***VAR-ARGS Method***

Var args method is in introduced in 1.5 version of Java.

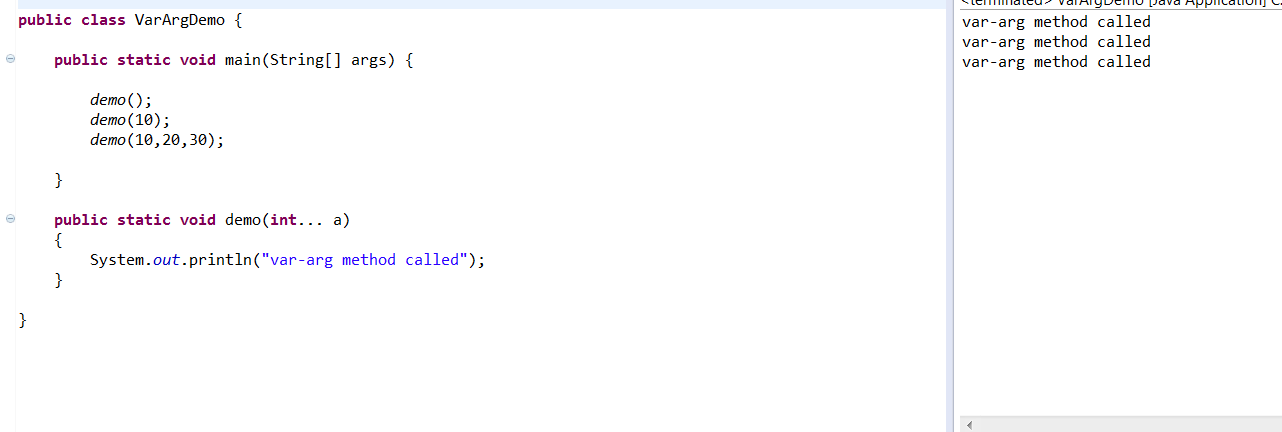
Till Java 1.4 version we can’t declare a method with variable number of arguments. If there is a change in no. of parameters we need to define a new method for it. So this approach increases the length of the code and reduces readability.

To solve this problem Sun people introduced var arg methods in 1.5 version of Java. Hence from 1.5 version of Java , we can declare a method with any number of parametes.

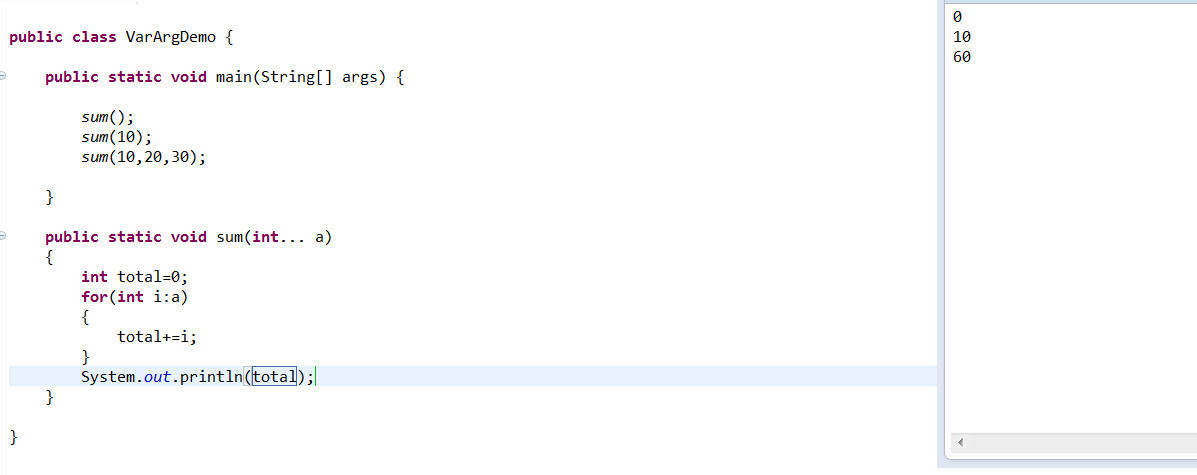
We can declare var-arg method as follows

**public static void methodDemo(int… a)**

we can call this method with any number of int parameters including 0 no. of parameters also.



Internally var-arg method is implemented using single dimensional array concept. Hence in var-arg method we can differenciate arguments by using index.



