

TIPOS DE MEMORIAS QUE UTILIZA JVM

Área de métodos

Todos los datos a nivel de clase, como el runtime constant pool, datos de los campos y los métodos, así como el código de los métodos y constructores, son almacenados en esta área. Si la memoria disponible en el área de métodos no puede satisfacer una petición de asignación de memoria por parte de un proceso, la JVM emite un `OutOfMemoryError`.



Area del montículo (heap)

Todos los objetos y sus correspondientes variables de instancia son almacenados en esta zona. El montículo es el área de datos de tiempo de ejecución en la que se aloja la memoria asignada a los arreglos e instancias de clases.

Nota: como las áreas del montículo y de métodos comparten la misma memoria para los diferentes hilos, habrá que gestionar cada hilo de forma segura para asegurar la integridad de los datos, como ya comentábamos en el apartado de inicialización de clases.



Area de pilas (stack)

A cada hilo creado en la JVM le corresponde en exclusiva una pila, creada al mismo tiempo que el propio hilo. Las variables locales, resultados parciales y llamadas a métodos se almacenan en esta zona de memoria conocida como pila.

Si el procesamiento que se está llevando a cabo en un hilo requiere un tamaño de pila mayor del permitido, la JVM emite un `StackOverflowError`.

Para cualquier llamada a método, se genera en la memoria de pila una entrada llamada marco de pila (stack frame). Cuando se completa la llamada al método, el marco de pila es destruido.

El marco de pila se divide en tres partes:

Variables locales – cada marco contiene un arreglo donde se almacenan las variables locales y sus valores.

La longitud de este arreglo se establece en tiempo de compilación.

Pila de operandos (operand stack) – esta estructura de tipo pila es utilizada para llevar a cabo las operaciones intermedias durante la llamada al método en cuestión. La profundidad máxima de esta pila se determina también en tiempo de compilación.

Marco de datos (frame data) – aquí se guardan todos los símbolos del método invocado, así como la información del bloque catch en caso de que se produzca alguna excepción.

Registros de contador de programa (PC, program counter)

La JVM admite múltiples hilos simultáneamente. Cada hilo tiene su propio registro contador de programa (PC) para guardar la dirección de la instrucción de la JVM ejecutándose en ese momento. Una vez ejecutada dicha instrucción, el registro PC es actualizado con la dirección de la próxima instrucción.



Pilas para métodos nativos

La JVM puede hacer uso de pilas que soporten métodos nativos, métodos escritos en lenguajes diferentes a Java, como C o C++. Cada hilo posee su propia pila de métodos nativos.

