Lambdas

Using Predicate with Lambda

What is Predicate?

- `Predicate` is a pre-defined functional interface provided by Java
- in order to use it `java.util.function` package must me imported
- `Predicate` has one abstract method, `test(T)`
  - which takes an argument of type `T` and returns `boolean`

```
@FunctionalInterface
public interface Predicate<T> {
  boolean test(T t);
  // ...
}
```

```
// in order to use Predicate you have to implement test(T) method
// and then you can use it


import java.util.function.*;
public class MyClass {
  public static void main(String[] args) {
    Predicate<Integer> gt10 = n -> n > 10;
    System.out.println(gt10.test(7));
    System.out.println(gt10.test(12));
  }
}
```

the implementation:

for a given n return n > 10 (true or false)

```
false

true
```

```
import java.util.function.*;
public class MyClass {
  static void myProbe (int n, Predicate<Integer> predicate) {
    if (predicate.test(n))
      System.out.println("The test has passed.");
    else
      System.out.println("The test has failed.");
  }
  public static void main(String args[]) {
    myProbe(5, n -> n > 2);
    myProbe(5, n -> n % 2 == 0);

    Predicate<Integer> myCriterium = n -> n*n + 5 > 100;
    myProbe(7, myCriterium);
    myProbe(11, myCriterium);
  }
}
```

```
The test has passed.
The test has failed.
The test has failed.
The test has passed.
```