

请大家阅读文档时，在视图里勾选导航窗格，在左边显示章节目录方便浏览。

一、填空题

1. 语句 `Class clz = null;` 的含义是__**clz 是 Class 类型的空引用，没有指向任何 Class 类型的对象**__。

2. 给定下列类的定义：

```
class GeometricObject {}
class Polygon extends GeometricObject {}
class Rectangle extends Polygon {}
GeometricObject o = new Rectangle ();
Class clz1 = o. getClass ();
```

(1) 声明一个指向 Polygon 及其子类的类型信息的引用变量 clz 的语句应该是

_____ **Class<? extends Polygon> clz4;** _____;

(2) `System.out.println(o.getClass().getSimpleName());`的输出结果是__**Rectangle**__;

(3) 下列语句中有错误的是__**②_③**__;

```
Class<Polygon> clz3 = null;
```

```
clz3 = Polygon.class;           ①
```

```
clz3 = Rectangle.class;         ②
```

```
Class<? extends Polygon> clz4 = null;
```

```
clz4 = GeometricObject.class;   ③
```

```
clz4 = Polygon.class;           ④
```

```
clz4 = Rectangle.class;         ⑤
```

错误原因是（按错误题号解释） _____

__**②错误原因是 clz3 只能指向 Polygon 类的 Class 对象**_____

__**③错误的原因是 clz4 只能指向 Polygon 及其子类的 Class 对象**_____。

3. 下面五条语句中，错误的有__**(2)_ (3)**_____。

(1) `ArrayList<String> lists = new ArrayList<String>();`

(2) `ArrayList<Object> lists = new ArrayList<String>();`

(3) `ArrayList<String> lists = new ArrayList<Object>();`

(4) ArrayList<String> lists = new ArrayList();

(5) ArrayList lists = new ArrayList<String>();

错误原因是 泛型没有协变性。

使用泛型通配符?将错误的语句修改正确的方法是_____

___ **(2) ArrayList<? extends Object> list2 = new ArrayList<String>();** _____,

___ **(3) ArrayList<? super String> list3 = new ArrayList<Object>();** _____。

4. 下面代码给出了泛型类和非泛型类的定义:

```
class Holder<T> {  
    T value;  
    public Holder (T value) {this.value = value;}  
    public T getValue () {return value;}  
}
```

```
class RawHolder {  
    Object value;  
    public RawHolder (Object value) {this.value = value;}  
    public Object getValue () {return value;}  
}
```

基于上面二个类的定义, 有下面四段代码:

<p>①</p> <pre>Holder<String> h1 = new Holder<>("aaa"); String s1 = h1. getValue (); System.out.println(s1);</pre>	<p>②</p> <pre>RawHolder h1 = new RawHolder("aaa"); String s1 = (String)h1. getValue (); System.out.println(s1);</pre>
<p>③</p> <pre>Holder<String> h1 = new Holder<> (Integer.valueOf(111)); String s1 = h1. getValue (); System.out.println(s1);</pre>	<p>④</p> <pre>RawHolder h1 = new RawHolder (Integer.valueOf(111)); String s1 = (String)h1. getValue (); System.out.println(s1);</pre>

上面四段代码中编译通过运行不出错的是 ①② _____,

上面四段代码中编译通过运行出错的是 ④ _____, 原因是 运行时 Integer 类型转 String 类型抛出异常, 编译时检查不出这种类型不一致 _____,

上面四段代码中编译不通过是 ③ _____, 原因是 h1 的声明类型是 Holder<String>, 但是构

造对象时构造函数的实参类型是 `Integer`，编译器可以检查出类型的不一致

_____，

这个例子说明泛型的作用是____在编译时检查出类型不一致,避免运行时发生类型转换异常_____。

二、单项选择题

1. 泛型参数<T>代表的是_____ **D** _____。
A. 任意类型
B. 某类型的子类型
C. 某类型的父类型
D. 固定指代某种类型
2. 泛型通配符<?>代表的是_____ **A** _____。
A. 任意类型
B. 某类型的子类型
C. 某类型的父类型
D. 固定指代某种类型
3. 下面泛型定义中不正确的是_____ **D** _____。
A. `class Test1<T> {}`
B. `interface Test2<T> {}`
C. `class Test3{<T> void test () {}}`
D. `class Test4{void <T> test () {}}`
4. 泛型通配符<? extends T>代表的是_____ **B** _____。
A. 任意类型
B. 某类型 T 的子类型
C. 某类型 T 的父类型
D. 固定指代某种类型
5. 泛型通配符<? super T>代表的是_____ **C** _____。
A. 任意类型
B. 某类型 T 的子类型
C. 某类型 T 的父类型
D. 固定指代某种类型

6. 关于下面代码, 描述正确的是___C___。

```
List<String> list = new ArrayList<String>();  
list.add("test");  
list.add("red");  
list.add(100);  
System.out.println(list.size());
```

- A. 输出 2
- B. 输出 3
- C. 编译错误
- D. 运行时报异常

7. 关于下面代码, 描述正确的是___B___。

```
List<Integer> ex_int= new ArrayList<Integer> ();  
List<Number> ex_num = ex_int;  
System.out.println(ex_num.size());
```

