|  |  |  |  |
| --- | --- | --- | --- |
| Predicate | Function | Consumer | Supplier |
| Accepts any Type, returns boolean | Accepts any Type, returns any Type | Accepts any Type, doesn’t return anything | No input, but returns some output |
|  |  |  |  |

@FunctionalInterface

**public** **interface** Consumer<T> {

**void** accept(T t);

---

}

**import** java.util.function.Consumer;

**public** **class** ExampleOne {

**public** **static** **void** main(String[] args) {

Consumer<String> c = (s) ->System.***out***.println(s);

c.accept("Hello World");

c.accept("Test");

Consumer<Integer> c1= (s) ->System.***out***.println(s\*s);

c1.accept(5);

c1.accept(60);

}

}

Consumer chaining

Consumer<Integer> c2= (s) ->System.***out***.print("Square of "+s);

Consumer<Integer> c3= (s) ->System.***out***.println(" is "+s\*s);

c2.andThen(c3).accept(50);

output

Square of 50 is 2500