

# 2017~2018 学年秋冬学期 《Java 应用技术》 期末试卷

## 一、判断题 (1% × 10)

1. JPanel must be placed inside a container. ✓
2. We can use `int a[][] = new int[2][]` to define array. ✓
3. For `final int[] ar = new int[10]` we cannot modify the content of the array. ✓
4. Interface methods can be static or final. ✗
5. When an object can be written to a stream using `ObjectOutputStream`, we can also use `ObjectOutputStream` to write the object of its super class. ✗
6. Box does not use the default layout `BorderLayout`. ✓
7. A static method cannot refer to `this` or `super` keywords in anyway. ✓
8. Private members of class can be inherited by a sub class, and become protected members in sub class. ✗
9. If constructor of class A is made private, objects of class A can be instantiated only within the class where it is declared. ✓
10. A thread that has called the `wait()` method of an object will release the lock of the object. ✓

## 二、单选题 (2% × 30)

- A 1. For object `o` and class `C`, which expression below is the right way to test if `o` is an object of `C`?

- ☒ A. `o instanceof C` ✓  
B. `C.isInstance(o)`  
C. `o.getClass() == C`  
D. `o.class == C`

2. Which one below is true about the `StringBuffer` class?

- ☒ A. An object of `StringBuffer` can be initialized using the `=` operator.  
☒ B. `StringBuffer` has `append()` method to form a larger string.  
C. An object of `StringBuffer` has a fixed size.  
D. `StringBuffer` inherits all the methods from `String`.

3. What will this code print?

```
String arr[] = new String[5];  
System.out.print(arr[0]);
```

- ☒ A. null  
☒ B. 0

- C. Class name@hashCode in hexadecimal form.

D. Exception thrown.

4. `wait()` and `notify()` are used to suspend and resume threads. They are defined as methods of:

☒ A. `Object`

B. `Thread`

C. `Runnable`

D. `Synchronized`

5. What best describes the appearance of an application with the following code?

```
public class App extends JFrame {
    public static void main(String argv[]) {
        App app = new App();
        app.setLayout(new FlowLayout());
        app.pack();
        app.setVisible(true);
    }

    App() {
        add(new JButton("One"));
        add(new JButton("Two"));
        add(new JButton("Three"));
        add(new JButton("Four"));
    }
}
```

☒ A. A frame with buttons marked One to Four placed one by one.

B. A frame with buttons marked One to Four placed in grids.

C. A frame with buttons marked One to Four placed at each edge.

D. A frame with one large button marked Four in the center.

6. Choose the best fill in the blanks.

```
class Hello2017 {
    public static void main(String[] args) {
        // __put the best here__
    }
}

class Century implements Runnable {
    String m = "Hello";

    Century(String m) {
        this.m = m;
    }

    public void run() {
        System.out.println(m);
    }
}
```

A. `new Thread("Hello").start();`

B. `new Century(new Thread("Hello")).start();`

C. `new Century("Hello").start();`

☒ D. `new Thread(new Century("Hello")).start();`

7. For code below:

```
public class Test {  
    public static void main(String[] args) {  
        try {  
            throw new B();  
        } catch (A a) {  
            System.out.println("Exception A");  
        } catch (B b) {  
            System.out.println("Exception B");  
        }  
    }  
}  
  
class A extends Exception {  
}  
  
class B extends A {  
}
```

It prints:

A. Exception B

☒ B. Compile error

C. Exception A

D. Compiled but exception raises at run-time

8. For code

```
int x = 0x80000000;  
System.out.println(Integer.toHexString(-x));
```

The result is:

A. overflow

B. -80000000

☒ C. 80000000

D. error (compilation or run-time)

9. Which of the following is NOT correct?

A. Cannot create an instance of a generic type. (i.e., `new E()` ).

B. Generic array creation is not allowed. (i.e., `new E[100]` ).

C. A generic type parameter of a class is allowed in a static context.

☒ D. Exception classes cannot be generic.

10. Which statement below is NOT correct?

A. A thread is an instance of `Thread` class.

B. A thread runs the `run()` method of the `Runnable` object.

☒ C. A new born thread can run immediately when `start()` is called.

D. Thread can access data of the `Runnable` object.

11. For `InputStream.read()`, the `read()` with no parameters, which statement below is correct?

A. `read()` returns `char`, because it reads a char from the stream.

☒ B. `read()` returns `int`, because it has to return EOF to indicate the end of the file.

