```java
import java.util.ArrayList;
import java.util.Arrays;
import java.util.List;
import java.util.stream.Stream;

public class CreateStreams {
    public static void main(String[] args) {
        String[] myArray = { "this", "is", "a", "sentence" };
        String result = Arrays.stream(myArray)
                .reduce("", (a,b) -> a + b);
        System.out.println(result);
    }
}
```

```java
import java.util.ArrayList;
import java.util.Arrays;
import java.util.List;
import java.util.stream.Stream;

public class CreateStreams2 {
    public static void main(String[] args) {
        List<Integer> myList = new ArrayList<Integer>();
        myList.add(1);
        myList.add(5);
        myList.add(8);

        Stream<Integer> myStream = myList.stream();


        Integer[] myArray = {1, 5, 8};

// Convert it into a Stream
        Stream<Integer> myStream2 = Arrays.stream(myArray);
    }
}
```

```java
import java.util.Arrays;
import java.util.List;
import java.util.stream.IntStream;
import java.util.stream.Stream;

public class Stream1 {
    public static void main(String args[]) {
        List<String> myList =
                Arrays.asList("a1", "a2", "b1", "c2", "c1");

        myList
                .stream()
                .filter(s -> s.startsWith("c"))
                .map(String::toUpperCase)
                .sorted()
                .forEach(System.out::println);

        Arrays.asList("a1", "a2", "a3")
                .stream()
                .findFirst()
                .ifPresent(System.out::println); // a1

        Stream.of("a1", "a2", "a3")
                .findFirst()
                .ifPresent(System.out::println); // a1

        IntStream.range(1, 4)
                .forEach(System.out::println);

        Arrays.stream(new int[] {1, 2, 3})
                .map(n -> 2 * n + 1)
                .average()
                .ifPresent(System.out::println); // 5.0

        Stream.of("a1", "a2", "a3")
                .map(s -> s.substring(1))
                .mapToInt(Integer::parseInt)
                .max()
                .ifPresent(System.out::println); // 3
    }
}
```