

+ Java Cơ Bản

+ Các Khái Niệm Java OOPs

+ Java String

+ Các Tính Năng Mới Trong

Java

☞ Giới thiệu về các tính năng mới trong java

+ Java 8

☞ Biểu thức Lambda

☞ Ví dụ biểu thức Lambda với forEach()

☞ Ví dụ biểu thức Lambda với Thread

☞ Ví dụ biểu thức Lambda với Comparator

☞ Ví dụ biểu thức Lambda với Filter Collection Data

☞ StringJoiner

☞ forEach

☞ Base64 Encode và Decode

☞ Stream trong java 8

+ Java 7

☞ Binary Literal

☞ String trong mệnh đề Switch

☞ Catch nhiều ngoại lệ

☞ Lệnh try-with-resources

☞ Suy luận tạo đối tượng Generic

☞ Chữ số Java với dấu gạch dưới

☞ Các cải tiến JDBC của Java 7

+ Java 4/5

☞ Varargs trong Java

☞ Enum trong java

☞ For-Each trong java

☞ Static Import trong java

☞ Autoboxing và Unboxing trong java

☞ Annotation trong java

☞ Tùy chỉnh Annotation trong java

☞ Generics trong java

+ Xử Lý Ngoại Lệ

Generics trong Java

☺ Tùy chỉnh Annotation trong java

Xử lý ngoại lệ trong java ☺

Java Generics được thêm vào ngôn ngữ lập trình Java từ Java 5. Generics trong java là một cách để xác định các kiểu cụ thể cho các lớp và phương thức trong ngữ cảnh khác nhau. Nghe có vẻ hơi trừu tượng, vì vậy chúng ta sẽ xem xét lần lượt các khái niệm và một số ví dụ cụ thể.

Cụ thể trong bài này chúng ta sẽ học:

- Lớp Generic trong java
- Phương thức Generic trong java
- Ký tự đại diện (?) trong Java Generics
- Sử dụng Generics trong các đối tượng Collection
- Lợi thế của Generics trong Java

1. Lớp Generic trong java

Một lớp có thể **tham chiếu** bất kỳ kiểu đối tượng nào được gọi là **lớp generic**.

Ví dụ

Hãy xem ví dụ đơn giản sau về việc tạo và sử dụng lớp generic:

1. Tạo lớp generic MyGeneric.java:

```
package vn.viettuts.generics;

class MyGeneric<T> {
    T obj;

    void add(T obj) {
        this.obj = obj;
    }

    T get() {
        return obj;
    }
}
```

Trong lớp trên, tham số <T> là một kiểu chung chung (bạn có thể sử dụng một ký tự bất kỳ khác T) đại

Recent Updates

☞ Câu lệnh DISTINCT trong MySQL

☞ Mệnh đề GROUP BY trong MySQL

☞ Mệnh đề ORDER BY trong MySQL

☞ Mệnh đề LIMIT trong MySQL

☞ Mệnh đề LIKE trong MySQL

☞ Câu lệnh DELETE trong MySQL

☞ Câu lệnh UPDATE trong MySQL

☞ Câu lệnh UPDATE trong SQL

☞ Toán tử AND và OR trong MySQL

☞ Mệnh đề WHERE trong MySQL

☞ Câu lệnh Select trong MySQL

☞ Câu lệnh INSERT trong MySQL

VietTuts on facebook



Học Java Cơ Bản Đến
2,327 lượt thích

it works..... Why?

- + Các Lớp Lồng Nhau
- + Đa Luồng (Multithreading)
- + Java AWT
- + Java Swing
- + Java I/O
- + Ví Dụ Java I/O
- + Lập Trình Mạng Với Java
- + Java Date
- + Chuyển Đổi Kiểu Dữ Liệu
- + Java Collections
- + Java JDBC
- + Bài Tập Java Có Lời Giải
- + Câu Hỏi Phỏng Vấn Java

63,000 đ

88,060 đ

55,000 đ

103,500 đ

87,500 đ

diện cho bất kỳ một kiểu cụ nào.

