

Special traits of java

Julie Ian Alejandro Brandon

February 10, 2015

1 Main Focus

The Java Language (special traits, why you would use it)

1.1 Question

- What the differences of using static and non-static?
- What is one difference between an Object Oriented Programing Language like Java and a Multi-paradigm Programming Language like Python?
- What is the relationship between the Java Programming Language, Bytecode, and the Java Virtual Machine?
- What is it about this ternary relationship that makes it very useful?
- Why in particular is Java's memory management such a huge achievement of this language in particular?

1.2 Answer

- A static method belongs to the class itself and a non-static (aka instance) method belongs to each object that is generated from that class. If your method does something that doesn't depend on the individual characteristics of its class, make it static (it will make the program's footprint smaller).
- In a language like Java, all objects must have a class to which they are children, and in Python, objects may live in thier frames.
- Together, these three things make up what is known as the Java platform. Code written in Java is compiled into Bytecode. Bytecode (which in essence is low-level code that is much more abstract than regular Java code) then acts as an instruction

set for the Java Virtual Machine. Finally, the Java Virtual Machine (quite literally a virtual computer) takes in all the bytecode and translates it into machine code, but not before proof checking the code first (hence the strict compilation regulations when programming in Java).

- Together, these three things make it possible for Java programs to essentially run (at least theoretically) on any machine that has the Java Runtime Environment, thus making the porting of applications from one operating system to the next much less strenuous.
- Java uses a technique called Garbage Collection. Programmers do not have to deal with the problem of memory allocation. This technique works by tracking live objects, and regarding everything else as garbage. When an object is no longer used, the garbage collector reclaims the memory and reuses it for future memory allocation. Once an object is no longer referenced and therefore is not reachable by code, the garbage collector removes it and reclaims the unused memory.