

Algorithm Classes

1. Arrays.
2. Collections.

@ Provide common functionalities can be applied on Array `int []` and Collection

`Collections.concurrentList();` // Thread safe equivalent



Java - 8

Functional-Programming

- # functional interface

- # default methods

- # static methods

- # lambdas

- # streams

- # methods references

DateTime API

Optional

Nashorn engine (javascript engine)

Extension in collection API

Traditional :(Pure OOPs) : Imperative

Functional : Declarative

Imperative :

How : focus

Pure OOPs

Embraces data mutability

Declarative

What : focus

Functional Programming (pure function)

Data immutability

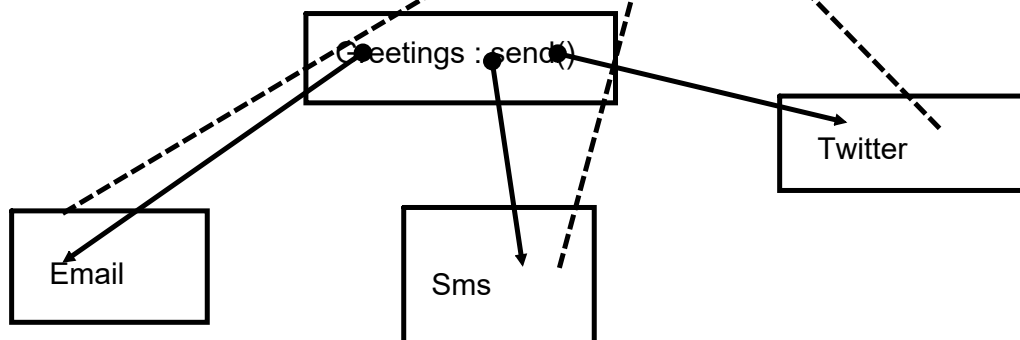
SQL style

Interfaces:

can have functions with definitions (default methods)

static methods with definitions

```
public void conveyMessage(String message, Greeting){  
    // call a single method send()  
}
```



LAmbda Syntax for pure function

1. not have any accessiblity modifier (nothing related to class)
2. Do not any name (anonymous function)
3. no return type
4. no param type
5. (<param>) -> { <definition> }

```
void fun(String str1, int n){  
}  
(str1,n) -> {  
}
```

```
void fun(String str1){  
}  
str1 -> {  
}
```

```
void fun(){  
}  
() -> {  
}
```

```
void fun(String str1){  
    // only single instruction  
}  
str1 -> single instruction
```

```
void fun(String str1){  
    // instruction  
    return a;  
}
```

```
str1 -> {  
    // instruction  
    return a;  
}
```

```
int add(int a, int b){  
    return a+b;  
}
```

// for single inst not bounded in braces return is default associated
(a, b) -> a+b;