



厦门大学《Java 程序设计》课程试卷

软件学院软件工程系年级软件工程专业

主考教师：王美红 试卷类型：(A 卷/B 卷)

1、 Single choice(10 points)

Identify the letter of the choice that best completes the statement or answers the question.

(1) The _____ class is inherited by every Java class.

- A. Class
- B. Object
- C. Number
- D. Comparable

(2) A subclass inherits _____ from its superclass.

- A. private method
- B. protected method
- C. A and B
- D. public method

E. B and D

(3) Which of the following is true about RuntimeException and its subclasses?

- A. If a method throws a RuntimeException, the use of the try/catch block is optional.
- B. The IOException class is a subclass of RuntimeException.
- C. In general, handling of RuntimeException should be done at compile time.
- D. In general, RuntimeException must be caught with a try/catch block.

(4) Analyze the following code.

```
class Test {  
    public static void main(String[] args) {  
        Object x = new Integer(2);  
        System.out.println(x.toString());  
    }  
}
```

- A. The program has syntax errors because an Integer object is assigned to x.
- B. When x.toString() is invoked, the toString() method in the Object class is used.
- C. When x.toString() is invoked, the toString() method in the Integer class is used.
- D. None of the above.

(5) Show the output of running the class Test in the following code:

```
interface A {  
    void print();  
}
```

```
class C {}
```

```
class B extends C implements A {
```

```
public void print() { }
}
```

```
public class Test{
    public static void main(String[] args) {
        B b = new B();
        if (b instanceof A)
            System.out.println("b is an instance of A");
        if (b instanceof C)
            System.out.println("b is an instance of C");
    }
}
```

- A. Nothing.
- B. b is an instance of A.
- C. b is an instance of C.
- D. b is an instance of A followed by b is an instance of C.

- (6) Suppose s is a string with the value "java". What will be assigned to x if you execute the following code?

```
char x = s.charAt(4);
```

- A. 'a'
- B. 'v'
- C. Nothing will be assigned to x, because the execution causes the runtime error `StringIndexOutOfBoundsException`.
- D. None of the above.

- (7) Which of the following data types does not implement the Collection interface?

- A. HashSet
- B. ArrayList
- C. Map
- D. TreeSet
- E. LinkedList

- (8) Which of the following statement will not result in compilation error?

- A. `import java.awt.*;`
`package Mypackage;`
`class myclass{}`
- B. `class myclass()`
`package mypackage;`
- C. `package mypackage;`
`import java.awt.*;`
`class myclass{}`
- D. `class myclass{}`
`import java.awt.*`

- (9) What is the output of the following code?

```
int a = 1, b = 2, c = 3;
a += b + c;
b += a + c;
```

```
c += a + b;
```

```
System.out.println(a + " " + b + " " + c);
```

A. 3 5 6

B. 5 4 3

C. 5 8 13

D. 6 11 20

E. 3 3 4

(10) If a method does not have a return statement, then

A. It will produce a syntax error when compiled.

B. it must be a void method.

C. it can not be called from outside the class that defined the method.

D. it must be defined to be a public method.

E. it must be an int, double, or String method.

2、 Multiple choice(10 points)

Identify the letters of the choices that best complete the statement or answer the question.

(1) Which of the following types can be used as the parameter for a switch statement?

A. char

B. boolean

C. byte

D. int

(2) What are not the main differences between a class and an interface in J2SE 8.0?

A. Classes can declare variables; interfaces cannot.

B. Classes must be public; interfaces can be public or private.

C. Classes give method definitions; interfaces only give the names and argument.

D. Classes can be used as data types for variables; interfaces cannot.

(3) Which of the following assignments are legal?

A. p = ug C. ug = new Student();

B. p = new Undergraduate();

C. ug = p;

D. s = new Person();

(4) which of the following statements correctly creates an array of five empty String?

A. String a[] = new String[5];

for(int i=0; i<5;i++)

a[i] = "";

B. String a[] = {"", "", "", "", ""};

C. String a[5];

D. String[5] a;

E. String[] a = new String[5];
a = {"", "", "", "", ""};

Array constants can only be used in initializers

(5) which statements will not result in compilation error in J2SE 8?

