super and this keywords

Tuesday, October 8, 2024 10:36 AM

super and this keywords

What is super keyword ?
-> super keyword is use to access parent class method, variable and constructors from child class

super keyword for constructor

-> by default every constructor have super keyword at their first line
-> when constructor is executed it firstly called parent class constructor because of super keyword at their first line
-> super keyword must present at first line in constructor
-> multiple super keywords not allowed in constructor
-> to initialize parent class instance variable execution of parent class constructor is mandatory that's why super keyword is present at first line in every constructor.

Test.java

```
1  package com.cjc.example1;
2
3  public class Test {
4
5      public static void main(String[] args) {
6          System.out.println("main-------start");
7          B b = new B();
8          System.out.println(b.x);
9          System.out.println("main-------ends");
10
11     }
12 }
13
```

A.java

```
1  package com.cjc.example1;
2
3  public class A {
4
5      int x = 10;
6
7      public A() {
8      System.out.println("A----constructor");
9      }
10 }
11
```

B.java

```
1  package com.cjc.example1;
2
3  public class B extends A{
4
5      int y = 20;
6
7      public B() {
8       System.out.println("B----constructor");
9      }
10 }
```

What is this keyword ?
-> this keywords represents current class object
-> this keyword is use to access method, variable and constructors of same class

this keyword for constructor

-> this keyword is use call same class constructor
-> this keyword we have to declare explicitly in constructor
-> this keyword must be present at first line of constructor
-> we can't declare super and this at a time in constructor
-> multiple this keyword in same constructor is not allowed

A.java

```java
package com.cjc.example1;

public class A {

    public A() {
    System.out.println("A----constructor");
    }

    public A(int y)
    {
         this();
        System.out.println("A----int---constructor");
    }
}

```

B.java

```java
package com.cjc.example1;

public class B extends A{


    public B() {
        this("ABC");
     System.out.println("B------no-arg---constructor");
    }
    public B(int x) {
        this();
        System.out.println("B------int ----constructor");
    }
    public B(String s)
    {
         super(100);
        System.out.println("B----String----Constructor");
    }

}

```

Test.java

```java
package com.cjc.example1;

public class Test {

    public static void main(String[] args) {
        System.out.println("main-------start");
        B b = new B(100);
        System.out.println("main-------ends");

    }
}

```

---

super and this for methods and variable

-> we can use any number of super and this for method and variable calling
-> we can declare super and this anywhere for method and variable calling

A.java

```java
package com.cjc.example2;

public class A {

    int x  = 10;

    public void m1()
    {
        System.out.println("m1----A");
    }
}
```

B.java

```java
package com.cjc.example2;

public class B extends A{

    int x = 20;

    public void m1() {

        System.out.println("m1---B");
    }

    public void m2() {
    // for variable calling
      int x = 30;
      System.out.println(super.x); // 10
      System.out.println(this.x); // 20
      System.out.println(x); // 20

      // for method calling
//      super.m1();  // m1----A
//      this.m1();  // m1----B
//      System.out.println("m2---B"); // m2--B
    }
}
```

Test.java

```java
package com.cjc.example2;

public class Test {

    public static void main(String[] args) {

        B b = new B();
        b.m2();

    }
}
```