

Difference Between Static Binding and Dynamic Binding

www.differencebetween.com

Key Difference - Static Binding vs Dynamic Binding

Programming languages such as Java, C# support Object Oriented Programming(OOP). It allows building software using objects. There are many objects in a software system or a program. These objects have attributes and methods. Attributes describe the characteristics. Methods describe the actions that can be performed by the object. Data is passed through objects using methods. The required values are sent through method calls with parameters. The actual method implementation is in the method definition. There is a link between a method call and method definition. It is known as binding. There are two types of bindings. They are static binding and dynamic binding. The **key difference** between static binding and dynamic binding is that **in static binding, the binding is resolved at the compile time while dynamic binding is resolved at the runtime, which is the actual time of execution.** This article discusses the difference between these two binding mechanisms.

What is Static Binding?

Binding is the link between a method call and method definitions.

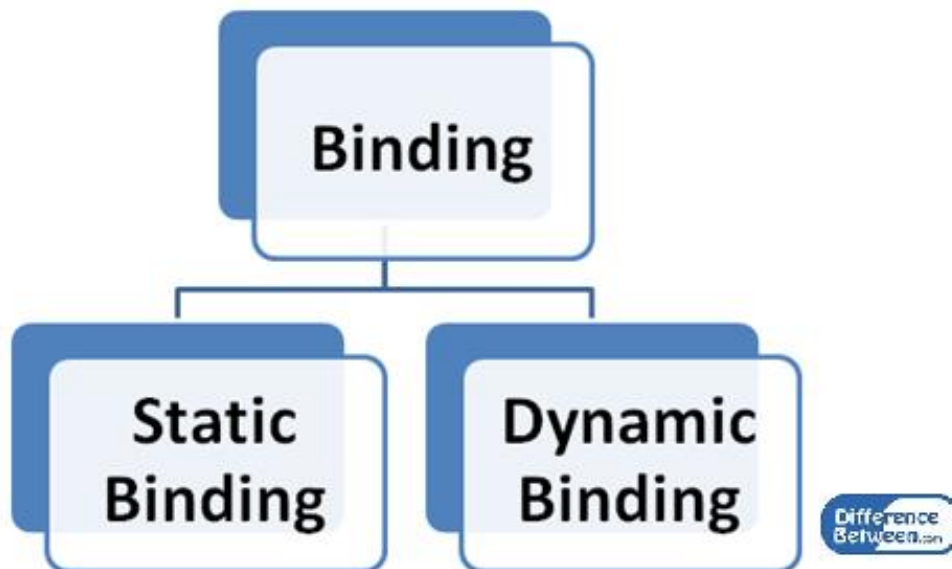


Figure 01: Static Binding and Dynamic Binding

Refer the below program written in Java.

```
public class A{  
  
    public void method1(){  
  
        System.out.println("Method1");  
  
    }  
  
    public void method2(){  
  
        System.out.println("Method2");  
  
    }  
  
    public static void main(String[] args){  
  
        A obj = new A();  
  
        obj.method1();  
  
        obj.method2();  
  
    }  
  
}
```

According to the above program, an object of type A is created. Then method1 and method2 are called. Identifying which method should call for execution is known as binding. Statement obj.method1() will call method1() and obj.method2() will call method2(). This link is binding.

In static binding, binding is resolved at compile time by the [compiler](#). It is also known as early binding. Binding happens before a program actually runs. Static binding occurs in method overloading. Refer the below program written in Java.

```
public void Calculation{  
  
    public void sum(int x, int y){  
  
        System.out.println("Sum is ", x+y);  
  
    }  
  
    public void sum(double x, double y){
```

```
System.out.println("Sum is ", x+y);  
}
```

```
public static void main(String[] args){  
    Calculation cal = new Calculation();  
    cal.sum(2,3);  
    cal.sum(5.1, 6.4);  
}
```

According to the above program, when passing the two integers, the method with two integers will be invoked. When passing two double values, the method corresponding to two double values will be invoked. This binding process occurs at the time of compilation. The compiler knows that it should call sum method with two integer values for `cal.sum(2,3)`. For `cal(5.1,6.4)`, it will call the sum method with two double values. All required information is known before runtime, so it increases the program efficiency and execution speed.

