<http://www.yiibai.com/java_generics/java_generics_type_parameters.html>

[](https://jq.qq.com/?_wv=1027&k=46RCS7y)

按照约定，类型参数名称命名为单个大写字母，以便可以在使用普通类或接口名称时能够容易地区分类型参数。以下是常用的类型参数名称列表 -

* E - 元素，主要由Java集合(Collections)框架使用。
* K - 键，主要用于表示映射中的键的参数类型。
* V - 值，主要用于表示映射中的值的参数类型。
* N - 数字，主要用于表示数字。
* T - 类型，主要用于表示第一类通用型参数。
* S - 类型，主要用于表示第二类通用类型参数。
* U - 类型，主要用于表示第三类通用类型参数。
* V - 类型，主要用于表示第四个通用类型参数。

以下示例将展示上述概念的使用。

**示例**

使用您喜欢的编辑器创建以下java程序，并保存到一个文件：*TypeParameterNamingConventions.java* 中，代码如下所示 -

package com.yiibai;

import java.util.ArrayList;

import java.util.HashMap;

import java.util.List;

import java.util.Map;

public class TypeParameterNamingConventions {

public static void main(String[] args) {

MyBox<Integer, String> box = new MyBox<Integer, String>();

box.add(Integer.valueOf(199), "Hello World");

System.out.printf("Integer Value :%d\n", box.getFirst());

System.out.printf("String Value :%s\n", box.getSecond());

Pair<String, Integer> pair = new Pair<String, Integer>();

pair.addKeyValue("1", Integer.valueOf(100));

System.out.printf("(Pair)Integer Value :%d\n", pair.getValue("1"));

CustomList<MyBox> list = new CustomList<MyBox>();

list.addItem(box);

System.out.printf("(CustomList)Integer Value :%d\n", list.getItem(0)

.getFirst());

}

}

class MyBox <T, S> {

private T t;

private S s;

public void add(T t, S s) {

this.t = t;

this.s = s;

}

public T getFirst() {

return t;

}

public S getSecond() {

return s;

}

}

class Pair<K, V> {

private Map<K, V> map = new HashMap<K, V>();

public void addKeyValue(K key, V value) {

map.put(key, value);

}

public V getValue(K key) {

return map.get(key);

}

}

class CustomList<E> {

private List<E> list = new ArrayList<E>();

public void addItem(E value) {

list.add(value);

}

public E getItem(int index) {

return list.get(index);

}

}

Java

执行上面示例代码，将得到以下结果 -

Integer Value :199

String Value :Hello World

(Pair)Integer Value :100

(CustomList)Integer Value :199