src\test\java\koans\KoansTest.java

```java
package koans;

import org.junit.*;
import static org.junit.Assert.*;

import java.util.Random;
import java.util.function.Function;
import java.util.function.Predicate;

public class KoansTest {


  @Test
  public void mapArrayPlusOne(){
    int[] a = {1,2,3,4};

    Koans.mapArray(a, i -> i+1);
    assertArrayEquals(new int[] {2,3,4,5}, a);
  }

  @Test
  public void mapArraySquare(){
    int[] a = {1,2,3,4};

    Koans.mapArray(a, i -> i*i);
    assertArrayEquals(new int[] {1,4,9,16}, a);
  }

  @Test
  public void iterateFunctionIncrease(){
    int[] a;
    a = Koans.iterateFunction(5,0, i->i+1);
    assertArrayEquals(new int[]{0,1,2,3,4}, a);

  }

  @Test
  public void iterateFunctionPowerTwo(){
    int[] a;
    a = Koans.iterateFunction(5,1, i->2*i);
    assertArrayEquals(new int[]{1,2,4,8,16}, a);

  }

  @Test
  public void forEachArrayConcatWithClosure(){
    String[] s = {"Never", "gonna", "give", "you", "up", "!"};
    StringBuilder b = new StringBuilder();

    Koans.forEachArray(s, string -> b.append(string));

    assertEquals("Nevergonnagiveyouup!", b.toString());
  }

  @Test
  public void fillArrayWithConstant(){
    double[] pis = Koans.fillArray(4, () -> Math.PI);

    assertArrayEquals(new double[] {Math.PI, Math.PI, Math.PI, Math.PI}, pis, 1e-10);
  }

  @Test
  public void fillArrayWithRandom(){
    Random r1 = new Random(0);
    Random r2 = new Random(0);
    double[] randomNumbers = Koans.fillArray(4, () -> r1.nextDouble());

    assertArrayEquals(new double[] {r2.nextDouble(), r2.nextDouble(), r2.nextDouble(), r2.nextDouble()}, randomNumbers, 1e-10);
  }

  @Test
  public void minInteger(){
    Integer[] numbers = {4,1,-1,2,0};
    int min = Koans.min(numbers, Integer::compare);
    assertEquals(-1, min);
  }

  @Test
  public void minStringLength(){
    String[] strings = {"Never", "gonna", "give", "you", "up", "!"};
    String min = Koans.min(strings, (s1,s2) -> Integer.compare(s1.length(), s2.length()));
    assertEquals(min, "!");
  }

  @Test
  public void createMultiplier(){

    Function<Double, Double> twice = Koans.createMultiplier(2d);
    Function<Double, Double> timesPi = Koans.createMultiplier(Math.PI);

    assertEquals(2*Math.PI, twice.apply(Math.PI), 1e-10);
    assertEquals(2*Math.PI, timesPi.apply(2d), 1e-10);
  }

  @Test
  public void duplicateChecker(){
    Predicate<String> dc = Koans.duplicateChecker();

    assertFalse(dc.test("Never"));
    assertFalse(dc.test("gonna"));
    assertFalse(dc.test("give"));
    assertFalse(dc.test("you"));
    assertFalse(dc.test("up"));
    assertFalse(dc.test("!"));
    assertTrue(dc.test("up"));
    assertTrue(dc.test("Never"));
    assertFalse(dc.test("let"));
  }


}

```