src\main\java\koans\Koans.java

```java
package koans;

import java.util.Comparator;
import java.util.HashSet;
import java.util.Set;
import java.util.function.*;

public class Koans {

  public static void mapArray(int[] array, IntUnaryOperator f){
    for (int i = 0; i < array.length; i++)
      array[i] = f.applyAsInt(array[i]);
  }

  public static double[] fillArray(int length, DoubleSupplier s){
    double[] result = new double[length];

    for (int i = 0; i < length; i++)
      result[i] = s.getAsDouble();

    return result;
  }

  public static int[] iterateFunction(int length, int first, IntUnaryOperator f){
    int array[] = new int[length];
    array[0] = first;
    for (int i = 1; i < array.length; i++)
      array[i] = f.applyAsInt(array[i-1]);
    return array;
  }


  public static <T> T min(T[] elements, Comparator<T> c){

    T min = null;

    for (T element : elements){
      if (element != null && (min == null || c.compare(element, min) < 0)){
        min = element;
      }
    }

    return min;
  }

  public static Function<Double,Double> createMultiplier(double d){
    return x -> d * x;
  }

  public static void forEachArray(String[] strings, Consumer<String> c){
    for (String s : strings)
      c.accept(s);
  }


  public static <T> Predicate<T> duplicateChecker(){
    Set<T> set = new HashSet<>();

    return x -> {
      if (set.contains(x))
        return true;
      else{
        set.add(x);
        return false;
      }
    };
  }

}

```