DSA Practice Report

|  |  |
| --- | --- |
| Student ID | 18511510019 |
| Student Name | MOAHMED NUR |
| Practice No. | 11 |
| Practice Title | inheritance and derivation of class |
| Date |  |
| Place |  |
| Mark |  |
| Checked by | Wingo WU |

# Inheritance and derivation of class

## Objects

1. to learn well of Composition and inheritance
2. to learn how to reuse code
3. To learn

## Problems

You are required to design 2D geometry software which can draw geometry shapes such as circle,ellipse,square,rectangle, triangle.

You are supposed to write classes for 2D point, circle, ellipse,square,rectangle, and triangle based on the their relationship in following figure.

The shape is virtual class.

shape

Circle

Square

Triangle

Ellipse

Rectangle

Point

Each object has unique id. The id start from 1. 1 for the first object, 2 for the second object,....

Each ojbect has unique name, such as Circle1, Ellipse2,Square3 and 1,2,3 are their respective id.

## results

==================By WingoBingo=====================================

enter your choice:10

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

Demo inheritence of class, virtual class, virtual function

shape, point, circle, ellipse,retangle, square

0.exit the program

1.create a circle

2.create a ellipse

3.create a square

4.create a rectangle

5.List geometry objects

Enter your choice::1

Input xc,yc,r of circle:

3 4 10

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

Demo inheritence of class, virtual class, virtual function

shape, point, circle, ellipse,retangle, square

0.exit the program

1.create a circle

2.create a ellipse

3.create a square

4.create a rectangle

5.List geometry objects

Enter your choice::2

Input xc,yc,r1,r2 of ellipse:

4 5 12 14

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

Demo inheritence of class, virtual class, virtual function

shape, point, circle, ellipse,retangle, square

0.exit the program

1.create a circle

2.create a ellipse

3.create a square

4.create a rectangle

5.List geometry objects

Enter your choice::3

Input xc,yc,side lenght of square:

3 10 12

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

Demo inheritence of class, virtual class, virtual function

shape, point, circle, ellipse,retangle, square

0.exit the program

1.create a circle

2.create a ellipse

3.create a square

4.create a rectangle

5.List geometry objects

Enter your choice::4

Input xc,yc,lenght and width of rectangle:

3 8 21 23

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

Demo inheritence of class, virtual class, virtual function

shape, point, circle, ellipse,retangle, square

0.exit the program

1.create a circle

2.create a ellipse

3.create a square

4.create a rectangle

5.List geometry objects

Enter your choice::5

id=1 name=Circle1 xc=3 yc=4 r=10

id=2 name=Ellipse2 xc=4 yc=5 r1=12 r2=14

id=3 name=Squared3 xc=3 yc=10 w=12

id=4 name=Rectangle4 xc=3 yc=8w=21 h=23

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

Demo inheritence of class, virtual class, virtual function

shape, point, circle, ellipse,retangle, square

0.exit the program

1.create a circle

2.create a ellipse

3.create a square

4.create a rectangle

5.List geometry objects

Enter your choice::

# myResult

## source code

*// ConsoleApplication2.cpp : Defines the entry point for the console application.*

*//*

#include <iostream>

#include <sstream>

using namespace std;

class Point {

private:

    double x, y;

public:

    Point(double newX = 0, double newY = 0) : x(newX), y(newY) {};

    Point(Point const &p) { *this*->setX(p.getX()); *this*->setY(p.getY()); };

    double getX() const { return x; } ;

    double getY() const { return y; } ;

    void setX(double const newX) { x = newX; };

    void setY(double const newY) { y = newY; };

    void operator=(double n){  *this*->setX(n)  ; *this*->setY(n); };

    friend ostream& operator<< (ostream&, Point&);

    friend istream& operator>> (istream&, Point&);

};

ostream& operator << (ostream& output, Point& p){

    output << p.x << " " << p.y;

    return output;

}

istream& operator>> (istream& input, Point& p){

    input >> p.x >> p.y;

    return input;

}

class Shape {

private:

    string id;

    string name;

public:

    Shape(string newId = "" ,string newName = ""):id(newId), name(newName){};

    void setId(string newId) { id = newId; };

    void setName(string newName) { name = newName; };

    string getId() { return id; };

    string getName() { return name; };

    virtual void show() = 0;

