

Tìm hiểu về các phép so sánh trong java

Contents

Khái quát	2
DataType primitive data type nguyên mẫu.	2
Data Type Reference.....	4
DataType Reference đã xây dựng trong java. Integer, String, Float, Double,....	4
DataType Reference tự định nghĩa	6

Khái quát

Trong java có 3 kiểu so sánh đặc trưng như sau:

+ Sử dụng toán tử `==` : return Boolean

Primitive thì so sánh giá trị thực,

Reference đã xây dựng sẵn Sử dụng ô nhớ địa chỉ để xét với nhau.

Reference tự định nghĩa So sánh bộ nhớ địa chỉ.

+ Sử dụng hàm `equals()` : return Boolean

Primitive so sánh giá trị thực,

Reference đã xây dựng sẵn sử dụng giá trị trực tiếp

Reference tự định nghĩa so sánh ô nhớ nếu ko override.

+ Sử dụng hàm `compareTo()`: dùng cho sort trả về 1 giá trị số.

Primitive so sánh giá trị thực,

Reference đã xây dựng sẵn sử dụng giá trị trực tiếp,

Reference tự định nghĩa ko có giá trị phải implement Comparable.

DataType primitive data type nguyên mẫu.

Sử dụng loại nào cũng được.

Primitive (int) vs Primitive (int) Chủ yếu là sử dụng `==` để so sánh các primitive với nhau.

workspace - comparefunctionality/src/main/java/comparefunctionality/datatypeprimitive/ComparePrimitive.java - Eclipse IDE

File Edit Source Refactor Navigate Search Project Run Window Help

The screenshot shows the Eclipse IDE interface. On the left, the Project Explorer displays the project structure: comparefunctionality (in compareinjava) [sharefu] > src/main/java > comparefunctionality.datatypeprimitive > ComparePrimitive.java. The main editor window shows the code for ComparePrimitive.java. The code is as follows:

```
1 package comparefunctionality.datatypeprimitive;
2
3
4 public class ComparePrimitive {
5
6     static void comparePrimitvevsPrimitive() {
7         int a=10;
8         int b=10;
9
10        // only use == that compare
11        if(a==b) {
12            System.out.println("use == ");
13        }
14    }
15
16    public static void main(String[] args) {
17        comparePrimitvevsPrimitive();
18    }
19
20 }
21
```

Primitive (int) vs Reference (Integer) so sánh thì có thể sử dụng hết 3 phương thức đều đúng.

```
ComparePrimitive.java
comparefunctionality > src/main/java > comparefunctionality.datatypeprimitive > ComparePrimitive > comparePrimitivevsP
13 }
14 }
15
16 static void comparePrimitivevsReference() {
17     int a=10;
18     Integer b=10;
19
20     // only use == that compare
21     if(a==b) {
22         System.out.println("Integer == int");
23     }
24
25     if(b.equals(a)) {
26         System.out.println("b.equals(a)");
27     }
28
29     if(b.compareTo(a) == 0) {
30         System.out.println("b.compareTo(a) == 0");
31     }
32 }
33
34 public static void main(String[] args) {
35     //comparePrimitivevsPrimitive();
36     comparePrimitivevsReference();
37 }

Problems Error Log Javadoc Progress Debug Shell Search Call Hierarchy Console Servers
<terminated> ComparePrimitive [Java Application] C:\Program Files\Java\jre1.8.0_271\bin\javaw.exe (May 10, 2021, 4:52:21 PM – 4:52:21 PM)
Integer == int
b.equals(a)
b.compareTo(a) == 0
```

Data Type Reference

Data Type Reference đã xây dựng trong java. Integer, String, Float, Double,....

Kết luận đó với loại này nên sử dụng equals() hoặc compareTo() để so sánh vì nó so sánh giá trị thật sự mà object đang nắm giữ.

