(int n) {
    if (n == 0)
        return 1;
    else
        return n * fattoriale(n - 1);
}
```