*// friend  istream& operator>> (istream&, Ellipse&); can not be declared virtual*

};

class Circle : public Shape {

private:

    Point p;

    double r;

public:

    Circle(Point newP = 0, double newR =0, string newId= "") :Shape( newId,  "circle"+newId), p(newP), r(newR) {

    };

    Point  getP() const {return p;}

    void show() ;

    friend ostream& operator<< (ostream&, Circle&);

};

void Circle::show() {

    cout << "id = " << *this*->getId() << " name = " << *this*->getName() << " xc = "  << *this*->p.getX()<< " yc = "  << *this*->p.getY() <<  " r = " <<  *this*->r << endl;

}

ostream& operator << (ostream& output, Circle& c){

    output <<  "xc = " << c.p.getX()<< " xy = "  << c.p.getY() <<  " r = " <<  c.r;

    return output;

}

class Ellipse : public Shape {

private:

    Point p;

    double r1, r2;

public:

    Ellipse(Point newP = 0, double newR1 =0, double newR2 =0, string newId= "") :Shape( newId,  "ellipse"+newId), p(newP), r1(newR1), r2(newR2) {

    };

    Point  getP() const {return p;}

    void show() ;

    friend ostream& operator<< (ostream&, Ellipse&);

    friend istream& operator>> (istream&, Ellipse&);

};

void Ellipse::show() {

    cout << "id = " << *this*->getId() << " name = " << *this*->getName() << " xe = "  << *this*->p.getX()<< " ye = "  << *this*->p.getY() <<  " r1 = " <<  *this*->r1 <<  " r2 = " <<  *this*->r2 << endl;

}

istream& operator>> (istream& input, Ellipse& e){

    input >> e.p >> e.r1  >> e.r2;

    return input;

}

ostream& operator << (ostream& output, Ellipse& c){

    output <<  "xc = " << c.p.getX()<< " xc = "  << c.p.getY() <<  " r1 = " <<  c.r1 <<  " r1 = " <<  c.r1;

    return output;

}

class Square : public Shape {

private:

    Point p;

    double l;

public:

    Square(Point newP = 0, double newL =0 , string newId= "") :Shape( newId,  "square"+newId), p(newP), l(newL) {

    };

    double getL() const {return l;}

    Point  getP() const {return p;}

    void show() ;

    friend ostream& operator<< (ostream&, Square&);

    friend istream& operator>> (istream&, Square&);

};

void Square::show() {

    cout << "id = " << *this*->getId() << " name = " << *this*->getName() << " xs = "  << *this*->p.getX()<< " ys = "  << *this*->p.getY() << "  length = " << *this*->getL() << endl;

}

istream& operator>> (istream& input, Square& e){

    input >> e.p >> e.l;

    return input;

}

ostream& operator << (ostream& output, Square& c){

    output <<  "xc = " << c.p.getX()<< " xc = "  << c.p.getY() << "length = "  << c.getL();

    return output;

}

class Retangle : public Shape {

private:

    Point p;

    double l, w;

public:

    Retangle(Point newP = 0, double newL =0 , double newW =0 , string newId= "") :Shape( newId,  "retangle"+newId), p(newP), l(newL), w(newW) {

    };

    double getL() const {return l;}

    double getW() const {return w;}

    Point  getP() const {return p;}

    void show() ;

    friend ostream& operator<< (ostream&, Retangle&);

    friend istream& operator>> (istream&, Retangle&);

};

void Retangle::show() {

    cout << "id = " << *this*->getId() << " name = " << *this*->getName() << " xs = "  << *this*->p.getX()<< " ys = "  << *this*->p.getY() << "  length = " << *this*->getL() << "  width = " << *this*->getW() << endl;

}

istream& operator>> (istream& input, Retangle& e){

    input >> e.p >> e.l;

    return input;

}

ostream& operator << (ostream& output, Retangle& c){

    output <<  "xc = " << c.p.getX()<< " xc = "  << c.p.getY() << "length = "  << c.getL();

    return output;

}

void display(Shape \*s){

    s->show();

}

string toString(int num){

    stringstream nums;

    nums << num;

    string stringNum;

        nums >> stringNum;

    return stringNum;

}

*// Circle& createCircle(){*

*//  string stringNum = toString(10);*

*//  Point p(1, 2);*

*//  Circle c(p, 5, toString(10));*

*//  return c;*

*// }*

int main()