C. `read()` returns `byte`, because it reads a byte from the stream.

D. `read()` returns `int`, as the number of bytes it just read.

12. Implements `Comparable` needs a function, (\_\_\_) is the one.

```
class Hello2016 implements Comparable {  
    public static void main(String[] args) {  
        }  
        // __put the best here__  
    }  
}
```

☒ A. `public int compareTo(Object b) {...}`

B. `public int equals(Object b) {...}`

C. `public int compare(Object b) {...}`

D. Need nothing for `Comparable`.

13. For code below:

```
ArrayList<Integer> a = new ArrayList<Integer>();  
ArrayList<Double> b = new ArrayList<Double>();
```

Which statement below is NOT correct?

A. `a.getClass() == b.getClass()` is true.

B. `a instanceof ArrayList` is true.

C. `a.getClass().equals(b.getClass())` is true.

☒ D. `a.getClass() == b.getClass()` is false.

14. What is the output of this program?

```
public class Output {  
    public static void main(String args[]) {  
        Integer i = new Integer(257);  
        byte x = i.byteValue();  
        System.out.print(x);  
    }  
}
```

☒ A. 1

B. 0

C. 256

D. 257

15. Which of these method waits for the thread to terminate?

A. `isAlive()`

B. `sleep()`

☒ C. `join()`

D. `stop()`

16. Which of the following declares an array that can support two rows and a variable number of columns?

A. `int myArray[][] = new int[2][];`

B. `int myArray[][] = new int[][2];`

C. `int myArray[][] = new int[2][2];`

D. `int myArray[][] = new int[][];`

17. The program needs a thread, (\_\_\_) is the one.

```
class Hello2016 {  
    public static void main(String[] args) {  
        // ___put the best here___  
    }  
}
```

A. `new Runnable(() -> System.out.println("Hi, 2017")).start();`

B. `new Thread(() -> System.out.println("Hi, 2017")).start();`

C. `new Thread(() -> System.out.println("Hi, 2017")).run();`

D. `new Runnable(() -> System.out.println("Hi, 2017")).run();`

18. For code below, the result would be printed?

```
String s1 = new String("hello");  
String s2 = new String("hello");  
System.out.println(s1 == s2);  
System.out.println(s1.equals(s2));
```

A. false, true

B. false, false

C. true, true

D. true, false

19. Given code below:

```
package his;  
public class My {}
```

Which statement below is NOT correct?

A. It has to be in a directory named `his`.

B. It has to be in a file named `My.java`.

C. It can be in any file but with no any other class definitions in the same file.

D. Any non-public classes can be defined in the same source file as it is in.

20. What is the output of the following code?

```

public class Test {
    public static void main(String[] args) {
        LinkedList list = new LinkedList<Integer>();
        for (int i = -3; i < 3; i++) {
            list.add(i);
        }
        for (int i = 0; i < 3; i++) {
            list.remove(i);
        }
        System.out.println(list);
    }
}

```

- ☒ A. [-2, 0, 2]
- B. [-3, -2, -1]
- C. [0, 1, 2]
- D. [-1, 0, 1]

21. What is the output of this program?

```

public class Output {
    public static void main(String args[]) {
        StringBuffer sb = new StringBuffer("Hello");
        sb.replace(1, 3, "Java");
        System.out.println(sb);
    }
}

```

- A. HJavaello
- B. HJavalo
- C. Hello
- D. HJavao

22. Which one below is NOT a valid Java identifier?

- ☒ A. goto
- B. Int
- C. 变量
- D. \$0

23. Which one below generates a random number in [1, 100]?

- ☒ A. `x = (int) (101 * Math.random()) + 1;`
- B. `x = (int) (100 * Math.random()) + 1;`
- C. `x = (int) (100 * Math.random());`
- D. `x = (int) (101 * Math.random());`

24. About inner class, which statement below is correct?

- ☒ A. No static members are allowed in an inner class. *static*
- B. Inner class cannot be defined as private.
- C. Objects of an inner class can be used in the outer class only.
- D. Inner class can access every member of the outer class.