Integer Trong java : == so sánh địa chỉ ô nhớ, equals và compareTo sử dụng Giá trị thật để so sánh.

```
ComparePrimitive.java  ReferenceBuiltBeforeJava.java
comparefunctionality > src/main/java > comparefunctionality.referencebuiltbefore >
1 package comparefunctionality.referencebuiltbefore;
2
3 public class ReferenceBuiltBeforeJava {
4
5     static Integer a = new Integer(10);
6     static Integer b = 10;
7     static Integer c = b;
8
9     static void IntegerdataTypeReferenceBuilt_in() {
10         // false
11         if (a == b) {
12             System.out.println("Integer == int");
13         }
14
15         // true
16         if (c == b) {
17             System.out.println("c==b");
18         }
19
20         // true
21         if (b.equals(a)) {
22             System.out.println("b.equals(a)");
23         }
24
25         // true
26         if (b.compareTo(a) == 0) {
27             System.out.println("b.compareTo(a) == 0");
28         }
29     }
30 }

Problems Error Log Javadoc Progress Debug Shell Search Call Hierarchy
<terminated> ReferenceBuiltBeforeJava [Java Application] C:\Program Files\Java\jre1.8.0_271\bin\
c==b
b.equals(a)
b.compareTo(a) == 0
```

String trong java cũng giống như interger

```

30
31 static String astr = new String("10");
32 static String bstr = "10";
33 static String cstr = bstr;
34
35 static void StringdataTypeReferenceBuilt_in() {
36     // false
37     if (astr == bstr) {
38         System.out.println("Integer == int");
39     }
40
41     // true
42     if (bstr == cstr) {
43         System.out.println("c==b");
44     }
45
46     // true
47     if (bstr.equals(astr)) {
48         System.out.println("b.equals(a)");
49     }
50
51     // true
52     if (bstr.compareTo(astr) == 0) {
53         System.out.println("b.compareTo(a) == 0");
54     }
55 }

```

Problems Error Log Javadoc Progress Debug Shell Search Call Hierarchy

<terminated> ReferenceBuiltBeforeJava [Java Application] C:\Program Files\Java\jre1.8.0_271\bin\j

c==b
b.equals(a)
b.compareTo(a) == 0

Data Type Reference tự định nghĩa

Object tự định nghĩa

== chỉ so sánh địa chỉ ô nhớ

Equals(): nếu không override thì nó sẽ so sánh ô nhớ.

compareTo: không được định nghĩa sẵn class phải implements Comparable

ComparePrimitive.java ReferenceBuiltBeforeJava.java CompareObjectCustomize.java ✕

comparefunctionality ▸ src/main/java ▸ comparefunctionality.objectcustomize ▸ Student ▸ ● getPoint() : Float

```
2
3 public class CompareObjectCustomize {
4
5 }
6
7 class Student {
8
9     private String name;
10    private int age;
11    private Float point;
12
13    public String getName() {
14        return name;
15    }
16    public void setName(String name) {
17        this.name = name;
18    }
19    public int getAge() {
20        return age;
21    }
22    public void setAge(int age) {
23        this.age = age;
24    }
25    public Float getPoint() {
26        return point;
27    }
28    public void setPoint(Float point) {
29        this.point = point;
30    }
31
32    Student(String name, int a, float f){
33        this.age=a;
34        this.name=name;
35        this.point=f;
36    }
37 }
```

```
ComparePrimitive.java  ReferenceBuiltBeforeJava.java  CompareObjectCustomize.java  Student.java
comparefunctionality  src/main/java  comparefunctionality.objectcustomize  CompareObjectCustomize

6
7     Student a = new Student("a", 10, 10.0f);
8     Student b = new Student("a", 10, 10.0f);
9     Student c = b;
10
11     // false
12     if(a==b) {
13         System.out.println("a==b");
14     }
15
16     // true
17     if(c==b) {
18         System.out.println("c==b");
19     }
20
21     // true
22     if(c.equals(b)) {
23         System.out.println("c.equals(b)");
24     }
25
26     // false
27     if(a.equals(b)) {
28         System.out.println("a.equals(b)");
29     }
30
31     //not available must implements Comparable
32     //if(a.compareTo(b)) {
33     //    System.out.println("a.compareTo(b)");
34     //}
35 }
```