

## Oracle.1z0-808.v2024-04-02.q150

試験コード:	1z0-808
試験名称:	Java SE 8 Programmer I
認定資格:	Oracle
無料問題数:	150
バージョン:	v2024-04-02
アクセス数:	474
ページビュー数:	1500
<a href="https://www.jpnpdf.com/Oracle.1z0-808.v2024-04-02.q150-mondaishu.html">https://www.jpnpdf.com/Oracle.1z0-808.v2024-04-02.q150-mondaishu.html</a>	

### 最新問題: 1

コードの断片を考えると、次のようになります。

```
public class Person {
    String name;
    int age = 25;

    public Person (String name) {
        this (); // //line n1
        setName (name);
    }
    public Person (String name, int age) {
        Person (name); //line n2
        setAge (age);
    }
    //setter and getter methods go here

    public String show () {
        return name + " " + age;
    }

    public static void main (String [] args) {
        Person p1 = new Person ("Jesse");
        Person p2 = new Person ("Walter", 52);
        System.out.println (p1.show () );
        System.out.println (p2.show () );
    }
}
```

結果は何ですか？

- A. 行 n1 と行 n2 の両方でコンパイルが失敗します。
- B. コンパイルは行 n1 でのみ失敗します。
- C. コンパイルは行 n2 でのみ失敗します。
- D. ジェシー 25 ウォルター 52

**Answer: D** ([メッセージを残す](#))

最新問題: 2

与えられる:

```

public class Triangle {
    static double area;
    int b = 2, h = 3;
    public static void main(String[] args) {
        double p, b, h;          //line n1
        if (area == 0) {
            b = 3;
            h = 4;
            p = 0.5;
            area = p * b * h;    //line n2
        }
        System.out.println("Area is " + area);
    }
}

```

結果は何ですか？

- A. 面積は 6.0
- B. n2 行目でコンパイルが失敗します。
- C. n1 行目でコンパイルが失敗します。
- D. 面積は 3.0

**Answer: B** ([メッセージを残す](#))

最新問題: 3

コードの断片を考えると、次のようになります。

```

public static void main(String[] args) {
    String[][] arr = {"A", "B", "C"}, {"D", "E"}
    for (int i = 0; i < arr.length; i++) {
        for (int j = 0; j < arr[i].length; j++) {
            System.out.print(arr[i][j] + " ");
            if (arr[i][j].equals("B")) {
                break;
            }
        }
        continue;
    }
}

```

結果は何ですか？

- A. ABC
- B. コンパイルは失敗します。
- C. ABCDE
- D. ABDE

**Answer: D** ([メッセージを残す](#))

最新問題: 4

与えられる:

```
1. import java.io.Error;
2.     public class TestApp {
3.     public static void main(String[] args) {
4.         TestApp t = new TestApp();
5.         try {
6.             t.doPrint();
7.             t.doList();
8.
9.         } catch (Exception e2) {
10.            System.out.println("Caught " + e2);
11.        }
12.    }
13.    public void doList() throws Exception {
14.        throw new Error("Error");
15.    }
16.    public void doPrint() throws Exception {
17.        throw new RuntimeException("Exception");
18.    }
19. }
```

ORACLE

結果は何ですか？

```
 A) Caught java.lang.RuntimeException: Exception
Exception in thread "main" java.lang.Error: Error
at TestApp.doList(TestApp.java: 14)
at TestApp.main(TestApp.java: 6)

 B) Exception in thread "main" java.lang.Error: Error
at TestApp.doList(TestApp.java: 14)
at TestApp.main(TestApp.java: 6)

 C) Caught java.lang.RuntimeException: Exception
Caught java.lang.Error: Error

 D) Caught java.lang.RuntimeException: Exception
```

ORACLE

A. オプション B

B. オプション A

C. オプション D

D. オプション C

Answer: D ([メッセージを残す](#))

最新問題: 5

与えられる:

```

class Vehicle {
    String type = "4W";
    int maxSpeed = 100;

    Vehicle(String type, int maxSpeed) {
        this.type = type;
        this.maxSpeed = maxSpeed;
    }
    Vehicle() {}
}

class Car extends Vehicle {
    String trans;

    Car(String trans) { //line n1
        this.trans = trans;
    }

    Car(String type, int maxSpeed, String trans) {
        super(type, maxSpeed); // line n2
        this.trans = trans;
    }
}

```

コードの断片を考えると、次のようになります。

```

7. Car c1 = new Car("Auto");
8. Car c2 = new Car("4W", 150, "Manual");
9. System.out.println(c1.type + " " + c1.maxSpeed + " " + c1.trans);
10. System.out.println(c2.type + " " + c2.maxSpeed + " " + c2.trans);

```

結果は何ですか？

- A. null 0 Auto4W 150 手動
- B. コンパイルは n2 行目でのみ失敗します
- C. 行 n1 と行 n2 の両方でコンパイルが失敗します。
- D. 4W 100 自動 4W 150 手動
- E. コンパイルは n1 行目でのみ失敗します

**Answer: C** ([メッセージを残す](#))

最新問題: 6

コードの断片を考えると、次のようになります。

```
LocalDateTime dt = LocalDateTime.of(2014, 7, 31, 1, 1);
dt.plusDays(30);
dt.plusMonths(1);
System.out.println(dt.format(DateTimeFormatter.ISO_DATE_TIME));
```

結果は何ですか？

- A.
- B. 実行時に例外がスローされます。  
2014-07-31T01:01:00
- C. 2014-07-31
- D. 2014-09-30T00:00:00

**Answer: C** ([メッセージを残す](#))

最新問題: 7

与えられた結果: 結果は何ですか？

```
public class MyFor3 {
    public static void main(String[] args) {
        int[] xx = null;
        for (int ii: xx) {
            System.out.println(ii);
        }
    }
}
```

- A. null
- B. コンパイルに失敗しました
- C. Java.lang.NullPointerException
- D. 0

**Answer: A** ([メッセージを残す](#))

配列変数 (ここでは xx) は null 値を持つ可能性があります。

注記 :

Null は、Java で void 参照、つまりへのポインタを表すために使用される予約定数です。

何もない。内部的には単なるバイナリの 0 ですが、高水準 Java 言語では魔法のようなものになります。

ゼロとはまったく異なる定数で、内部的には任意の表現が可能です。

最新問題: 8

MyException クラスと Test クラスの次のコードを考えます。

```

public class MyException extends RuntimeException {}

public class Test {
    public static void main(String[] args) {
        try {
            method1();
        }
        catch (MyException ne) {
            System.out.print("A");
        }
    }
    public static void method1() { // line n1
        try {
            throw Math.random() > 0.5 ? new MyException() : new RuntimeException();
        }
        catch (RuntimeException re) {
            System.out.print("B");
        }
    }
}

```

結果は何ですか？

- A. A
- B. A または B のいずれか
- C. n1 行目でコンパイル時エラーが発生しました
- D. B
- E. AB

**Answer: D** ([メッセージを残す](#))

最新問題: 9

コードの断片を考えると、次のようになります。

```

if (aVar++ < 10) {
    System.out.println(aVar + " Hello World!");
} else {
    System.out.println(aVar + " Hello Universe!");
}

```

整数 aVar が 9 の場合、結果はどうなりますか？

- A. こんにちは、ユニバース!
- B. コンパイルは失敗します。
- C. こんにちは、世界!
- D. 10 Hello World!

**Answer: D** ([メッセージを残す](#))

最新問題: 10

与えられる:



```
public static void main(String[] args) {  
  
    String[][] chs = new String[2][];  
    chs[0] = new String[2];  
    chs[1] = new String[5];  
    int i = 97;  
  
    for (int a = 0; a < chs.length; a++) {  
        for (int b = 0; b < chs[a].length; b++)  
            chs[a][b] = "" + i;  
        i++;  
    }  
  
    for (String[] ca : chs) {  
        for (String c : ca) {  
            System.out.print(c + " ");  
        }  
        System.out.println();  
    }  
}
```

結果は何ですか？

- A. 97 9899 100 101 102 103
- B. コンパイル レール。
- C. 実行時に NullPointerException がスローされます。
- D. 97 9899 100 null null null
- E. 実行時に ArrayIndexOutOfBoundsException がスローされます。

**Answer: D** ([メッセージを残す](#))

最新問題: 11

Java クラスの構造に関して正しい 3 つの記述はどれですか？

- A. クラスはプライベート コンストラクターを 1 つだけ持つことができます。
- B. メソッドはフィールドと同じ名前を持つことができます。
- C. クラスにはオーバーロードされた静的メソッドを含めることができます。
- D. パブリック クラスには main メソッドが必要です。
- E. メソッドはクラスの必須コンポーネントです。
- F. フィールドは使用前に初期化する必要はありません。

**Answer: A,B,C** ([メッセージを残す](#))

A: プライベート コンストラクターは、クラスがそのクラスによって明示的にインスタンス化されることを防ぎます。

電話をかける人。

プログラマがクラスのコンストラクタを提供しない場合、システムは常に

デフォルトの引数なしの public コンストラクターを提供します。このデフォルトのコンストラクターを無効にするには、引数のないプライベート コンストラクターをクラスに追加するだけです。このプライベート コンストラクターは次のとおりです。

空の。

B: 以下は正常に動作します。

```
int ケーキ() {  
int ケーキ = 0;  
(1) を返します。  
}
```

C: Java では静的メソッドをオーバーロードできます。メソッドのオーバーロード静的メソッドに関して

これらは通常の方法と同じであり、静的メソッドをオーバーロードするには、次のことを提供する必要があります。

同じ名前を持つ別の静的メソッドですが、メソッド シグネチャが異なります。

正しくない :

D ではありません: アプリケーション内のパブリック クラスのみが main メソッドを持つ必要があります。

注記 :

例 :

クラスA

```
{  
パブリック文字列何か。  
パブリック int a;  
}
```

Q: メソッドのないクラスを何と呼びますか?

ほとんどの場合: アンチパターンです。

なぜ? Operator クラスとデータを使用した手続き型プログラミングが容易になるため構造物。データと動作を分離することは、必ずしも良い OOP とは言えません。

多くの場合: DTO (データ転送オブジェクト)

ビジネス/ドメイン オブジェクトから派生した、データ交換を目的とした読み取り専用のデータ構造。

場合によっては: データ構造だけ。

まあ、場合によっては、単純なデータを保持するための構造が必要になることもあります。

シンプルで操作はありません。

F ではない: フィールドを初期化する必要があります。そうでない場合、コードはコンパイルされません。

例 :

コンパイルできないソース コード - 変数 x が初期化されていない可能性があります

### 最新問題: 12

与えられる:

```
int x = 10;

if (x > 10) {

    System.out.println(">");

} else if (x < 10) {

    System.out.println("<");

} else {

    System.out.println("=");

}
```

上記のコード部分と同等のものは次のうちどれですか？

- A. `System.out.println(x>10?">"; "<"; "=");`
- B. `System.out.println(x>10? ">"? "<": "=");`
- C. `System.out.println(x>10?">":x<10?"<": "=");`
- D. `System.out.println(x>10?">"? "<"? "=");`
- E. 上記のどれでもない

**Answer: B** ([メッセージを残す](#))

オプション A は、非抽象メソッドで抽象を使用できないため不正解です (このメソッドにはメソッド本体があります)。オプション C は、メソッドをオーバーライドするときに、より制限的なアクセス修飾子を使用できないため、プライベートを使用してデフォルトをオーバーライドしようとしているため、不正解です。access Level メソッドを使用するとコンパイル時にエラーが発生します。デフォルトのメソッド (デフォルトのアクセス レベルのメソッドではない) はインターフェイスでのみ許可されるため、オプション D は不正解です。オプション E は、メソッド all ready の戻り値の型が void であるため、int を追加できないため、不正解です。オプション B は正解です。メソッドは非抽象 <https://docs.oracle.com/javase/tutorial/java/landl/polymorphism.html> であるため、そこで Final を使用できます。

### 最新問題: 13

コードの断片を考えると、次のようになります。

```

public class App {
    public static void main(String[] args) {
        String str1 = "Java";
        String str2 = new String("java");
        //line n1
        {
            System.out.println("Equal");
        } else {
            System.out.println("Not Equal");
        }
    }
}

```

n1 行目に挿入されると、App クラスが Equal を出力できるようにするコードフラグメントはどれですか？

- A) `str1.toLowerCase();`  
`if (str1 == str2)`
- B) `if (str2.equals(str1.toLowerCase()))`
- C) `str1.toLowerCase();`  
`if (str1.equals(str1.toLowerCase()))`
- D) `if (str1.toLowerCase() == str2.toLowerCase())`

- A. オプション A
- B. オプション B
- C. オプション D
- D. オプション C

**Answer: B** ([メッセージを残す](#))

#### 最新問題: 14

次のコードがあるとします。このコードが実行された後の intArr の各要素の値は何ですか？

```

int[] intArr = {15, 30, 45, 60, 75};
intArr[2] = intArr[4];
intArr[4] = 90;

```

- A. 15、60、45、90、75
- B. 15、90、45、90、75
- C. 15、30、75、60、90
- D. 15、30、90、60、90
- E. 15、4、45、60、90

**Answer: B** ([メッセージを残す](#))

注: 私は混乱しています。インターラーが 4 のときに、C が正しいと言えるでしょうか?

**最新問題: 15**

次のコードの場合: 出力は何ですか?

```
public static void main(String[] args){  
    String[] planets = {"Mercury", "Venus", "Earth", "Mars"};  
  
    System.out.println(planets.length);  
    System.out.println(planets[1].length());  
}
```

- A. 4 4
- B. 5 4
- C. 4 7
- D. 4 21
- E. 4 5
- F. 3 5

**Answer: B** ([メッセージを残す](#))

**最新問題: 16**

与えられる:

```
class Caller {
    private void init () {
        System.out.println("Initialized");
    }

    private void start () {
        init();
        System.out.println("Started");
    }
}

public class TestCall {
    public static void main(String[] args) {
        Caller c = new Caller();
        c.start(); // line n1
        c.init(); // line n2
    }
}
```

ORACLE®

結果は何ですか？

- A. n1 行目でコンパイルが失敗します。
- B. n2 行目でコンパイルが失敗します。
- C. 初期化済み  
開始しました  
初期化されました
- D. 初期化されました  
開始しました

Answer: ([解答を表示する](#))

有効な **1z0-808** 問題集は GoShiken.com が提供された合格しやすい 1z0-808 試験問題集！  
GoShiken.com が最新の **1z0-808** 試験問題集を提供しています。GoShiken.com 1z0-808 試験  
問題は最新で、解答が正確でございます。最新の GoShiken.com 1z0-808 問題集をゲットする

人はこちら: <https://www.goshiken.com/Oracle/1z0-808-mondaishu.html> (39830%OFF問題集  
と正解付きで 30%w 特別割引コード: **Freepdfdumps**)

最新問題: 17

public クラス内の Field 宣言の protected 修飾子は、フィールド \_\_\_\_\_ を意味し  
ます。

- A. 変更できません
- B. クラス外からの読み取りは可能ですが、書き込みはできません
- C. 同じパッケージ内でのみ、このクラスとそのサブクラスから読み書きできます。
- D. このクラスと、任意のパッケージで定義されたそのサブクラスから読み書きできます。

Answer: D (メッセージを残す)

参考: <http://beginnersbook.com/2013/05/java-access-modifiers/>

最新問題: 18

public クラス内の Field 宣言の protected 修飾子は、フィールド \_\_\_\_\_ を意味し  
ます。

- A. 変更できません
- B. クラス外からの読み取りは可能ですが、書き込みはできません
- C. 同じパッケージ内でのみ、このクラスとそのサブクラスから読み書きできます。
- D. このクラスと、任意のパッケージで定義されたそのサブクラスから読み書きできます。

Answer: (解答を表示する)

参照 :

<http://beginnersbook.com/2013/05/java-access-modifiers/>

最新問題: 19

次のクラス宣言があるとします。

パブリック抽象クラス Animal パブリック インターフェイス Hunter パブリック クラス Cat  
は、Animal を実装します Hunter パブリック クラス Tiger は、Cat を拡張します  
コンパイルに失敗する答えはどれですか?