25. Which component is used to compile, debug and execute java program?

- A. JVM
- ☒ B. JDK
- C. JIT
- D. JRE

26. Given the following code:

```
class Background extends Thread {  
    public int run() {  
        while (true) {  
            System.out.println("hello");  
        }  
        return 0;  
    }  
  
    public static void main(String[] args) {  
        new Thread(new Background()).start();  
    }  
}
```

What will happen when you attempt to compile and run the code?(2分)

- A. It compiles and prints out nothing.
- ☒ B. It does not compile because of the function signature of `run()`.
- C. It compiles and prints out "hello" repeatedly.
- D. It does not compile because of the expression inside `main()`.

27. What is the output of this program?

```
public class Test {  
    public static void main(String[] args) throws Exception {  
        String str = "zju2018";  
        Method m = str.getClass().getMethod("toUpperCase");  
        m.invoke(str);  
        System.out.println(str);  
    }  
}
```

- A. compilation error
- B. ZJU2018
- ☒ C. zju2018
- D. runtime error

28. What is the output of this program?

```

public class Hello2017 {
    public static void main(String args[]) {
        boolean b1 = true;
        if ((b1 == true) || place(true)) {
            System.out.print("Hello01, ");
        }
        System.out.println("HelloWorld.");
    }

    public static boolean place(Boolean location) {
        if (location == true) System.out.print("Hello02, ");
        if (location = true) System.out.print("Hello03, ");
        return location;
    }
}

```

- A. HelloWorld.
- B. Hello01, HelloWorld.
- C. Hello02, Hello01, HelloWorld.
- D. Hello02, Hello03, Hello01, HelloWorld.

29. Which of the following statements is NOT true?

- A. Strings can be initialized using the `=` operator with a string literal value.
- B. The `toString()` method can be used to return a `String` value from an object of any class.
- C. All strings are terminated with a null (`'\0'`) character.
- D. It is impossible to change the contents of a `String` object.

30. Given code below:

```

List<String> ls = new ArrayList<String>();
List<Object> lo = ls;
lo.add(new Object());
String s = ls.get(0);

```

Which statement below is correct?

- A. It compiles but exception raises at line 3
- ☒ B. It does not compile
- C. It compiles but exception raises at line 2
- D. It compiles but exception raises at line 4

### 三、填空题 (3% × 10)

1. What will this code output?



```
public class Test {  
    public static void main(String[] args) {  
        Double a = new Double(127);  
        Double b = 127d;  
        Double c = Double.valueOf("127");  
  
        System.out.println(a == b); //1  
        System.out.println(a == c); //2  
        System.out.println(b == c); //3  
    }  
}
```

The output of //1 is: (1%)

The output of //2 is: (1%)

The output of //3 is: (1%)

2. The code below will print three lines.

```

class Pet {
}

class Dog extends Pet {
}

class Pug extends Dog {
}

class Cat extends Pet {
}

class Rodent extends Pet {
}

class Gerbil extends Rodent {
}

class Hamster extends Rodent {
}

class Main {
    static HashMap<Integer, Class<? extends Pet>> map =
        new HashMap<Integer, Class<? extends Pet>>();

    static {
        map.put(Pet.class.getName().length(), Pet.class);
        map.put(Dog.class.getName().length(), Dog.class);
        map.put(Pug.class.getName().length(), Pug.class);
        map.put(Cat.class.getName().length(), Cat.class);
        map.put(Rodent.class.getName().length(), Rodent.class);
        map.put(Gerbil.class.getName().length(), Gerbil.class);
        map.put(Hamster.class.getName().length(), Hamster.class);
    }

    public static void main(String[] args) {
        for (Integer i : map.keySet())
            System.out.println(map.get(i).getName());
    }
}

```

1st line: (1%)

2nd line: (1%)

3rd line: (1%)

3. What will this code output?

```

class M {
    void f(M m) {
        System.out.println("in M.f");
    }

    void g(M m) {
        System.out.println("in M.g");
    }
}

class C extends M {
    void f(C c) {
        System.out.println("in C.f");
    }

    void g(M c) {
        System.out.println("in C.g");
    }
}

class H extends C {
    void f(H h) {
        System.out.println("in H.f");
    }

    void g(M h) {
        System.out.println("in H.g");
    }
}

public class T {
    public static void main(String[] args) {
        M h = new H();
        C c = new H();
        c.f(h); //1
        h.g(c); //2
    }
}

```

The output of //1 is: (1%)

The output of //2 is: (2%)