Case 1: valid var-arg method declarations

* demo(int… a) Valid
* demo(int a…) Not valid
* demo(int …a) Valid
* demo(int. ..a) Not valid
* demo(int .a..) Not valid

Case 2: we can mix var arg parameters with normal parameter also.

demo(int a, int… b) Valid

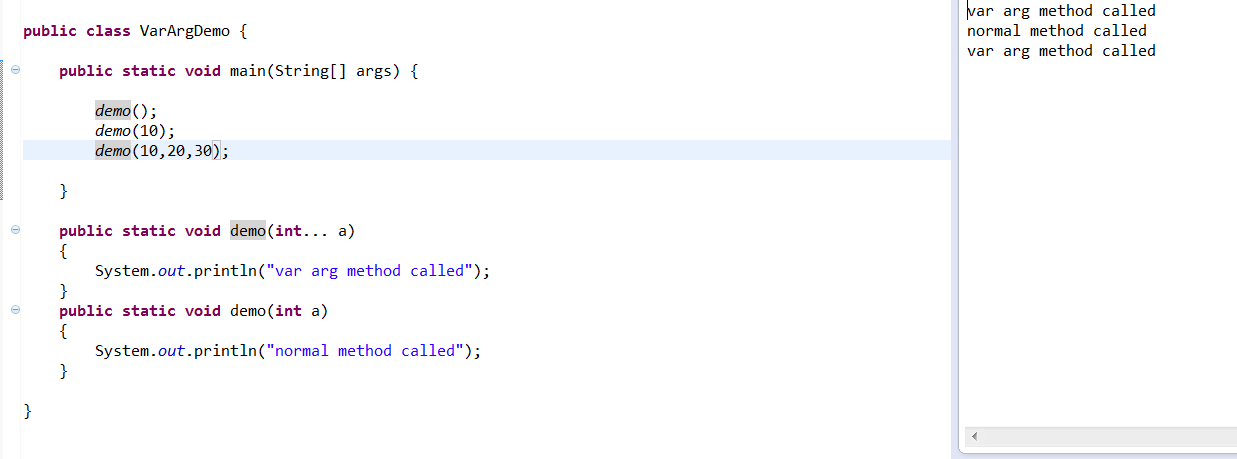
Case 3: If we are mixing var arg parameter with normal parameter , then var arg parameter should be last parameter.

demo(int… a, int b) Not valid

Case 4: In any var arg method we can take only 1 var arg parameter.

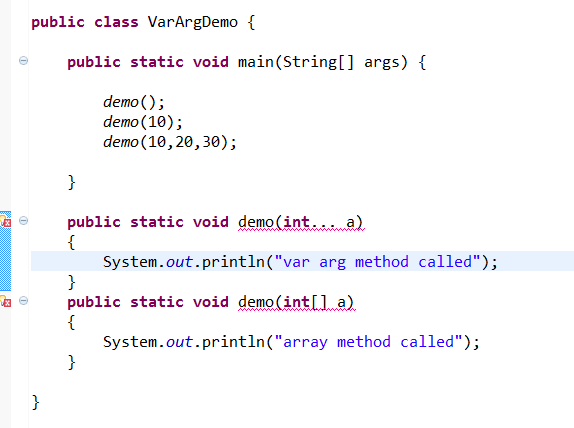
demo(int… a, int… b) Not valid

Case 5: Normal method is give more importance when compared to var arg method.



In general, var arg method has given least priority i.e., when no other method matches then only var arg method will be called. This is similar to default case in switch.

Case 6: we can’t declare demo(int… a) and demo(int[] a) in same class.



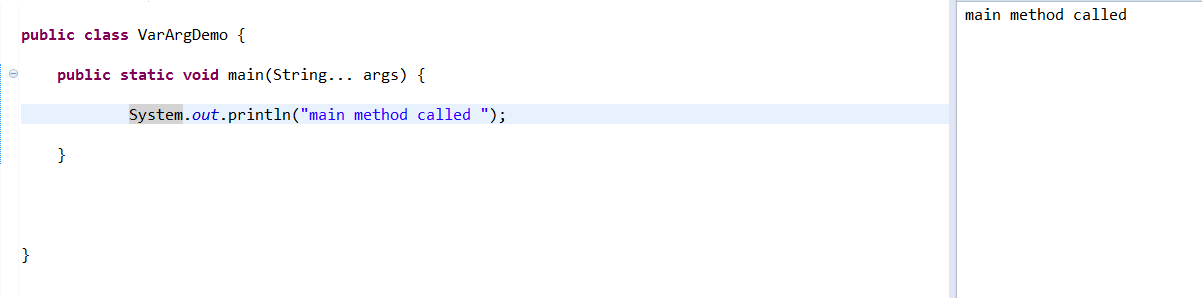
**Var arg vs single dimensional array**

Case 1: whenever single dimensional array present we can replace it with var arg method.