----

- A) `ArrayList<Animal> myList = new ArrayList<>();  
myList.add(new Tiger());`
- B) `ArrayList<Hunter> myList = new ArrayList<>();  
myList.add(new Cat());`
- C) `ArrayList<Hunter> myList = new ArrayList<>();  
myList.add(new Tiger());`
- D) `ArrayList<Tiger> myList = new ArrayList<>();  
myList.add(new Cat());`
- E) `ArrayList<Animal> myList = new ArrayList<>();  
myList.add(new Cat());`

ORACLE

- A. オプション A
- B. オプション B
- C. オプション C
- D. オプション D
- E. オプション E

**Answer:** ([解答を表示する](#))

ArrayList ではなく ArrayList() 宣言の右側に注目してください。

最新問題: 20

与えられる:

ACC.java:

```
package p1;  
public class Acc {  
    int p;  
    private int q;  
    protected int r;  
    public int s;  
}
```

Test.java:

```
package p2;  
import p1.Acc;  
public class Test extends Acc {  
    public static void main(String[] args) {  
        Acc obj = new Test();  
    }  
}
```

どの発言が真実ですか？

- A. obj 経由でアクセスできるのは s のみです。
- B. p と s の両方に obj 経由でアクセスできます。
- C. p、r、s には obj 経由でアクセスできます。
- D. r と s の両方に obj 経由でアクセスできます。

**Answer:** ([解答を表示する](#))

最新問題: 21

与えられる:

```

interface Readable {
    public void readBook();
    public void setBookMark();
}

abstract class Book implements Readable { // line n1
    public void readBook() { }
    // line n2
}

class EBook extends Book { // line n3
    public void readBook() { }
    // line n4
}

```

コードのコンパイルを有効にするオプションはどれですか？

- A) Replace the code fragment at line n1 with:  
class Book implements Readable {
- B) At line n2 insert:  
public abstract void setBookMark();
- C) Replace the code fragment at line n3 with:  
abstract class EBook extends Book {
- D) At line n4 insert:  
public void setBookMark() { }

- A. オプション B
- B. オプション A
- C. オプション D
- D. オプション C

**Answer: D** ([メッセージを残す](#))

最新問題: 22

与えられる:

```

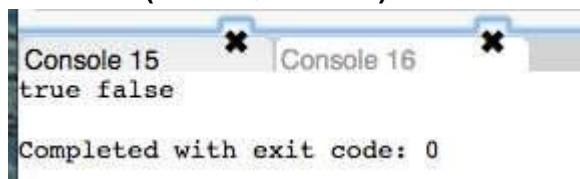
public class Test {
    public static void main(String[] args) {
        Test ts = new Test();
        System.out.print(isAvailable + " ");
        isAvailable= ts.doStuff();
        System.out.println(isAvailable);
    }
    public static boolean doStuff() {
        return !isAvailable;
    }
    static boolean isAvailable = true;
}

```

結果は何ですか？

- A. コンパイルは失敗します。
- B. false true
- C. true false
- D. true true
- E. false false

Answer: ([解答を表示する](#))



```

Console 15 Console 16
true false
Completed with exit code: 0

```

最新問題: 23

与えられた結果: 結果は何ですか？

```

public class Product {
    int id;
    String name;
    public Product(int id, String name) {
        this.id = id;
        this.name = name;
    }
}

```

And given the code fragment:

```

4. Product p1 = new Product (101, "Pen");
5. Product p2 = new Product (101, "Pen");
6. Product p3 = p1;
7. boolean ans1 = p1 == p2;
8. boolean ans2 = p1.name.equals(p2.name);
9. System.out.print(ans1 + ":" - ans2);

```

- A. true:false
- B. true:true
- C. false:false

D. false:true

Answer: ([解答を表示する](#))

最新問題: 24

次の配列があるとします。

```
int[] intArr = {8, 16, 32, 64, 128};
```

この配列の各要素を独立して出力する2つのコード部分はどれですか？

- A) 

```
for (int i : intArr) {  
    System.out.print(intArr[i] + " ");  
}
```
- B) 

```
for (int i : intArr) {  
    System.out.print(i + " ");  
}
```
- C) 

```
for (int i=0 : intArr) {  
    System.out.print(intArr[i] + " ");  
    i++;  
}
```
- D) 

```
for (int i=0; i < intArr.length; i++)  
    System.out.print(i + " ");  
}
```
- E) 

```
for (int i=0; i < intArr.length; i++)  
    System.out.print(intArr[i] + " ");  
}
```
- F) 

```
for (int i; i < intArr.length; i++) {  
    System.out.print(intArr[i] + " ");  
}
```

A. オプション C

B. オプション E

C. オプション F

D. オプション D

E. オプション B

F. オプション A

Answer: ([解答を表示する](#))

最新問題: 25

コードの断片を考えると、次のようになります。

```
public static void main (String [] args) {  
    ArrayList<Integer> points = new ArrayList<> ();  
    points.add (1);  
    points.add (2);  
    points.add (3);  
    points.add (4);  
    points.add (null);  
    points.remove (2);  
    points.remove (null);  
    System.out.println(points);  
}
```

結果は何ですか？

- A. 実行時に NullPointerException がスローされます。
- B. [1, 2, 4]
- C. [1, 2, 4, null]
- D. [1, 3, 4, null]
- E. [1, 3, 4]
- F. コンパイルは失敗します。

**Answer: F** ([メッセージを残す](#))

説明/参照:

Explanation:

The screenshot shows an IDE window with the following content:

```
1- public static void main (String [] args) {
2   ArrayList<Integer> points = new ArrayList<> ();
3   points.add (1);
4   points.add (2);
5   points.add (3);
6   points.add (4);
7   points.add (null);
8   points.remove (null);
9   System.out.println (points);
10 }
```

External Libraries ...

cs1.keyboard

Input Arguments (args of Main Method)...

Interactive mode :  OFF

Stdin Inputs...

Result...  
compiled and executed in 0 second(s)

No "public class" found to execute

**最新問題: 26**

Java クラスの構造に関して正しい 3 つの記述はどれですか？ (3つお選びください。)

- A. クラスにそのフィールドと同じ名前を付けることはできません。
- B. メソッドとフィールドはクラスのオプションのコンポーネントです。
- C. フィールドは使用前に初期化する必要があります。
- D. パブリック クラスには main メソッドが必要です。
- E. クラスにはオーバーロードされたプライベート コンストラクターを含めることができます。
- F. クラスには、final static メソッドを含めることができます。

**Answer: C,D,E (メッセージを残す)**

**最新問題: 27**

与えられる:

```
class X {
    int x1, x2, x3;
}
class Y extends X {
    int y1;
    Y() {
        x1 = 1;
        x2 = 2;
        y1 = 10;
    }
}

class Z extends Y {
    int z1;
    Z() {
        x1 = 3;
        y1 = 20;
        z1 = 100;
    }
}

And,

public class Test3 {
    public static void main(String[] args) {
        Z obj = new Z();
        System.out.println(obj.x3 + ", " + obj.y1 + ", " + obj.z1);
    }
}
```

変数 x3 を初期化するコンストラクターはどれですか？

- A. クラス Z の引数なしのコンストラクターのみ
- B. オブジェクトクラスのデフォルトコンストラクターのみ
- C. クラス X のデフォルトのコンストラクターのみ
- D. クラス Y の引数なしのコンストラクターのみ

Answer: A ([メッセージを残す](#))

最新問題: 28

コードの断片を考えると、次のようになります。

A.java:

```
package p1;
public class A {
}
```

B.java:

```
package p1.p2;
//line n1
public class B {
    public void doStuff() {
        A b = new A ();
    }
}
```

C.java:

```
package p3;
//line n2
public class C {
    public static void main(String[] args) {
        A o1 = new A();
        B o2 = new B();
    }
}
```

コードをコンパイルできるようにする変更はどれですか？

- A. 行 n1 を次のように置き換えます。import p1; 行 n2 を次のように置き換えます。p1.p2 をインポートします。
- B. 行 n1 を次のように置き換えます。行 n2 を次のように置き換えます。import p1.\*;
- C. 行 n1 を次のように置き換えます: import p1.\*; 行 n2 を次のように置き換えます。import p1.p2.\*;
- D. 行 n1 を次のように置き換えます。import p1.A; 行 n2 を次のように置き換えます。p1.p2.B をインポートします。

**Answer: C** ([メッセージを残す](#))

最新問題: 29

与えられる:

```
1. public class SampleClass {
2.     public static void main(String[] args){
3.         AnotherSampleClass asc = new AnotherSampleClass();
4.         SampleClass sc = new SampleClass();
5.         //insert code here
6.     }
7. }
8. class AnotherSampleClass extends SampleClass {
9. }
```

5 行目に挿入された場合、有効な変更となるのはどのステートメントですか？

- A. asc = sc;
- B. sc = asc;
- C. asc = (オブジェクト) sc;
- D. asc = sc.clone ()

**Answer: B** ([メッセージを残す](#))

正常に動作します。

最新問題: 30

与えられる:

```
public class App {
    public static void main(String[] args) {
        Boolean[] bool = new Boolean[2];

        bool[0] = new Boolean(Boolean.parseBoolean("true"));
        bool[1] = new Boolean(null);

        System.out.println(bool[0] + " " + bool[1]);
    }
}
```

結果は何ですか？

- A. 実行時に NullPointerException がスローされる
- B. 真の null
- C. 真偽
- D. コンパイルに失敗しました

**Answer: C** ([メッセージを残す](#))

最新問題: 31

次の 2 つのクラスがあるとします。

```

public class Customer {
    ElectricAccount acct = new ElectricAccount()

    public void useElectricity(double kWh) {
        acct.addKWh(kWh);
    }
}

public class ElectricAccount {
    private double kWh;
    private double rate = 0.07;
    private double bill;

    //line n1
}

```

メンバー変数 bill がメンバー変数 kWh の値にメンバー変数 rate を乗算した値と常に等しくなるように、ElectricAccount クラスの n1 行目にメソッドを記述するにはどうすればよいでしょうか？顧客（顧客クラスのインスタンスで表される）が使用する電力量は、useElectricity メソッドを通じて顧客の請求書（メンバー変数 bill で表される）に寄与する必要があります。customer クラスのインスタンスは、メンバー変数 bill の値を決して改ざんしたり、値を減らしたりしてはなりません。

- A) public void addKWh(double kWh) {
 this.kWh += kWh;
 this.bill = this.kWh\*this.rate;
 }
- B) public void addKWh(double kWh) {
 if (kWh > 0){
 this.kWh += kWh;
 this.bill = this.kWh \* this.rate;
 }
 }
- C) private void addKWh(double kWh) {
 if (kWh > 0) {
 this.kWh += kWh;
 this.bill = this.kWh\*this.rate;
 }
 }
- D) public void addKWh(double kWh) {
 if(kWh > 0) {
 this.kWh += kWh;
 setBill(this.kWh);
 }
 }
 public void setBill(double kWh) {
 bill = kWh\*rate;
 }

- A. オプション C
- B. オプション D
- C. オプション B
- D. オプション A

Answer: D ([メッセージを残す](#))

有効な 1z0-808 問題集は GoShiken.com が提供された合格しやすい 1z0-808 試験問題集！  
GoShiken.com が最新の 1z0-808 試験問題集を提供しています。GoShiken.com 1z0-808 試験問題は最新で、解答が正確でございます。最新の GoShiken.com 1z0-808 問題集をゲットする人はこちら: <https://www.goshiken.com/Oracle/1z0-808-mondaishu.html> (39830%OFF問題集溶と正解付きで 30%w 特別割引コード: **Freepdfdumps**)

最新問題: 32

与えられる:

ACC.java:

```
package p1;
public class Acc {
    int p;
    private int q;
    protected int r;
    public int s;
}
```

ORACLE®

Test.java:

```
package p2;
import p1.Acc;
public class Test extends Acc {
    public static void main(String[] args) {
        Acc obj = new Test();
    }
}
```

どの発言が真実ですか？

- A. p、r、および s は obj からアクセスできます。
- B. obj からアクセスできるのは s のみです。
- C. p と s の両方に obj からアクセスできます。
- D. r と s の両方に obj からアクセスできます。

Answer: ([解答を表示する](#))

最新問題: 33

与えられる:

```
1. public class Whizlabs{
2.
3.     public static void main(String[] args){
4.         try{
5.             Double number = Double.valueOf("120D");
6.         }catch(NumberFormatException ex){
7.         }
8.         System.out.println(number);
9.     }
10. }
```

結果は何ですか？

- A. 120
- B. 120D
- C. NumberFormatException がスローされます。
- D. 5 行目のエラーによりコンパイルが失敗します。
- E. 8 行目のエラーによりコンパイルが終了します。

**Answer:** ([解答を表示する](#))

5 行目で、120D を渡すことによって double のラッパー オブジェクトを作成しました。

Double に変換できるため、例外はありません。しかし、よく確認してみると、変数番号が try ブロック内で宣言されていることがわかります。そのため、変数番号の範囲はそのブロックに限定されており、外部からアクセスしようとするときコンパイル時にエラーが発生します。<https://docs.oracle.com/javase/tutorial/iava/nutsandbolts/variables.html>

**最新問題: 34**

Java バイト コードに関して正しいのはどれですか？

- A. 任意のプラットフォームで実行できます。
- B. プラットフォーム用にコンパイルされている場合にのみ、どのプラットフォームでも実行できます。
- C. Java ランタイム環境を備えた任意のプラットフォームで実行できます。
- D. Java コンパイラを備えた任意のプラットフォームで実行できます。
- E. プラットフォームに Java ランタイム環境と Java コンパイラの両方がある場合にのみ、プラットフォーム上で実行できます。

**Answer: D** ([メッセージを残す](#))

Java バイトコードは、「一度書けばどこでも実行できる」ことを可能にします。Java コンパイラを備えた任意のプラットフォームでプログラムをバイトコードにコンパイルできます。その後、バイトコードは Java VM の任意の実装上で実行できます。つまり、コンピュータに Java VM が搭載されている限り、Java プログラミング言語で書かれた同じプログラムを Windows 2000、Solaris ワークステーション、または iMac 上で実行できるということです。

参照: <http://www.math.uni-hamburg.de/doc/java/tutorial/getStarted/intro/definition.html>

最新問題: 35

コードの断片を考えると、次のようになります。

```
int wd = 0;
String days[] = {"sun", "mon", "wed", "sat"};
for (String s:days) {
    switch (s) {
        case "sat":
        case "sun":
            wd -= 1;
            break;
        case "mon":
            wd -= 1;
            break;
        case "wed":
            wd += 2;
    }
}
System.out.println(wd);
```

