Method references

[Method references](https://javarevisited.blogspot.com/2017/03/what-is-method-references-in-java-8-example.html) provide a shorthand notation for lambda expressions that only call a single method. Method references can be used to simplify the code and make it more readable.

There are four types of method references in Java 8:

1. Reference to a static method
2. Reference to an instance method of an object
3. Reference to an instance method of an arbitrary object of a particular type
4. Reference to a constructor

**Reference to a static method:**

 A reference to a static method can be created using the syntax ClassName::methodName. For example:

interface IntBinaryOperator{

int applyAsInt(int a,int b);

}

class MathUtils {

public static int multiply(int a, int b) {

return a \* b;

}

}

public class Main {

public static void main(String[] args) {

IntBinaryOperator operator = MathUtils::multiply;

int result = operator.applyAsInt(4, 5);

System.out.println(result);

}

}

**Reference to an instance method of an object:**

A reference to an instance method of an object can be created using the syntax object::methodName. For example:

interface Supplier<T>

{

String get();

}

class Person {

private String name;

public Person(String name) {

this.name = name;

}

public String getName() {

return name;

}

}

public class Main1 {

public static void main(String[] args) {

Person person = new Person("John");

Supplier<String> supplier = person::getName;

String name = supplier.get();

System.out.println(name);

}

}

**Reference to an instance method of an arbitrary object of a particular type:**

A reference to an instance method of an arbitrary object of a particular type can be created using the syntax ClassName::methodName. For example:

interface Predicate<T>{

boolean test(T t);

}

class StringUtils {

public static boolean startsWithUppercase(String s) {

return Character.isUpperCase(s.charAt(0));

}

}

class Main2 {

public static void main(String[] args) {

Predicate<String> predicate = StringUtils::startsWithUppercase;

boolean result = predicate.test("hello");

System.out.println(result);

}

}

**Reference to a constructor:** A reference to a constructor can be created using the syntax ClassName::new. For example:

interface EmployeeFactory{

public abstract Employee getEmployee(String name, Integer age);

}

class Employee{

String name;

Integer age;

//Contructor of employee

public Employee(String name, Integer age){

this.name=name;

this.age=age;

}

}

public class Main4{

public static void main(String args[]){

EmployeeFactory empFactory=Employee::new;

Employee emp= empFactory.getEmployee("John Hammond", 25);

System.out.println(emp.name);

System.out.println(emp.age);

}

}