

書名：SCJP 6.0 認證教戰手冊 作者：黃彬華

完全擬真試題 201-244（共 244 題）

第201題

Given:

```

1.  import java.util.*;
2.  class A{}
3.  class B extends A{}
4.  public class Test{
5.      public static void main(Strang[] args){
6.          List<A> listA = new LinkedList<A>();
7.          List<B> listB = new LinkedList<B>();
8.          List<Obect> listO = new LinkedList<Obect>();
9.          //insert code here
10.     }
11.     public static void m1(List<? extends A> list){}
12.     public static void m2(List<A> list){}

```

Place a result onto each method call to indicate what would happen if the method call were inserted at line 9.

Note: Results can be used more than once.

Method Calls	Result
m1(listA); m2(listA);	Does not compile.
m1(listB); m2(listB);	Compiles and runs without error.
m1(listO); m2(listO);	An exception is thrown at runtime.

答案：

m1(listA);	Compiles and runs without error.
m1(listB);	Compiles and runs without error.
m1(listO);	Does not compile.
m2(listA);	Compiles and runs without error.
m2(listB);	Does not compile.
m2(listO);	Does not compile.

參考：12-3 泛型

第202題

Given:

```

NumberNames nn = new NumberNames();
nn.put("one", 1);
System.out.println(nn.getNames());

```

Place the code into position to create a class that maps from Strings to integer values. The result of execution must be [one]. Some options may be used more than once.

```

public class NumberNames{
    private HashMap< Place here , Place here > map =
        new HashMap< Place here , Place here Place here >;

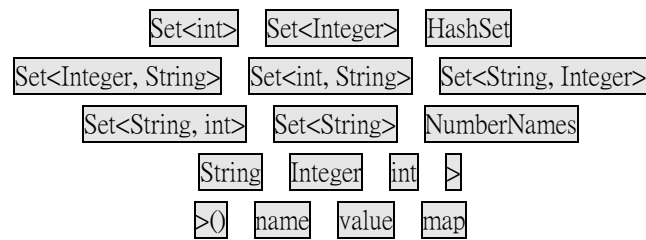
```

```

public void put(String name, int value){
    map.put(Place here, Place here);
}
public Place here getNames(){
    return map.keySet();
}
}

```

Code



答案：

```

public class NumberNames{
    private HashMap<String, Integer> map =
        new HashMap<String, Integer>>0;
    public void put(String name, int value){
        map.put(name, value);
    }
    public Set<String> getNames(){
        return map.keySet();
    }
}

```

參考：12-5-1 HashMap

第203題

Given:

```

5. import java.util.*;
6. public class SortOf{
7.     public static void main(String[] args){
8.         ArrayList<Integer> a = new ArrayList<Integer>();
9.         a.add(1); a.add(5); a.add(3);
10.        Collections.sort(a);
11.        a.add(2);
12.        Collections.reverse(a);
13.        System.out.println(a);
14.    }
15. }

```

What is the result?

- A. [1, 2, 3, 5]
- B. [2, 1, 3, 5]
- C. [2, 5, 3, 1]

- D. [5, 3, 2, 1]
- E. [1, 3, 5, 2]
- F. Compilation fails.
- G. An exception is thrown at runtime.

答案：C

參考：12-2-2 Collections 類別、12-4-5 List 集合

第204題

Given:

```
11. public class Person{
12.     private name;
13.     public Person(String name){
14.         this.name = name;
15.     }
16.     public int hashCode(){
17.         return 420;
18.     }
19. }
```

Which statement is true?

- A. The time to find the value from HashMap with a Person key depends on the size of the map.
- B. Deleting a Person key from a HashMap will delete all map entries for all keys of type Person.
- C. Inserting a second Person object into a HashSet will cause the first Person object to be removed as a duplicate.
- D. The time to determine whether a Person object is contained in a HashSet is constant and does NOT depend on the size of the map.

答案：A

參考：12-4-2 equals()、hashCode()方法的改寫

第205題

Given:

```
12. import java.util.*;
13. public class Explorer2{
14.     public static void main(String[] args){
15.         TreeSet<Integer> s = new TreeSet<Integer>();
16.         TreeSet<Integer> subs = new TreeSet<Integer>();
17.         for(int i=606; i<613; i++){
18.             if(i%2 == 0) s.add(i);
19.             subs = (TreeSet)s.subSet(608, true, 611, true);
20.             s.add(629);
21.             System.out.println(s + " " + subs);
22.         }
23. }
```

What is the result?