結果は何ですか？

- A. コンパイルは失敗します。
- B. -1
- C. 3
- D. 0

Answer: ([解答を表示する](#))

最新問題: 36

与えられる:

```
public class Fieldinit {
    char c;
    boolean b;
    float f;
    void printAll() {
        System.out.println ("c = " + c);
        System.out.println ("b = " + b);
        System.out.println ("f = " + f);
    }
    public static void main (String [] args) {
        FieldInit f = new FieldInit ();
        f.printAll ();
    }
}
```

結果は何ですか？

- A. c= nullb = truef = 0.0
- B. c= nullb = falsef = 0.0F
- C. c=b = falsef = 0.0
- D. c=0b = falsef = 0.0f

**Answer: C** ([メッセージを残す](#))

最新問題: 37

与えられた結果: 結果は何ですか？

```
class Vehicle {
    int x;
    Vehicle() {
        this(10); // line n1
    }
    Vehicle(int x) {
        this.x = x;
    }
}

class Car extends Vehicle {
    int y;
    Car() {
        super();
        this(20); // line n2
    }
    Car(int y) {
        this.y = y;
    }
    public String toString() {
        return super.x + ":" + this.y;
    }
}
```

And given the code fragment:

And given the code fragment:

```
Vehicle y = new Car();
System.out.println(y);
```

- A. n2 行目でコンパイルが失敗します。
- B. n1 行目でコンパイルが失敗します。
- C. 0:20
- D. 10:20

**Answer: D** ([メッセージを残す](#))

最新問題: 38

次のクラスがあるとします。

```
public class Employee {
    public int salary;
}

public class Manager extends Employee {
    public int budget;
}

public class Director extends Manager {
    public int stockOptions;
}

And given the following main method:

public static void main(String[] args) {
    Employee employee = new Employee();
    Manager manager = new Manager();
    Director director = new Director();
    //line n1
}
```

main メソッドの n1 行目に配置するとコンパイルに失敗する 2 つのオプションはどれですか? (2 つお選びください。)

- A. director.stockOptions = 1\_000;
- B. 従業員.予算 = 200\_000;
- C. 従業員.給与 = 50\_000;
- D. マネージャー.ストックオプション = 500;
- E. ディレクター.給与 = 80\_000;
- F. マネージャー.予算 = 1\_000\_000;

**Answer:** ([解答を表示する](#))

最新問題: 39

MyException クラスと Test クラスの次のコードを考えます。

```
public class MyException extends RuntimeException {}

public class Test {
    public static void main(String[] args) {
        try {
            method1();
        }
        catch (MyException ne) {
            System.out.print("A");
        }
    }
    public static void method1() { // line n1
        try {
            throw Math.random() > 0.5 ? new MyException() : new RuntimeException();
        }
        catch (RuntimeException re) {
            System.out.print("B");
        }
    }
}
```



結果は何ですか？

- A. A または B のいずれか
- B. AB
- C. n1 行目でコンパイル時エラーが発生しました
- D. B
- E. A

Answer: D ([メッセージを残す](#))

最新問題: 40

与えられる:

```
class A {
    public A() {
        System.out.print("A ");
    }
}

class B extends A {
    public B() { //line n1
        System.out.print("B ");
    }
}

class C extends B {
    public C() { //line n2
        System.out.print("C ");
    }
    public static void main(String[] args) {
        C c = new C();
    }
}
```

結果は何ですか？

- A. CBA
- B. 行 n1 と行 n2 でコンパイルが失敗します。
- C. ABC
- D. C

**Answer:** ([解答を表示する](#))

最新問題: 41

与えられる:

ORACLE

```
class Patient {
    String name;
    public Patient (String name) {
        this.name = name;
    }
}
```

コードの断片は次のとおりです。

ORACLE

```
8. public class Test {
9.     public static void main (String [] args) {
10.         List ps = new ArrayList ();
11.         Patient p2 = new Patient ("Mike");
12.         ps.add(p2);
13.
14.         // insert code here
15.
16.         if (f >= 0) {
17.             System.out.print ("Mike Found");
18.         }
19.     }
20. }
```

14 行目に挿入すると、Mike Found を出力できるコード フラグメントはどれですか？

A

```
int f = ps.indexOf (p2);
```

B

```
int f = ps.indexOf (Patient ("Mike") );
```

C

```
int f = ps.indexOf (new Patient "Mike" );
```

D

```
Patient p = new Patient ("Mike");  
int f = ps.indexOf (p)
```

A. オプション C

B. オプション A

C. オプション B

D. オプション D

Answer: B ([メッセージを残す](#))

最新問題: 42

与えられる:

```
MainTest.java:  
  
public class MainTest {  
  
    public static void main(int[] args) {  
        System.out.println("int main " + args[0]);  
    }  
    public static void main(Object[] args) {  
        System.out.println("Object main " + args[0]);  
    }  
    public static void main(String[] args) {  
        System.out.println("String main " + args[0]);  
    }  
}  
  
and commands:  
  
javac MainTest.java  
java MainTest 1 2 3
```

結果は何ですか？

A. 文字列メイン 1

B. 実行時に例外がスローされます

C. コンパイルに失敗しました

D. オブジェクトメイン 1

E. int メイン 1

Answer: ([解答を表示する](#))

最新問題: 43

コードの断片を考えると、次のようになります。

```
public static void main(String[] args) {
    String date = LocalDate
        .parse("2014-05-04")
        .format(DateTimeFormatter.ISO_DATE_TIME);
    System.out.println(date);
}
```

結果は何ですか？

- A. 実行時に例外がスローされます。
- B. 2014-05-04T00:00:00.000
- C. 5/4/14T00:00:00.000
- D. 2014 年 5 月 4 日T00:00:00.000

Answer: A ([メッセージを残す](#))

最新問題: 44

コードの断片を考えると、次のようになります。

```
1. public class Test {
2.     public static void main(String[] args) {
3.         /* insert code here */
4.         array[0]=10;
5.         array[1]=20;
6.         System.out.print(array[0]+":"+array[1]);
7.     }
8. }
```

3 行目に挿入すると、コードが 10:20 を出力できるようにするコード フラグメントはどれですか？

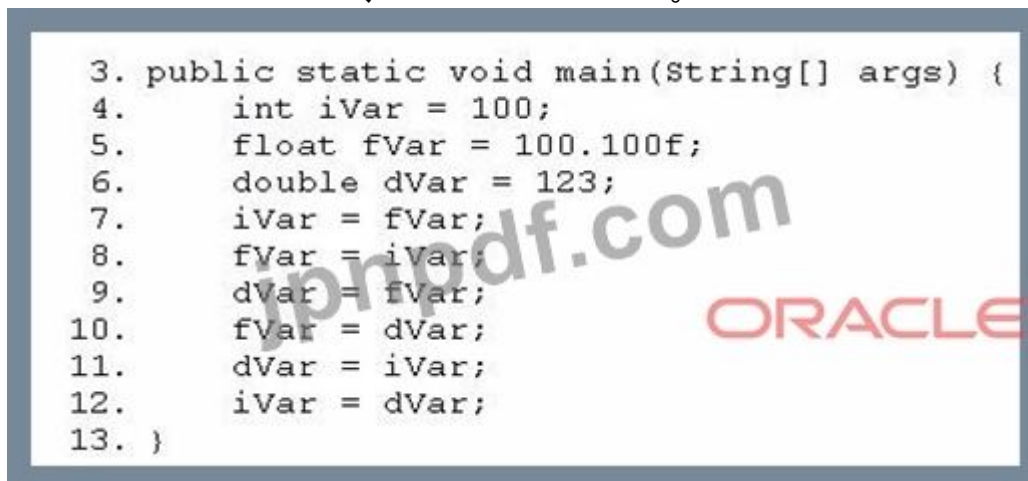
- A. int[] 配列 = new int[2];
- B. int[] 配列;配列 = int[2];
- C. int 配列 = new int[2];
- D. int 配列 [2];

Answer: A ([メッセージを残す](#))



最新問題: 45

コードの断片を考えると、次のようになります。



コンパイルに失敗する3行はどれですか? \$つお選びください。)

- A. 10 行目
- B. 7 行目
- C. 11 行目
- D. 9 行目
- E. 12 行目
- F. 8 行目

Answer: A,B,E (メッセージを残す)

最新問題: 46

与えられる:

```
public class MyField {
    int x;
    int y;
    public void doStuff(int x, int y) {
        this.x = x;
        y = this.y;
    }
    public void display () {
        System.out.print(x + " " + y + ":");
    }
    public static void main(String[] args) {
        MyField m1 = new MyField();
        m1.x = 100;
        m1.y = 200;
        MyField m2 = new MyField();
        m2.doStuff(m1.x, m1.y);
        m1.display();
        m2.display();
    }
}
```

結果は何ですか？

A. 100200 : 100 200 :

B. 100 200 : 100 0 :

C. 100 0 : 100 200:

D. 100 0 : 100 0 :

Answer: D ([メッセージを残す](#))

有効な 1z0-808 問題集は GoShiken.com が提供された合格しやすい 1z0-808 試験問題集！  
GoShiken.com が最新の 1z0-808 試験問題集を提供しています。GoShiken.com 1z0-808 試験  
問題は最新で、解答が正確でございます。最新の GoShiken.com 1z0-808 問題集をゲットする

人はこちら: <https://www.goshiken.com/Oracle/1z0-808-mondaishu.html> (39830%OFF問題集  
と正解付きで 30%w 特別割引コード: **Freepdfdumps**)

最新問題: 47

与えられる:

```
public class Test {  
  
    public static void main(String[] args) {  
        if (args[0].equals("Hello") ? false : true) {  
            System.out.println("Success");  
        } else {  
            System.out.println("Failure");  
        }  
    }  
}
```

そして次のコマンドを与えます:

```
javac Test.Java  
Java Test Hello
```

結果は何ですか?

- A. 実行時に例外がスローされます
- B. コンパイルは失敗します。
- C. 失敗
- D. 成功

Answer: ([解答を表示する](#))

最新問題: 48

与えられる:

```
public class App {  
  
    public static void main(String[] args) {  
        Boolean[] bool = new Boolean[2];  
  
        bool[0] = new Boolean(Boolean.parseBoolean("true"));  
        bool[1] = new Boolean(null);  
  
        System.out.println(bool[0] + " " + bool[1]);  
    }  
}
```

結果は何ですか?

- A. 真偽
- B. コンパイルに失敗しました

- C. 実行時に NullPointerException がスローされる
- D. 真の null

Answer: A ([メッセージを残す](#))

最新問題: 49

与えられた結果: 結果は何ですか?

```
class Test {
    public static void main(String[] args) {
        int numbers[];
        numbers = new int[2];
        numbers[0] = 10;
        numbers[1] = 20;

        numbers = new int[4];
        numbers[2] = 30;
        numbers[3] = 40;
        for (int x : numbers) {
            System.out.print(" "+x);
        }
    }
}
```

- A. 実行時に例外がスローされます
- B. 0 0 30 40
- C. コンパイルに失敗しました
- D. 10 20 30 40

Answer: ([解答を表示する](#))

最新問題: 50

コードの断片を考えると、次のようになります。

```
public static void main(String[] args) {
    String[][] arr = {"A", "B", "C"}, {"D", "E"};
    for (int i = 0; i < arr.length; i++) {
        for (int j = 0; j < arr[i].length; j++) {
            System.out.print(arr[i][j] + " ");
            if (arr[i][j].equals("B")) {
                break;
            }
        }
        continue;
    }
}
```

結果は何ですか?

- A. ABCDE
- B. コンパイルは失敗します。
- C. ABDE
- D. ABC

Answer: ([解答を表示する](#))

最新問題: 51

カプセル化を最もよく説明しているのはどれですか？

- A. カプセル化により、抽象として宣言された一部のフィールドとメソッドを使用してクラスを設計できるようになります。
- B. カプセル化により、クラスのメソッドが継承可能になるようにクラスを設計できます。
- C. カプセル化により、メソッドに引数 MyType x がある場合に、MyType のサブクラスをそのメソッドに渡すことができるようにクラスを設計できるようになります。
- D. カプセル化により、オブジェクトの特定のフィールドとメソッドのみが他のオブジェクトからアクセスできるようにクラスを設計できるようになります。

Answer: ([解答を表示する](#))

最新問題: 52

コードの断片を考えると、次のようになります。

```
System.out.println ("結果: " +3+5);  
System.out.println ("結果: " + (3+5));
```

結果は何ですか？

- A. 結果: 8 結果: 8
- B. 結果: 35 結果: 8
- C. 結果: 8 結果: 35
- D. 結果: 35 結果: 35

Answer: B ([メッセージを残す](#))

最初のステートメントでは、3 と 5 は文字列として扱われ、単純に連結されます。最初のステートメントでは、3 と 5 が整数として扱われ、それらの合計が計算されます。

最新問題: 53

コンパイルされた Java ソース ファイルのコード スニペットを考えると、次のようになります。

```
public class MyFile  
{  
    public static void main (String[] args)  
    {  
        String arg1 = args[0];  
        String arg2 = args[1];  
        String arg3 = args[2];  
        System.out.println("Arg is " + arg3);  
    }  
}
```

そしてこの出力:

```
Arg is 2
```

この出力を取得するにはどのコマンドを実行する必要がありますか？

- A. Java MyFile 2
- B. Java MyFile 1 2 3 4

C. java MyFile 1 2 2

D. Java MyFile 2 2

**Answer: A** ([メッセージを残す](#))

説明/参照:

**最新問題: 54**

コンパイルに失敗する 2 つのクラス定義はどれですか？

A. 最終クラス A1 {public A1(){}}

B. 最終抽象クラス A5 {protected static int i;void doStuff(){abstract void dolt();}

C. 抽象クラス A3 {private static int i;public void doStuff(){public A3(){}}

D. パブリック クラス A2 {private static int i;private A2(){}}

E. クラス A4 {protected static Final int i;private void doStuff(){}}

**Answer: B,E** ([メッセージを残す](#))

**最新問題: 55**

次の配列があるとします。この配列内の各要素を独立して出力する 2 つのコード フラグメントはどれですか？

```
int[] intArr = {8, 16, 32, 64, 128};
```

- ORACLE®
- A) 

```
for (int i : intArr) {
    System.out.print(intArr[i] + " ");
}
```
  - B) 

```
for (int i : intArr) {
    System.out.print(i + " ");
}
```
  - C) 

```
for (int i=0 : intArr) {
    System.out.print(intArr[i] + " ");
    i++;
}
```
  - D) 

```
for (int i=0; i < intArr.length; i++) {
    System.out.print(i + " ");
}
```
  - E) 

```
for (int i=0; i < intArr.length; i++) {
    System.out.print(intArr[i] + " ");
}
```
  - F) 

```
for (int i; i < intArr.length; i++) {
    System.out.print(intArr[i] + " ");
}
```

- A. オプション A
- B. オプション B
- C. オプション C
- D. オプション D
- E. オプション E
- F. オプション F

**Answer: B,E** ([メッセージを残す](#))

残りのすべてのオプションに構文エラーがあります

最新問題: 56

コードの断片を考えると、次のようになります。

```
public class App {
    public static void main(String[] args) {
        String str1 = "Java";
        String str2 = new String("java");
        //line n1
        {
            System.out.println("Equal");
        } else {
            System.out.println("Not Equal");
        }
    }
}
```

n1 行目に挿入されると、App クラスが Equal を出力できるようにするコードフラグメントはどれですか？

- A) str1.toLowerCase();  
if (str1 == str2)
- B) if (str2.equals(str1.toLowerCase()))
- C) str1.toLowerCase();  
if (str1.equals(str2))
- D) if (str1.toLowerCase() == str2.toLowerCase())

- A. オプション C
- B. オプション A
- C. オプション B
- D. オプション D

Answer: ([解答を表示する](#))

最新問題: 57

コードの断片を考えると、次のようになります。

```

public class Test {
    void readCard(int cardNo) throws Exception {
        System.out.println("Reading Card");
    }

    void checkCard(int cardNo) throws RuntimeException { // line n1
        System.out.println("Checking Card");
    }

    public static void main(String[] args) {
        Test ex = new Test();
        int cardNo = 12344;
        ex.readCard(cardNo); //line n2
        ex.checkCard(cardNo); //line n3
    }
}

```

結果は何ですか？

Reading Card

A. Checking Card

B. コンパイルは n3 行目でのみ失敗します。

C. コンパイルは行 n1 でのみ失敗します。

D. コンパイルは行 n2 でのみ失敗します。

E. 行 n2 と行 n3 の両方でコンパイルが失敗します。

**Answer: B** ([メッセージを残す](#))

最新問題: 58

コードの断片を考えると、次のようになります。

```

int nums1[] = new int[3];
int nums2[] = {1, 2, 3, 4, 5};
nums1 = nums2;
for (int x : nums1){
    System.out.print(x + ":");
}

```

結果は何ですか？

A. コンパイルは失敗します。

B. 実行時に ArrayOutOfBoundsException がスローされます。

C. 1:2:3:4:5:

D. 1:2:3:

**Answer: C** ([メッセージを残す](#))

最新問題: 59

コードの断片を考えると、次のようになります。

```

public class Game {
    public static void menu() {
        system.out.println("1. Left 2. Right 0. Stop");
    }
    public static void main(String[] args) {
        int option;
        /* insert code here */
    }
}

```

そしてアプリケーションの要件:

※メニューを表示する必要があります。

\* 選択したオプションを印刷する必要があります。

\* '0' が読み取られるまで実行を継続する必要があります。

要件を満たすにはどのコードフラグメントを使用できますか?