A. (int x, int y) -> {return x + y;}

B. (x, y) -> {return x + y;}

C. (x, y) -> x + y

D. value -> System.out.printf("%d ", value) 可能需要加一个 ;

3、 True/False(10 points)

Indicate whether the sentence or statement is true or false.

- (1) If myArray is an array variable, it is possible to make the array grow or shrink(缩小) by changing the value of *myArray.length*. F, 可以reassigning it to the new one
- (2) Local variables and parameters can be *final*, but they cannot be *static*. T
- (3) A *try* block can only have one *catch* clause(语句). F
- (4) *NullPointerException* and *ArrayIndexOutOfBoundsException* are checked exceptions(受控异常). F
- (5) Overloaded methods can have different return values, and must have different parameter lists. Two methods differing only by return type will result in a compilation error. T
- (6) A lambda expression represents an anonymous(匿名) method—a shorthand notation (符号) for implementing a functional interface. T
- (7) A generic class declaration looks like a nongeneric class declaration, except that the class name is followed by a type-parameter section. T
- (8) Class *Number* is the superclass of both *Integer* and *Double*. T
- (9) A *Set* is an ordered *Collection* that contains no duplicate elements. F, unordered
- (10) An *Arrays* is similar to an array but can be dynamically resized. T, 静态的

4、 Short answer(40 points)

- (1) (4 points) what's *Functional Interfaces* ?
- (2) (4 points) what's *dynamic binding*?
- (3) (6 points) what's the difference between *reference type* and *primitive type*?
- (4) (6 points) Please list the new features of *interface* in j2SE 8.
- (5) (6 points) what's the difference between keywords *throw* and *throws* ?
- (6) (6 points) what's the difference between *List*, *Set* and *Map*?
- (7) (4 points) when and how to use wildcard (?) in the method's header?
- (8) (4 points) Suppose A is an abstract class, what is wrong in the following code?

```
A[] list = new A[10];
```

```
list[0] = new A();
```

5、Completion(30 points)

(9) (2 points) A(n) 省略 in the parameter list of a method indicates that the method can receive a variable number of arguments.

for(double d : numbers)

(10) (3 points) Use the enhanced for statement _____ to walk through double array *numbers*.

(11) (2 points) 123 is displayed by System.out.println("1" + new Integer(2) + 3);

(12) (2 points) To get an iterator from a set, you may use the iterator method.

(13) (4 points) Consider the following code segment:

```
List<String> list = new LinkedList<String>();
list.add("["); list.add("A"); list.add("]");
System.out.println(list);
ListIterator<String> it = list.listIterator();
while(it.hasNext())
{
    if ("[".equals(it.next()) || "].equals(it.next()))
        it.remove();
    else
        it.add("*");
}
System.out.println(list);
```

The first output line is [[, A,]]

The Second output line is [A]

(14) (2 points) What is the printout for the following code? x[0] is 0

```
class Test {
    public static void main(String[] args) {
        int[] x = new int[3];
        System.out.println("x[0] is " + x[0]);
    }
}
```

属性

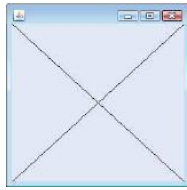
(15) (2 points) An attribute that is shared by all objects of the class is coded using a static method.

(16) (8 points) Design a method *duplicate(List L)*, which, when given a list of objects, can return the set of all objects which occur more than one time in L.

```
static <E> Set<E> duplicate(List<? extends E> L){
```

```
}
```

(17) (5 points) Fill in the blanks so that the program can run as shown in the following figure.



```
//DrawPanelTest.java
import javax.swing.JFrame;
public class DrawPanelTest
{
    public static void main(String[] args)
    {
        DrawPanel panel = new DrawPanel();
        JFrame application = new JFrame();
        application.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
        application.add(panel);
        application.setSize(250, 250);
        application.setVisible(true);
    }
} // end class DrawPanelTest

//DrawPanel.java
import java.awt.Graphics;
import javax.swing.JPanel;
public class DrawPanel extends JPanel
{
    // draws an X from the corners of the panel
    _____

} // end class DrawPanel
```