

Modello della memoria di Java

ambiente delle classi (workspace) codice di metodi e variabili statiche;

stack record di attivazione dei metodi (e variabili locali);

heap oggetti raggiungibili tramite riferimenti;

riferimenti creati tramite **new**, assegnamento e passaggio di parametri; ammettono aliasing

memoria dinamica gestita interamente dalla JVM: non c'è **free** esplicito, gli oggetti vengono liberati dal GC quando rimangono orfani (nessun cammino di accesso).

Oggetti

Memorizzati come segue:

- almeno un riferimento all'oggetto nello stack (altrimenti viene liberato dal GC);
- sull'heap, descrittore di dato e variabili d'istanza;
- al descrittore di dato corrisponde una *tabella dei metodi (dispatch table)* nella *method area*.

Tabella dei metodi

A differenza di linguaggi come Smalltalk non contiene solo i metodi della classe più un puntatore alla tabella della superclasse, ma è una copia di questa, con i metodi aggiuntivi inseriti in coda e quelli sovrascritti mantenuti nella stessa posizione. Questo riduce la dinamicità ma migliora la performance: invece di una ricerca nella gerarchia (offset e profondità) è sufficiente conoscere l'offset all'interno della singola tabella. Inoltre, conservare la posizione anche dei metodi sovrascritti significa che l'offset di un metodo in una classe è valido anche per trovare lo stesso metodo nelle sottoclassi. Infine, le entrate della tabella contengono solo un puntatore al codice, anziché il codice stesso, per permetterne la condivisione tra le classi che contengono quel metodo (*structural sharing*).