A. `for (option = 0; option != 0; option = //code that reads the option goes here) {`  
`/* code that print the option go here */`  
`}`

B. `while (option != 0) {`  
`menu();`  
`option = // code that reads the option goes here`  
`/* code that print the option go here */`  
`}`

C. `do {`  
`menu();`  
`option = // code that reads the option goes here`  
`/* code that print the option go here */`  
`} while (option != 0);`

D. `while (option >= 0) {`  
`menu ();`  
`option = // code that reads the option goes here`  
`/* code that print the option go here */`  
`}`

A. オプション C

B. オプション A

C. オプション D

D. オプション B

Answer: [\(解答を表示する\)](#)

与えられる:

```
public class App {
    int count;
    public static void displayMsg () {
        count++; // line n1
        System.out.println ("Welcome "+ "Visit Count: "+count); // line n2
    }
    public static void main (String [] args) {
        App.displayMsg (); // line n3
        App.displayMsg (); // line n4
    }
}
```

結果は何ですか？

- A. 行 n1 と行 n2 でコンパイルが失敗します。
- B. 行 n3 と行 n4 でコンパイルが失敗します。
- C. ようこそ訪問数:1  
ようこそ訪問数: 2
- D. ようこそ訪問数:1  
ようこそ訪問数: 1

Answer: A ([メッセージを残す](#))

最新問題: 61

与えられる:

```
public class Test {
    public static void main(String[] args) {
        int x = 1;
        int y = 0;
        if(x++ > ++y) {
            System.out.print("Hello ");
        } else {
            System.out.print("Welcome ");
        }
        System.out.print("Log " + x + ":" + y);
    }
}
```

結果は何ですか？

- A. こんにちはログ 2:1
- B. こんにちはログ 1:0
- C. ウェルカム ログ 1:0
- D. ウェルカムログ 2:1


Answer: D ([メッセージを残す](#))

有効な 1z0-808 問題集は GoShiken.com が提供された合格しやすい 1z0-808 試験問題集！  
GoShiken.com が最新の 1z0-808 試験問題集を提供しています。GoShiken.com 1z0-808 試験問題は最新で、解答が正確でございます。最新の GoShiken.com 1z0-808 問題集をゲットする人はこちら: <https://www.goshiken.com/Oracle/1z0-808-mondaishu.html> (39830%OFF問題集溶と正解付きで 30%w 特別割引コード: **Freepdfdumps**)

最新問題: 62

コードの断片を考えると、次のようになります。

```
public static void main (String[] args) {
    String[] arr = ("Hi", "How", "Are", "You");
    List<String> arrList = new ArrayList<>(Arrays.asList(arr));
    if (arrList.removeIf((String s) -> (return s.length() <= 2;))) {
        System.out.println(s + "removed")
    }
}
```



結果は何ですか？

- A. コンパイルは失敗します。
- B. こんにちは、削除されました
- C. 実行時に UnsupportedOperationException がスローされます。
- D. プログラムはコンパイルされますが、何も出力されません。

Answer: ([解答を表示する](#))

最新問題: 63

与えられる:

```

class X {
    static int i;
    int j;
    public static void main(String[] args) {
        X x1 = new X();
        X x2 = new X();
        x1.i = 3;
        x1.j = 4;
        x2.i = 5;
        x2.j = 6;
        System.out.println(
            x1.i + " " +
            x1.j + " " +
            x2.i + " " +
            x2.j);
    }
}

```

結果は何ですか？

- A. 3 4 3 6
- B. 3 6 4 6
- C. 5 4 5 6
- D. 3 4 5 6

Answer: C ([メッセージを残す](#))

最新問題: 64

次のクラスがあるとします。

```

public class CheckingAccount {
    public int amount;
    // line n1
}

```

And given the following main method, located in another class:

```

public static void main (String [] args) {
    CheckingAccount acct = new CheckingAccount ();
    //line n2
}

```

独立して挿入された場合、amount の値が 100 に設定される 3 つのコードはどれですか？

```
A. At line n2 insert:
    amount = 100;
B. At line n2 insert:
    This. amount = 100
C. At line n2 insert:
    acct.amount = 100
D. At line n1 insert:
    public CheckingAccount () {
        amount = 100;
ORACLE }
E. At line n1 insert:
    public CheckingAccount () {
        this.amount = 100;
    }
F. At line n1 insert:
    public CheckingAccount () {
        acct.amount = 100;
    }
```

- A. オプション E
- B. オプション A
- C. オプション B
- D. オプション D
- E. オプション F
- F. オプション C

Answer: A,D,F ([メッセージを残す](#))

最新問題: 65

次のクラスがあるとします。

```

public class Employee {
    public int salary;
}

public class Manager extends Employee {
    public int budget;
}

public class Director extends Manager {
    public int stockOptions;
}

```

And given the following main method:

```

public static void main(String[] args) {
    Employee employee = new Employee();
    Manager manager = new Manager();
    Director director = new Director();
    //line n1
}

```

main メソッドの n1 行目に配置するとコンパイルに失敗する 2 つのオプションはどれですか？

- A. manager.stockOption = 500;
- B. 従業員.予算 = 200\_000;
- C. マネージャー.予算 = 1\_000\_000;
- D. director.stockOptions = 1\_000;
- E. ディレクター.給与 = 80\_000;
- F. 従業員.給与 = 50\_000;

**Answer: A,B** ([メッセージを残す](#))

最新問題: 66

次のコードがあるとします。

```

public static void main(String[] args) {
    String[] planets = {"Mercury", "Venus", "Earth", "Mars"};

    System.out.println(planets.length);
    System.out.println(planets[1].length());
}

```

出力は何ですか？

- A. 421
- B. 47
- C. 54
- D. 35
- E. 44
- F. 45

**Answer: (**[解答を表示する](#)**)**

最新問題: 67

あなたは銀行モジュールを開発しています。Maskcc メソッドを持つ ccMask という名前のクラスを開発しました。

コードの断片を考えると、次のようになります。

```
class CCMask {
    public static String maskCC(String creditCard) {
        String x = "XXXX-XXXX-XXXX-";
        //line n1
    }

    public static void main(String[] args) {
        System.out.println(maskCC("1234-5678-9101-1121"));
    }
}
```

クレジットカード番号の最後の 4 桁 (および 4 桁の各グループを区切るハイフン) を除くすべての桁を非表示にする文字列が、maskcc メソッドから返されることを確認する必要があります。この要件を達成するには、行 n1 でどの 2 つのコード フラグメントを独立して使用する必要がありますか? (2つお選びください。)

- A) `StringBuilder sb = new StringBuilder(creditCard);`  
`sb.substring(15, 19);`  
`return x + sb;`
- B) `return x + creditCard.substring(15, 19);`
- C) `StringBuilder sb = new StringBuilder(x);`  
`sb.append(creditCard, 15, 19);`  
`return sb.toString();`
- D) `StringBuilder sb = new StringBuilder(creditCard);`  
`StringBuilder s = sb.insert(0, x);`  
`return s.toString();`

- A. オプション C
- B. オプション A
- C. オプション D
- D. オプション B

Answer: ([解答を表示する](#))

最新問題: 68

コードの断片を考えると、次のようになります。

```
LocalDate Time dt= LocalDateTime.of (2014, 7, 31, 1, 1);
dt.plusDays (30);
dt. plusMonths (1);
System.out.print (dt.format (DateTimeFormatter. ISO_DATE) );
```

結果は何ですか？

- A. 実行時に例外がスローされます。
- B. 2014 年 7 月 31 日
- C. 2014-09-30
- D. 2014-07-31

Answer: A ([メッセージを残す](#))

最新問題: 69

与えられる:

```
class Vehicle {
    String type = "4W";
    int maxSpeed = 100;

    Vehicle(String type, int maxSpeed) {
        this.type = type;
        this.maxSpeed = maxSpeed;
    }
}

class Car extends Vehicle {
    String trans;

    Car(String trans) { //line n1
        this.trans = trans;
    }

    Car(String type, int maxSpeed, String trans) {
        super(type, maxSpeed);
        this(trans); //line n2
    }
}
```

コードの断片を考えると、次のようになります。

```
7. Car c1 = new Car("Auto");
8. Car c2 = new Car("4W", 150, "Manual");
9. System.out.println(c1.type + " " + c1.maxSpeed + " " + c1.trans);
10. System.out.println(c2.type + " " + c2.maxSpeed + " " + c2.trans);
```

結果は何ですか？

- A. コンパイルは n2 行目でのみ失敗します

- B. Null 0 自動 4W 150 手動
- C. コンパイルは n1 行目でのみ失敗します
- D. 行 n1 と行 n2 の両方でコンパイルが失敗します。
- E. 4W 100 自動 4W 150 手動

Answer: A ([メッセージを残す](#))

#### 最新問題: 70

コードの断片を考えると、次のようになります。

```
7.  StringBuilder sb1 = new StringBuilder("Duke");
8.  String str1 = sb1.toString();
9.  // insert code here
10. System.out.print(str1 == str2);
```

9 行目に挿入すると、コードが true と表示されるコード フラグメントはどれですか?

- A. 文字列 str2 = "公爵";
- B. 文字列 str2 = 新しい文字列 (str1);
- C. 文字列 str2 = str1;
- D. 文字列 str2 = sb1.toString();

Answer: ([解答を表示する](#))

#### 最新問題: 71

コードの断片を考えると、次のようになります。

```
public static void main(String[] args) {
    String[] arr = {"A", "B", "C", "D"};
    for (int i = 0; i < arr.length; i++)
        System.out.print(arr[i] + " ");
        if (arr[i].equals("C")) {
            continue;
        }
        System.out.println("Work done");
        break;
}
```

結果は何ですか?

- A. ABC 作業完了
- B. コンパイルに失敗しました
- C. 完了した作業
- D. ABCD 作業が完了しました

Answer: ([解答を表示する](#))

#### 最新問題: 72

与えられる:

```
public class SumTest {  
  
    public static void doSum(Integer x, Integer y) {  
        System.out.println("Integer sum is " + (x + y));  
    }  
  
    public static void doSum(double x, double y) {  
        System.out.println("double sum is " + (x + y));  
    }  
  
    public static void doSum(float x, float y) {  
        System.out.println("float sum is " + (x + y));  
    }  
  
    public static void doSum(int x, int y) {  
        System.out.println("int sum is " + (x + y));  
    }  
  
    public static void main(String[] args) {  
        doSum(10, 20);  
        doSum(10.0, 20.0);  
    }  
}
```

ORACLE®

結果は何ですか？

- ORACLE
- A) int sum is 30  
float sum is 30.0
  - B) int sum is 30  
double sum is 30
  - C) Integer sum is 30  
double sum is 30.0
  - D) Integer sum is 30  
float sum is 30.0

- A. オプション D
- B. オプション C
- C. オプション A
- D. オプション B

Answer: D ([メッセージを残す](#))

コードの断片を考えると、次のようになります。

```
String[] strs = new String[2];
int idx = 0;
for (String s : strs) {
    strs[idx].concat("element " + idx);
    idx++;
}
for (idx = 0; idx < strs.length; idx++) {
    System.out.println(strs[idx]);
}
```

結果は何ですか？

- A. Null Null
- B. 実行時に NullPointerException がスローされます。
- C. Null 要素 0 Null 要素 1
- D. 要素 0 要素 1

Answer: ([解答を表示する](#))

最新問題: 74

コード フラグメントが与えられた場合: どのコード フラグメントを n1 行目に挿入すると、コードでハックを出力できるようになりますか？

Person.java:



```
public class Person {
    String name;
    int age;

    public Person(String n, int a) {
        name = n;
        age = a;
    }

    public String getName() {
        return name;
    }

    public int getAge() {
        return age;
    }
}
```

Test.java:

```
public static void checkAge(List<Person> list, Predicate<Person> predicate) {
    for (Person p : list) {
        if (predicate.test(p)) {
            System.out.println(p.name + " ");
        }
    }
}

public static void main(String[] args) {
    List<Person> iList = Arrays.asList(new Person("Hank", 45),
                                       new Person("Charlie", 40),
                                       new Person("Smith", 38));

    //line n1
}
```

- A. checkAge (iList, p -> p.getAge() > 40);
- B. checkAge(iList, (人 p) -> { p.getAge() > 40; });
- C. checkAge (iList, ( ) -> p. get Age ( ) > 40);
- D. checkAge(iList, 人 p -> p.getAge() > 40);

**Answer:** [解答を表示する](#)

最新問題: 75

コードの断片を考えると、次のようになります。

```
1. public class Test {
2.     public static void main(String[] args) {
3.         /* insert code here */
4.         array[0]=10;
5.         array[1]=20;
6.         System.out.print (array[0]+": "+array[1]);
7.     }
8. }
```

3行目に挿入すると、コードが 10:20 を出力できるようにするコード フラグメントはどれですか？

- A. int 配列 [2];
- B. int[] 配列;配列 = int[2];
- C. int[] 配列 n= new int[2];
- D. int 配列 = new int[2];

**Answer: D** ([メッセージを残す](#))

最新問題: 76

与えられる:

```
public class App {
    public static void main(String[] args) {
        int i = 10;
        int j = 20;
        int k = j += i - 5;
        System.out.print(i + " : " + j + " : " + k);
    }
}
```

結果は何ですか？

- A. 10 : 30 : 6
- B. 10 : 22 : 22
- C. 10 : 22 : 20
- D. 10 : 22 : 6

**Answer: (**[解答を表示する](#)**)**

説明

説明

```
1- public class App {
2-     public static void main (String[] args) {
3-         int i = 10;
4-         int j = 20;
5-         int k = j + i / 5;
6-         System.out.print (i + " : " + j + " : " + k);
7-     }
8- }
9
```

External Libraries ... [Add External Library \(from Maven Repo\)](#)

CommandLine Arguments ...