- A. 0
- B. 编译错误
- C. 运行时报异常
- D. 1

8. 下列语句编译时不出错的是___D___。

- A. List<?> c1 = new ArrayList<String> (); c1.add (new Object ());
- B. List<?> c2 = new ArrayList<String> (); c2.add (new String ("1"));
- C. List<?> c3 = new ArrayList<String> (); c3.add ("1");
- D. List<?> c4 = new ArrayList<String> (); c4.add(null);

9. 给定下列代码:

```
class Shape {}  
class Circle extends Shape {}  
class Triangle extends Shape {}  
public class Test2_9 {  
    public static void main (String [] args) {  
        List<? extends Shape> list1 = new ArrayList< Triangle> ();  
        List<? extends Shape> list2 = new ArrayList<Circle> ();  
  
        System.out.println(list1 instanceof List< Triangle>); ①  
        System.out.println(list2 instanceof List); ②  
        System.out.println(list1.getClass() == list2.getClass()); ③  
    }  
}
```

则关于语句①②③说法正确的是: ___D___。

- A. ①②③输出结果为 true、false、false

- B. ①②③输出结果为 true、true、true
- C. ①编译出错, ②③输出结果为 false、false
- D. ①编译出错, ②③输出结果为 true、true

三、多项选择题（一个或多个正确选项）

1. 对于泛型类 `class A<T> { ... }`, `T` 在 `A` 类里可以用作不同的地方, 在 `A` 类类体内, 下面语句正确的有 ABDG。

- A. `T x;`
- B. `T m1() {return null;}`
- C. `static T y;`
- D. `void m2(T i) {}`
- E. `static T s1() {return null;}`
- F. `static void s2(T i) {}`
- G. `static <T1> void s3(T1 i, T1 j){}`

2. 下列语句编译时不出错的是 AEFH。

- A. `List<? super Integer> x1 = new ArrayList<Number> ();`
- B. `List<? super Number> x2 = new ArrayList<Integer> ();`
- C. `List<? super Number> x3 = new ArrayList<Short> ();`
- D. `List<? super Integer> x4 = new ArrayList<Short> ();`
- E. `List<? extends Number> x5 = new ArrayList<Integer> ();`
- F. `List<? extends Number> x6 = new ArrayList<Object> ();`
- G. `List<Number> x7 = new ArrayList<> ();`
- H. `List<? extends Comparable<Double>> x8 = new ArrayList<Double> ();`
- I. `List<? extends Number> x9 = new ArrayList<int> ();`

3. 下面泛型类是 `List<?>` 的子类的是 ABCD。

- A. `List<String>`
- B. `List<Object>`
- C. `List<Integer>`
- D. `List<Float>`

4. 泛型参数应该写的位置是 BD。

- A. 类名前
- B. 类名后
- C. 方法名前
- D. 方法返回值类型前

5. 关于 java 泛型, 下面描述正确的是 ABCD。

- A. 泛型的类型参数只能是类类型（包括自定义类），不能是基本类型
- B. 泛型的类型参数可以有多个

C. 不能对泛型的具体实例类型使用 instanceof 操作, 如 o instanceof ArrayList<String>, 否则编译时会出错。

D. 不能创建一个泛型的具体实例类型的数组, 如 new ArrayList<String>[10], 否则编译时会出错。

6. 给定下列类和泛型方法的定义:

```
class A {}
class B extends A {}
class C extends B {}
class D extends C {}
public class Test2_9 {
    public static <T> void m (List<? super T> list1, List<? extends T> list2) {}
}
```

则下面 6 段代码编译出错的是 CEF。

A.

```
List<B> l1 = new ArrayList<> ();
List<B> l2 = new ArrayList<> ();
Test2_9.m (l1, l2);
```

B.

```
List<B> l3 = new ArrayList<> ();
List<D> l4 = new ArrayList<> ();
Test2_9.m (l3, l4);
```

C.

```
List<B> l5 = new ArrayList<> ();
List<A> l6 = new ArrayList<> ();
Test2_9.m (l5, l6);
```

D.

```
List<C> l7 = new ArrayList<> ();
List<D> l8 = new ArrayList<> ();
Test2_9.m (l7, l8);
```

E.

```
List<C> l7 = new ArrayList<> ();
List<D> l8 = new ArrayList<> ();
Test2_9.<B>m (l7, l8);
```

F.

```
List<D> l9 = new ArrayList<> ();
List<C> l10 = new ArrayList<> ();
Test2_9.m (l9, l10);
```


四、问答题

阅读下列程序，并填写表格

```
import java.util.*;
class A {}
class B extends A {}
class Test {
    public static void m1(List<? extends A> list) {}
    public static void m2(List<A> list) {}
    public static void m3(List<? super A> list) {}
    public static void main (String [] args) {
        List<A> listA = new ArrayList<A> ();
        List<B> listB = new ArrayList<B> ();
        List<Object> listO = new ArrayList<Object> ();

        // insert code here
    }
}
```

在上面代码插入点插入的代码	结果（从下面结果选项中选择）
m1(listA);	C 编译通过，运行不出错
m2(listA);	C 编译通过，运行不出错
m3(listA);	C 编译通过，运行不出错
m1(listB);	C 编译通过，运行不出错
m2(listB);	A 编译出错
m3(listB);	A 编译出错
m1(listO);	A 编译出错
m2(listO);	A 编译出错
m3(listO);	C 编译通过，运行不出错
结果选项	
A. 编译出错	
B. 编译正确，运行出错	
C.编译正确，运行正确	