

import javax.swing.JOptionPane;

public class TestStatus {

public static void main(String[] args) {

//perfrom polymorphism in main.

String in, out, name;

double length, width;

int age, weight, id, examCount;

//double[] examScores;

//create array of statuses. create status object

//can not do new to create object but can use status references

//an array of status

Status [] stat = new Status [6]; //polymorphism

//an array of different objects

int iStat=0;

for (int i =0; i < 3 ; i++){

//input, get the l,w

in = JOptionPane.showInputDialog("Enter length");

length = Double.parseDouble(in);

in = JOptionPane.showInputDialog("Enter width");

width = Double.parseDouble(in);

//create the object

new Rectangle (length, width);

//where to store it

stat[iStat] = new Rectangle (length,width);

iStat++;

}

for (int i =0; i < 2 ; i++){

//input, get the l,w

name = JOptionPane.showInputDialog("Enter name");

in = JOptionPane.showInputDialog("Enter age");

age = Integer.parseInt(in);

in = JOptionPane.showInputDialog("Enter weight");

weight = Integer.parseInt(in);

//create the object

new Sibling (name, age, weight);

//where to store it

stat[iStat] = new Sibling (name, age, weight );

iStat++;

}

for (int i =0; i < 1 ; i++){

in = JOptionPane.showInputDialog("Enter id");

id = Integer.parseInt(in);

name = JOptionPane.showInputDialog("Enter name");

in = JOptionPane.showInputDialog("Enter exams count");

examCount = Integer.parseInt(in);

//object

double[] examScores = new double [examCount];

for (int j =0; j < examCount ; j++){

in = JOptionPane.showInputDialog("Enter score");

examScores[j]=Double.parseDouble(in);

}

stat[iStat] = new Student (id,name,examScores);

iStat++;

}

//polymorphic loop. helds the statuses.

String status = " ";

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

status += stat[i].getStatus();

}

JOptionPane.showMessageDialog(null,status);

}

}

import javax.swing.JOptionPane;

public class Student implements Status{

private int id;

private String name;

private double[] examsScores;

//we need to create and object so we need a counstroctor.

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

super();

this.id = id;

this.name = name;

this.examsScores = examsScores;

}

@Override

public String getStatus() {

String status = "\nstudent:\n";

status = status + "id " +id + "\nname: " +name + "\n exam scores " + examsScores;

status = status + "scores: ";

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

status = status + examsScores[i] + " , ";

}

return status;

}

@Override

public void displayStatus() {

String status = getStatus();

JOptionPane.showMessageDialog(null, status);

}

}

public interface Status {

//two methods . The show method and the display method. get the status

//and return the status

public String getStatus();

public void displayStatus();

}

import javax.swing.JOptionPane;

public class Sibling implements Status{

private String name;

private int age;

private int weight;

public Sibling(String name, int age, int weight) {

super();

this.name = name;

this.age = age;

this.weight = weight;

}

@Override

public String getStatus() {

String status = "\n sibling:\n";

status = status + "name: " +name + "\nage " +age + "\nweight " + weight;

return status;

}

@Override

public void displayStatus() {

String status = getStatus();

JOptionPane.showMessageDialog(null, status);

}

//create constructors. Can be created by the eclipse source

//implement status. Provide the methods in the body.

}

import javax.swing.JOptionPane;

public class Rectangle implements Status {

private double length;

private double width;

//constructors

public Rectangle(double length, double width) {

super();

this.length = length;

this.width = width;

}

public String getStatus() {

String status = "\nRectangle: \n";

status = status + "length = " + length + ", width = " + width;

return status;

}

public void displayStatus() {

String status = getStatus();

JOptionPane.showMessageDialog(null, status);

}

//put the rectangle in the polymorphic array

//define an interface and the methods and implement

}