**\*BTVN**

1. **Từ khóa super**

Từ khóa super trong Java là một biến tham chiếu mà được sử dụng để tham chiếu đến đối tượng lớp cha gần nhất. Bất cứ khi nào bạn tạo instance (sự thể hiện) của lớp con, một instance của lớp cha được tạo ngầm định, được tham chiếu bởi biến super.

Sự sử dụng của từ khóa super trong Java

* super được sử dụng để tham chiếu biến instance của lớp cha gần nhất.
* super() được sử dụng để triệu hồi Constructor của lớp cha gần nhất.
* super được sử dụng để triệu hồi phương thức của lớp cha gần nhất.

Ví dụ:

super được để tham chiếu biến instance của lớp cha gần nhất

*Vấn đề xảy ra nếu không có từ khóa super:*

class Vehicle{

int speed=50;

}

class Bike3 extends Vehicle{

int speed=100;

void display(){

System.out.println(speed); //se in speed cua Bike

}

public static void main(String args[]){

Bike3 b=new Bike3();

b.display();

}

}

Trong ví dụ trên, cả hai lớp Vehicle và Bike có cùng thuộc tính speed chung. Biến instance của lớp hiện tại được tham chiếu bởi instance theo mặc định, nhưng mình phải tham chiếu tới biến instance của lớp cha, và đó là tại sao chúng ta sử dụng từ khóa super để phân biệt giữa biến instance của lớp cha và biến instance của lớp hiện tại.

*Xử lý vấn đề với từ khóa super trong Java:*

//Chuong trinh vi du tu khoa super

class Vehicle{

int speed=50;

}

class Bike4 extends Vehicle{

int speed=100;

void display(){

System.out.println(super.speed); //bay gio se in speed cua Vehicle

}

public static void main(String args[]){

Bike4 b=new Bike4();

b.display();

}

}

super() được sử dụng để triệu hồi Constructor của lớp cha gần nhất

Bạn theo dõi ví dụ sau:

class Vehicle{

Vehicle(){System.out.println("Vehicle duoc tao");}

}

class Bike5 extends Vehicle{

Bike5(){

super();//se trieu hoi constructor cua lop cha

System.out.println("Bike duoc tao");

}

public static void main(String args[]){

Bike5 b=new Bike5();

}

}

**Ghi chú**: super() được tự động thêm vào trong mỗi Constructor của lớp bởi Compiler.

Qua các chương trước, chúng ta đã biết rằng constructor mặc định được cung cấp bởi Compiler nhưng nó cũng thêm super() cho lệnh đầu tiên. Nếu bạn đang tạo constructor cho riêng mình và bạn không có this() hoặc super() như là lệnh đầu tiên, thì Compiler sẽ cung cấp super() như là lệnh đầu tiên của Constructor đó.

Một ví dụ khác về từ khóa **super**, trong ví dụ này super() được cung cấp **ngầm định** bởi Compiler.

class Vehicle{

Vehicle(){System.out.println("Vehicle duoc tao");}

}

class Bike6 extends Vehicle{

int speed;

Bike6(int speed){

this.speed=speed;

System.out.println(speed);

}

public static void main(String args[]){

Bike6 b=new Bike6(10);

}

}

super được sử dụng để triệu hồi phương thức của lớp cha gần nhất

Từ khóa super cũng có thể được sử dụng để triệu hồi phương thức lớp cha gần nhất. Nó nên được sử dụng trong lớp con mà có chứa cùng phương thức như lớp cha, như trong ví dụ sau:

class Person{

void message(){System.out.println("Chao mung");}

}

class Student16 extends Person{

void message(){System.out.println("Chao mung ban den voi java");}

void display(){

message();//se trieu hoi phuong thuc message() cua lop hien tai

super.message();//se trieu hoi phuong thuc message() cua lop cha

}

public static void main(String args[]){

Student16 s=new Student16();

s.display();

}

}

Trong ví dụ trên, cả hai lớp Student và Person cùng có phương thức message(), nếu chúng ta gọi phương thức message() từ lớp Student, nó sẽ gọi phương thức message() của lớp Student chứ không phải của lớp Person bởi vì quyền ưu tiên cục bộ.

Trong tình huống mà không có phương thức nào của lớp con giống lớp cha, thì không cần sử dụng từ khóa super. Trong ví dụ dưới đây, phương thức message() được triệu hồi từ lớp Student, nhưng lớp Student không có phương thức message() này, vì thế bạn có thể trực tiếp gọi phương thức message().

