Method References

Marco Faella - University of Naples Federico II

Function Pointers Come to Java!

A **method reference** is an expression denoting a method

Similar to function pointers in C/C++

More efficient than reflection (i.e., Method class)

Forms of Method References

- Static method — `Employee::getMaxSalary`
- Instance method, unspecified instance — `Employee::getSalary`
- Instance method, specified instance — `mike::getSalary`
- Constructor — `Employee::new`

Less common:

- Instance method of super-class — `super::foo`
- Array constructor — `A[]::new`

Method References have no Intrinsic Type

Type inference assigns a type based on context

Hence, a telling receiving context is needed

Method References are Typed to a FI

The context must identify a **Functional Interface**

Legal Contexts for Method References

- The same as for lambda expressions:
  - RHS of assignment
  - Actual parameter of a method or constructor
  - Argument of ‘return’
  - Argument of a cast

Legal Contexts for Method References

The receiving type T must be a functional interface:

	Context	Example	Receiving type
1	RHS of assignment	`T var = <method ref>`	`T`
2	Actual parameter of a method or constructor	`foo(<method ref>)`	Type of the corresponding formal parameter
3	Argument of 'return'	`return <method ref>`	Return type of current method
4	Argument of a cast	`(T) <method ref>`	`T`

Lambdas and method references are handy ways to *pass code around*

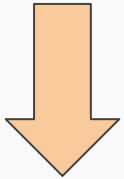
Lambdas for short one-shot snippets

Method references for more general cases

From classes to method references

Regular class

```
class Printer implements Consumer<String> {
   @Override
   public int accept(String s) {
      System.out.println(s);
   }
}
Consumer<String> printer = new Printer();
```



Anonymous class

```
Consumer<String> printer = new Consumer<String>() {
   @Override
   public int accept(String s) {
      System.out.println(s);
   }
};
```

Lambda expression

```
Consumer<String> printer = s -> System.out.println(s);
```

Java 8+, only functional intf.

Method reference

```
Consumer<String> printer = System.out::println;
```

Java 8+, only functional intf.

The Function FI

```
@FunctionalInterface
interface Function<S, T> {
   T apply(S s);
}
```

Examples

`MethodReferences.java`

From: https://bitbucket.org/mfaella/functionaljava

Summary

Method references are expressions denoting a method

Used in the same contexts as lambdas