Interactive mode :  OFF Version: JDK 9.0.1

Stdin Inputs...

[Execute](#) [Save](#) [My Projects](#) [Recent](#) [Collaborate](#) [More Options](#)

Result...  
CPU Time: 0.20 sec(s), Memory: 32080 kilobyte(s) compiled and executed in 1.229 sec(s)

10 : 22 : 22



有効な **1z0-808** 問題集は GoShiken.com が提供された合格しやすい 1z0-808 試験問題集！  
GoShiken.com が最新の **1z0-808** 試験問題集を提供しています。GoShiken.com 1z0-808 試験問題は最新で、解答が正確でございます。最新の GoShiken.com 1z0-808 問題集をゲットする人はこちら: <https://www.goshiken.com/Oracle/1z0-808-mondaishu.html> (**39830%OFF**問題集溶と正解付きで **30%w**特別割引コード: **Freepdfdumps**)

#### 最新問題: 77

コードの断片を考えると、次のようになります。

```
int[] array = {1, 2, 3, 4, 5};
```

そして、次のような要件が与えられます。

1. 配列のすべての要素を入力順に処理します。
2. 配列のすべての要素を入力 inverse の順序で処理します。
3. 配列の要素を入力順に交互に処理します。

正しい 2 つの記述はどれですか? (2つお選びください。)

- A. 要件 1、2、および 3 は、標準の for ループを使用して実装できます。
- B. 要件 1 は、拡張された for ループを使用して実装できます。
- C. 要件 3 は、拡張 for ループまたは標準 for ループのいずれかを使用して実装することはできません。
- D. 要件 1、2、および 3 は、拡張された for ループを使用して実装できます。

E. 要件 2 と 3 は、標準の for ループを使用して実装することはできません。

**Answer: B,C** ([メッセージを残す](#))

最新問題: 78

与えられる:

```
public class MyField {
    int x;
    int y;
    public void doStuff(int x, int y) {
        x = x;
        y = this.y;
    }
    public void display () {
        System.out.print(x + " " + y + " : ");
    }
    public static void main(String[] args) {
        MyField m1 = new MyField();
        m1.x = 100;
        m1.y = 200;
        MyField m2 = new MyField();
        m2.doStuff(m1.x, m1.y);
        m1.display();
        m2.display();
    }
}
```



結果は何ですか？

A. 100 200 : 100 0 :

B. 100 200 : 100 200 :

C. 0 0 : 100 0 :

D. 100 200 : 0 0 :

**Answer: A** ([メッセージを残す](#))

最新問題: 79

コードの断片を考えると、次のようになります。

```
public class Test{

    void readCard(int cardNo) throws Exception {
        System.out.println("Reading Card");
    }

    void checkCard(int cardNo) throws RuntimeException { // line n1
        System.out.println("Checking Card");
    }

    public static void main(String[] args) {
        Test ex = new Test();
        int cardNo = 12344;
        ex.checkCard(cardNo); //line n2
        ex.readCard(cardNo); //line n3
    }
}
```

ORACLE®

結果は何ですか？

A:

Reading Card  
Checking Card

B: コンパイルは n1 行目でのみ失敗します。

C: n2 行目でのみコンパイルが失敗します。

D: n3 行目でのみコンパイルが失敗します。

E: n2 行目と n3 行目でコンパイルに失敗します。

A. オプション E

B. オプション B

C. オプション D

D. オプション C

E. オプション A

**Answer: C** ([メッセージを残す](#))

最新問題: 80

与えられる:

```

1. public class Whizlabs {
2.
3.     public static void main(String[] args) {
4.         String s = "A";
5.
6.         switch (s) {
7.             case "a":
8.                 System.out.print("simple A ");
9.             default:
10.                System.out.print("default ");
11.             case "A":
12.                 System.out.print("Capital A ");
13.         }
14.     }
15. }

```

結果は何ですか？

- A. 単純な A
- B. 大文字 A
- C. 単純なデフォルトの大文字 A
- D. 単純なデフォルト
- E. コンパイルに失敗しました。

**Answer:** [\(解答を表示する\)](#)

ここでは2つの三項演算子を組み合わせて使用する必要があります。最初に確認するために使用できます

次のような  $x > 10$  の条件。

$x > 10 ? ">":$  (条件が false の場合) 次に、次のように別の関数を使用して  $x < 10$  かどうかを確認する必要があります。

$x < 10 ? "V:"=$  最後の三項ステートメントを false に入れることで、これら2つを組み合わせることができます。

最初の三項ステートメントの位置は次のようになります。

$x > 10 ? ">": x < 10 ? "<:"$

<https://docs.oracle.com/javase/tutorial/java/nutsandbolts/if.html>

最新問題: 81

コードの断片を考えると、次のようになります。

```
public class Test {  
  
    static int count = 0  
    int i = 0;  
  
    public void changeCount () {  
        while (i<5) {  
            i++;  
            count++;  
        }  
    }  
  
    public static void main (String [] args) {  
        Test check1 = new Test ();  
        Test check2 = new Test ();  
        check1.changeCount ();  
        check2.changeCount ();  
        System.out. print (check1.count + " : " + check2.count);  
    }  
}
```

結果は何ですか？

- A. コンパイルは失敗します。
- B. 10 : 10
- C. 5 : 10
- D. 5 : 5

**Answer: B** ([メッセージを残す](#))

最新問題: 82

コードの断片を考えると、次のようになります。

```
4. public static void main(String[] args) {  
5.     boolean opt = true;  
6.     switch (opt) {  
7.         case true:  
8.             System.out.print("True");  
9.             break;  
10.        default:  
11.            System.out.print("****");  
12.        }  
13.        System.out.println("Done");  
14. }
```

コード フラグメントで TrueDone を出力できるようにする変更はどれですか？

- A. 5 行目をブール値 opt = 1 に置き換えます。7 行目をケース 1 に置き換えます。
- B. デフォルトのセクションを削除します。
- C. 9 行目で、break ステートメントを削除します。
- D. 行 5 を文字列 opt = "true"; に置き換えます。7 行目を case "true" に置き換えます。

Answer: D ([メッセージを残す](#))

最新問題: 83

コードの断片を考えると、次のようになります。

```
13. List colors = new ArrayList();
14. colors.add("green");
15. colors.add("red");
16. colors.add("blue");
17. colors.add("yellow");
18. colors.remove(2);
19. colors.add(3, "cyan");
20. System.out.print(colors);
```

結果は何ですか？

- A. 実行時に IndexOutOfBoundsException がスローされます。
- B. (緑、赤、黄、シアン)
- C. (緑、赤、シアン、黄)
- D. (緑、青、黄、シアン)

Answer: C ([メッセージを残す](#))

最新問題: 84

与えられる:

```
class Alpha {
    int ns;
    static int s;
    Alpha(int ns) {
        if (s < ns) {
            s = ns;
            this.ns = ns;
        }
    }
    void doPrint() {
        System.out.println("ns = " + ns + " s = " + s);
    }
}
```

And,

```
public class TestA {
    public static void main(String[] args) {
        Alpha ref1 = new Alpha(50);
        Alpha ref2 = new Alpha(125);
        Alpha ref3 = new Alpha(100);
        ref1.doPrint();
        ref2.doPrint();
        ref3.doPrint();
    }
}
```

結果は何ですか？

A) ns = 50 s = 125  
ns = 125 s = 125  
ns = 100 s = 125

B) ns = 50 s = 125  
ns = 125 s = 125  
ns = 0 s = 125

C) ns = 50 s = 50  
ns = 125 s = 125  
ns = 100 s = 100

D) ns = 50 s = 50  
ns = 125 s = 125  
ns = 0 s = 125

A. オプション A

B. オプション B

C. オプション D

D. オプション C

Answer: B ([メッセージを残す](#))

最新問題: 85

コードの断片を考えると、次のようになります。

```

public static void main(String[] args) {
    String[][] arr = {"A", "B", "C"}, {"D", "E"};
    for (int i = 0; i < arr.length; i++) {
        for (int j = 0; j < arr[i].length; j++) {
            System.out.print(arr[i][j] + " ");
            if (arr[i][j].equals("B")) {
                continue;
            }
        }
        continue;
    }
}

```

結果は何ですか？

- A. コンパイルは失敗します。
- B. ABDE
- C. ABCDE
- D. ABC

**Answer:** ([解答を表示する](#))

最新問題: 86

与えられる:

```

1. public class Whizlabs {
2.     public static void main(String[] args) {
3.         int sum = 0;
4.
5.         for(int x = 0; x <= 10; x++)
6.             sum += x;
7.         System.out.print("Sum for 0 to " + x)
8.         System.out.println(" " + sum);
9.     }
10. }

```

どれが本当ですか？

- A. 0 ~ 0 の合計 = 55
- B. 0 ~ 10 の合計 = 55
- C. 6 行目のエラーによりコンパイルが失敗します。
- D. 7 行目のエラーによりコンパイルが失敗します。
- E. 実行時に例外がスローされます。

**Answer: D** ([メッセージを残す](#))

ループ変数のスコープは、それを囲んでいるループに制限されます。したがって、この場合、ループのスコープは

変数 x は 5 行目で宣言されており、for ループに限定されます。7 行目の変数にアクセスしようとすると、

これは変数 x の範囲外であるため、コンパイル時エラーが発生します。したがってコンパイルは失敗します

7行目のエラーが原因です。したがって、オプションDが正しいです。  
コードがコンパイルに失敗するため、オプションAとBは正しくありません。  
参考: <https://docs.oracle.com/javase/tutorial/java/nutsandbolts/variables.html>

#### 最新問題: 87

このグリッドは2D配列の状態を示しています。

0	0	
	X	0
X		X

グリッドは次のコードで作成されます。

```
char[][] grid = new char[3][3];
grid[1][1] = 'X';
grid[0][0] = '0';
grid[2][0] = 'X';
grid[0][1] = '0';
grid[2][2] = 'X';
grid[1][2] = '0';
//line n1
```

//line n1 の代わりに挿入されるコード行は、グリッドに X を追加して、グリッドに 3 つの連続する X が含まれるようにしますか?

- A. グリッド[3][2] = 'X';
- B. グリッド[2][1] = 'X';
- C. グリッド[2][3] = 'X';
- D. グリッド[3][1] = 'X';

Answer: ([解答を表示する](#))

#### 最新問題: 88

コードの断片を考えると、次のようになります。

```
public static void main(String[] args) {
    StringBuilder sb = new StringBuilder("Java");
    String s = "Java";

    if (sb.toString().equals(s.toString())) {
        System.out.println("Match 1");
    } else if (sb.equals(s)) {
        System.out.println("Match 2");
    } else {
        System.out.println("No Match");
    }
}
```

結果は何ですか?

- A. 実行時に NullPointerException がスローされます。
- B. 一致しません
- C. マッチ 2
- D. 一致 1

Answer: D ([メッセージを残す](#))

最新問題: 89

ショッピングアプリケーションのプログラムを開発するように求められ、次の情報が与えられます。

アプリケーションには、Toy、EduToy、および ConsToy クラスが含まれている必要があります。Toy クラスはスーパークラスです

他の 2 つのクラスのうち。

int cacuiatePrice (Toy t) メソッドは、おもちゃの価格を計算します。

void printToy (Toy t) メソッドは、おもちゃの詳細を出力します。

Toy クラスのどの定義が、有効な抽象化層をクラス階層に追加しますか？

- A) 

```
public abstract class Toy {
    public abstract int calculatePrice(Toy t);
    public void printToy(Toy t) { /* code goes here */ }
}
```
- B) 

```
public abstract class Toy {
    public int calculatePrice(Toy t);
    public void printToy(Toy t);
}
```
- C) 

```
public abstract class Toy {
    public int calculatePrice(Toy t);
    public final void printToy(Toy t){ /* code goes here */ }
}
```
- D) 

```
public abstract class Toy {
    public abstract int calculatePrice(Toy t) { /* code goes here */ }
    public abstract void printToy(Toy t) { /* code goes here */ }
}
```

- A. オプション B
- B. オプション D
- C. オプション C
- D. オプション A

Answer: D ([メッセージを残す](#))

最新問題: 90

次のクラスがあるとします。

```
public class CheckingAccount {  
    public int amount:  
    // line n1  
}
```

And given the following main method, located in another class:

```
public static void main (String [] args) {  
    CheckingAccount acct = new CheckingAccount ();  
    //line n2  
}
```

独立して挿入された場合、amount の値が 100 に設定される 3 つのコードはどれですか？

At line n1 insert:

```
public CheckingAccount() {  
    amount = 100;  
}
```

At line n2 insert:

```
this.amount = 100;
```

At line n2 insert:

```
amount = 100;
```

ORACLE®

At line n1 insert:

```
public CheckingAccount() {  
    this.amount = 100;  
}
```

At line n2 insert:

```
acct.amount = 100;
```

At line n1 insert:

```
public CheckingAccount() {  
    acct.amount = 100;  
}
```

- A. オプション E
- B. オプション A
- C. オプション C
- D. オプション D
- E. オプション B
- F. オプション F

Answer: A,C,D ([メッセージを残す](#))

最新問題: 91

与えられる:

```
public class SumTest {  
  
    public static void doSum(Integer x, Integer y) {  
        System.out.println("Integer sum is " + (x + y));  
    }  
  
    public static void doSum(double x, double y) {  
        System.out.println("double sum is " + (x + y));  
    }  
  
    public static void doSum(float x, float y) {  
        System.out.println("float sum is " + (x + y));  
    }  
  
    public static void doSum(int x, int y) {  
        System.out.println("int sum is " + (x + y));  
    }  
  
    public static void main(String[] args) {  
        doSum(10, 20);  
        doSum(10.0, 20.0);  
    }  
}
```

ORACLE

結果は何ですか？

A  
int sum is 30  
float sum is 30.0

B  
int sum is 30  
double sum is 30.0

C  
integer sum is 30  
double sum is 30.0

D  
integer sum is 30  
float sum is 30.0

ORACLE

A. オプション D

- B. オプション C
- C. オプション A
- D. オプション B

Answer: ([解答を表示する](#))

有効な 1z0-808 問題集は GoShiken.com が提供された合格しやすい 1z0-808 試験問題集！  
GoShiken.com が最新の 1z0-808 試験問題集を提供しています。GoShiken.com 1z0-808 試験問題は最新で、解答が正確でございます。最新の GoShiken.com 1z0-808 問題集をゲットする人はこちら: <https://www.goshiken.com/Oracle/1z0-808-mondaishu.html> (39830%OFF問題集溶と正解付きで 30%w 特別割引コード: **Freepdfdumps**)

最新問題: 92

与えられる:

```
class X {
    static int i;
    int j;
    public static void main(String[] args) {
        X x1 = new X();
        X x2 = new X();
        x1.i = 3;
        x1.j = 4;
        x2.i = 5;
        x2.j = 6;
        System.out.println(
            x1.i + " " +
            x1.j + " " +
            x2.i + " " +
            x2.j);
    }
}
```

結果は何ですか？

- A. 3 4 5 6
- B. 3 6 4 6
- C. 3 4 3 6
- D. 5 4 5 6

Answer: D ([メッセージを残す](#))

最新問題: 93

コードの断片を考えると、次のようになります。

```
public static void main (String [ ] args) {
    int [] stack = {10,20,30}
    int size = 3;
    int idx = 0;
    /*line n1 */
    System.out.print ("The Top element: " + stack [idx] );
}
```

n1 行目に挿入されたコードの断片のうち、The Top 要素: 30 を示すものはどれですか？

```
A. do {
    idx++;
} while (idx >=size);
B. while (idx < size) {
    idx++;
}
C. do {
    idx++;
} while (idx <size -1);
D. do {
    idx++;
} while (idx<= size);
E. while (idx <= size -1) {
    idx++
}
```

- A. オプション E
- B. オプション B
- C. オプション D
- D. オプション C
- E. オプション A

Answer: E ([メッセージを残す](#))

最新問題: 94

コードの断片を考えると、結果はどうなるでしょうか？

```
public static void main(String[] args) {
    String[][] arr = {"A", "B", "C"}, {"D", "E"};
    for (int i = 0; i < arr.length; i++) {
        for (int j = 0; j < arr[i].length; j++) {
            System.out.print(arr[i][j] + " ");
            if (arr[i][j].equals("B")) {
                break;
            }
        }
        continue;
    }
}
```

- A. ABC
- B. ABCDE
- C. ABDE
- D. コンパイルは失敗します。

**Answer: C** ([メッセージを残す](#))

最新問題: 95

与えられる:

```
public class MyClass {
    public static void main(String[] args) {
        String s = "Java SE 8 1";
        int len = s.trim().length();
        System.out.print(len);
    }
}
```

結果は何ですか?

