

Oracle.1Z0-819.v2022-09-06.q170

試験コード:	1Z0-819
試験名称:	Java SE 11 Developer
認定資格:	Oracle
無料問題数:	170
バージョン:	v2022-09-06
アクセス数:	1590
ページビュー数:	1700
https://www.jpnpdf.com/Oracle.1Z0-819.v2022-09-06.q170-mondaishu.html	

最新問題: 1

機密情報を保護するために使用される3つのガイドラインはどれですか？ 3つ選択してください。)

- A. 診断を改善するために情報を透過的に処理します。
- B. 機密情報をカプセル化します。
- C. 機密情報やその他の情報を一律に管理します。
- D. 機密情報を保持しているオブジェクトへのアクセスを制限します。
- E. ユーザー入力を通常の情報として扱います。
- F. 機密情報を保存する前に入力を検証します。
- G. 機密情報を明確に識別してラベルを付けます。

Answer: A,D,F (メッセージを残す)

最新問題: 2

与えられた：

```
public class Tester {
    public static void main(String[] args) {
        int x = 0, y = 6;
        for( ; x < y ; x++, y--) { // line 1
            if (x%2 == 0) {
                continue;
            }
            System.out.println(x+"-"+y);
        }
    }
}
```

結果はどうなりますか？

- 2-4
- A. 0-6
- B. 1-5

2-4

1-5

C. 1-5

D. 2-4

E. 1行目のエラーが原因でコンパイルが失敗します。

0-6

F. 0-6

G. 2-4

Answer: C ([メッセージを残す](#))

The screenshot shows a Java IDE window with a code editor and a console output area. The code editor contains the following Java code:

```
1 public class Tester {  
2     public static void main(String[] args) {  
3         int x = 0, y = 6;  
4         for (; x < y ; x++, y--) { //line 1  
5             if (x%2 == 0) {  
6                 continue;  
7             }  
8             System.out.println(x+"-"+y);  
9         }  
10    }  
11 }
```

Below the code editor, there is a section titled "Execute Mode, Version, Inputs & Arguments" with a dropdown menu showing "JDK 11.0.4". Below this is a section titled "CommandLine Arguments" which is currently empty.

The "Result" section shows the execution output:

CPU Time: 0.27 sec(s), Memory: 35356 kilobyte(s)

1-5

最新問題: 3

Javaクラスファイルの構造で可能なステートメントの正しい順序はどれですか？

- A. インポート、パッケージ、クラス
- B. クラス、パッケージ、インポート
- C. パッケージ、インポート、クラス
- D. インポート、クラス、パッケージ
- E. パッケージ、クラス、インポート

Answer: C ([メッセージを残す](#))

最新問題: 4

与えられた :

```
public class Test {
    public static void main(String[] args) {
        int x;
        int y = 5;
        if (y > 2) {
            x = ++y;
            y = x + 7;
        } else {
            y++;
        }
        System.out.print(x + " " + y);
    }
}
```

結果はどうなりますか？

- A. コンパイルエラー
- B. 0 5
- C. 6 13
- D. 5 12

Answer: A ([メッセージを残す](#))

```
1 public class Test {
2     public static void main (String[] args) {
3         int x;
4         int y = 5;
5         if (y > 2) {
6             x = ++y;
7             y = x + 7;
8         } else {
9             y++;
10        }
11        System.out.print(x + " "+y);
12    }
13 }
```

✖ variable x might not have been initialized

最新問題: 5

与えられた :

```

List<String> list1 = new LinkedList<String>();
Set<String> hs1 = new HashSet<String>();
String[] v = {"a", "b", "c", "b", "a"};
for (String s: v) {
    list1.add(s);
    hs1.add(s);
}
System.out.print(hs1.size() + " " + list1.size() + " ");
HashSet hs2 = new HashSet(list1);
LinkedList list2 = new LinkedList(hs1);
System.out.print(hs2.size() + " " + list2.size());

