```java
import java.util.Scanner;

/**
 * Person represents a single line from the file Experiment-Student.txt
 *   trait: the first column (0 or 1)
 *   without: the second column (without treatment)
 *   with: the third column (with treatment)
 *
 * @author Nick Jones
 * @version 12/3/2014
 */
public class Person {
    int index;
    Double without;
    Double with;
    int trait;

    /**
     * Constructor
     * @param index which line in the file is it?
     * @param without result without treatment
     * @param with result with treatment
     * @param trait trait of person
     */
    public Person(int index, Double without, Double with, int trait) {
        this.index = index;
        this.without = without;
        this.with = with;
        this.trait = trait;
    }


    /**
     * Get a string of this Person
     * @return index: trait without with
     */
    @Override
    public String toString() {
        return String.format("%4d:  %1d  %3.8f  %3.8f", index, trait, without, with);
    }


    /**
     * What is the mean of the values of the people in the array
     * @param ar  The array of people to find the mean of
     * @param control Is this the control group?
     * @return If control, the mean of without. Else, the mean of with.
     */
    public static Double mean(ArrayList<Person> ar, boolean control) {
        Double result = 0.0;

        if (control)
            for (Person p : ar)
                result += p.without;
        else
            for (Person p : ar)
                result += p.with;

        return result/ar.size();
    }

    /**
     * What is the standard deviation of the values of the people in the array.
     * @param ar The array of people
     * @param control Is this the control group.
     * @return If control, std(without). Else, std(with).
     */
    public static Double std(ArrayList<Person> ar, boolean control) {
        Double result = 0.0;
        Double mean = Person.mean(ar, control);

        if (control)
            for (Person p : ar)
                result += Math.pow(p.without - mean, 2);
        else
            for (Person p : ar)
                result += Math.pow(p.with - mean, 2);

        return Math.sqrt(result/ar.size());
    }

    /**
     * What percent of the Person have a trait of zero?
     * @param ar The array of people
     * @return What percent of people have zero as their trait. [0.0, 100.0]
     */
    public static Double percentZero(ArrayList<Person> ar) {
        Double result = 0.0;

        for(Person p : ar)
            result += (p.trait + 1) % 2;

        return result / ar.size() * 100;
    }

    /**
     * Main Method
     *
     * Read in Experiment-Student.txt and perform statistical analysis on it.
     *
     * @param args UNUSED
     */
    public static void main(String[] args) {
        int numControl = 20; // Number of people in control group
        int numTest =  20; // Number of people in treatment group.

        assert(numControl + numTest <= 1000);

        try {
            Scanner scnr = new Scanner(new File("Experiment-Student.txt"));

            ArrayList<Person> ar = new ArrayList<Person>(); // All people in file
            ArrayList<Person> control = new ArrayList<Person>(); // Control group
            ArrayList<Person> test = new ArrayList<Person>(); // Treatment group

            int i = 1;

            while(scnr.hasNextLine()) {
                String line = scnr.nextLine();
                Scanner lScnr = new Scanner(line); // scans the line

                // PARSE LINE FROM FILE
                int trait = lScnr.nextInt();
                Double without = lScnr.nextDouble();
                Double with = lScnr.nextDouble();
                Person p = new Person(i++, without, with, trait);

                // ADD TO ARRAY OF ALL PEOPLE
                ar.add(p);
            }


            // SELECT PEOPLE FOR CONTROL GROUP
            for (i=0; i<numControl; i++) {
                int rand = (int)(ar.size() * Math.random());
                control.add(ar.get(rand));
                ar.remove(rand);
            }

            // SELECT PEOPLE FOR TREATMENT GROUP
            for (i=0; i<numTest; i++) {
                int rand = (int)(ar.size() * Math.random());
                test.add(ar.get(rand));
                ar.remove(rand);
            }

            // DISPLAY CONTROL PEOPLE
            System.out.println("\nControl:\n--------------------------");
            i=1;
            for(Person p : control)
                System.out.printf("%2d: %s\n", i++, p);

            // DISPLAY TREATMENT PEOPLE
            System.out.println("\nTest:\n--------------------------");
            i=1;
            for(Person p : test)
                System.out.printf("%2d: %s\n", i++, p);

            // CALCULATE CONTROL STATS
            Double controlMean = Person.mean(control, true);
            Double controlSTD = Person.std(control, true);
            Double controlZeroPercent = Person.percentZero(control);

            // CALCULATE TREATMENT STATS
            Double testMean = Person.mean(test, false);
            Double testSTD = Person.std(test, false);
            Double testZeroPercent = Person.percentZero(test);

            // FIND TOTAL PERCENT ZERO
            ArrayList<Person> all = new ArrayList<Person>(control);
            all.addAll(test);
```

```
            Double totalZeroPercent = Person.percentZero(all);

            // DISPLAY CONTROL STATS
            System.out.println("\nControl:\n--------------------------");
            System.out.println("mean: " + controlMean);
            System.out.println("std: " + controlSTD);
            System.out.println("%0: " + controlZeroPercent);

            // DISPLAY TREATMENT STATS
            System.out.println("\nTest:\n--------------------------");
            System.out.println("mean: " + testMean);
            System.out.println("std: " + testSTD);
            System.out.println("%0: " + testZeroPercent);

            // DISPLAY TOTAL PERCENT ZERO
            System.out.println("\nTotal %0:\n--------------------------");
            System.out.println("" + totalZeroPercent);

            // CALCULATE CONCLUSION
            Double ts = (testMean - controlMean) /
                    Math.sqrt(Math.pow(testSTD,2)/test.size() +
                              Math.pow(controlSTD,2)/control.size());
            Double cv = 1.96;
            Boolean concl = Math.abs(ts) > cv;

            // DISPLAY CONCLUSION
            System.out.println("\nResult:\n--------------------------");
            System.out.printf("|%f| %s %f => %s H0",
                    ts,
                    concl ? ">" : "<=",
                    cv,
                    concl ? "reject" : "fail to reject");

            // CLOSE SCANNER
            scnr.close();
        }
        catch (FileNotFoundException e) { System.out.println("File not found"); System.exit(1); }
    }
}
```