What is Dynamic Binding?

In Dynamic Binding the compiler does not resolve the binding at compile time. Binding occurs at run time. It is also known as late binding. Dynamic Binding occurs in method [overriding](#). Refer to program written in Java.

```
public class Shape(){  
    public void draw(){  
        System.out.println("Draw shape");  
    }  
}  
  
public class Circle() extends Shape{  
    public void draw(){
```

```

System.out.println("Draw circle");
}
}

public class Triangle() extends Shape{
public void draw(){
System.out.println("Draw triangle");
}
}

public class Test{
public static void main(String [] args){
Shape s;
s = new Shape();
s.draw();
s= new Circle();
s.draw();
s = new Triangle();
s.draw();
}
}

```

According to the above program, class Shape has a method draw(). Class Circle and class Triangle extends Shape class. Class Circle and class Triangle can inherit the attributes and methods of class Shape. Therefore, class Shape is the [super class](#) or parent class. Class Circle and Class Triangle are [sub classes](#) or derived classes. These classes also have draw() method with their own implementations. Therefore, the draw() method in the super class is overridden.

In the main method, different objects are invoked. There is a reference variable of Shape type, which is s. Then, s invokes the method according to the specific class. At compile time, the compiler will only refer the superclass draw method. When actual execution begins, it will lead to the execution of different draw methods. First, s will be pointing to the object of type Shape. Therefore, it will invoke the draw method in Shape class. Then the s will be pointing the object of type Circle, and it will invoke the draw method of Circle class. Finally, s will be referring to the object of type Triangle, and it will invoke the draw method in Triangle class. Even though the reference variable is of type Shape, the binding happens to depend on the object type. This concept is known as Dynamic Binding. The information is provided at runtime, so the speed of execution is slower comparing to static binding.

What is the Similarity Between Static Binding and Dynamic Binding?

- Both these are related to a [polymorphism](#) that allows an object to behave in multiple ways.

What is the Difference Between Static Binding and Dynamic Binding?

Static Binding vs Dynamic Binding	
Static Binding is the type of binding that collects all required information to call a function during compile time.	Dynamic Binding is the type of binding that collects all required information to call a function during runtime.
Time of Binding	
Static Binding occurs at compile time.	Dynamic binding occurs at runtime.
Functionality	
Static Binding uses type information for binding.	Dynamic Binding uses objects to resolve to bind.
Actual Object	
Static binding does not use an actual object for binding.	Dynamic binding, use the actual object for binding.
Synonyms	
Static binding is also known as early binding.	Dynamic binding is also known as late binding.

Execution	
The execution speed is fast in static binding.	The execution speed is low in dynamic binding.
Example	
Static binding is used in method overloading.	Dynamic binding is used in method overriding.

Summary - Static Binding vs Dynamic Binding

There is a link between a method call and method definition. It is known as binding. There are two types of bindings called static binding and dynamic binding. The difference between static binding and dynamic binding is that in static binding, the binding is resolved at the compile time while dynamic binding is resolved at the runtime, which is the actual time of execution. As the required information is provided before runtime, static binding is fast in execution comparing to dynamic binding.

Reference:

1. Java Interview 04 - Static Binding Vs Dynamic Binding, Mahika Tutorials, 27 Dec. 2017. [Available here](#)

How to Cite this Article?

APA: Difference Between Static Binding and Dynamic Binding.(2018 February 01). Retrieved (date), from <http://differencebetween.com/difference-between-static-binding-and-vs-dynamic-binding/>

MLA: "Difference Between Static Binding and Dynamic Binding" Difference Between.Com. 01 February 2018. Web.

Chicago: "Difference Between Static Binding and Dynamic Binding." Difference Between.Com. <http://differencebetween.com/difference-between-static-binding-and-vs-dynamic-binding/> accessed (accessed [date]).



Copyright © 2010-2017 Difference Between. All rights reserved