

1

Java

```
ArrayList<String> arrayLi st1=new ArrayLi st<Object>();//
ArrayList<Object> arrayLi st1=new ArrayLi st<String>();//
```

```
ArrayList<Object> arrayLi st1=new ArrayLi st<Object>();
arrayLi st1.add(new Object());
arrayLi st1.add(new Object());
ArrayList<String> arrayLi st2=arrayLi st1;//
```

4

arrayList2 get() String
Object ClassCastException
Java

```
ArrayList<String> arrayLi st1=new ArrayLi st<String>();
arrayLi st1.add(new String());
arrayLi st1.add(new String());
ArrayList<Object> arrayLi st2=arrayLi st1;//
```

ClassCastException String Object arrayList2

java String arrayList2 add()
Object

2

ArrayList<double> ArrayList<Double> ArrayList
T Object Object double

3

```
ArrayList<String> arrayLi st=new ArrayLi st<String>();if( arrayLi st instanceof ArrayLi
```

ArrayList<String> String

4

```
public class Test2<T> {
    public static T one; //
    public static T show(T one){ //
        return null;
    }
}
```

```

    }
}

```

ArrayList<Integer>

```

public class Test2<T> {
    public static <T >T show(T one){//
        return null;
    }
}

```

T

T

T

1. Java

?

?

ClassCastException

2 Java

?

?

List<String>

List

3.

?

<? extends T>

T

<? super T>

T

<?>

<?>

4. List<? extends T>

List <? super T>

?

List

List<? extends T>

T

List

List<? super T>

T

List

List<?

extends Number>

List<Integer>

List<Float>

5.

?

T, E or K,V

Java

:

```

public V put(K key, V value) {
    return cache.put(key, value);
}

```

6. Java

?

JDK

7. LRU ?

Java

LinkedHashMap

LRU

LRU

LinkedHashMap

removeEldestEntry()

put() putAll()

8. List<String>

List<Object>

()

Java

String

Object List<String>

List<Object>

Java

List<Object>

String, Integer

List<String>

Strings

9. Array

?

Java

Array

Joshua Bloch

Effective Java

List

Array

List

Array

10. Java

?

Java 5 javac

```
List<String> rawList = new ArrayList()
```

```
: Hello.java
```

```
;
```

```
@SuppressWarnings("unchecked")
```

11 Java List<Object> List ?

<Object>

Object

String Integer

List

List<String>

List<Object>

12 Java List<?> List<Object>

?

List<?>

List

List<Object>

List

List<String>, List<Integer>

List<?>

List<String>

List<Object>

```
List<?> listOfType;
```

```
List<Object> listOfObject = new ArrayList<Object>();
```

```
List<String> listOfString = new ArrayList<String>();
```

```
List<Integer> listOfInteger = new ArrayList<Integer>();
```

```
listOfType = listOfString; //legal
```

```
listOfType = listOfInteger; //legal
```

```
listOfObjectType = (List<Object>) listOfString; //compiler error - in-convertible type
```

13 List<String>

List

.

1/2

1/2

List

String

Object

String

List

List

```
List listOfRawTypes = new ArrayList();
listOfRawTypes.add("abc");
listOfRawTypes.add(123); // -
String item = (String) listOfRawTypes.get(0); //
item = (String) listOfRawTypes.get(1); // ClassCastException Integer

List<String> listOfString = new ArrayList();
listOfString.add("abcd");
listOfString.add(1234); //
item = listOfString.get(0); // -
```

```
public class Test {
    public static void printIntValue(List<? extends Number> list) {
        for (Number number : list) {
            System.out.print(number.intValue()+" ");
        }
        System.out.println();
    }
    public static void main(String[] args) {
        List<Integer> integerList=new ArrayList<Integer>();
        integerList.add(2);
        integerList.add(2);
        printIntValue(integerList);
        List<Float> floatList=new ArrayList<Float>();
        floatList.add((float) 3.3);
        floatList.add((float) 0.3);
        printIntValue(floatList);
    }
}
```

2 2

3 0

```
public class Test {
    public static void fillNumberList(List<? extends Number> list) {
        list.add(new Integer(0)); //
        list.add(new Float(1.0)); //
    }
}
```

List<? extends Number>	List<Integer>	List<Float>	Integer
Float			
	List<Integer>	Integer	

list1	list2	fillNumberList
-------	-------	----------------

List<? extends T>

Number

List<? superT>	List	Object
----------------	------	--------

List<?> 1/2 1/2

```
List<Object>    List<?>
                null
```

List<Object> List<?>

List<?> list

```
public static void printList(List<Object> list) {  
    for (Object elem : list)  
        System.out.println(elem + "");  
    System.out.println();  
}
```

```
public static void printList(List<?> list) {  
    for (Object elem: list)  
        System.out.print(elem + "");  
    System.out.println();  
}
```

List<Object> ,

```
List<Integer> li = Arrays.asList(1, 2, 3);  
List<String> ls = Arrays.asList("one", "two", "three");  
printList(li);  
printList(ls);
```