11.1

package demo2;

import java.util.GregorianCalendar;

import java.util.Scanner;

class LinearEquation {

private double a;

private double b;

private double c;

private double d;

private double e;

private double f;

LinearEquation(double a,double b,double c,double d,double e,double f){

this.a=a;

this.b=b;

this.c=c;

this.d=d;

this.e=e;

this.f=f;

}

public double getA() {

return a;

}

public double getB() {

return b;

}

public double getC() {

return c;

}

public double getD() {

return d;

}

public double getE() {

return e;

}

public double getF() {

return f;

}

boolean isSolvable() {

if((a\*d-b\*c)==0)

return false;

else

return true;

}

double getX() {

double x;

x=(e\*d-b\*f)/(a\*d-b\*c);

return x;

}

double getY() {

double y;

y=(a\*f-e\*c)/(a\*d-b\*c);

return y;

}

}

class SimpleGeometricObject {

private String color = "white";

private boolean filled;

private java.util.Date dateCreated;

/\*\* Construct a default geometric object \*/

public SimpleGeometricObject(){

dateCreated = new java.util.Date();

}

}

/\*\* Construct a geometric object with the specified color

\* and filled value \*/

public SimpleGeometricObject(String color, boolean filled){

dateCreated=new java.util.Date();

this.color = color;

this.filled=filled;

}

/\* Return color \*/

public String getColorf(){

return color;

}

/\*\* Set a new color \*/

public void setColor(String color){

this.color ■ color;

}

/\*\* Return filled. Since filled is boolean,

its getter method is named isFilled \*/

public boolean isFilled(){

return filled;

}

/\*\* Set a new filled \*/

public void setFilied(boolean filled){

this.filled=filled;

}

/\*\* Get dateCreated \*/

public java.util.Date getDateCreated(){

return dateCreated;

}

/\*\* Return a string representation of this object\*\*/

public String toString() {

return "created on " + dateCreated + "\ncolor: " + color +

" and filled: " + filled;

}

}

}

11.2

**package** demo2;

**import** java.util.GregorianCalendar;

**import** java.util.Scanner;

**public** **class** CircleFromSimpleCeometricObject

**extends** SimpleGeometricObject

**private** **double** radius;

**public** CircleFromSimpleGeometricObject() {}

**public** CircleFromSimpleGeometricObject(**double** radius) {

**this**.radius = radius;

}

**public** CircleFromSimpleGeometricObject(**double** radius,

String color, **boolean** filled) {

**this**.radius = radius;

setColor(color);

setFilled(filled);

}

/\*\* Return radius \*/

**public** **double** getRadius(){

**return** radius;

}

/\*\* Set a new radius \*/

**public** **void** setRadius(**double** radius) {

**this**.radius = radius;

}

/\*\* Return area \*/

**public** **double** getArea() {

**return** radius \* radius \* Math.***PI***;

}

/\*\* Return diameter \*/

**public** **double** getDiameter() {

**return** 2 \* radius;

}

/\*\* Return perimeter \*\*/

**public** **double** getPerimeter() {

**return** 2 \* radius \* Math . ***PI***;

}

/\*\* Print the circle info \*/

**public** **void** printCirclef(){

System.***out***.println("The circle is created "+getDateCreated()+

" and the radius is " + radius);

}

}

11.3

**package** demo2;

**import** java.util.GregorianCalendar;

**import** java.util.Scanner;

**public** **class** RectangleFromSimpleCeometricObject

**extends** SimpleCeometricObject{

**private** **double** width;

**private** **double** height;

**public** RectangleFromSimpleGeometricObject() {

}

**public** RectangleFromSimpleGeometricObject(

**double** width, **double** height) {

**this**.width=width;

**this**.height = height;

}

**public** RectangleFromSimpleGeometricObject(

**double** width, **double** height, String color, **boolean** filled) {

**this**.width = width;

**this**.height=height;

setColor(color);

setFilled(filled);

}

/\*\* Return width \*/

**public** **double** getWidth() {

**return** width;

}

/\*\* Set a new width \*/

**public** **void** setWidth(**double** width) {

**this**.width =width;

}

/\* Return height\*/

**public** **double** getHeight() {

**return** height;

}

/\*\*Set a new height\*/

**public** **void** setHeight(**double** height) {

**this**.height=height;

}

/\*\*Return area \*/

**public** **double** getArea() {

**return** width \*height;

}

/\*\* Return perimeter \*/

**public** **double** getPerimeter() {

**return** 2 \* (width + height);

}

}

11.4

**package** demo2;

**import** java.util.GregorianCalendar;

**import** java.util.Scanner;

**public** **class** TestCircleRectangle {

**public** **static** **void** main(String[] args) {

CircleFromSimpleGeometricObject circle =

**new** CircleFromSimpleGeometricObject(1);

System.***out***.println(".A circle " + circle.toString());

System.***out***.println("The color is " + circle.getColor());

System.***out***.println("The radius is " + circle.getRadius());

System.***out***.println("The area is " + circle.getArea());

System.***out***.println("The diameter is " + circle.getDiameter());

RectangleFromSimpleCeometricObject rectangle =

**new** RectangleFromSimpleGeometricObject(2, 4);

System.***out***.println("\nA rectangle " + rectangle.toStringO);

System.***out***.println("The area is " + rectangle.getAreaO);

System.***out***.println("The perimeter is " +

rectangle.getPerimeter());

}

}

11.5

**package** demo2;

**import** java.util.GregorianCalendar;

**import** java.util.Scanner;

**public** **class** PolymorphismDemo {

/\*\* Main method \*/

**public** **static** **void** main(String[] args){

// Display circle and rectangle properties

*displayObject*(**new** CircleFromSimpleCeometricObject

(1, "red",**false**));