- A. Compilation fails.

- B. An exception is thrown at runtime.
- C. [608, 610, 612, 629] [608, 610]
- D. [608, 610, 612, 629] [608, 610, 629]
- E. [606, 608, 610, 612, 629] [608, 610]
- F. [606, 608, 610, 612, 629] [608, 610, 629]

答案：E

參考：12-4-4 子集檢視

第206題

Given:

```
1. public class Drink implements Comparable{
2.     public String name;
3.     public int compareTo(Object o){
4.         return 0;
5.     }
6. }
```

and:

```
20. Drink one = new Drink();
21. Drink two = new Drink();
22. one.name = "Coffee";
23. two.name = "Tea";
24. TreeSet set = new TreeSet();
25. set.add(one);
26. set.add(two);
```

A programmer iterates over the TreeSet and prints the name of each Drink object.

What is the result?

- A. Tea
- B. Coffee
- C. Coffee Tea
- D. Compilation fails.
- E. The code runs with no output.
- F. An exception is thrown at runtime.

答案：B

參考：12-4-3 SortedSet 集合

第207題

A programmer must create a generic class MinMax and the type parameter of MinMax must implement Comparable. Which implementation of MinMax will compile?

- A.

```
class MinMax<E extends Comparable<E>>{
    E min = null;
    E max = null;
    public MinMax(){}
```

- ```
public void put(E value){/* store min or max */}
```
- B. class MinMax<E implements Comparable<E>>{  
 E min = null;  
 E max = null;  
 public MinMax(){ }  
 public void put(E value){/\* store min or max \*/}

C. class MinMax<E extends Comparable<E>>{  
 <E> E min = null;  
 <E> E max = null;  
 public MinMax(){ }  
 public <E> void put(E value){/\* store min or max \*/}

D. class MinMax<E implements Comparable<E>>{  
 <E> E min = null;  
 <E> E max = null;  
 public MinMax(){ }  
 public <E> void put(E value){/\* store min or max \*/}

答案：A

參考：12-3 泛型

## 第208題

Given:

```
1. import java.util.*;
2. public class Example{
3. public static void main(String[] args){
4. //insert code here
5. set.add(new Integer(2));
6. set.add(new Integer(1)),
7. System.out.println(set);
8. }
9. }
```

Which code, inserted at line 4, guarantees that this program will output [1, 2]?

- A. Set set = new TreeSet();
- B. Set set = new HashSet();
- C. Set set = new SortedSet();
- D. List set = new SortedList();
- E. Set set = new LinkedHashSet();

答案：A

參考：12-4-3 SortedSet 集合

## 第209題

Given:

```

1. import java.util.*;
2. public class TestSet{
3. enum Example{ONE, TWO, THREE}
4. public static void main(String[] args){
5. Collection coll = new ArrayList();
6. coll.add(Example.THREE);
7. coll.add(Example.THREE);
8. coll.add(Example.THREE);
9. coll.add(Example.TWO);
10. coll.add(Example.TWO);
11. coll.add(Example.ONE);
12. Set set = new HashSet(coll);
13. }
14. }

```

Which statement is true about the set variable on line 12?

- A. The set variable contains all six elements from the coll collection, and the order is guaranteed to be preserved.
- B. The set variable contains only three elements from the coll collection, and the order is guaranteed to be preserved.
- C. The set variable contains all six elements from the coll collection, but the order is NOT guaranteed to be preserved.
- D. The set variable contains only three elements from the coll collection, but the order is NOT guaranteed to be preserved.

答案：D

參考：12-4-1 Set 集合、12-4-5 List 集合

## 第210題

Given:

```

11. public class Person{
12. private String name, comment;
13. private int age;
14. public Person(String n, int a, String c){
15. name = n; age = a; comment = c;
16. }
17. public boolean equals(Object o){
18. if (!(o instanceof Person)) return false;
19. Person p = (Person)o;
20. return age == p.age && name.equals(p.name);
21. }
22. }

```

What is the appropriate definition of the hashCode method in class Person?

- A. return super.hashCode();
- B. return name.hashCode() + age \* 7;
- C. return name.hashCode() + comment.hashCode() / 2;
- D. return name.hashCode() + comment.hashCode() / 2 - age \* 3;

答案：B

參考：12-4-2 equals()、hashCode()方法的改寫

### 第211題

Given:

```
11. public class Key{
12. private long id1,
13. private long id2;
14.
15. //class Key methods
16. }
```

A programmer is developing a class Key, that will be used as a key in a standard java.util.HashMap.