- A. 9
- B. 10
- C. 8
- D. 11
- E. コンパイルは失敗します。

**Answer: (**[解答を表示する](#)**)**

最新問題: 96

コードの断片を考えると、次のようになります。

```

public static void main (String[ ] args) {
    int data [] = {2010, 2013, 2014, 2015, 2014};
    int key = 2014;
    int count = 0;
    for (int e: data) {
        if (e! = key) {
            continue;
            count++;
        }
    }
    System.out.print (count + "Found");
}

```

ORACLE®

結果は何ですか？

- A. コンパイルは失敗します。
- B. 1 件見つかりました
- C. 3 件見つかりました
- D. 0 件見つかりました

Answer: A ([メッセージを残す](#))

最新問題: 97

コードの断片を考えると、次のようになります。

```

public static void main (String [] args) {
    ArrayList<Integer> points = new ArrayList<> ();
    points.add (1);
    points.add (2);
    points.add (3);
    points.add (4);
    points.add (null);
    points.remove (2);
    points.remove (null);
    System.out.println(points);
}

```

結果は何ですか？

- A. 実行時に NullPointerException がスローされます。
- B. [1, 2, 4]
- C. [1, 2, 4, null]
- D. [1, 3, 4, null]

E. [1, 3, 4]

F. コンパイルは失敗します。

**Answer: F** ([メッセージを残す](#))

説明/参照:

Explanation:

The screenshot shows an IDE interface with the following components:

- Your Code ...**: A code editor containing the following Java code:

```
1- public static void main (String [] args) {
2   ArrayList<Integer> points = new ArrayList<> ();
3   points.add (1) ;
4   points.add (2) ;
5   points.add (3) ;
6   points.add (4) ;
7   points.add (null) ;
8   points.remove (null) ;
9   System.out.println (points) ;
10 }
```
- External Libraries ...**: A section with a button "Add External Library (from Maven Repo)" and the text "cs1.keyboard".
- Input Arguments (args of Main Method)...**: An empty text input field.
- Interactive mode :** A toggle switch set to "OFF".
- Stdin Inputs...**: An empty text input field.
- Buttons:** A row of buttons: "Execute", "Save", "My Projects", "Recent", "Collaborate", "Others", and "Goto Another Language/DB".
- Result...**: A section with the text "compiled and executed in 0 second(s)".
- Error Message:** A black box containing the text "No 'public class' found to execute".

最新問題: 98

コードの断片を考えると、次のようになります。

```

public static void main(String[] args) {
    try {
        int num = 10;
        int div = 0;
        int ans = num / div;
    } catch (ArithmeticException ae) {
        ans = 0 // line n1
    } catch (Exception e) {
        System.out.println("Invalid calculation");
    }
    System.out.println("Answer = " + ans); // line n2
}

```

ORACLE®

結果は何ですか？

- A. 無効な計算
- B. コンパイルは行 n1 でのみ失敗します。
- C. コンパイルは行 n1 と行 2 でのみ失敗します。
- D. コンパイルは行 n2 でのみ失敗します。
- E. 答え = 0

Answer: ([解答を表示する](#))

最新問題: 99

与えられる:

```

String stuff = "TV";
String res = null;

if (stuff.equals ("TV")) {
    res = "Walter";
} else if (stuff.equals ("Movie") ) {
    res= "White";
} else {
    res= "No Result";
}

```

if ブロックを置き換えることができるコードの断片はどれですか？

- A. `res = stuff.equals ("TV") ? "Walter" else stuff.equals ("Movie") ? "White" : "No Result";`
- B. `stuff.equals ("TV") ? res= "Walter" : stuff.equals ("Movie") ? res = "White" : res = "No Result";`


- ```
res = stuff.equals ("TV") ? stuff.equals ("Movie")? "Walter" :  
"White" : "No Result";
```
- C.
- ```
res = stuff.equals ("TV")? "Walter" : stuff.equals ("Movie")?  
"White" : "No Result";
```
- D.

**Answer: C** ([メッセージを残す](#))

最新問題: 100

与えられる:

```
public class Test {  
  
    public static void main(String[] args) {  
        if (args[0].equals("Hello") ? true : false) {  
            System.out.println("Success");  
        } else {  
            System.out.println("Failure");  
        }  
    }  
}
```



そして次のコマンドを与えます:

```
javac Test.java  
Java Test Hello
```

結果は何ですか?


- A. 失敗
- B. コンパイルは失敗します。
- C. 実行時に例外がスローされます
- D. 成功

**Answer: D** ([メッセージを残す](#))

最新問題: 101

与えられる:

```
int i;  
static int j;  
public static void main(String[] args)  
    X x1 = new X();  
    X x2 = new X();  
    x1.i = 3;  
    x1.j = 4;  
    x2.i = 5;  
    x2.j = 6;  
    System.out.println(  
        x1.i + " " +  
        x1.j + " " +  
        x2.i + " " +  
        x2.j);  
}
```



結果は何ですか？

- A. 3 4 5 6
- B. 3 4 3 6
- C. 5 4 5 6
- D. 3 6 5 6

Answer: [\(解答を表示する\)](#)

```
3 6 5 6
Completed with exit code: 0
```

最新問題: 102

コードの断片を考えると、次のようになります。

```
Interface Exportable {
    Void export();
}

class Tool implements Exportable {
    protected void export () {           //line n1
        System.out.println("Tool::export");
    }
}

class ReportTool extends Tool implements Exportable {

    public void export() {                //line n2
        System.out.println("RTool::export");
    }

    public static void main(String[] args) {
        Tool aTool = new ReportTool();
        Tool bTool = new Tool();
        callExport(aTool);
        callExport(bTool);
    }

    public static void callExport (Exportable ex) {
        ex.export();
    }
}
```

結果は何ですか？

- A. コンパイルは行 n1 でのみ失敗します。
- B. 行 n1 と行 n2 の両方でコンパイルが失敗します。
- C. コンパイルは行 n2 でのみ失敗します。
- D. RTool::exportTool::export
- E. ツール::エクスポートツール:エクスポート

Answer: B ([メッセージを残す](#))

最新問題: 103

コードの断片を考えると、次のようになります。

```
public class Employee {
    String name;
    boolean contract;
    double salary;
    Employee() {
        // line n1
    }
    public String toString() {
        return name + ":" + contract + ":" + salary;
    }
    public static void main(String[] args) {
        Employee e = new Employee();
        // line n2
        System.out.print(e);
    }
}
```

独立して行った場合、コードで `joe:true: 100.0` を出力できるようにする 2 つの変更はどれですか？ (2つお選びください。)

- A) Replace line n2 with:  
e.name = "Joe";  
e.contract = true;  
e.salary = 100;
- B) Replace line n2 with:  
this.name = "Joe";  
this.contract = true;  
this.salary = 100;
- C) Replace line n1 with:  
this.name = new String("Joe");  
this.contract = new Boolean(true);  
this.salary = new Double(100);
- D) Replace line n1 with:  
name = "Joe";  
contract = TRUE;  
salary = 100.0f;
- E) Replace line n1 with:  
this("Joe", true, 100);

- A. オプション C
- B. オプション D

- C. オプション E
- D. オプション A
- E. オプション B

Answer: A,D ([メッセージを残す](#))

最新問題: 104

コードの断片を考えると、次のようになります。

```
public static void main(String[] args) {  
    LocalDate date = LocalDate.of(2012, 1, 30);  
    date.plusDays(10);  
    System.out.println(date);  
}
```

結果は何ですか？

- A. 2012-02-10
- B. 2012-02-10 00:00
- C. 2012-01-30
- D. 実行時に DateTimeException がスローされます。

Answer: B ([メッセージを残す](#))

最新問題: 105

MyString クラスと Test クラスの定義を考えると、次のようになります。

MyString.java:

```
package p1;  
class MyString {  
    String msg;  
    MyString(String msg) {  
        this.msg = msg;  
    }  
}
```

Test.java:

```
package p1;  
public class Test {  
    public static void main(String[] args) {  
        System.out.println("Hello " + new StringBuilder("Java SE 8"));  
        System.out.println("Hello " + new MyString("Java SE 8"));  
    }  
}
```

結果は何ですか？

- A) Hello Java SE 8  
Hello Java SE 8
- B) Hello java.lang.StringBuilder@<<hashcode1>>  
Hello p1.MyString@<<hashcode2>>
- C) Hello Java SE 8  
Hello p1.MyString@<<hashcode>>
- D) Compilation fails at the Test class.

- A. オプション C  
B. オプション A  
C. オプション D  
D. オプション B

Answer: A ([メッセージを残す](#))

最新問題: 106

与えられる:

```
public class App {  
    public static void main(String[] args) {  
        int i = 10;  
        int j = 20;  
        int k = j += i / 5;  
        System.out.print(i + " : " + j + " : " + k);  
    }  
}
```

結果は何ですか？

- A. 10 : 22 : 6  
B. 10 : 22 : 20  
C. 10 : 30 : 6  
D. 10 : 22 : 22

Answer: D ([メッセージを残す](#))

有効な **1z0-808** 問題集は GoShiken.com が提供された合格しやすい 1z0-808 試験問題集！  
GoShiken.com が最新の **1z0-808** 試験問題集を提供しています。GoShiken.com 1z0-808 試験問題は最新で、解答が正確でございます。最新の GoShiken.com 1z0-808 問題集をゲットする

人はこちら: <https://www.goshiken.com/Oracle/1z0-808-mondaishu.html> (39830%OFF問題集  
と正解付きで 30%w 特別割引コード: **Freepdfdumps**)

#### 最新問題: 107

あなたは銀行モジュールを開発しています。Maskcc メソッドを持つ ccMask という名前のクラスを開発しました。

コードの断片を考えると、次のようになります。

```
class CCMask {
    public static String maskCC(String creditCard) {
        String x = "XXXX-XXXX-XXXX ";
        //line n1
    }

    public static void main(String[] args) {
        System.out.println(maskCC("1234-5678-9101-1121"));
    }
}
```

クレジットカード番号の最後の 4 桁 (および 4 桁の各グループを区切るハイフン) を除くすべての桁を非表示にする文字列が、maskcc メソッドから返されることを確認する必要があります。この要件を達成するには、行 n1 でどの 2 つのコード フラグメントを独立して使用する必要がありますか？

- A) `StringBuilder sb = new StringBuilder(creditCard);`  
`sb.substring(15, 19);`  
`return x + sb;`
- B) `return x + creditCard.substring(15, 19);`
- C) `StringBuilder sb = new StringBuilder(x);`  
`sb.append(creditCard, 15, 19);`  
`return sb.toString();`
- D) `StringBuilder sb = new StringBuilder(creditCard);`  
`StringBuilder s = sb.insert(0, x);`  
`return s.toString();`

- A. オプション A
- B. オプション C
- C. オプション B
- D. オプション D

Answer: B,C ([メッセージを残す](#))

#### 最新問題: 108

与えられた結果: 結果は何ですか？

```
class X {
    static void m(int i) {
        i += 7;
    }
    public static void main(String[] args) {
        int j = 12;
        m(j);
        System.out.println(j);
    }
}
```

- A. 実行時に例外がスローされます
- B. コンパイルに失敗しました
- C. 7
- D. 19
- E. 12

Answer: E ([メッセージを残す](#))

最新問題: 109

与えられる:

```
public static void main(String[] args) {
    String ta = "A ";
    ta = ta.concat("B ");
    String tb = "C ";
    ta = ta.concat(tb);
    ta.replace('C', 'D');
    ta = ta.concat(tb);
    System.out.println(ta);
}
```

結果は何ですか?

- A. ACD
- B. ABD
- C. ABCC
- D. ABDC
- E. ABCD

Answer: B ([メッセージを残す](#))

最新問題: 110

与えられる:

```
public class Test {
    public static final int MIN =1;
    public static void main (String [] args) {
        int x = args.length;
        if (checkLimit (x)) { //line n1
            System.out.println ("Java SE");
        } else {
            System.out.println ("Java EE");
        }
    }
    public static boolean checkLimit (int x) {
        return (x > = MIN) ? true : false;
    }
}
```

ORACLE®

そして次のコマンドを与えます:

```
javac Test.java
java Test
```

結果は何ですか?