4. What will this code output?

```

public class Test {
    public static void main(String[] args) {
        String s1 = "ZJU";
        String s2 = new String("ZJU");
        String s3 = "ZJ";
        s3 += "U";
        String s4 = s2.intern();

        System.out.println(s1 == s2); //1
        System.out.println(s1 == s3); //2
        System.out.println(s1 == s4); //3
    }
}

```

The output of //1 is: (1%)

The output of //2 is: (1%)

The output of //3 is: (1%)

5. The value of the expression below is:

```

IntStream.range(2, 20)
    .filter(x -> IntStream.range(2, x).filter(k -> x % k == 0).sum() > 0)
    .sum()

```

6. For the code segment below, after all the lines here, the value of sum is:

```

Integer[] a = {1, 2, 3, 4, 5, 6, 7, 8, 9, 10};
for (int k : a) {
    k++;
}
int sum = 0;
for (int k : a) {
    sum += k;
}

```

7. The output of the code below is:

```

enum A {
    JAN(31), FEB(28) {
        public int getDays(int year) {
            return (year % 400 == 0 || (year % 4 == 0 && year % 100 != 0)) ? 29 : 28;
        }
    }, MAR(31), APR(30), MAY(31), JUN(30), JUL(31), AUG(31), SEP(30), OCT(31),
    NOV(30), DEC(31);

    A(int d) {
        days = d;
    }

    private int days;

    public int getDays(int year) {
        return days;
    }

    public static void main(String[] args) {
        int sum = 0;
        for (A e : A.values()) {
            sum += e.getDays(2018);
        }
        System.out.println(sum);
    }
}

```

8. The code below will print three lines, they are:

```

package hello;

class A {
    public int data = 5;
    private int pd = 6;

    public void print() {
        System.out.println(data + pd);
        f();
    }

    protected void f() {
        System.out.println("A::f()");
    }
}

class B extends A {
    public int data = 2;
    private int pd = 3;

    public void print() {
        super.print();
        System.out.println(data + pd);
    }

    protected void f() {
        System.out.println("B::f()");
    }
}

public class TestAB {
    public static void main(String[] args) {
        A a = new B();
        a.print();
    }
}

```

1st line: (1%)

2nd line: (1%)

3rd line: (1%)

9. For code below, the output should be:

```
static void f() throws Exception {  
    throw new RuntimeException();  
}  
  
public static void main(String[] args) {  
    try {  
        f();  
        System.out.print("A");  
    } catch (RuntimeException ex) {  
        System.out.print("B");  
    } catch (Exception ex1) {  
        System.out.print("C");  
    } finally {  
        System.out.print("D");  
    }  
    System.out.print("E");  
}
```

10. What will this code output?

```

public class Test {
    public static void main(String[] args) {
        CloneT c = new CloneT();
        CloneT c1 = (CloneT) c.clone();
        c1.b.setA(3);
        c1.ii = 3;
        System.out.println(c1 == c); //1
        System.out.println(c1.b == c.b); //2
        System.out.println(c.toString() + c1.toString()); //3
    }
}

class Base implements Cloneable {
    int a = 1;

    public String toString() {
        return String.valueOf(a);
    }

    public void setA(int a) {
        this.a = a;
    }

    public int getA() {
        return a;
    }
}

class CloneT implements Cloneable {
    transient int i;
    private int pi;
    static int num;
    Integer ii = new Integer(1);
    transient Base b = new Base();

    public CloneT() {
        num++;
    }

    public Object clone() {
        try {
            return super.clone();
        } catch (CloneNotSupportedException e) {
            System.out.println("clone not supported!");
            return null;
        }
    }

    public String toString() {
        return (
            String.valueOf(i) + String.valueOf(pi) + String.valueOf(num) +
            String.valueOf(ii) + String.valueOf(b.getA())
        );
    }
}

```



```
}  
}
```

The output of //1 is: (1%)

The output of //2 is: (1%)

The output of //3 is: (1%)

## 答案

一、 TTFFF TTFTT

二、 ABAAA DBCCC BADAC ABACA BABDB BCBCB

三、 (每空答案以 / 分隔)

1. false / false / false
2. Cat / Gerbil / Hamster
3. in M.f / in H.g
4. false / false / true
5. 112
6. 55
7. 365
8. 11 / B::f() / 5
9. BDE
10. false / true / 0011300133