Which two methods should be overridden to assure that Key works correctly as a key? (Choose two.)

- A. public int hashCode()
- B. public void hashCode()
- C. public int compareTo(Object o)
- D. public boolean equals(Object o)
- E. public boolean compareTo(Key k)

答案：AD

參考：12-4-2 equals()、hashCode()方法的改寫

### 第212題

Given:

```
3. import java.util.*;
4. public class Hancock{
5. //insert code here
6. list.add("foo");
7. }
8. }
```

Which two code fragments, inserted independently at line 5, will compile without warnings? (Choose two.)

- A. public void addStrings(List list){
- B. public void addStrings(List<String> list){
- C. public void addStrings(List<? super String> list){
- D. public void addStrings(List<? extends String> list){

答案：BC

參考：12-4-2 equals()、hashCode()方法的改寫

### 第213題

Given a class whose instances, when found in a collection of objects, are sorted by using the compareTo() method, which two statements are true? (choose two.)

- A. The class implements java.lang.Comparable.
- B. The class implements java.util.Comparator.
- C. The interface used to implement sorting allows this class to define only one sort sequence.

D. The interface used to implement sorting allows this class to define many different sort sequences.

答案：AC

參考：12-4-3 SortedSet 集合

#### 第214題

Given:

```
12. import java.util.*;
13. public class Explorer3{
14. public static void main(String[] args){
15. TreeSet<Integer> s = new TreeSet<Integer>();
16. TreeSet<Integer> subs = new TreeSet<Integer>();
17. for(int i=606; i<613; i++)
18. if(i%2 == 0) s.add(i);
19. subs = (TreeSet)s.subSet(608, true, 611, true);
20. subs.add(629);
21. System.out.println(s + " " + subs);
22. }
23. }
```

What is the result?

- A. Compilation fails.
- B. An exception is thrown at runtime.
- C. [608, 610, 612, 629] [608, 610]
- D. [608, 610, 612, 629] [608, 610, 629]
- E. [606, 608, 610, 612, 629] [608, 610]
- F. [606, 608, 610, 612, 629] [608, 610, 629]

答案：B

參考：12-4-4 子集檢視

#### 第215題

Given:

```
1. import java.util.*;
2.
3. public class LetterASort{
4. public static void main(String[] args){
5. ArrayList<String> strings = new ArrayList<String>();
6. strings.add("aAaA");
7. strings.add("AaA");
8. strings.add("aAa");
9. strings.add("AAaa");
10. Collections.sort(strings);
11. for(String s : strings){System.out.print(s + " ");}
12. }
13. }
```

What is the result?

- A. Compilation fails.



- B. aAaA aAa AAaa AaA
- C. AAaa AaA aAa aAaA
- D. AaA AAaa aAaA aAa
- E. aAa AaA aAaA AAaa
- F. An exception is thrown at runtime.

答案：C

參考：12-2-2 Collections 類別、12-4-5 List 集合

## 第216題

Given:

```

1. import java.util.*;
2. public class TestGenericConversion{
3. public static void main(String[] args){
4. List list = new LinkedList();
5. list.add("one");
6. list.add("two");
7. System.out.print(((String)list.get(0)).length());
8. }
9. }
```

Refactor this class to use generics without changing the code's behavior.

```

1. import java.util.*;
2. public class TestGenericConversion{
3. public static void main(String[] args){
4. Place here
5. list.add("one");
6. list.add("two");
7. Place here
8. }
9. }
```

### Code

|                                               |                                                       |
|-----------------------------------------------|-------------------------------------------------------|
| List list = new LinkedList();                 | System.out.print(list.get(0).length());               |
| List<String> list = new LinkedList<String>(); | System.out.print(list.get<String>(0).length());       |
| List<String> list = new LinkedList();         | System.out.print(<String>list.get(0).length());       |
| List list = new LinkedList<String>();         | System.out.print((List<String>)list.get(0).length()); |

答案：

```

1. import java.util.*;
2. public class TestGenericConversion{
3. public static void main(String[] args){
4. List<String> list = new LinkedList<String>();
5. list.add("one");
6. list.add("two");
7. System.out.print(list.get(0).length());
8. }
9. }
```

參考：12-3 泛型

## 第217題

Place the code into the GenericB class definition to make the class compile successfully.

