判断题

1. Each container must have a layout manager. (1分)
2. All member variables in an interface are `static`. (1分)
3. A static variable in an object can not be serialized. (1分)
4. In MVC architecture, Controller will notifiy the change of data to the listeners. (1分)
5. A thread object has a method called wait(). (1分)
6. A method declared static cannot access non-static class members. A static method does not have `this` reference because static class variables and static methods exist independent of any objects of a class. (1分)
7. Constructors are called before static variables are initialized. (1分)
8. There are no derived classes of a `final` class. (1分)
9. We can use `array.size()`to get array’s length. (1分)
10. FileOutputStream cannot write an `int` directly to a file.(1分)

选择题

1. Which one below is a feature of Java? (2分)

   A. Operator overloading

   B. Generic programming

   C. Virtual inheritance

   D. Multi-inheritance

2. About JIT JVM, which statement below is correct? (2分)

   A. Java Native Invoking is the way JIT JVM does to make program run faster.

   B. There is a special translator to convert byte code into native code before its loading into the JVM.

   C. It is the JVM that translates the byte code into native during the loading of the class.

   D. The compiler generates native code for JIT JVM.

3. The program needs a Thread, (*__*) is the one.。(2分)

```
class Hello2016
{    public static void main(String[] args)
 {
     (__//put the best here__);
 }
}
```

1. new Runnable(run()-> System.out.println("Hi,2017")).start();
2. new Thread(run()-> System.out.println("Hi,2017")).start();
3. new Runnable(()-> System.out.println("Hi,2017")).start();
4. new Thread(()-> System.out.println("Hi,2017")).start();

4. `synchonize` needs a "lock" object. Object of which class can be used as that lock? (2 分)

1. Synchronized
2. Object
3. Lock
4. Thread

5. Which switch-case below is NOT correct? (2 分)

1. boolean b; switch (b) { case...}
2. char c; switch (c) { case...}
3. String s; switch (s) { case...}
4. int i; switch (i) { case...}

6. Object of which class below can be real parameter of constructor of DataInputStream? ( ) (2 分)

1. FileOutputStream
2. FilterInputStream
3. String
4. File

7. Given:

```Java
Integer.valueOf(1).equals(Long.valueOf(1))
```

which statement below is right? (2 分)

1. It throws exception in run-time.
2. The value is true.
3. The value is false.
4. It doesn’t compile.

8. What does the following line of code do? quantity = new int[3]; .分值为2分。 (2分)

1. Creates storage for unlimited int
2. Creates storage for unlimited doubles.
3. Creates storage for three int.
4. Creates storage for three doubles.

9. Given code below:

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

Which statement below is correct? (2分)

1. It compiles but exception raises at line 3
2. It compiles but exception raises at line 4
3. It compiles but exception raises at line 2
4. It does not compile

10. Which one below generates a random number in [0,50)? (2分)

1. x=(int)(51*Math.random())+1;
2. x=(int)(51*Math.random());
3. x=(int)(50*Math.random());
4. x=(int)(50*Math.random())+1;

11. What best describes the appearance of an application with the following code? (2分)

```Java
public class FlowAp extends Frame{
    public static void main(String argv[]){
        FlowAp fa=new FlowAp();
```

```
        fa.setLayout(new FlowLayout());
        fa.add(new Button("One"));
        fa.add(new Button("Two"));
        fa.add(new Button("Three"));
        fa.add(new Button("Four"));
        fa.pack();
        fa.setVisible(true);
}}
```

1. A window with one large button marked Four in the center.
2. A window with buttons marked One to Four occupy a quarter of the window evenly.
3. A window with buttons marked One to Four placed from the top to bottom.
4. A window with buttons marked One to Four placed on each edge.

12. About Java containers, which statement below is NOT correct? (2分)

1. `Iterator` can deal with `List`, `Set` and `Map`
2. `Map` has group of key-value object pairs
3. `Set` cannot have any duplicate elements
4. `List` holds the elements in a particular sequence

13. Given code below:

```
package his;
public class My {
    void f() {};
}
```

Which statement below is NOT correct? (2分)

1. Only methods of classes in package his can access the f() function.
2. The f() function is default to private.
3. It has to be in a directory named his.
4. It has to be in a file named My.java.

14. The output of the code below is: (2分)

```Java
System.out.println(
```

```
true?Integer.valueOf(1):Double.valueOf(2));
```

1. 2
2. 1.0
3. 2.0
4. 1

15. About String in Java, which statement below is NOT correct? (2 分)

1. A String object can be altered using the = operator with a string literal.
2. Class Object defines toString() function to generate a String represents the object.
3. A String object can be initialized using the = operator with a string literal.
4. A String object is immutable.

16. Given code below：

```Java
class RunTest implements Runnable{
    public static void main(String[] arg) {
        RunTest rt = new RunTest();
        Thread t = new Thread(rt);
        //  here
    }
    public void run() {
        while ( true )
            System.out.println("running");
    }
    void go() { start(1); }
    void start(int i) {}
}
```

Put which statement below at //here is able to print running endlessly? (2 分)

1. `rt.start(1);`
2. `rt.go();`
3. `t.start();`
4. `System.out.println("running");`

17. Which one below is checked exception that should be declared if a method throws it? (2 分)

1. IndexOutOfBoundsException
2. FileNotFoundException
3. ClassCastException
4. NullPointerException

18. Which of the following will output -3.0 (2 分)

1. System.out.println(Math.min(-3.7));
2. System.out.println(Math.round(-3.7));
3. System.out.println(Math.ceil(-3.7));
4. System.out.println(Math.floor(-3.7));

19. `wait()` and `notify()` are used to suspend and resume threads. They are defined as methods of: (2 分)

1. Object
2. Synchronized
3. Runnable
4. Thread

20. What will happen if you try to compile and run the following code?(2 分)

```Java
public class Q {
    public static void main(String argv[]){
        int anar[]=new int[5];
        System.out.println(anar[0]);
    }
}
```

1. 5
2. 0
3. null
4. Error: anar is referenced before it is initialized

21. What is the output of this program? (2 分)

```
public class Test {
```

```
  public static void main(String[] args) throws Exception{
  String str = "ZJU2018";
  Method m = str.getClass().getMethod("toLowerCase");
  m.invoke(str);
  System.out.println(str);
 }
}
```

1. runtime error
2. zju2018
3. compilation error
4. ZJU2018

22. About classes in Java, which one below is correct? (2 分)

1. Every Java class is derived from the root class Class.
2. Every Java class is derived from the root class Root.
3. Every Java class is derived from the root class Object.
4. There is no root class in Java.