- A. Java SE
- B. Java EE
- C. n1 行目でコンパイルが失敗します。
- D. 実行時に NullPointerException がスローされます。

**Answer: B** ([メッセージを残す](#))

説明/参照:

最新問題: 111

与えられる:

```

class A {
    public A() {
        System.out.print("A ");
    }
}

class B extends A {
    public B() { //line n1
        System.out.print("B ");
    }
}

class C extends B {
    public C() { //line n2
        System.out.print("C ");
    }
    public static void main(String[] args) {
        C c = new C();
    }
}

```

結果は何ですか？

- A. n1 行目と n2 行目でコンパイルが失敗します。
- B. CBA
- C. C
- D. ABC

Answer: D ([メッセージを残す](#))

最新問題: 112

コードの断片を考えると、次のようになります。

```

int num[][] = new int[1][3];
for (int i = 0; i < num.length; i++) {
    for (int j = 0; j < num[i].length; j++) {
        num[i][j] = 10;
    }
}

```

外側のループが正常に完了した後の num 配列の状態を表すオプションはどれですか？

A ORACLE

```
num[0][0]=10  
num[0][1]=10  
num[0][2]=10
```

B

```
num[0][0]=10  
num[1][0]=10  
num[2][0]=10
```

C

```
num[0][0]=10  
num[0][1]=0  
num[0][2]=0
```

D

```
num[0][0]=10  
num[0][1]=10  
num[0][2]=10  
num[0][3]=10  
num[1][0]=0  
num[1][1]=0  
num[1][2]=0  
num[1][3]=0
```

A. オプション C

B. オプション D

C. オプション B

D. オプション A

Answer: ([解答を表示する](#))

最新問題: 113

コードの断片を考えると、次のようになります。

```
String shirts[][] = new String[2][2];  
shirts[0][0] = "red";  
shirts[0][1] = "blue";  
shirts[1][0] = "small";  
shirts[1][1] = "medium";
```

どのコード部分が赤、青、小、中と表示されますか？

- A) 

```
for (int index = 1; index < 2; index++) {
    for (int idx = 1; idx < 2; idx++) {
        System.out.print (shirts[index][idx] + ":");
    }
}
```
- B) 

```
for (int index = 0; index < 2; ++index) {
    for (int idx = 0; idx < index; ++idx) {
        System.out.print (shirts[index][idx] + ":");
    }
}
```
- C) 

```
for (String c : colors) {
    for (String s : sizes) {
        System.out.println(s + ":");
    }
}
```
- D) 

```
for (int index = 0; index < 2;) {
    for (int idx = 0; idx < 2;) {
        System.out.print (shirts[index][idx] + ":");
        idx++;
    }
    index++;
}
```

ORACLE®

A. オプション B

B. オプション A

C. オプション D

D. オプション C

Answer: C ([メッセージを残す](#))

最新問題: 114

与えられる:

```
package p1;
public class Acc {
    int p;
    private int q;
    protected int r;
    public int s;
}

Test.java:

package p2;
import p1.Acc;
public class Test extends Acc {
    public static void main(String[] args) {
        Acc obj = new Test();
    }
}
```

どの発言が真実ですか？

- A. p、r、および s は obj からアクセスできます。
- B. obj からアクセスできるのは s のみです。
- C. p と s の両方に obj からアクセスできます。
- D. r と s の両方に obj からアクセスできます。

**Answer: B** ([メッセージを残す](#))

最新問題: 115

コードの断片を考えると、次のようになります。

```
abstract class Planet {  
    protected void revolve() { //line n1  
    }  
  
    abstract void rotate(); //line n2  
}  
  
class Earth extends Planet {  
    void revolve() { //line n3  
    }  
  
    protected void rotate() { //line n4  
    }  
}
```

独立して行われた 2 つの変更により、コードのコンパイルが可能になるのはどれですか？

- A. n1 行目のメソッドをパブリックにします。
- B. n3 行目のメソッドを public にします。
- C. n4 行目のメソッドを public にします。
- D. n3 行目のメソッドを保護します。
- E. n2 行目のメソッドを public にします。

**Answer: B,D** ([メッセージを残す](#))

最新問題: 116

コードの断片を考えると、次のようになります。

結果は何ですか？

- A. 実行時に例外がスローされます。
- B. 0
- C. -1
- D. 5

**Answer: D** ([メッセージを残す](#))

最新問題: 117

次のコードがあるとします。

```
int[] intArr = {15, 30, 45, 60, 75};  
intArr[2] = intArr[4];  
intArr[4] = 90;
```

このコードが実行された後の intArr の各要素の値は何ですか？

- A. 15、30、75、60、90
- B. 15、60、45、90、75
- C. 15、4、45、60、90
- D. 15、90、45、90、75
- E. 15、30、90、60、90

**Answer: A** ([メッセージを残す](#))

最新問題: 118

このクラスを考えると:

```
public class Rectangle {  
    private double length;  
    private double height;  
    private double area;  
  
    public void setLength(double length) {  
        this.length = length;  
    }  
    public void setHeight(double height) {  
        this.height = height;  
    }  
    public void setArea() {  
        area = length*height;  
    }  
}
```

このクラスをカプセル化し、Rectangle クラスが使用されるときは常に area フィールドが常に  $\text{length} * \text{height}$  に等しくなるようにする 2 つの変更はどれですか？

- A. setHeight メソッドの最後で setArea メソッドを呼び出します。
- B. setHeight メソッドの先頭で setArea メソッドを呼び出します。
- C. setLength メソッドの先頭で setArea メソッドを呼び出します。
- D. setArea メソッドをプライベートに変更します。
- E. エリアフィールドをパブリックに変更します。
- F. setLength メソッドの最後で setArea メソッドを呼び出します。

**Answer: A,D** ([メッセージを残す](#))

最新問題: 119

与えられる:

ACC.java:

```
package p1;
public class Acc {
    int p;
    private int q;
    protected int r;
    public int s;
}
```

Test.java:

```
package p2;
import p1.Acc;
public class Test extends Acc {
    public static void main(String[] args) {
        Acc obj = new Test();
    }
}
```

どの発言が真実ですか？

- A. p と s の両方に obj からアクセスできます。
- B. p、r、s は obj からアクセスできます。
- C. obj からアクセスできるのは s のみです。
- D. r と s の両方に obj からアクセスできます。

**Answer:** ([解答を表示する](#))

最新問題: 120

コードの断片を考えると、次のようになります。

```
int[] 配列 = {1, 2, 3, 4, 5};
```

そして、次のような要件が与えられます。

1. 配列のすべての要素を入力順に処理します。
2. 配列のすべての要素を入力 inverse の順序で処理します。
3. 配列の要素を入力順に交互に処理します。

正しい 2 つの記述はどれですか？

- A. 要件 1 は、拡張された for ループを使用して実装できます。
- B. 要件 2 と 3 は、標準の for ループを使用して実装することはできません。
- C. 要件 3 は、拡張 for ループまたは標準 for ループのいずれかを使用して実装することはできません。
- D. 要件 1、2、および 3 は、拡張された for ループを使用して実装できます。
- E. 要件 1、2、および 3 は、標準の for ループを使用して実装できます。

**Answer:** A,C ([メッセージを残す](#))

最新問題: 121

コードの断片を考えると、次のようになります。

```
public static void main (String [] args) {
    String myStr = "Hello World";
    myStr.trim ()
    int i1 = myStr.indexOf (" ");
    System.out.println (i1);
}
```

結果は何ですか？

- A. 5
- B. -1
- C. 0
- D. 実行時に例外がスローされます。

Answer: D ([メッセージを残す](#))

有効な 1z0-808 問題集は GoShiken.com が提供された合格しやすい 1z0-808 試験問題集！  
GoShiken.com が最新の 1z0-808 試験問題集を提供しています。GoShiken.com 1z0-808 試験問題は最新で、解答が正確でございます。最新の GoShiken.com 1z0-808 問題集をゲットする人はこちら: <https://www.goshiken.com/Oracle/1z0-808-mondaishu.html> (39830%OFF問題集溶と正解付きで 30%w 特別割引コード: **Freepdfdumps**)

最新問題: 122

コードの断片を考えると、次のようになります。

```
int[] array = {1, 2, 3, 4, 5};
```

そして、次のような要件が与えられます。

1. 配列のすべての要素を入力順に処理します。
2. 配列のすべての要素を入力 inverse の順序で処理します。
3. 配列の要素を入力順に交互に処理します。

正しい 2 つの記述はどれですか? (2つお選びください。)

- A. 要件 3 は、拡張 for ループまたは標準 for ループのいずれかを使用して実装することはできません。
- B. 要件 1、2、および 3 は、拡張された for ループを使用して実装できます。
- C. 要件 1 は、拡張された for ループを使用して実装できます。
- D. 要件 2 と 3 は、標準の for ループを使用して実装することはできません。
- E. 要件 1、2、および 3 は、標準の for ループを使用して実装できます。

Answer: A,C ([メッセージを残す](#))

最新問題: 123

与えられる:

```
class C2 {
    public void displayC2 () {
        System.out.print("C2");
    }
}
interface I {
    public void displayI();
}
class C1 extends C2 implements I {
    public void displayI() {
        System.out.print("C1");
    }
}
```

コードの断片を考えると、次のようになります。

```
C2 obj1 = new C1();
I obj2 = new C1();

C2 s = obj2;
I t = obj1;

t.displayI();
s.displayC2();
```

結果は何ですか？

- A. C1C1
- B. C2C2
- C. C1C2
- D. コンパイルに失敗しました

**Answer:** ([解答を表示する](#))

最新問題: 124

展示品をご覧ください。

```

class MissingInfoException extends Exception ( )
class AgeOutOfRangeException extends Exception ( )
class Candidate {
    String name;
    int age;
    Candidate(String name, int age) throws Exception
        if (name == null) {
            throw new MissingInfoException();
        } else if (age <= 10 || age >= 150) {
            throw new AgeOutOfRangeException();
        } else {
            this.name = name;
            this.age = age;
        }
    }
    public String toString() {
        return name + " age: " + age;
    }
}

```

コードの断片を考えると、次のようになります。

```

4. public class Test {
5.     public static void main(String[] args) {
6.         Candidate c = new Candidate("James", 20);
7.         Candidate c1 = new Candidate("Williams", 32);
8.         System.out.println(c);
9.         System.out.println(c1);
10.    }
11. }

```

コードで次の出力が可能になるのはどの変更ですか？

ジェームス 年齢: 20

ウィリアムズ 年齢: 32

- A. 行 6 と行 7 を try ブロック内で囲み、次のように追加します。catch(Exception e1) { //コードはここにあります} catch (missingInfoException e2) { //コードはここにあります} catch (AgeOutOfRangeException e3) { //コードはここにあります}
- B. 5 行目を public static void main (String [] args) throws.Exception { に置き換えます。
- C. 行 5 を public static void main (String [] args) に置き換える  
と、MissingInfoException、AgeOutOfRangeException がスローされます。
- D. 行 6 と行 7 を try ブロック内で囲み、次のように追加します。catch (missingInfoException e2) { //コードはここにあります} catch (AgeOutOfRangeException e3) { //コードはここにあります}

**Answer:** ([解答を表示する](#))

最新問題: 125

コードの一部を考えると、整数 aVar が 9 の場合、結果はどうなるでしょうか？

```
if (aVar++ < 10) {
    System.out.println(aVar + " Hello World!");
} else {
    System.out.println(aVar + " Hello Universe!");
}
```

- A. 9 Hello World!
- B. 10 Hello World!
- C. コンパイルは失敗します。
- D. 10 こんにちは、ユニバース!

Answer: B ([メッセージを残す](#))

#### 最新問題: 126

コードの断片を考えると、次のようになります。

```
LocalDate date1 = LocalDate.now();
LocalDate date2 = LocalDate.of(2014, 6, 20);
LocalDate date3 = LocalDate.parse("2014-06-20", DateTimeFormatter.ISO_DATE);
System.out.println("date1 = " + date1);
System.out.println("date2 = " + date2);
System.out.println("date3 = " + date3);
```

システムの日付が 2014 年 6 月 20 日であると仮定します。結果はどうなるでしょうか？

- A) date1 = 2014-06-20  
date2 = 2014-06-20  
date3 = 2014-06-20
- B) date1 = 06/20/2014  
date2 = 2014-06-20  
date3 = Jun 20, 2014
- C) Compilation fails.
- D) A DateParseExcpetion is thrown at runtime.

- A. オプション A
- B. オプション D
- C. オプション C
- D. オプション B

Answer: B ([メッセージを残す](#))

#### 最新問題: 127

コードの断片を考えると、次のようになります。

```

public class Test{

    void readCard(int cardNo) throws Exception {
        System.out.println("Reading Card");
    }

    void checkCard(int cardNo) throws RuntimeException { // line n1
        System.out.println("Checking Card");
    }

    public static void main(String[] args) {
        Test ex = new Test();
        int cardNo = 12344;
        ex.checkCard(cardNo); //line n2
        ex.readCard(cardNo); //line n3
    }
}

```

結果は何ですか？

- A. コンパイルは行 n1 でのみ失敗します。
- B. コンパイルは行 n2 でのみ失敗します。
- C. カードの読み取りカードの確認
- D. 行 n2 と行 n3 の両方でコンパイルが失敗します。
- E. コンパイルは n3 行目でのみ失敗します。

**Answer: E** ([メッセージを残す](#))

最新問題: 128

MyString クラスと Test クラスの定義を考えると、次のようになります。

```

MyString.java:

package p1;
class MyString {
    String msg;
    MyString(String msg) {
        this.msg = msg;
    }
}

Test.java:

package p1;
public class Test {
    public static void main(String[] args) {
        System.out.println("Hello " + new StringBuilder("Java SE 8"));
        System.out.println("Hello " + new MyString("Java SE 8"));
    }
}

```

結果は何ですか？

- A) Hello Java SE 8  
Hello Java SE 8
- B) Hello java.lang.StringBuilder@<<hashcode1>>  
Hello p1.MyString@<<hashcode2>>
- C) Hello Java SE 8  
Hello p1.MyString@<<hashcode>>
- D) Compilation fails at the Test class.

- A. オプション C
- B. オプション B
- C. オプション D
- D. オプション A

Answer: D ([メッセージを残す](#))

最新問題: 129

与えられる:

```
public class App {  
  
    public static void main(String[] args) {  
        Boolean[] bool = new Boolean[2];  
  
        bool[0] = new Boolean(Boolean.parseBoolean("true"));  
        bool[1] = new Boolean(null);  
  
        System.out.println(bool[0] + " " + bool[1]);  
    }  
}
```

結果は何ですか？

- A. 真の null
- B. 真偽
- C. コンパイルに失敗しました
- D. 実行時に NullPointerException がスローされる

Answer: B ([メッセージを残す](#))

最新問題: 130

与えられる:

```

public class App {
    int count;
    public static void displayMsg() {
        System.out.println("Welcome Visit Count: " + count++); // line n1
    }
    public static void main(String[] args) {
        App.displayMsg();
        displayMsg(); // line n2
    }
}

```

結果は何ですか？

- A. ようこそ訪問数:0 ようこそ訪問数: 1
- B. n2 行目でコンパイルが失敗します。
- C. n1 行目でコンパイルが失敗します。
- D. ようこそ訪問数:0 ようこそ訪問数: 0

Answer: ([解答を表示する](#))

```

1
2 public class App {
3     int count;
4     public static void displayMsg() {
5         System.out.println("Welcome Visit Count: " + count ++); //line n1
6     }
7     public static void main(String[] args) {
8         App.displayMsg();
9         displayMsg();
10    }
11 }
12

```

最新問題: 131

次のコード部分があるとします。

```

ArrayList<Vehicle> myList = new ArrayList<>();
myList.add(new Motorcycle());

```

どちらかが true の場合、コードがコンパイルされるのはどの 2 つのステートメントですか? (2つお選びください。)

- A. Vehicle は Motorcycle のスーパークラスです。
- B. オートバイは車両のスーパークラスです。
- C. Vehicle は Motorcycle クラスによって実装されるインターフェイスです。
- D. Vehicle と Motorcycle は両方とも Transportation インターフェイスを実装します
- E. Motorcycle は Vehicleclass を実装するインターフェイスです。
- F. Vehicle と Motorcycle はどちらも Transportation スーパークラスを拡張します。

Answer: A,C ([メッセージを残す](#))

最新問題: 132

コードの断片を考えると、次のようになります。

```

public static void main(String[] args) {
    double discount = 0;
    int qty = Integer.parseInt(args[0]);
    //line n1;
}