|                                                                                                                                                                                                                                                                                                                                                                                                                                              |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                    |
|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <pre>import java.util.*;  public class GenericB&lt;Place&gt;{      public Place foo.     public void setFoo(Place foo){         this.foo = foo;     }     public Place getFoo(){         return foo;     }     public static void main(String[] args){         GenericB&lt;Cat&gt; bar = new GenericB&lt;Cat&gt;();         bar.setFoo(new Cat());         Cat c = bar.getFoo();     } }  interface Pet{}  class Cat implements Pet{ }</pre> | <p style="text-align: center;"><b>Code</b></p> <div style="border: 1px solid black; padding: 2px; margin-bottom: 2px;">? extends Pet</div> <div style="border: 1px solid black; padding: 2px; margin-bottom: 2px;">T extends Pet</div> <div style="border: 1px solid black; padding: 2px; margin-bottom: 2px;">? implements Pet</div> <div style="border: 1px solid black; padding: 2px; margin-bottom: 2px;">T implements Pet</div> <div style="border: 1px solid black; padding: 2px; margin-bottom: 2px;">Pet extends T</div> <div style="border: 1px solid black; padding: 2px; margin-bottom: 2px;">?</div> <div style="border: 1px solid black; padding: 2px; margin-bottom: 2px;">T</div> <div style="border: 1px solid black; padding: 2px; margin-bottom: 2px;">&lt;?&gt;</div> <div style="border: 1px solid black; padding: 2px; margin-bottom: 2px;">&lt;Pet&gt;</div> |
|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|

答案：

|                                                                                                                                                                                                                                                                                                                                                                                                                                          |
|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <pre>import java.util.*;  public class GenericB&lt;T extends Pet&gt;{      public T foo.     public void setFoo(T foo){         this.foo = foo;     }     public T getFoo(){         return foo;     }     public static void main(String[] args){         GenericB&lt;Cat&gt; bar = new GenericB&lt;Cat&gt;();         bar.setFoo(new Cat());         Cat c = bar.getFoo();     } }  interface Pet{}  class Cat implements Pet{ }</pre> |
|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|

參考：12-3 泛型

## 第218題

Place the code elements in position so that the Flags2 class will compile and make appropriate use of the wait/notify mechanism. Note You may reuse code elements.

```
class Flags2{
 private boolean isReady = false;
 public Place here void produce(){
 isReady = true;
 Place here ;
 }
 public Place here void consume(){
 while(!isReady){
 try{
 Place here ;
 }catch(Exception ex){}
 }
 isReady = Place here ;
 }
}
```

**Code Elements**

synchronized

true

false

wait()

volatile

synchronized()

notifyAll()

synchronize

答案：

```
class Flags2{
 private boolean isReady = false;
 public synchronized void produce(){
 isReady = true;
 notifyAll() ;
 }
 public synchronized void consume(){
 while(!isReady){
 try{
 wait() ;
 }catch(Exception ex){}
 }
 isReady = false ;
 }
}
```

參考：13-5 執行緒的互動處理

## 第219題

Given:

1. public class Threads2 implements Runnable{
- 2.
3. public void run(){

```

4. System.out.println("run.");
5. throw new RuntimeException("Problem");
6. }
7. public static void main(String[] args){
8. Thread t = new Thread(new Threads2());
9. t.start();
10. System.out.println("End of method.");
11. }
12. }
```

Which two can be results? (Choose two.)

- A. java.lang.RuntimeException: Problem
- B. run.  
java.lang.RuntimeException: Problem
- C. End of method.  
java.lang.RuntimeException: Problem
- D. End of method.  
run.  
java.lang.RuntimeException: Problem
- E. run.  
java.lang.RuntimeException: Problem  
End of method.

答案：DE

參考：13-2 Java 執行緒與 Thread 類別

#### 第220題

Which factor or factors

- A. It is possible for more than two threads to deadlock at once.
- B. The JVM implementation guarantees that multiple threads cannot enter into a deadlocked state.
- C. Deadlocked threads release once their sleep() method's sleep duration has expired.
- D. Deadlocking can occur only when the wait(), notify() and notifyAll() methods are used incorrectly.
- E. It is possible for a single-threaded application to deadlock if synchronized blocks are used incorrectly.
- F. If a piece of code is capable of deadlocking, you cannot eliminate the possibility of deadlocking by inserting invocations of Thread.yield().