Ví dụ chương trình *không cần* từ khóa super:

class Person{

void message(){System.out.println("Chao mung");}

}

class Student17 extends Person{

void display(){

message();//se trieu hoi phuong thuc message() cua lop cha

}

public static void main(String args[]){

Student17 s=new Student17();

s.display();

}

}

1. **Kế thừa trong hàm khởi tạo**

Khi lớp con [kế thừa](https://freetuts.net/tinh-ke-thua-trong-php-30.html) từ lớp cha thì khi ta tạo một đối tượng thuộc lớp con thì sẽ xảy ra một trong các trường hợp sau đây:

**Trường hợp 1:** Nếu lớp Con có hàm khởi tạo và lớp cha cũng có hàm khởi tạo

Trường hợp này hàm khởi tạo của lớp con sẽ được chạy, còn hàm khởi tạo ở lớp cha không được chạy.

**Ví dụ:**

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8  9  10  11  12  13  14  15 | class A {        function \_\_construct() {          echo 'Lớp A được khởi tạo';      }  }    class B extends A {        function \_\_construct() {          echo 'Lớp B được khởi tạo';      }  }    $a = new B(); // Kết quả là Lớp B được khởi tạo |

Kết quả xuất ra màn hình là “Lớp B được khởi tạo”.

Trường hợp 2: Nếu lớp con không có hàm khởi tạo, lớp Cha có hàm khởi tạo

Trường hợp này hàm khởi tạo ở lớp cha sẽ được chạy.

Ví dụ:

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8  9  10  11  12  13  14  15 | // Lớp A  class A {        function \_\_construct() {          echo 'Lớp A được khởi tạo';      }  }    // Lớp B  class B extends A {    }    // Khởi Tạo Lớp B  $a = new B(); // Kết quả là Lớp A Chạy |

Kết quả xuất ra màn hình là “Lớp A được khởi tạo”

Trường hợp 3: Lớp Con có hàm khởi tạo, lớp cha không có hàm khởi tạo

Trường hợp này hàm khởi tạo lớp con sẽ được chạy.

 Ví dụ:

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8  9  10  11  12  13  14  15 | // Lớp A  class A {    }    // Lớp B  class B extends A  {      function \_\_construct() {          echo 'Lớp B được khởi tạo';      }  }    // Khởi Tạo Lớp B  $a = new B(); // Kết quả là Lớp B Chạy |

Kết quả xuất ra màn hình là “Lớp B được khởi tạo”

Trường hợp 4: Cả 2 lớp chà và lớp con đều không có hàm khởi tạo

Trường hợp này đương nhiên là sẽ không có hàm nào được chạy

Ví dụ:

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8  9  10  11  12 | // Lớp A  class A {    }    // Lớp B  class B extends A {    }    // Khởi Tạo Lớp B  $a = new B(); // Kết quả là Không có gì |

1. **Lớp tĩnh**

* **Biến static:** Khi bạn khai báo một biến là static, thì biến đó được gọi là biến tĩnh, hay biến static.
* Biến static có thể được sử dụng để tham chiếu thuộc tính chung của tất cả đối tượng (mà không là duy nhất cho mỗi đối tượng), ví dụ như tên công ty của nhân viên, tên trường học của các sinh viên, ...
* Biến static lấy bộ nhớ chỉ một lần trong Class Area tại thời gian tải lớp đó.
* **Phương thức static:** Khi bạn khai báo một phương thức là static, thì phương thức đó gọi là phương thức static.
* Một phương thức static thuộc lớp chứ không phải đối tượng của lớp.
* Một phương thức static gọi mà không cần tạo một instance của một lớp.
* Phương thức static có thể truy cập biến static và có thể thay đổi giá trị của nó.
* **Khối static:** Được sử dụng để khởi tạo thành viên dữ liệu static. Nó được thực thi trước phương thức main tại lúc tải lớp.
* **Import static:** Nó tạo điều kiện cho lập trình viên java truy cập trực tiếp vào các thành viên static của lớp được import, mà không cần phải sử dụng thông qua tên lớp. Tiết kiệm code khi bạn phải truy cập trực tiếp tới bất kỳ phương thức nào của lớp khác.