Ví dụ khi bạn định nghĩa `MyGeneric<Integer>` tức là `T = Integer`, lúc này bạn có thể hình dung lớp `MyGeneric` có dạng như sau:

```
package vn.viettuts.generics;

class MyGeneric {
    Integer obj;

    void add(Integer obj) {
        this.obj = obj;
    }

    Integer get() {
        return obj;
    }
}
```

2. Tạo lớp TestGenerics3.java

```
package vn.viettuts.generics;

public class TestGenerics3 {
    public static void main(String args[]) {
        // use Integer
        MyGeneric<Integer> m1 = new MyGeneric<Integer>();
        m1.add(2);
        System.out.println(m1.get());

        // use String
        MyGeneric<String> m2 = new MyGeneric<String>();
        m2.add("Hello");
        System.out.println(m2.get());
    }
}
```

Trong lớp trên, tại dòng lệnh thứ 6 chúng ta khai báo như sau `MyGeneric<Integer>`, có nghĩa là kiểu chung `T` sẽ là `Integer`. Tại dòng lệnh thứ 11 chúng ta khai báo như sau `MyGeneric<String>`, có nghĩa là kiểu chung `T` sẽ là `String`.

Kết quả:

```
2
Hello
```

Quy ước đặt tên tham số

Việc đặt tên tham số là rất quan trọng để học generics. Các tham số thông thường như sau:

1. T – Type
2. E – Element
3. K – Key
4. N – Number

Thích Trang
 Gửi tin nhắn

Đây là người đầu tiên trong số bạn bè của bạn thích nội dung này

2. Phương thức Generic trong java

Giống như lớp generic, chúng ta có thể tạo phương thức generic có thể chấp nhận bất kỳ kiểu đối số nào.

Chúng ta hãy xem một ví dụ đơn giản về phương thức generic trong java để in các phần tử của mảng. Chúng ta sử dụng **E** để biểu thị phần tử.

```
package vn.viettuts.generics;

public class TestGenerics4 {

    public static <E> void printArray(E[] elements) {
        for (E element : elements) {
            System.out.print(element + " ");
        }
        System.out.println();
    }

    public static void main(String args[]) {
        Integer[] intArray = { 10, 20, 30, 40, 50 };
        Character[] charArray = { 'J', 'A', 'V', 'A' };

        System.out.print("In mảng số nguyên: ");
        printArray(intArray);

        System.out.print("In mảng ký tự: ");
        printArray(charArray);
    }
}
```

Kết quả:

```
In mảng số nguyên: 10 20 30 40 50
In mảng ký tự: J A V A
```

3. Ký tự đại diện trong Java Generics

Ký tự ? (dấu chấm hỏi) tượng trưng cho phần tử ký tự đại diện. Nó có nghĩa là bất kỳ loại nào. Nếu chúng ta viết <? mở rộng Number>, nó có nghĩa là bất kỳ lớp con của Number ví dụ như Integer, Float, Double vv. Bây giờ chúng ta có thể gọi phương thức của lớp Number qua bất kỳ đối tượng lớp con nào.

Hãy hiểu nó bằng ví dụ dưới đây:

```
package vn.viettuts.generics;

import java.util.ArrayList;
import java.util.List;

abstract class Shape {
    abstract void draw();
}

class Rectangle extends Shape {
```

```

        void draw() {
            System.out.println("ve hình chu nhât.");
        }
    }

    class Circle extends Shape {
        void draw() {
            System.out.println("ve hình tron.");
        }
    }

    public class GenericTest5 {
        // tao phuong thuc chi chap nhan tham so la lop con cua Shape
        public static void drawShapes(List<? extends Shape> lists) {
            for (Shape s : lists) {
                s.draw();// goi phuong thuc cua lop Shape boi lop con
            }
        }

        public static void main(String args[]) {
            List<Rectangle> list1 = new ArrayList<Rectangle>();
            list1.add(new Rectangle());

            List<Circle> list2 = new ArrayList<Circle>();
            list2.add(new Circle());
            list2.add(new Circle());

            drawShapes(list1);
            drawShapes(list2);
        }
    }
}

```

Kết quả:

```

ve hình chu nhât.
ve hình tron.
ve hình tron.

```

4. Sử dụng Generics trong các đối tượng Collection

Trước Generics, chúng ta có thể lưu trữ bất kỳ loại đối tượng nào trong collection như Non-generic. Với generics, được các lập trình java áp dụng để lưu trữ các kiểu cụ thể của các đối tượng.