答案：AF

參考：13-5 執行緒的互動處理

#### 第221題

Given:

```

7. void waitForSignal(){
8. Object obj = new Object();
9. synchronized(Thread.currentThread()){
10. obj.wait();
```

```

11. obj.notify();
12. }
13. }

```

Which statement is true?

- A. This code can throw an InterruptedException.
- B. This code can throw an IllegalMonitorStateException.
- C. This code can throw a TimeoutException after ten minutes.
- D. Reversing the order of obj.wait() and obj.notify() might cause this method to complete normally.
- E. A call to notify() or notifyAll() from another thread might cause this method to complete normally.
- F. This code does NOT compile unless "obj.wait()" is replaced with "((Thread) obj).wait()".

答案：B

參考：13-5 執行緒的互動處理

## 第222題

Given:

```

10. public class Starter extends Thread{
11. private int x = 2;
12. public static void main(String[] args) throws Exception{
13. new Starter().makeItSo();
14. }
15. public Starter(){
16. x = 5;
17. start();
18. }
19. public void makeItSo() throws Exception{
20. join();
21. x = x - 1;
22. System.out.println(x);
23. }
24. public void run(){x *= 2;}
25. }

```

What is the output if the main() method is run?

- A. 4
- B. 5
- C. 8
- D. 9
- E. Compilation fails.
- F. An exception is thrown at runhime.
- G. It is impossible to determine for certain.

答案：D

參考：13-2 Java 執行緒與 Thread 類別、13-3 Runnable 介面

## 第223題

Given:

```

11. class PingPong2{
12. synchronized void hit(long n){
13. for(int i=1; i<3; i++)
14. System.out.print(n + "-" + i + " ");
15. }
16. }
17. public class Tester implements Runnable{
18. static PingPong2 pp2 = new PingPong2();
19. public static void main(String[] args){
20. new Thread(new Tester()).start();
21. new Thread(new Tester()).start();
22. }
23. public void run(){pp2.hit(Thread.currentThread.getId());}
24. }

```

Which statement is true?

- A. The output could be 5-1 6-1 6-2 5-2
- B. The output could be 6-1 6-2 5-1 5-2
- C. The output could be 6-1 5-2 6-2 5-1
- D. The output could be 6-1 6-2 5-1 7-1

答案：B

參考：13-4 執行緒的同步性與安全性

#### 第224題

Given:

```

1. public class Threads4{
2. public static void main(String[] args){
3. new Threads4.go();
4. }
5. public void go(){
6. Runnable r = new Runnable(){
7. public void run(){
8. System.out.print("foo");
9. }
10. };
11. Thread t = new Thread(r);
12. t.start();
13. t.start();
14. }
15. }

```

What is the result?

- A. Compilation fails.
- B. An exception is thrown at runtime.
- C. The code executes normally and prints "foo";
- D. The code executes normally, but nothing is printed.

答案：B

參考：13-4 執行緒的同步性與安全性

## 第225題

Given:

```

10. Runnable r = new Runnable(){
11. public void run(){
12. try{
13. Thread.sleep(1000);
14. }catch(InterruptedException e){
15. System.out.println("interrupted");
16. }
17. System.out.println("ran");
18. }
19. };
20. Thread t = new Thread(r);
21. t.start();
22. System.out.println("started");
23. t.sleep(2000);
24. System.out.println("interrupting");
25. t.interrupt();
26. System.out.println("ended");

```

Assume that sleep(n) executes in exactly n milliseconds, and all other code executes in an insignificant amount of time.

Place the fragments in the output area to show the result of running this code.

| Output     | Fragments            |
|------------|----------------------|
| Place here | interrupted          |
| Place here | ran                  |
| Place here | started              |
| Place here | interrupting         |
| Place here | ended                |
|            | InterruptedException |
|            | (no more output)     |

答案：

| Output           |
|------------------|
| (no more output) |
| started          |
| ran              |
| interrupting     |
| ended            |

參考：13-3 Runnable 介面

## 第226題

Which two statements are true? (Choose two.)

A. It is possible to synchronize static methods.

- B. When a thread has yielded as a result of `yield()`, it releases its locks.
- C. When a thread is sleeping as a result of `sleep()`, it releases its locks.
- D. The `Object.wait()` method can be invoked only from a synchronized context.
- E. The `Thread.sleep()` method can be invoked only from a synchronized context.
- F. When the thread scheduler receives a `notify()` request, and notifies a thread, that thread immediately releases its lock.