Example:

**demo(int[] a)** should be replaced with **demo(int… a)**

**main(String[] args)** should be replaced with **main(String… args)**

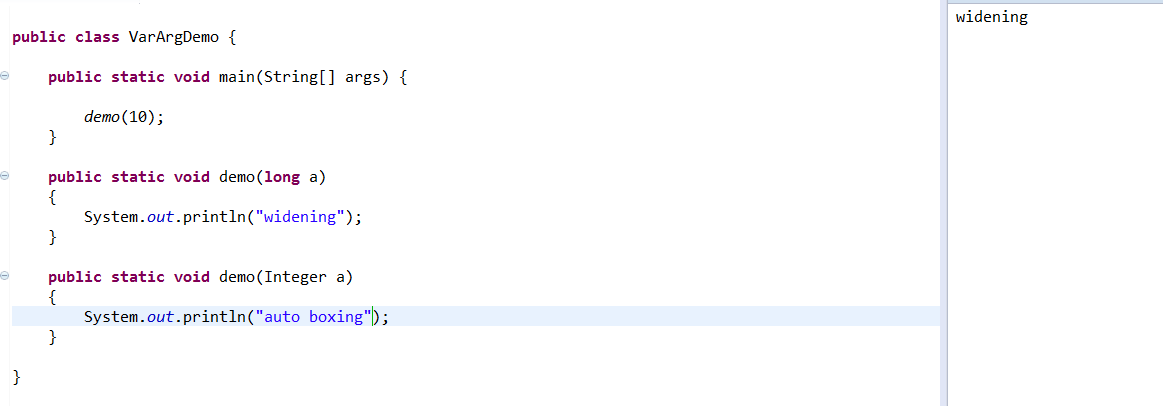


Case 2: whenever var arg method is present, we can’t replace with single dimensional array.

demo(int… a) can’t be replaced with demo(int[] a)

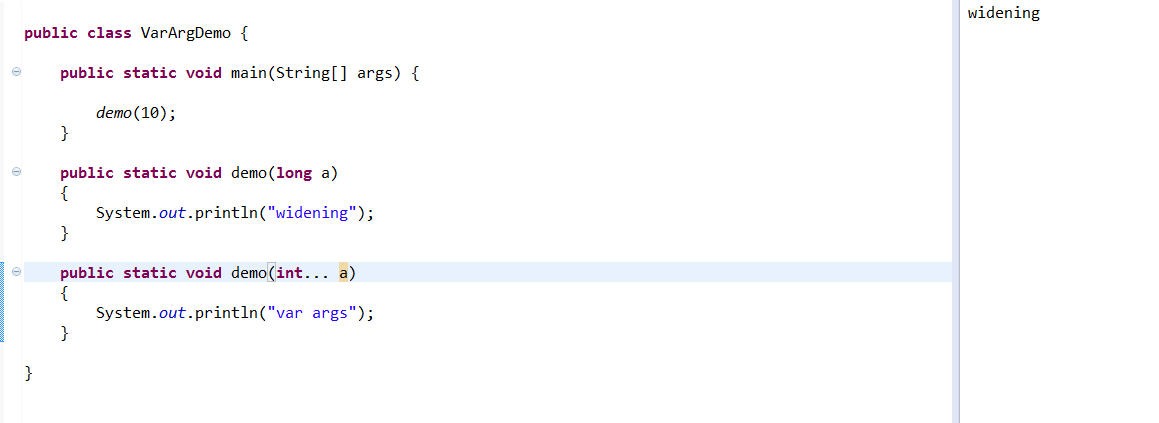
**Overloading w.r.t Auto boxing, widening and var arg method:**

Case 1: widening and auto boxing



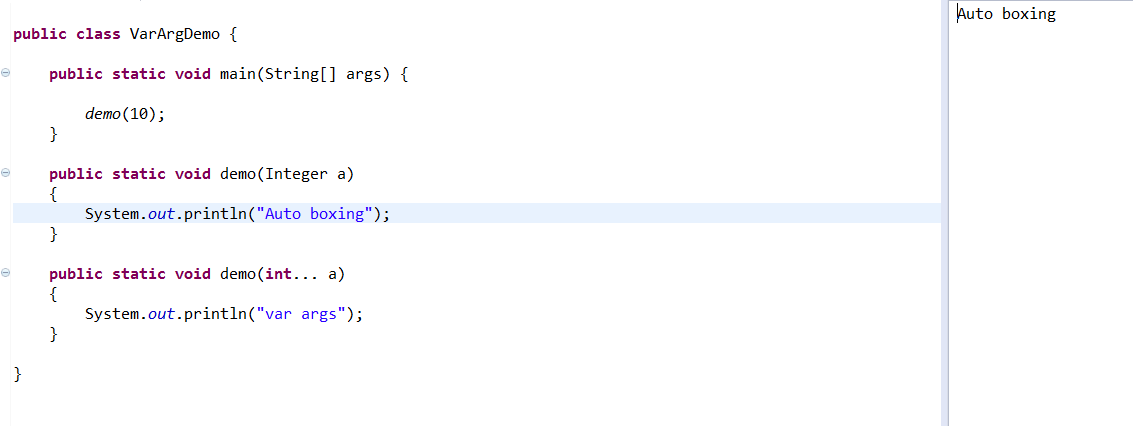
Widening dominates auto boxing.