*displayObject*(**new** RectangleFromSimpleCeometricObject

(1, 1,"black", **true**));

}

/\*\* Display geometric object properties \*/

**public** **static** **void** displayObject(SimpleCeometricObject object){

System out.println("Created on "+ object.getDateCreated() +

". Color is " + object.getColor());

}

}

11.6

**package** demo2;

**import** java.util.GregorianCalendar;

**import** java.util.Scanner;

**public** **class** helloworld {

**public** **static** **void** main(String[] args) {

*m*(**new** GraduateStudent());

*m*(**new** Student());

*m*(**new** Person());

*m*(**new** Object());

}

**public** **static** **void** m(Object x) {

System.***out***.println(x.toString());

}

}

**class** GraduateStudent **extends** Student {

}

**class** Student **extends** Person {

@Override

**public** String toString() {

**return** "Student" ;

}

}

**class** Person **extends** Object {

@Override

**public** String toString() {

**return** "Person" ;

}

}

11.7

**package** demo2;

**import** java.util.GregorianCalendar;

**import** java.util.Scanner;

**public** **class** helloworld {

/\*\* Main method \*/

**public** **static** **void** main(String[] args) {

// Create and initialize two objects

Object objectl = **new** CircleFromSimpleGeometricObject(l);

Object object2 = **new** RectangleFromSimpleCeometricObject(l, 1);

// Display circle and rectangle

*displayObject*(objectl);

*displayObject*(object2);

}

/\*\* A method for displaying an object \*/

**public** **static** **void** displayObject(Object object) {

**if** (object **instanceof** CircleFromSimpleGeometricObject) {

System.***out***.println("The circle area is " +

((CircleFromSimpleGeometricObject)object).getArea());

System.***out***.println("The circle diameter is " +

((CircleFromSimpleGeometricObject)object).getDiamete());

}

**else** **if** (object **instanceof**

RectangleFromSimpleGeometricObject) {

System.***out***.println ("The rectangle area is " +

((RectangleFromSimpleCeometricObject)object).getArea());

}

}

}

11.8

**package** demo2;

**import** java.util.GregorianCalendar;

**import** java.util.ArrayList;

**import** java.util.Scanner;

**public** **class** helloworld {

**public** **static** **void** main (String[] args) {

// Create a list to store cities

ArrayList<String> cityList = **new** ArrayList<>();

// Add some cities in the list

cityList.add ("London");

// cityList now contains [London]

cityList.add("Denver");

// cityList now contains [London , Denver]

cityList.add("Paris");

// cityList now contains [London, Denver, Paris]

cityList.add("Miami");

// cityList now contains [London, Denver, Paris, Miami]

cityList.add("Seoul");

// Contains [London, Denver, Paris, Miami, Seoul]

cityList.add("Tokyo");

// Contains [London, Denver, Paris, Miami, Seoul , Tokyo]

System.out.println("List size? " + cityList.size());

System.out.println("Is Miami in the list? " +

cityList.contains("Miami "));

System.out.println("The location of Denver in the list? "

+ cityList.indexOf("Denver"))；

System.out.println("Is the list empty? " +

cityList.isEmpty()); // Print false

// Insert a new city at index 2

cityList.add(2, "Xian");

// Contains [London, Denver, Xian, Paris, Miami , Seoulf Tokyo]

// Remove a city from the list

cityList.remove("Miami");

// Contains [London, Denver, Xian, Paris, Seoulf Tokyo]

// Remove a city at index 1

cityList.remove(1);

// Contains [London, Xian, Paris, Seoul, Tokyo]

// Display the contents in the list

System.out.println(cityList.toString());

// Display the contents in the list in reverse order

**for** (**int** i = cityList.sizeO - 1；i >= 0; i--)

System.out.print(c1tyList.get(i) + " ");

System.out.println();

// Create a list to store two circles

ArrayList<CircleFromSimpleCeometricObject> list

= **new** ArrayListoO;

// Add two circles

list,add(**new** CircleFromSimpleCeoinetric0bject(2));

list.add(**new** CircleFromSimpleCeometric0bject(3));

// Display the area of the first circle in the list

System.out.println("The area of the circle? M +

list.get(0).getArea());

}

}

11.9

**package** demo2;

**import** java.util.GregorianCalendar;

**import** java.util.ArrayList;

**import** java.util.Scanner;

**public** **class** helloworld {

**public** **static** **void** main(String[] args) {

ArrayList<Integer> list = **new** ArrayList<>();

Scanner input=**new** Scanner(System.***in***);

System.***out***.print("Enter integers (input ends with 0): ");

**int** value;

**do** {

value = input.nextInt(); // Read a value from the input

**if** (!list.contains(value) && value != 0)

list.add(value); // Add the value if it is not in the \*1ist

}**while** (value != 0);

// Display the distinct numbers

**for** (**int** i = 0; i < list.size(); i++) {

System.***out***.print(list.get(i) + " ");

}

}

}

11.10

**package** demo2;

**import** java.util.GregorianCalendar;

**import** java.util.ArrayList;

**import** java.util.Scanner;

**public** **class** helloworld {

**private** ArrayList<Object> list =**new** ArrayList<>();

**public** **boolean** isEmpty() {

**return** list.isEmpty();

}

**public** **int** getSize() {

**return** list.size();

}

**public** Object peek() {

**return** list.get(getSize() - 1);

}

**public** Object pop() {

Object o =list.get(getSize() - 1);

list.remove(getSize() - 1);

**return** o;

}

**public** **void** push(Object o) {

list.add(o);

}

@Override

**public** String toString(){

**return** "stack: " + list.toString();

}

}