

Function Interface

Create a `Function<Integer, String>` that takes an integer and returns it as a string prefixed with "Value: ".

```
import java.util.function.Function;

public class FunctionExample {
    public static void main(String[] args) {
        // Write your implementation here...

        System.out.println(intToString.apply(5)); // Expected
        ↪ output: Value: 5
    }
}
```

Predicate Interface

Create a `Predicate<Integer>` that checks if the given integer is even.

```
import java.util.function.Predicate;

public class PredicateExample {
    public static void main(String[] args) {
        // Write your implementation here...

        System.out.println(isEven.test(4)); // Expected output:
        ↪ true
        System.out.println(isEven.test(7)); // Expected output:
        ↪ false
    }
}
```

Consumer Interface

Create a `Consumer<String>` that prints a message, appending " is awesome!" to the string passed to it.

```
import java.util.function.Consumer;

public class ConsumerExample {
    public static void main(String[] args) {
        // Write your implementation here...

        printMessage.accept("Java"); // Expected output: Java is
        ↪ awesome!
    }
}
```

Supplier Interface

Create a Supplier<String> that returns a random greeting message.

```
import java.util.function.Supplier;
import java.util.Random;

public class SupplierExample {
    public static void main(String[] args) {
        // Write your implementation here...
        // Hint: create an array of Strings inside of the
        ↪ Supplier lambda body

        System.out.println(randomGreeting.get()); // Expected
        ↪ output: one of the greetings randomly chosen
    }
}
```