答案：AD

參考：13-2 Java 執行緒與 Thread 類別、13-3 Runnable 介面、13-5 執行緒的互動處理

#### 第227題

Given:

```
1. public class TestOne implements Runnable{
2. public static void main (String[] args)throws Exception{
3. Thread t = new Thread(new TestOne());
4. t.start();
5. System.out.prit("Started");
6. t.join();
7. System.out.print("Complete");
8. }
9. public void run(){
10. for(int i=0; i<4; i++){
11. System.out.print(i);
12. }
13. }
14. }
```

What can be a result?

- A. Compilation fails.
- B. An exception is thrown at runtime.
- C. The code executes and prints "StartedComplete".
- D. The code executes and prints "StartedComplete0123".
- E. The code executes and prints "Started0123Complete".

答案：E

參考：13-2 Java 執行緒與 Thread 類別、13-3 Runnable 介面

#### 第228題

Which three will compile and run without exception? (Choose three.)

- A. `private synchronized Object o;`
- B. `void go(){  
 synchronized{/* code here */}`
- C. `public synchronized void go(){/* code here */}`
- D. `private synchronized(this) void go(){/* code here */}`
- E. `void go(){`



```
synchronized(Object.class){/* code here */}
```

```
F. void go(){
 Object o = new Object();
 synchronized(o){/* code here */}
```

答案：CEF

參考：13-2 Java 執行緒與 Thread 類別、13-3 Runnable 介面、13-5 執行緒的互動處理

## 第229題

Given:

```
1. public class TestFive{
2. private int x;
3. public void foo(){
4. int current = x;
5. x = current + 1;
6. }
7. public void go(){
8. for(int i=0; i<5; i++){
9. new Thread(){
10. public void run(){
11. foo();
12. System.out.print(x + ", ");
13. }.start();
14. }}
15. }
```

Which two changes, taken together, would guarantee the output: 1, 2, 3, 4, 5, ? (Choose two.)

- A. move the line 12 print statement into the foo() method
- B. change line 7 to public synchronized void go(){
- C. change the variable declaration on line 2 to private volatile int x;
- D. wrap the code inside the foo() method with a synchronized(this) block
- E. wrap the for loop code inside the go() method with a synchronized block synchronized(this){//for loop code here}

答案：AD

參考：8-5-4 匿名內部類別、13-2 Java 執行緒與 Thread 類別、13-3 Runnable 介面

## 第230題

Given that t1 is a reference to a live thread, which is true?

- A. The Thread.sleep() method can take t1 as an argument.
- B. The Object.notify() method can take t1 as an argument.
- C. The Thread.yield() method can take t1 as an argument.
- D. The Thread.setPriority() method can take t1 as an argument.
- E. The Object.notify() method arbitrarily chooses which thread to notify.

答案：E

參考：13-2 Java 執行緒與 Thread 類別、13-3 Runnable 介面、13-5 執行緒的互動處理

第231題

Given:

```
11. Runnable r = new Runnable(){
12. public void run(){
13. System.out.print("Cat");
14. }
15. };
16. Thread t = new Thread(r){
17. public void run(){
18. System.outprint("Dog");
19. }
20. };
21. t.start();
```

What is the result?

- A. Cat
- B. Dog
- C. Compilation fails.
- D. The code runs with no output.
- E. An exception is thrown at runtime.

答案：B

參考：8-5-4 匿名內部類別、13-2 Java 執行緒與 Thread 類別、13-3 Runnable 介面

第232題

Given:

```
1. public class Threads5{
2. public static void main(String[] args){
3. new Thread(new Runnable(){
4. public void run(){
5. System.out.print("bar");
6. }).start();
7. }
8. }
```

What is the result?

- A. Compilation fails.
- B. An exception is thrown at runtime.
- C. The code executes normally and prints "bar".
- D. The code executes normally, but nothing prints.

答案：C

參考：8-5-4 匿名內部類別、13-2 Java 執行緒與 Thread 類別、13-3 Runnable 介面

第233題

Given:

```
1. public class Threads1{
```

```

2. int x = 0;
3. public class Runner implements Runnable{
4. public void run(){
5. int current = 0;
6. for(int i=0; i<4; i++){
7. current = x;
8. System.out.print(current + ", ");
9. x = current + 2;
10. }
11. }
12. }
13.
14. public static void main(String[] args){
15. new Threads1().go();
16. }
17.
18. public void go(){
19. Runnable rl = new Runner();
20. new Thread(rl).start();
21. new Thread(rl).start();
22. }
23. }

```

Which two are possible results? (Choose two.)

