

METHOD OVERLOADING IN JAVA

By

M. BABY ANUSHA,
ASST.PROF IN CSE DEPT.,
RGUKT,NUZVID

Introduction :

- ❑ If a class has multiple methods having same name but different in parameters, it is known as **Method Overloading**.
- ❑ If we have to perform only one operation, having same name of the methods increases the readability of the program.

Continued...

- ❑ Suppose you have to perform addition of the given numbers but there can be any number of arguments, if you write the method such as `a(int,int)` for two parameters, and `b(int,int,int)` for three parameters then it may be difficult for you as well as other programmers to understand the behavior of the method because its name differs.
- ❑ So, we perform method overloading to figure out the program quickly.

Advantage of method overloading :

Method overloading *increases the readability of the program*. There are two ways to overload the method in java.

- By changing number of arguments.
- By changing the data type of arguments.

Method Overloading: changing no. of arguments :

- In this example, we have created two methods, first add() method performs addition of two numbers and second add method performs addition of three numbers.
- In this example, we are creating static methods so that we don't need to create instance for calling methods.

Example Program :

```
class Adder
{
static int add(int a,int b)
{
    return a+b;
}
static int add(int a,int b,int c)
{
    return a+b+c;}
}
```

Example Program :

```
class TestOverloading1
{
    public static void main(String[] args)
    {
        System.out.println(Adder.add(11,11));
        System.out.println(Adder.add(11,11,11));
    }
}
```

Output:

22

33

Method Overloading: changing data type of arguments

- In this example, we have created two methods that differs in data type.
- The first add method receives two integer arguments and second add method receives two double arguments.

Example Program :

```
class Adder  
{  
    static int add(int a, int b)  
    {  
        return a+b;  
    }  
    static double add(double a, double b)  
    {  
        return a+b;  
    }  
}
```

Example Program :

```
class TestOverloading2
{
    public static void main(String[] args)
    {
        System.out.println(Adder.add(11,11));
        System.out.println(Adder.add(12.3,12.6));
    }
}
```

Output:

22

24.9

Method Overloading is not possible by changing the return type of method only?

- In java, method overloading is not possible by changing the return type of the method only because of ambiguity.
- Let's see how ambiguity may occur:

Example Program :

```
class Adder  
{  
    static int add(int a,int b)  
    {  
        return a+b;  
    }  
    static double add(int a,int b)  
    {  
        return a+b;  
    }  
}
```

Example Program :

```
class TestOverloading3
{
    public static void main(String[] args)
    {
        System.out.println(Adder.add(11,11));//
        ambiguity
    }
}
```

Output:

**Compile Time Error: method add(int,int) is already
defined in class Adder**

Can we overload java main() method?

- Yes, by method overloading. You can have any number of main methods in a class by method overloading.
- But JVM calls main() method which receives string array as arguments only. Let's see the simple example:

Example Program :

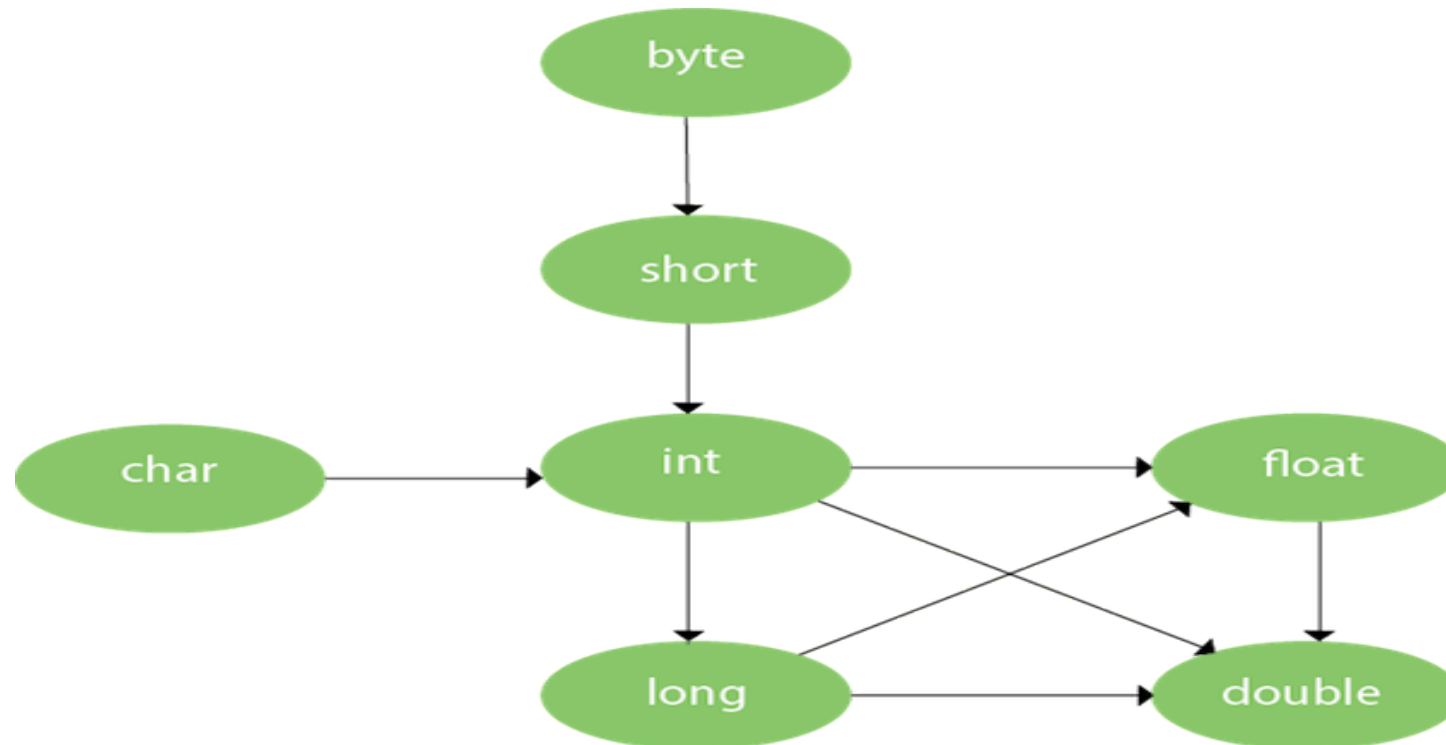
```
class TestOverloading4
{
public static void main(String[] args)
{System.out.println("main with String[]");}
public static void main(String args)
{System.out.println("main with String");}
public static void main()
{System.out.println("main without args");}
}
```

Output:

main with String[]

Method Overloading and Type Promotion :

- One type is promoted to another implicitly if no matching datatype is found. Let's understand the concept by the figure given below:



Method Overloading and Type Promotion :

- As displayed in the above diagram, byte can be promoted to short, int, long, float or double.
- The short datatype can be promoted to int, long, float or double.
- The char datatype can be promoted to int, long, float or double and so on.

Example of Method Overloading with TypePromotion :

```
class OverloadingCalculation1
{
    void sum(int a,long b)
    {
        System.out.println(a+b);
    }
    void sum(int a,int b,int c)
    {
        System.out.println(a+b+c);
    }
}
```

Example of Method Overloading with TypePromotion :

```
public static void main(String args[])
{
    OverloadingCalculation1 obj=new OverloadingCalculation1();

    obj.sum(20,20);//
    now second int literal will be promoted to long

    obj.sum(20,20,20);
}
```

Output:

40

60



Thank you!

shutterstock.com · 567687052