```

そして、次のような要件が与えられます。

-qty 変数の値が 90 以上の場合、割引 = 0.5 -qty 変数の値が 80 ~ 90 の場合、割引 = 0.2要件？

- A) `if (qty >= 90) { discount = 0.5; }`  
     `if (qty > 80 && qty < 90) { discount = 0.2; }`
- B) `discount = (qty >= 90) ? 0.5 : 0;`  
     `discount = (qty > 80) ? 0.2 : 0;`
- C) `discount = (qty >= 90) ? 0.5 : (qty > 80) ? 0.2 : 0;`
- D) `if (qty > 80 && qty < 90) {`  
     `discount = 0.2;`  
     `}` `else {`  
     `discount = 0;`  
     `}`  
     `if (qty >= 90) {`  
     `discount = 0.5;`  
     `}` `else {`  
     `discount = 0;`  
     `}`
- E) `discount = (qty > 80) ? 0.2 : (qty >= 90) ? 0.5 : 0;`

A. オプション C

B. オプション A

C. オプション D

D. オプション B

E. オプション E

**Answer: C** ([メッセージを残す](#))

最新問題: 133

コードの断片を考えると、次のようになります。

```
public class Employee {
    String name;
    boolean contract;
    double salary;
    Employee() {
        // line n1
    }
    public String toString() {
        return name + ":" + contract + ":" + salary;
    }
    public static void main(String[] args) {
        Employee e = new Employee();
        // line n2
        System.out.print(e);
    }
}
```

独立して行った場合、コードで `joe:true: 100.0` を出力できるようにする 2 つの変更はどれですか？

- A) Replace line n2 with:  
e.name = "Joe";  
e.contract = true;  
e.salary = 100;
- B) Replace line n2 with:  
this.name = "Joe";  
this.contract = true;  
this.salary = 100;
- C) Replace line n1 with:  
this.name = new String("Joe");  
this.contract = new Boolean(true);  
this.salary = new Double(100);
- D) Replace line n1 with:  
name = "Joe";  
contract = TRUE;  
salary = 100.0f;
- E) Replace line n1 with:  
this("Joe", true, 100);

- A. オプション E
- B. オプション D
- C. オプション B
- D. オプション C
- E. オプション A

Answer: ([解答を表示する](#))

最新問題: 134

Planet オブジェクトの次のコードを考えると、次のようになります。

```
public class Planet {
    public String name;
    public int moons;

    public Planet(String name, int moons) {
        this.name = name;
        this.moons = moons;
    }
}
```

And this method:

```
public static void main(String[] args){
    Planet[] planets = {
        new Planet("Mercury", 0),
        new Planet("Venus", 0),
        new Planet("Earth", 1),
        new Planet("Mars", 2)
    };

    System.out.println(planets);
    System.out.println(planets[2].name);
    System.out.println(planets[2].moons);
}
```

ORACLE®

出力は何ですか？

- A
- ```
planets
Earth
1
```
- B
- ```
[LPlanets.Planet;@15db9742
Earth
1
```
- C
- ```
[LPlanets.Planet;@15db9742
Planets.Planet@6d06d69c
1
```
- D
- ```
[LPlanets.Planet;@15db9742
Planets.Planet@6d06d69c
[LPlanets.Moon;@7852e922
```
- E
- ```
[LPlanets.Planet;@15db9742
Venus
0
```

- A. オプション B  
 B. オプション A  
 C. オプション C  
 D. オプション E  
 E. オプション D

Answer: C ([メッセージを残す](#))

最新問題: 135

コードの断片を考えると、次のようになります。

```
public static void main(String[] args) {
    String[] arr = {"A", "B", "C", "D"};
    for (int i = 0; i < arr.length; i++) {
        System.out.print(arr[i] + " ");
        if (arr[i].equals("C")) {
            continue;
        }
        System.out.println("Work done");
        break;
    }
}
```

結果は何ですか？

- A. ABC 仕事完了  
 B. ABCD 作業が完了しました  
 C. 完了した作業

D. コンパイルに失敗しました

Answer: ([解答を表示する](#))

最新問題: 136

与えられる:

```
class S1 {  
    protected void display(int x) {  
        System.out.print("Parent" + x);  
    }  
}  
class S2 extends S1 {  
    public void display(int x, int y) {  
        this.display(x);  
        display(y);  
        super.display(y);  
    }  
    public void display(int x) {  
        System.out.println("Child " + x);  
    }  
}
```

そしてコードの断片:

```
S2 subj = 新しい S2(); subj.display(10, 100);
```

結果は何ですか?

A. 子 10

こども100

親100

B. 親 10

子供10

親 1000

C. 子 10

親100

親100

D. コンパイル時エラーが発生します。

Answer: D ([メッセージを残す](#))

```
Error: Main method not found in class S1, please define the main method as:  
public static void main(String[] args)  
or a JavaFX application class must extend javafx.application.Application
```

有効な 1z0-808 問題集は GoShiken.com が提供された合格しやすい 1z0-808 試験問題集！  
GoShiken.com が最新の 1z0-808 試験問題集を提供しています。GoShiken.com 1z0-808 試験問題は最新で、解答が正確でございます。最新の GoShiken.com 1z0-808 問題集をゲットする人はこちら: <https://www.goshiken.com/Oracle/1z0-808-mondaishu.html> (39830%OFF問題集溶と正解付きで 30%w 特別割引コード: **Freepdfdumps**)

### 最新問題: 137

MyString クラスと Test クラスの定義を考えると、次のようになります。

MyString.java:

```
package p1;
class MyString {
    String msg;
    MyString(String msg) {
        this.msg = msg;
    }
}
```

Test.java:

```
package p1;
public class Test {
    public static void main(String[] args) {
        System.out.println("Hello " + new StringBuilder("Java SE 8"));
        System.out.println("Hello " + new MyString("Java SE 8"));
    }
}
```

結果は何ですか？

- A  
Hello Java SE 8  
Hello Java SE 8
- B  
Hello java.lang.StringBuilder@<<hashcode1>>  
Hello p1.MyString@<<hashcode2>>
- C  
Hello Java SE 8  
Hello p1.MyString@<<hashcode>>
- D  
Compilation fails at the Test class

- A. オプション A
- B. オプション C
- C. オプション D
- D. オプション B

Answer: ([解答を表示する](#))

最新問題: 138

コードの断片を考えると、次のようになります。

```
public static void main(String[] args) {
    int[][] arr = new int [2] [4];
    arr[0] = new int [] {1, 3, 5, 7};
    arr[1] = new int [] {1, 3};
    for (int[] a : arr) {
        for (int i : a) {
            System.out.print(i+ " ");
        }
        System.out.println();
    }
}
```

結果は何ですか？

A. コンパイルは失敗します。

1 3

B. 1 3

1 3

C. followed by an ArrayIndexOutOfBoundsException

1 3

D. 1 3 0 0

1 3 5 7

E. 1 3

Answer: E (メッセージを残す)

The screenshot shows an IDE window with the following content:

- Your Code ...**: A code editor containing the Java code from the question.
- External Libraries ...**: A button to "Add External Library (from Maven Repo)".
- CommandLine Arguments ...**: An empty text input field.
- Interactive mode :** A dropdown menu set to "OFF".
- Version:** A dropdown menu set to "JDK 9.0.1".
- Stdin Inputs...**: An empty text input field.
- Execute**, **Save**, **My Projects**, **Recent**, **Collaborate**, **More Options -**: A row of buttons.
- Result...**: A section showing execution details: "CPU Time: 0.13 sec(s), Memory: 30680 kilobyte(s)" and "compiled and executed in 0.785 sec(s)".
- Output**: A black box with white text showing the output: "1 3 7" on the first line and "1 3" on the second line.
- ORACLE**: The Oracle logo is visible in the bottom right corner.

最新問題: 139

メソッドは3つの引数を取るように宣言されます。プログラムはこのメソッドを呼び出し、2つの引数だけを渡します。結果は何ですか？

- A. 3番目の引数には値 void が与えられます。
- B. 3番目の引数には値 0 が与えられます。
- C. コンパイルは失敗します。
- D. 3番目の引数には、宣言された型に適切な false 値が与えられます。F) メソッドが3番目の引数にアクセスしようとする、例外が発生します。
- E. 3番目の引数には値 null が与えられます。

Answer: C ([メッセージを残す](#))

最新問題: 140

与えられる:

```
public class MyClass {
    public static void main(String[] args) {
        String s = "Java Duke";
        int len = s.trim().length();
        System.out.print(len);
    }
}
```

結果は何ですか？

- A. 9
- B. 10
- C. コンパイルに失敗しました。
- D. 11
- E. 8

Answer: ([解答を表示する](#))

最新問題: 141

コードの断片を考えると、次のようになります。

```
13. List colors = new ArrayList();
14. colors.add("green");
15. colors.add("blue");
16. colors.add("red");
17. colors.add("yellow");
18. colors.remove(2);
19. colors.add(3, "cyan");
20. System.out.print(colors);
```

結果は何ですか？

[緑、赤、黄、シアン]

A.

B. 実行時に `IndexOutOfBoundsException` がスローされます。

C. [緑、青、黄、シアン]

D. [緑、赤、シアン、黄]

Answer: B ([メッセージを残す](#))

最新問題: 142

与えられる:

```
ORACLE
class Patient {
    String name;
    public Patient (String name) {
        this.name = name;
    }
}
```

コードの断片は次のとおりです。

```
8. public class Test {
9.     public static void main (String [] args) {
10.         List ps = new ArrayList ();
11.         Patient p2 = new Patient ("Mike");
12.         ps.add(p2);
13.
14.         // insert code here
15.
16.         if (f >= 0) {
17.             System.out.print ("Mike Found");
18.         }
19.     }
20. }
```

14 行目に挿入すると、Mike Found を出力できるコード フラグメントはどれですか？

A. `int f = ps.indexOf (p2);`

`Patient p = new Patient ("Mike");`

`int f = ps.indexOf (p)`

B.

- C. `int f = ps.indexOf (Patient ("Mike") );`
- D. `int f = ps.indexOf (new Patient "Mike") );`

Answer: A ([メッセージを残す](#))

最新問題: 143

Java 言語のオブジェクト指向機能を説明している 3 つのステートメントはどれですか? 3つお選びください。)

- A. main メソッドはすべてのクラスで宣言する必要があります。
- B. オブジェクトは、他のすべてのオブジェクトのルート クラスです。
- C. オブジェクトは他のオブジェクトと動作を共有できます。
- D. オブジェクトは再利用できません。
- E. サブクラスはスーパークラスから継承できます。
- F. パッケージには複数のクラスが含まれている必要があります。

Answer: A,C,E ([メッセージを残す](#))

最新問題: 144

これらのクラスを考えると、次のようになります。

```
public class Employee {
    public int salary;
}

public class Manager extends Employee {
    public int budget;
}

public class Director extends Manager {
    public int stockOptions;
}
```

そして、このメインメソッドを考えると、次のようになります。

```
public static void main(String[] args) {
    Employee employee = new Employee();
    Employee manager = new Manager();
    Employee director = new Director();
    //line n1
}
```

main メソッドの n1 行目に配置されたときにコンパイルされる 2 つのオプションはどれですか? (2つお選びください。)

- A. `従業員.給与 = 50_000;`
- B. `マネージャー.予算 = 1_000_000;`
- C. `director.stockOptions = 1_000;`
- D. `マネージャー.ストックオプション = 500;`

E. ディレクター.給与 = 80\_000;

F. 従業員.予算 = 200\_000;

Answer: ([解答を表示する](#))

最新問題: 145

コンパイルされた Java ソース ファイルのコード スニペットを考えると、次のようになります。

```
public class MyFile
{
    public static void main (String[] args)
    {
        String arg1 = args[1];
        String arg2 = args[2];
        String arg3 = args[3];
        System.out.println("Arg is " + arg3);
    }
}
```

次の出力を得るには、どのコマンドライン引数をプログラムに渡す必要がありますか？

引数は 2

A. java MyFile 0 1 2 3

B. Java MyFile 2 2 2

C. java MyFile 1 3 2 2

D. Java MyFile 1 2 2 3 4

Answer: C ([メッセージを残す](#))

最新問題: 146

与えられる:

```
class Test
    int a1;

    public static void doProduct(int a) {
        a = a * a;
    }

    public static void doString(String s) {
        s.concat (" " + s);
    }

    public static void main(String[] args) {
        Test item = new Test();
        item.a1 = 11;
        String sb = "Hello";
        Integer i = 10;
        doProduct(i);
        doString(sb);
        doProduct(item.a1);
        System.out.println(i + " " + sb + " " + item.a1);
    }
}
```

結果は何ですか？

- A. 10 こんにちは、こんにちは 11
- B. 100 ハローハロー 121
- C. 100 こんにちは 121
- D. 10 こんにちは 11
- E. 10 ハローハロー 121

Answer: D ([メッセージを残す](#))

最新問題: 147

コードの断片を考えると、次のようになります。

```
3. public static void main(String[] args) {
4.     int iVar = 100;
5.     float fVar = 100.100f;
6.     double dVar = 123;
7.     iVar = fVar;
8.     fVar = iVar;
9.     dVar = fVar;
10.    fVar = dVar;
11.    dVar = iVar;
12.    iVar = dVar;
13. }
```

コンパイルに失敗する 3 行はどれですか？ 3つお選びください。)

- A. 7 行目
- B. 10 行目
- C. 11 行目
- D. 12 行目
- E. 9 行目
- F. 8 行目

Answer: [\(解答を表示する\)](#)

最新問題: 148

sb という名前の StringBuilder 変数の内容を空にするステートメントはどれですか？

- A. sb. delete (0, sb. length ());
- B. sb. すべて削除 ( )
- C. sb. delete (0, sb.size ());
- D. sb. すべて削除する ( )

Answer: A ([メッセージを残す](#))

最新問題: 149

与えられる:

```
class Product {
    double price;
}

public class Test {
    public void updatePrice(Product product, double price) {
        price = price * 2;
        product.price = product.price + price;
    }
    public static void main(String[] args) {
        Product prt = new Product ();
        prt.price = 200;
        double newPrice = 100;

        Test t = new Test ();
        t.updatePrice(prt, newPrice);
        System.out.println(prt.price + " : " + newPrice);
    }
}
```

結果は何ですか？

- A. 400.0 : 200.0
- B. 400.0 : 100.0
- C. コンパイルは失敗します。
- D. 200.0 : 100.0

Answer: B ([メッセージを残す](#))

最新問題: 150

コードの断片を考えると、次のようになります。

```
int nums1[] = new int[3];  
int nums2[] = {1, 2, 3, 4, 5};  
nums1 = nums2;  
for (int x : nums1) {  
    System.out.print(x + ":");  
}
```

結果は何ですか？

- A. コンパイルは失敗します。
- B. 1:2:3:4:5:
- C. 1:2:3:
- D. 実行時に ArrayOutOfBoundsException がスローされます。

Answer: B ([メッセージを残す](#))

**Valid 1z0-808 Dumps** shared by GoShiken.com for Helping Passing 1z0-808 Exam!  
GoShiken.com now offer the **newest 1z0-808 exam dumps**, the GoShiken.com 1z0-808 exam **questions have been updated** and **answers have been corrected** get the **newest** GoShiken.com 1z0-808 dumps with Test Engine here:

<https://www.goshiken.com/Oracle/1z0-808-mondaishu.html> (398 Q&As Dumps, **30%OFF**)

Special Discount: **Freepdfdumps**)