Case 2: widening vs var args()



Widening dominates var args()

Case 3: Auto boxing vs var args()

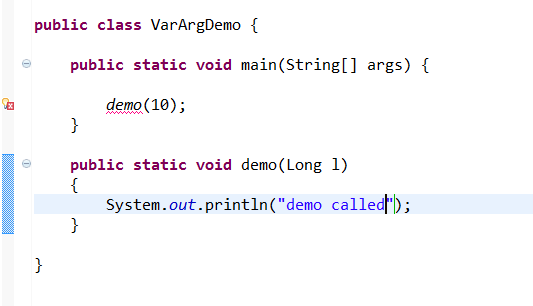


In general var args() will get least priority, if no other method matched then only var arg method will be executed.

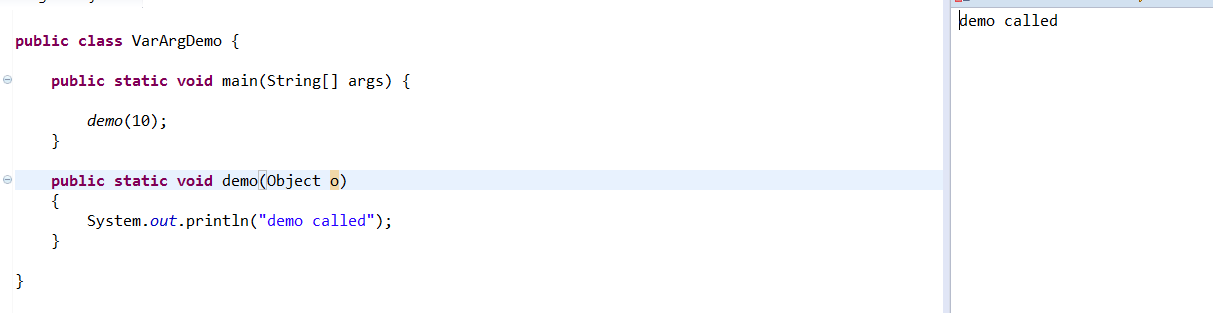
While resolving overloaded methods compiler will always keeps the precedence in the following order

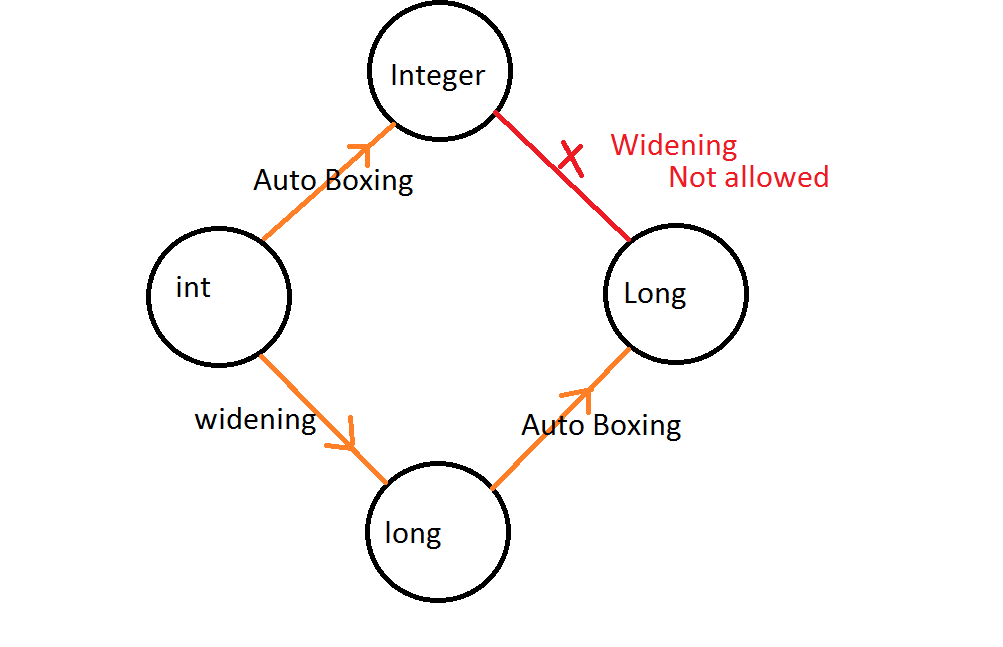
* widening
* auto boxing
* var arg method

Case 4:



Widening followed by auto boxing is not allowed in java where as auto boxing followed by widening is allowed.





Sanjeev Kumar