```java
import java.util.ArrayList;

/**
 * COMP215-Programming Project 3: 01 Knapsack Problem Analysis.
 * ALGORITMTESTER is the testing class for various solutions to the 01 Knapsack Problem.
 * This class takes a solution as a parameter and writes solution times to CSV files.
 * @author Andrew Parsons
 * @version 09 March 2017
 */
class AlgorithmTester {

    private Solution solution;
    private Stopwatch stopwatch;
    private MultiFileWriter multiFileWriter = KnapsackMain.multiFileWriter;

    AlgorithmTester(Solution solution){ this.solution = solution; }

    void testSolution(double[] value, double[] weight, int repetitions) {

        ArrayList<Long> timeList = new ArrayList<>();
        double takenValue = 0;

        for (int r = 0; r < repetitions; r++) {

            stopwatch = new Stopwatch();
            takenValue = solution.knapsack(value, weight, KnapsackMain.capacity);
            timeList.add(stopwatch.elapsedTime());
        }
        try {
            multiFileWriter.processTestResult(new TestResult(value.length,
            calculateMean(timeList)), solution);
        } catch (Exception e) {
            System.out.println("Problem with the MultiFileWriter!");
            e.printStackTrace();
        }
        System.out.printf("%1$-4s %2$-4s %3$-10s %4$10f  %5$10f",
        solution.getClass().getCanonicalName(), ": ", "takenValue: ", takenValue, ((double)
        calculateMean(timeList)/1E6));
        System.out.println();

    }

    /**
     * Calculates the mean time to perform an operation.
     * @param arrayOfTimes, an array of times.
     * @return long, the mean time listed in the parametrized array.
     */
    private static long calculateMean(ArrayList<Long> arrayOfTimes) {

        long sum = 0;
        long size = arrayOfTimes.size();

        if (arrayOfTimes.isEmpty())
            return 0;

        for (long time: arrayOfTimes) {
            sum += time;
        }
        return sum / size;
    }

}
```