- A. 0, 2, 4, 4, 6, 8, 10, 6,
- B. 0, 2, 4, 6, 8, 10, 2, 4,
- C. 0, 2, 4, 6, 8, 10, 12, 14,
- D. 0, 0, 2, 2, 4, 4, 6, 6, 8, 8, 10, 10, 12, 12, 14, 14,
- E. 0, 2, 4, 6, 8, 10, 12, 14, 0, 2, 4, 6, 8, 10, 12, 14,

答案：AC

參考：8-5-4 匿名內部類別、13-2 Java 執行緒與 Thread 類別、13-3 Runnable 介面

#### 第234題

Given:

foo and bar are public references available to many other threads, foo refers to a Thread and bar is an Object.

The thread foo is currently executing bar.wait().

From another thread, what provides the most reliable way to ensure that foo will stop executing wait()?

- A. foo.notify();
- B. bar.notify();
- C. foo.notifyAll();
- D. Thread.notify();
- E. bar.notifyAll();
- F. Object.notify();

答案：E

參考：13-5 執行緒的互動處理

## 第235題

Given:

```

11. public class PingPong implements Runnable{
12. synchronized void hit(long n){
13. for(int i=1; i<3; i++)
14. System.out.print(n + "-" + i + " ");
15. }
16. public static void main(String[] args){
17. new Thread(new PingPong()).start();
18. new Thread(new PingPong()).start();
19. }
20. public void run(){
21. hit(Thread.currentThread().getId);
22. }
23. }
```

Which two statements are true? (Choose two.)

- A. The output could be 8-1 7-2 8-2 7-1
- B. The output could be 7-1 7-2 8-1 6-1
- C. The output could be 8-1 7-1 7-2 8-2
- D. The output could be 8-1 8-2 7-1 7-2

答案：CD

參考：13-4 執行緒的同步性與安全性

## 第236題

Given:

```

1. class Computation extends Thread{
2.
3. private int num;
4. private boolean isComplete;
5. private int result;
6.
7. public Computation(int num){this.num = num;}
8.
9. public synchronized void run(){
10. result = num * 2;
11. isComplete = true;
12. notify();
13. }
14.
15. public synchronized int getResult(){
16. while(!isComplete){
17. try{
18. wait();
19. }catch(InterruptedException e){}
20. }
21. return result;
22. }
23.
24. public static void main(String[] args){
```

```

25. Computation[] computations = new Computation[4];
26. for(int i=0; i<computations.length; i++){
27. computations[i] = new Computation(i);
28. computations[i].start();
29. }
30. for(Computation c : computations)
31. System.out.print(c.getResult() + " ");
32. }
33. }

```

What is the result?

- A. The code will deadlock.
- B. The code may run with no output.
- C. An exception is thrown at runtime.
- D. The code may run with output "0 6".
- E. The code may run with output "2 0 6 4".
- F. The code may run with output "0 2 4 6".

答案：F

參考：13-4 執行緒的同步性與安全性

### 第237題

Place the code elements into the class so that the code compiles and prints "Run. Run. doIt. " in exactly that order. Note that there may be more than one correct solution.

```

class TestTwo extends Thread{
 public static void main(String[] a) throws Exception{
 TestTwo t = new TestTwo();
 t.start();
 Place here
 Place here
 Place here
 }
 public void run(){
 System.out.print("Run. ");
 }
 public void doIt(){
 System.out.print("doIt. ");
 }
}

```

#### Code Elements

答案：

```

class TestTwo extends Thread{
 public static void main(String[] a) throws Exception{
 TestTwo t = new TestTwo();
 t.start();

```

```

 t.run();
 t.join();
 t.doIt();
 }
 public void run(){
 System.out.print("Run. ");
 }
 public void doIt(){
 System.out.print("doIt. ");
 }
}

```

參考：13-2 Java 執行緒與 Thread 類別

### 第238題

Which two code fragments will execute the method doStuff() in a separate thread? (Choose two.)

- A. `new Thread(){  
 public void run(){doStuff();}  
};`
- B. `new Thread(){  
 public void start(){doStuff();}  
};`
- C. `new Thread(){  
 public void start(){doStuff();}  
}.run();`
- D. `new Thread(){  
 public void run(){doStuff();}  
}.start();`
- E. `new Thread(new Runnable(){  
 public void run(){doStuff();}  
}).run();`
- F. `new Thread(new Runnable(){  
 public void run(){doStuff();}  
}).start();`

答案：DF

參考：8-5-4 匿名內部類別、13-2 Java 執行緒與 Thread 類別、13-3 Runnable 介面

### 第239題

Given:

1. `public class TestOne{`
2.  `public static void main(String[] args) throws Exception{`
3.  `Thread.sleep(3000);`

```

4. System.out.println("sleep");
5. }
6. }

```

What is the result?