{

    int choose;

    cout << "Demo inheritance of class, virtual class, virtual function \n shape, point, circle, ellipse,retangle, square \n 0.exit the program \n 1.create a circle \n 2.create a ellipse \n 3.create a square \n 4.create a rectangle \n 5.List geometry objects \n";

    cout << "Enter your choice : ";

    cin >> choose;

    Shape\*\* base = new Shape\*[20];

    int index = 0;

    while(choose != 0){

        if(choose == 1){

*// string stringNum = toString(10);*

            Point p;

            double r;

            cout << "Input xc,yc,r of circle: ";

            cin >> p >> r;

            base[index] = new Circle(p, r, toString(index+1));

            display(base[index]);

            index++;

        }

        else if(choose == 2){

            Point p;

            double r1, r2;

            cout << "Input xe,ye,r1,r2 of ellipse: ";

            cin >> p >> r1>> r2;

            base[index] = new Ellipse(p, r1,r2, toString(index + 1));

            display(base[index]);

            index++;

        }

        else if(choose == 3){

            Point p;

            double l;

            cout << "Input xe,ye,l of square : ";

            cin >> p >> l ;

            base[index] = new Square(p, l , toString(index+  1));

            display(base[index]);

            index++;

        }

        else if(choose == 4){

            Point p;

            double l, w;

            cout << "Input xc,yc, length and width of rectangle: ";

            cin >> p >> l >> w ;

            base[index] = new Retangle(p, l, w , toString(index+  1));

            display(base[index]);

            index++;

        }

        else if(choose == 5){

            for (int  i = 0; i < index; i++)

            {

                display(base[i]);

            }

        }

        cout << "Demo inheritance of class, virtual class, virtual function \n shape, point, circle, ellipse,retangle, square \n 0.exit the program \n 1.create a circle \n 2.create a ellipse \n 3.create a square \n 4.create a rectangle \n 5.List geometry objects \n";

        cout << "Enter your choice : ";

        cin >> choose;

    }

    return 0;

}

## output

Demo inheritance of class, virtual class, virtual function

shape, point, circle, ellipse,retangle, square

0.exit the program

1.create a circle

2.create a ellipse

3.create a square

4.create a rectangle

5.List geometry objects

Enter your choice : 1

Input xc,yc,r of circle: 1 2 3

id = 1 name = circle1 xc = 1 yc = 2 r = 3

Demo inheritance of class, virtual class, virtual function

shape, point, circle, ellipse,retangle, square

0.exit the program

1.create a circle

2.create a ellipse

3.create a square

4.create a rectangle

5.List geometry objects

Enter your choice : 2

Input xe,ye,r1,r2 of ellipse: 1 2 3 4

id = 2 name = ellipse2 xe = 1 ye = 2 r1 = 3 r2

= 4

Demo inheritance of class, virtual class, virtual function

shape, point, circle, ellipse,retangle, square

0.exit the program

1.create a circle

2.create a ellipse

3.create a square

4.create a rectangle

5.List geometry objects

Enter your choice : 3

Input xe,ye,l of square : 1 2 3

id = 3 name = square3 xs = 1 ys = 2 length = 3

Demo inheritance of class, virtual class, virtual function

shape, point, circle, ellipse,retangle, square

0.exit the program

1.create a circle

2.create a ellipse

3.create a square

4.create a rectangle

5.List geometry objects

Enter your choice : 4

Input xc,yc, length and width of rectangle: 1 2 3

4

id = 4 name = retangle4 xs = 1 ys = 2 length = 3 width = 4

s, virtual function

shape, point, circle, ellipse,retangle, square

0.exit the program

1.create a circle

2.create a ellipse

3.create a square

4.create a rectangle

5.List geometry objects

Enter your choice : 5

id = 1 name = circle1 xc = 1 yc = 2 r = 3

id = 2 name = ellipse2 xe = 1 ye = 2 r1 = 3 r2 = 4

id = 3 name = square3 xs = 1 ys = 2 length = 3

id = 4 name = retangle4 xs = 1 ys = 2 length = 3 width = 4

Demo inheritance of class, virtual class, virtual function

shape, point, circle, ellipse,retangle, square

0.exit the program

1.create a circle

2.create a ellipse

3.create a square

4.create a rectangle

5.List geometry objects

Enter your choice :