1. **Override phương thức**

* Nếu lớp con có cùng phương thức như đã được khai báo trong lớp cha, thì đó gọi là **Ghi đè phương thức (Method Overriding)** trong Java. Nói cách khác, nếu lớp con cung cấp trình triển khai cụ thể của phương thức mà đã được cung cấp bởi một trong các lớp cha của nó, thì đó là ghi đè phương thức.
* Sự sử dụng của ghi đè phương thức trong Java
* Ghi đè phương thức được sử dụng để cung cấp trình triển khai cụ thể của một phương thức mà đã được cung cấp bởi lớp cha của nó.
* Ghi đè phương thức được sử dụng để thu được tính đa hình tại runtime.
* Qui tắc cho ghi đè phương thức trong Java
* Phương thức phải có cùng tên như trong lớp cha.
* Phương thức phải có cùng tham số như trong lớp cha.
* Phải là quan hệ IS-A (kế thừa).

Bài tập hình học

package KhoPhuTungXeMay.baitaphinhhoc;

public class hinhchunhat extends hinhhoc{

    public double Dai;

    public double Rong;

    public hinhchunhat(){}

    public void nhapChieuDai(double dai){

        this.Dai = dai;

    }

    public void nhapChieuRong(double rong){

        this.Rong = rong;

    }

    public void tinhChuVi(){

        this.ChuVi = (this.Dai + this.Rong)\*2;

    }

    public void tinhDienTich(){

        this.DienTich = this.Dai\*this.Rong;

    }

}

package KhoPhuTungXeMay.baitaphinhhoc;

public class hinhhoc {

    public double PI = 3.14;

    public String Ten;

    public double ChuVi, DienTich, TheTich;

    public hinhhoc(){}

    public void xuatTen(){

        System.out.println("ten la: "+ this.Ten);

    }

    public void inChuVi(){

        System.out.println("Chu vi la: "+ this.ChuVi);

    }

    public void inDienTich(){

        System.out.println("dien tich la: "+ this.DienTich);

    }

    public void inTheTich(){

        System.out.println("the tich la: "+ this.TheTich);

    }

}

package KhoPhuTungXeMay.baitaphinhhoc;

import java.util.Scanner;

public class hinhhocmain {

    public static void main(String[] args) {

        Scanner input = new Scanner(System.in);

        int a;

        System.out.println("1:tinh chu vi hinh chu nhat.");

        System.out.println("2:tinh dien tich hinh chu nhat.");

        System.out.println("3:tinh chu vi hinh vuong.");

        System.out.println("4:tinh dien tich hinh vuong.");

        System.out.println("5:tinh chu vi hinh tron.");

        System.out.println("6:tinh dien tich hinh tron.");

        System.out.println("7:tinh the tich hinh tru.");

        System.out.println("hay nhap vao con so tuong ung: ");

        do{

            a = input.nextInt();

            if(a<0 || a>4){

                System.out.println("-------------------------");

                System.out.println("vui long nhap lai!!!");

                System.out.println("1:tinh chu vi hinh chu nhat.");

                System.out.println("2:tinh dien tich hinh chu nhat.");

                System.out.println("3:tinh chu vi hinh vuong.");

                System.out.println("4:tinh dien tich hinh vuong.");

                System.out.println("5:tinh chu vi hinh tron.");

                System.out.println("6:tinh dien tich hinh tron.");

                System.out.println("7:tinh the tich hinh tru.");

                System.out.println("hay nhap vao con so tuong ung: ");

            }

        }while(a<0 || a>7);

        switch(a){

            case 1:{

                hinhchunhat hinhchunhat1 = new hinhchunhat();

                System.out.println("hay vao chieu dai hinh chu nhat: ");

                hinhchunhat1.nhapChieuDai(input.nextDouble());

                System.out.println("hay nhap chieu rong hinh chu nhat: ");

                hinhchunhat1.nhapChieuRong(input.nextDouble());

                hinhchunhat1.tinhChuVi();

                System.out.println("chu vi hình chu nhat la: ");

                hinhchunhat1.inChuVi();

                break;

            }

            case 2:{

                hinhchunhat hinhchunhat2 = new hinhchunhat();

                System.out.println("hay nhap vao chieu dai: ");

                hinhchunhat2.nhapChieuDai(input.nextDouble());

                System.out.println("hay nhap vao chieu rong: ");

                hinhchunhat2.nhapChieuRong(input.nextDouble());

                hinhchunhat2.tinhDienTich();

                System.out.println("dien tich hinh chu nhat la: ");

                hinhchunhat2.inDienTich();

                break;

            }

            case 3:{

                hinhvuong hinhvuong1 = new hinhvuong();