- A. Compilation fails.
- B. An exception is thrown at runtime.
- C. The code executes normally and prints "sleep".
- D. The code executes normally, but nothing is printed.

答案：C

參考：13-2 Java 執行緒與 Thread 類別

#### 第240題

Place a Class on each method that is declared in the class.

| Method Name           | Class                         |
|-----------------------|-------------------------------|
| <code>run()</code>    | <code>java.lang.Object</code> |
| <code>wait()</code>   | <code>java.lang.Thread</code> |
| <code>notify()</code> |                               |
| <code>sleep()</code>  |                               |
| <code>start()</code>  |                               |
| <code>join()</code>   |                               |

答案：

|                       |                               |
|-----------------------|-------------------------------|
| <code>run()</code>    | <code>java.lang.Thread</code> |
| <code>wait()</code>   | <code>java.lang.Object</code> |
| <code>notify()</code> | <code>java.lang.Object</code> |
| <code>sleep()</code>  | <code>java.lang.Thread</code> |
| <code>start()</code>  | <code>java.lang.Thread</code> |
| <code>join()</code>   | <code>java.lang.Thread</code> |

參考：13-2 Java 執行緒與 Thread 類別、13-3 Runnable 介面、13-5 執行緒的互動處理

#### 第241題

Given:

```

1. public class Threads3 implements Runnable{
2. public void run(){
3. System.out.print("running");
4. }
5. public static void main(String[] args){
6. Thread t = new Thread(new Threads3());
7. t.run();
8. t.run();
9. t.start();
10. }
11. }

```

What is the result?

- A. Compilation fails.
- B. An exception is thrown at runtime.
- C. The code executes and prints "running".
- D. The code executes and prints "runningrunning".
- E. The code executes and prints "runningrunningrunning".

答案：E

參考：13-3 Runnable 介面

#### 第242題

Given:

```
public class NamedCounter{
 private final String name;
 private int count;
 public NamedCounter(String name){this.name = name;}
 public String getName(){return name;}
 public void increment(){count++;}
 public int getCount(){return count;}
 public void reset(){count = 0;}
```

Which three changes should be made to adapt this class to be used safely by multiple threads? (Choose three.)

- A. declare reset() using the synchronized keyword
- B. declare getName() using the synchronized keyword
- C. declare getCount() using the synchronized keyword
- D. declare the constructor using the synchronized keyword
- E. declare increment() using the synchronized keyword

答案：ACE

參考：13-3 Runnable 介面

#### 第243題

Given that Triangle implements Runnable, and:

```
31. void go() throws Exception{
32. Thread t = new Thread(new Triangle());
33. t.start();
34. for(int x=1; x<100000; x++){
35. //insert code here
36. if(x%100 == 0) System.out.print("g");
37. }}
38. public void run(){
39. try{
40. for(int x=1; x<100000; x++){
41. //insert the same code here
42. if(x%100 == 0) System.out.print("t");
43. }
```



```
44. }catch(Exception e){}
45. }
```

Which two statements, inserted independently at both lines 35 and 41, tend to allow both threads to temporarily pause and allow the other thread to execute? (Choose two.)

- A. Thread.wait();
- B. Thread.join();
- C. Thread.yield();
- D. Thread.sleep(1);
- E. Thread.notify();

答案：CD

參考：13-2 Java 執行緒與 Thread 類別、13-3 Runnable 介面

#### 第244題

Given:

```
1. public class TestSeven extends Thread{
2. private static int x;
3. public synchronized void doThings(){
4. int current = x;
5. current++;
6. x = current;
7. }
8. public void run(){
9. doThings();
10. }
11. }
```

Which statement is true?

- A. Compilation fails.
- B. An exception is thrown at runtime.
- C. Synchronizing the run() method would make the class thread-safe.
- D. The data in variable "x" are protected from concurrent access problems.
- E. Declaring the doThings() method as static would make the class thread-safe.
- F. Wrapping the statements within doThings() in a synchronized(new Object()){ } block would make the class thread-safe.

答案：E

參考：13-4 執行緒的同步性與安全性