Java Stream Operations

Different Ways to Create Java Streams

1. From a Collection

```
---------------------
List<String> list = List.of("Java", "Python", "Go");
Stream<String> stream = list.stream();
Stream<String> parallelStream = list.parallelStream();
```

2. From an Array

```
-----------------
String[] languages = {"Java", "Python", "Go"};
Stream<String> stream = Arrays.stream(languages);

int[] numbers = {1, 2, 3};
IntStream intStream = Arrays.stream(numbers);
```

3. Using Stream.of()

```
---------------------
Stream<String> stream = Stream.of("Apple", "Banana", "Cherry");
```

4. Using Stream.iterate()

```
--------------------------
Stream<Integer> evenNumbers = Stream.iterate(0, n -> n + 2).limit(10);
```

5. Using Stream.generate()

```
---------------------------
Stream<Double> randoms = Stream.generate(Math::random).limit(5);
Stream<String> helloStream = Stream.generate(() -> "Hello").limit(3);
```

6. From Files (I/O Streams)

```
----------------------------
```

```
try (Stream<String> lines = Files.lines(Paths.get("myfile.txt"))) {
    lines.forEach(System.out::println);
}


7. Using Stream.builder()
--------------------------
Stream<String> stream = Stream.<String>builder()
    .add("one")
    .add("two")
    .add("three")
    .build();


8. Primitive Streams (IntStream, LongStream, DoubleStream)
------------------------------------------------------------
IntStream range = IntStream.range(1, 5);         // 1 to 4
IntStream rangeClosed = IntStream.rangeClosed(1, 5); // 1 to 5
```