Method Reference

Method Reference

method and constructor reference

- It came into existence to reuse existing code
- Method reference and constructor reference are alternative to [[Lambda Expression]]
- Functional interface refering another method is called method reference
- **For static method**
    - class A { static void m1(){ System.out.println("Hello world"); } }
    - FunctionalInterface variableName=A::m1;
    - variableName.methodcall // functional interface
- In method reference always check
- **arguments**
- if the return type is different , you don't need to worry
- ## For non static mehtod
- **by using method reference name**
- className var=new className();
- FunctionalInterface varName=var::methodName // without ()
- varName.methodCall() // use the method in functional interface
-
- ## constructor reference
    - FunctionalInterface i1=className::new;
    - functional interface method should be void / className

- ```
  What is method reference?
      Referencing a method of another class
      for implementing a functional inteface
      is called method reference.

      It means for implementing a functional interface
      we must call a method of another class

  Why method reference?
      It is a Java 8 new feature.
      It is given to create LE in
      more short-cut way with less syntax.
  ```
  ```

When can we use method reference?

Inside LE body if we are just invoking a
method of another class to implement a FI
we can use MR.

When we can not use MR?

In addtion to reusing a method of another class,
if we need to place atleast one more statement in LE body,
we can not use MR, we must continue using LE.

How many ways we can develop MR?

We have three types of method references

1) static method reference
2) instance method refernce
3) constructor reference

static method reference syntax:

ContainingClassName::StaticMethodName

For example:

```
A::m1

I1 i1 = A::m1;
m2(A::m1);
I1 m3(){
    return A::m1;
}
```

instance method reference syntax:

ContaingClassObjectRef::InstanceMethodName

For example:

```
A a1 = new A();
a1::m1
new A()::m1

I1 i1 = new A()::m1;
m2(new A()::m1);
I1 m3() {
  return new A()::m1;
   }
```

Constrctor reference syntax

```
ContaingClassName::new

For example:
    A::new

I1 i1 = A::new;
m2(A::new);
I1 m3(){
    return A::new;
}
```

What are the rules to implement MR?

1) This referencing method/constructor
   parameter type, list, and order must be
   either same or widening to FI method parameter.

2) This referencing method/constructor
   return type must be
   either same or narrowing to FI method return type.

3) We can not develop constructor reference
   if FI method return is not void

   - We can develop constructor reference
     to a FI whose return type is
     either void or the constructor class name as return type

4) If FI method is a void method,
   then we can use either void method or non-void type or constructor as
   as method reference because the returned value or object will not be used
   via FI method call, because FI method is void we will call it directlty
   as single statement.

5) If FI method is non-void method,
   then we can not use void method or constructor as method reference.
   We msut use only non-void method as method reference.
   The referening method return type must be either same or lesser type
   then FI method return type.

If parameter is not mathced or return type is not mathced

we can not use this other class method or constructor as method reference.

-
-
-
-
-