- Java8 introduce a new feature called method reference.

- Method reference is an implementation of functional interface.

- It is a simple way of defining lambda function/expression.

- If lambda expression required to provide implementation of existing method we use method reference.

**Types of method references**

- Static Method Reference (Fi ⟶ static Method)

- Non Static Method Reference (Fi ⟶ Non-static Method)

- Constructor Method Reference (Fi ⟶ Constructor)

# Stream API

➤ Java 8 streams are used to manipulate data of collections.

➤ A stream is not a data structure instead it takes input from the Collections.

➤ It provide functional style approach to manipulate data.

➤ Streams are used for reading elements but not for storing elements.

➤ When used properly, StreamAPI allows us to reduce a huge amount of code.

**Stream API methods** ⇒

stream()
filter()
Map()
count()
sorted()
collect()
Collectors Class

# Consumer Vs Predicate Vs Function Vs Supplier

| Type | Abstract methods | Input | Returns Result as |
|---|---|---|---|
| Consumer | void accept(T t) | Accepts one Argument | It returns no result |
| Predicate | boolean test(T t) | Accepts one Argument | It returns a value ie., either true or false |
| Function | R apply(T t) | Accepts one Argument | It returns some value |
| Supplier | T get() | NILL | It returns some value |

**NOTE**: All these are functional Interfaces, which are present in **java.util.function** package which has been introduced in Java 1.8