                System.out.println("hay nhap vao canh hinh vuong:");

                hinhvuong1.nhapCanh(input.nextDouble());

                hinhvuong1.tinhChuVi();

                System.out.println("chu vi hinh vuong la: ");

                hinhvuong1.inChuVi();

                break;

            }

            case 4:{

                hinhvuong hinhvuong2 = new hinhvuong();

                System.out.println("hay nhap vao canh hinh vuong: ");

                hinhvuong2.nhapCanh(input.nextDouble());

                hinhvuong2.tinhDienTich();

                System.out.println("den tich hinh vuong la: ");

                hinhvuong2.inDienTich();

                break;

            }

            case 5:{

                hinhtron hinhtron1 = new hinhtron();

                System.out.println("hay nhap vao ban kinh: ");

                hinhtron1.nhapBanKinh(input.nextDouble());

                hinhtron1.tinhChuVi();

                System.out.println("chu vi hinh tron la: ");

                hinhtron1.inChuVi();

                break;

            }

            case 6:{

                hinhtron hinhtron2 = new hinhtron();

                System.out.println("hay nhap vao ban kinh: ");

                hinhtron2.nhapBanKinh(input.nextDouble());

                hinhtron2.tinhDienTich();

                System.out.println("dien tich hinh tron la: ");

                hinhtron2.inDienTich();

                break;

            }

            case 7:{

                hinhtru hinhtru1 = new hinhtru();

                System.out.println("hay nhap chieu cao: ");

                hinhtru1.nhapChieuCao(input.nextDouble());

                System.out.println("hay nhap vao ban kinh: ");

                hinhtru1.nhapBanKinh(input.nextDouble());

                hinhtru1.tinhTheTich();

                System.out.println("the tich hinh tru la: ");

                hinhtru1.inTheTich();

                break;

            }

        }

        input.close();

    }

}

package KhoPhuTungXeMay.baitaphinhhoc;

public class hinhtron extends hinhhoc{

    public double BanKinh;

    public hinhtron(){}

    public void nhapBanKinh(double bankinh){

        this.BanKinh = bankinh;

    }

    public void tinhChuVi(){

        this.ChuVi = 2 \* this.BanKinh \* PI;

    }

    public void tinhDienTich(){

        this.DienTich=PI\*this.BanKinh\*this.BanKinh;

    }

}

package KhoPhuTungXeMay.baitaphinhhoc;

public class hinhtru extends hinhtron{

    public double chieuCao;

    public hinhtru(){}

    public void nhapChieuCao(double chieucao){

        this.chieuCao = chieucao;

    }

    public void tinhTheTich(){

        this.TheTich = this.chieuCao \*this.BanKinh \* this.BanKinh \* PI;

    }

}

package KhoPhuTungXeMay.baitaphinhhoc;

public class hinhvuong extends hinhchunhat{

    public hinhvuong(){}

    public void nhapCanh(double canh){

        this.Dai = this.Rong = canh;

    }

}

Bài tập lab4

package lab4;

public class Circle {

    public double radius;

    public String color;

    public Circle(Double radius,String color){

        this.radius = radius;

        this.color = color;

    }

    public double getRadius(){

        return this.radius;

    }

    public void setRadius(double radius){

        this.radius = radius;

    }

    public String getcolor(){

        return this.color;

    }

    public void setColor(String color){

        this.color = color;

    }

    public double getArea(){

        return 3.14\*this.radius\*this.radius;

    }

    public String toString(){

        return this.radius +"/"+ this.color;

    }

}

package lab4;

import java.util.Scanner;

public class classmain {

    public static void main(String[] args) {

        Cylinder c1=new Cylinder();

        Scanner input=new Scanner(System.in);

        System.out.println("nhap radius");

        c1.setRadius(input.nextDouble());

        System.out.println("nhap height");

        c1.setHeight(input.nextDouble());

        System.out.println("ket qua la: "+c1.getVolume());

        c1.inRadius();

    }

}

package lab4;

public class Cylinder extends Circle{

    private double height;

    public Cylinder(){}

    public Cylinder(String color, double radius,double height) {

        super(color, radius);

        this.height=height;

    }

    public double getHeight() {

        return height;

    }

    public void setHeight(double height) {

        this.height = height;

    }

    public double getVolume() {

        return this.getArea()\*height;

    }

    public void inRadius() {

        System.out.println("radius la:"+this.getRadius());

    }

}