23. About inner class, which statement below is NOT correct? (2 分)

1. Inner class cannot be defined as private.
2. Inner classes defined inside a function can access any local variables in that function.
3. Inner class can access every member of the outer class.
4. Non-static member inner class can not be used in the static functions.

24. Given code below:

```Java
void f(int port) {
    return new Package(port) {
        private int pt = port;
        public getPort() { return pt; }
    };
}
```

which statement below is correct? (2 分)

1. It compiles but exception raises at line 3

2. Compile fails because of line 2.
3. Compile fails because of line 4.
4. It compiles and runs smoothly.

25. About the scope of member variables, which statement is correct? (2 分)

1. A private variable can be used directly in the public static void main(String[] args).
2. static variables can be used in static methods only.
3. All member variables are accessible in all non-static member methods.
4. Only public variables can be used in all methods.

26. Which one below is NOT a valid Java identifier? (2 分)

1. 名字
2. goto
3. $2
4. Each

27. Which one below is NOT correct to declare a variable of array? (2 分)

1. int[]a[]
2. int a[]
3. int[][] a
4. int[] a

28. What will this code print?(2 分)

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

1. Exception thrown.
2. Class name@ hashcode in hexadecimal form.
3. null
4. 0

29. Given a Runnable:

```
```

```
class TheRunnable implements Runnable
{
    public void run(){}
    public static void main(String[] args) {
        TheRunnable r = new TheRunnable();
        // put your line here
        System.out.println("Hello World!");
    }
}
```

There's a missing line in TheRunnable.main(), To begin a thread, which is the proper code: (2 分)

1. new Thread(r).start();
2. new Thread(this).start();
3. r.start();
4. r.run();

30. What is the output of this program?(2 分)

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

1. JHelloa
2. JHello
3. JHelloava
4. Java

填空题

1. he output of the code below is:

```Java
enum EnumTry {
```

```
   MON, TUE, WED, THU, FRI;
   public static void main(String[] args)  {
      for (EnumTry e : EnumTry.values()) {
         System.out.println(
            e +":"+ e.toString() +":"+ e.ordinal() +":"+ e.name());
}}}
```

(4 分)

2. The output of the code below is:

```Java
class Z {
       public static void main(String args[]) {
              System.out.println("AAA"+new Z());
       }
       public String toString() {
              System.out.println("###");
              return "Z";
       }
  }
```

3. The output of the code below is:

```Java
class Mammal{
       Mammal(){
               System.out.println("Creating Mammal");
       }
}

public class Human extends Mammal{
public static void main(String argv[]){
       Human h = new Human();
    }
    Human(){
       System.out.println("Creating Human");
    }
}
```

4. The output of the code below is:

```Java
class Test {
  public static void main(String[] args) {
    try {
      System.out.println("Welcome to Java");
      int i = 0;
      int y = 2 / i;
      System.out.println("Welcome to HTML");
    }
    finally {
      System.out.println("The finally clause is executed");
    }
    System.out.println("End of the block");
}}
```

5. The output of the code below is:

```Java
class Test {
    public static void main(String[] args) {
        Integer a = new Integer(3);
        Integer b = 3;
        int c = 3;
        System.out.println(a == b);
        System.out.println(a == c);
}}
```

6. 请写出以下程序运行结果：

```Java
public class Test {
 private int i = 0;
 Test increment() {
     i++;
     return this.clone();
 }
 public Test clone() {
     Test t = new Test();
     t.i = i;
     return t;
 }
```

```
 public Test() { System.out.print(i); }
 void print() {
     System.out.printf("i = %d", i);
 }
 public static void main(String[] args) {
     Test x = new Test();
     for ( int i=0; i<10; i++ )
         x = x.increment();
     x.print();
 }
}
```

7. The output of the code below is:

```Java
class Value {
    int i;
    public static void main(String[] args) {
        Value[] a = new Value[10];
        for ( int i=0; i<a.length; i++ ) {
            a[i] = new Value();
        }
        for ( Value v : a ) {
            v.i += 5;
        }
        for ( int k = 0; k<a.length; k++ ) {
            System.out.print(a[k].i);
        }
    }
}
```

8. The output of the code below is:

```Java
public class Test {
 private static int i = 40;
 private int t=8;
 static {
     System.out.println("Curl world");
     i = 10;
 }
 public static void main(String[] args) {
```

```
        new Test().numberPlay();
        new Test().numberPlay();
    }
    public void numberPlay() {
        i *= 2;
        System.out.println(t+i);
        t /= 2;
    }
}
```

9. The output of the code below is:

```Java
public class Test {
    public static void main(String[] argv){
        String s = "16 abc 32 de\next 64";
        Scanner sc = new Scanner(s);
        while ( sc.hasNext() ) {
            if ( sc.hasNextInt() ) {
                System.out.printf("%02X",sc.nextInt());
            } else {
                System.out.print(sc.next());
            }
        }
    }
}
```