Cú pháp sử dụng generic collection

```
ClassOrInterface<Type>
```

Ví dụ sử dụng Generics trong java:

```
ArrayList<String>
```

Ví dụ đầy đủ về Generics trong java

Ở đây, chúng ta sử dụng lớp ArrayList, nhưng bạn có thể sử dụng bất kỳ lớp Collection khác như

```

package vn.viettuts.generics;

import java.util.ArrayList;
import java.util.Iterator;

class TestGenerics1 {
    public static void main(String args[]) {
        ArrayList<String> list = new ArrayList<String>();
        list.add("Java");
        list.add("PHP");
        list.add("C++");
        // list.add(32);//compile time error

        // show list
        Iterator<String> itr = list.iterator();
        while (itr.hasNext()) {
            System.out.println(itr.next());
        }
    }
}

```

Kết quả:

```

Java
PHP
C++

```

Ví dụ về Generics trong java sử dụng Map

Bây giờ chúng ta sẽ sử dụng Map để lưu trữ dữ liệu bằng cách sử dụng generics. Ở đây, chúng ta cần phải truyền key và value:

```

package vn.viettuts.generics;

import java.util.HashMap;
import java.util.Iterator;
import java.util.Map;
import java.util.Set;

class TestGenerics2 {
    public static void main(String args[]) {
        Map<Integer, String> map = new HashMap<Integer, String>();
        map.put(1, "Java");
        map.put(4, "PHP");
        map.put(2, "C++");

        // Now use Map.Entry for Set and Iterator
        Set<Map.Entry<Integer, String>> set = map.entrySet();

        Iterator<Map.Entry<Integer, String>> itr = set.iterator();
        while (itr.hasNext()) {
            Map.Entry e = itr.next(); // no need to typecast
            System.out.println(e.getKey() + " " + e.getValue());
        }
    }
}

```

Kết quả:

```
1 Java
2 C++
4 PHP
```

5. Lợi thế của Generics trong Java

Có 3 ưu điểm chính của generics trong java, như sau:

1. Kiểu đối tượng an toàn: Chúng ta chỉ có thể lưu một kiểu đối tượng duy nhất trong generics. Nó không cho phép lưu trữ 2 đối tượng có kiểu khác nhau.

2. Không cần phải ép kiểu:

Trước Generics chúng ta cần phải ép kiểu, ví dụ:

```
List list = new ArrayList();
list.add("hello");
String s = (String) list.get(0); // ép kiểu
```

Sau Generics chúng ta **không** cần phải ép kiểu đối tượng, ví dụ:

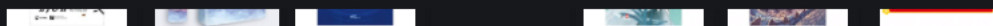
```
List<String> list = new ArrayList<String>();
list.add("hello");
String s = list.get(0);
```

3. Kiểm tra lúc biên dịch: nó kiểm tra lỗi khi biên dịch nên sẽ không bị lỗi lúc runtime.

```
List<String> list = new ArrayList<String>();
list.add("hello");
list.add(32); // Compile Time Error
```

🕒 Tùy chỉnh Annotation trong java

Xử lý ngoại lệ trong java 🕒




0 bình luận

Sắp xếp theo

Cũ nhất ↕



Thêm bình luận...

 Plugin bình luận trên Facebook

Danh sách bài học

[Học java](#)
[Học servlet](#)
[Học jsp](#)
[Học Hibernate](#)
[Học Struts2](#)
[Học Spring](#)
[Học SQL](#)

Câu hỏi phỏng vấn

[201 câu hỏi phỏng vấn java](#)
[25 câu hỏi phỏng vấn servlet](#)
[75 câu hỏi phỏng vấn jsp](#)
[52 câu hỏi phỏng vấn Hibernate](#)
[70 câu hỏi phỏng vấn Struts2](#)
[70 câu hỏi phỏng vấn Spring](#)
[57 câu hỏi phỏng vấn SQL](#)

About VietTuts.Vn

Hệ thống bài học trên VietTuts.Vn bao gồm các bài lý thuyết và thực hành về các công nghệ java và công nghệ web. Các bài lý thuyết trên hệ thống VietTuts.Vn được tham khảo và tổng hợp từ các trang <http://javatpoint.com>, <http://www.tutorialspoint.com>, <http://docs.oracle.com/en> ...