

import java.util.StringTokenizer;

import javax.swing.JOptionPane;

public class TestStudentExt {

public static void main(String[] args) {

String in, out;

String grade = null;

String stData;

String grade\_final;

//create 10 ex student objects.

in = JOptionPane.showInputDialog("enter number of student "); //enter student data

int stCount = Integer.parseInt(in);

//create an array of references . just like create 10 ints. reference type has no type, they have target type.

StudentExt [] st = new StudentExt[stCount];

//create a student objects. in a for loop. st1, st2,..st9

String token;

for (int i =0; i < st.length ; i++){ //st.length?

in = JOptionPane.showInputDialog("enter one student data");

StringTokenizer stk = new StringTokenizer(in, ",");

token = stk.nextToken().trim(); //eliminte the spaces

int id = Integer.parseInt(token);

token = stk.nextToken().trim();

String name = token;

token = stk.nextToken().trim();

int examCount = Integer.parseInt(token);

//create an exam array. //formtt data type variable = new data\_type[variable2]

double[] exam = new double[examCount];

//populate the array

for (int j=0; j< exam.length ; j++){

token = stk.nextToken().trim();

exam[j] = Double.parseDouble(token);

}

//create student object

st[i] = new StudentExt (id,name,exam, name);

}

//organize the students with similar grades

String outA = "", outB = "",outC = "", outD = "", outF = "", outAll = "";

for (int i =0; i<st.length; i++){

int id = st[i]. getId();

String name = st[i].getName();

grade = st[i].findGrade();

grade\_final =st[i]. findGradeType();

stData = "" + id + " " + name + " " + ("final grade: ") + grade + "\n";

if (grade.equalsIgnoreCase("A")) {

outA= outA+stData;

}

else if (grade.equalsIgnoreCase("B")) {

outB= outB+stData;

}

if (grade.equalsIgnoreCase("C")) {

outC= outC+stData;

}

if (grade.equalsIgnoreCase("D")) {

outD= outD+stData;

}

if (grade.equalsIgnoreCase("F")) {

outF= outF+stData;

}

}

outAll = outA+outB+outC+outD+outF;

JOptionPane.showMessageDialog(null, outAll );

}

}

**public** **class** StudentExt **extends** Student {

**private** String gradeType;

**private** String grade\_final;

**public** StudentExt(**int** id, String name, **double** [] exam, String gradeType){

**super**(id, name, exam);

**this**.gradeType = gradeType;

}

//@override

**public** String findGradeType() {

String grade = **super**.findGrade();

**if**(gradeType.equalsIgnoreCase("Credit"))

{

**if** (grade.equals("A") ||

(grade.equals("B")) ||

(grade.equals("C")) )

grade = "CR";

**else** {

grade = "NCR";

}

}

**return** grade;

}

}

**public** **class** Student {

**private** **int** id;

**private** String name;

//array. data\_type [] variable

**private** **double**[] exam;

//constructor. created by the source. click on the left mouse

**public** Student(**int** id, String name, **double**[] exam) {

**super**();

**this**.id = id;

**this**.name = name;

**this**.exam = exam.clone(); //clones the vlaue. creates a new array, sends a copy

}

**public** String findGrade () {

**double** sum=0;

**for** (**int** i= 0; i<exam.length; i++) {

sum = sum + exam[i];

}

**double** avg=sum/exam.length;

String grade;

**if** (avg >= 90.0) {

grade = "A";

}

**else** **if** (avg >= 80.0) {

grade = "B";

}

**else** **if** (avg >= 70.0){

grade = "C";

}

**else** **if** (avg >= 60.0){

grade = "D";

}

**else** {

grade = "F";

}

**return** grade;

}

//need to create get methods. go to any field constructor. use the source. choose geenerate getter and setters

**public** **int** getId() {

**return** id;

}

**public** String getName() {

**return** name;

}

**public** **double**[] getExam() {

**return** exam.clone();

}

}