```

結果はどうなりますか？

A. 3 5 3 3

B. 3 3 3 3

C. 3 5 3 5

D. 5 5 3 3

Answer: A (メッセージを残す)

```

1  import java.util.*;
2  import java.io.*;
3  import java.lang.Thread;
4  import java.util.ArrayList;
5  import java.util.LinkedList;
6  import java.util.List;
7  import java.util.function.Consumer;
8  import java.util.stream.Stream;
9  import java.util.stream.IntStream;
10 import java.util.Optional;
11
12
13 public class Main {
14     public static void main(String[] args) {
15         List<String> list1 = new LinkedList<String>();
16         Set<String> hs1 = new HashSet<String>();
17         String[] v = {"a", "b", "c", "b", "a"};
18         for (String s: v) {
19             list1.add(s);
20             hs1.add(s);
21         }
22         System.out.println(hs1.size() + " " + list1.size() + " ");
23         HashSet hs2 = new HashSet(list1);
24         LinkedList list2 = new LinkedList(hs1);
25         System.out.print(hs2.size() + " " + list2.size());
26
27     }
28 }

```

Result

CPU Time: 0.28 sec(s), Memory: 36204 kilobyte(s)

35
33

最新問題: 6

与えられた:

```
import java.util.function.BiFunction;
public class Pair<T> {
    final BiFunction<T, T, Boolean> validator;
    T left = null;
    T right = null;
    private Pair() {
        validator=null;
    }
    Pair(BiFunction<T, T, Boolean> v, T x, T y) {
        validator = v;
        set(x, y);
    }
    void set(T x, T y) {
        if (!validator.apply(x, y)) throw new IllegalArgumentException();
        setLeft(x);
        setRight(y);
    }
    void setLeft(T x) {
        left = x;
    }
    void setRight(T y) {
        right = y;
    }
    final boolean isValid() {
        return validator.apply(left, right);
    }
}
```

ペアのpインスタンスの場合、p.isValid()はtrueを返す必要があります。

この要件が満たされることを保証するための可視性の変更の最小セットはどれですか？

A. left、right、setLeft、およびsetRightはプライベートである必要があります。

B. setLeftとsetRightを保護する必要があります。

C. isValidはパブリックである必要があります。

D. 左右はプライベートである必要があります。

Answer: D ([メッセージを残す](#))

最新問題: 7

コードフラグメントが与えられた場合:

```
List<整数> list = List.of(1,12,13,12,13);
```

コンパイル時エラーの原因となるステートメントはどれですか？

- A. 整数a = Integer.valueOf (list.get (0))
- B. double f = list.get (0);
- C. Double d = list.get (0);
- D. int c = list.get (0);
- E. Double e = Double.valueOf (list.get (0))
- F. 整数b = list.get (0);

Answer: ([解答を表示する](#))

最新問題: 8

与えられた :

```
public class Tester {
    static class Person implements Comparable<Person> {
        private String name;
        Person(String name) { this.name = name; }
        /* line 2 */
    }
    public static void main(String[] args) {
        Person[] people = {new Person("Joe"),
                           new Person("Jane"),
                           new Person("John")};
        Arrays.sort(people);
        for(Person person: people) {
            System.out.println(person.name);
        }
    }
}
```

コードで次の出力を生成する必要があります。

ジョン

ジョー

ジェーン

出力を生成するには、1行目と2行目にどのコードフラグメントを挿入する必要がありますか？

- A. 1行目にComparator<Person>を挿入します。

入れる

```
public int compareTo (Person person){
    person.name.compareTo (this.name);を返します。
}
```

2行目。

- B. 1行目にComparator<Person>を挿入します。

入れる

```
public int compare (Person person){
    person.name.compare (this.name);を返します。
}
```

2行目。

C. 1行目にComparator<Person>を挿入します。

入れる

```
public int compare (Person p1, Person p2){  
    p1.name.compare (p2.name);を返します。  
}
```

2行目。

D. 1行目にComparable<Person>を挿入します。

入れる

```
public int compare (Person p1, Person p2){  
    p1.name.compare (p2.name);を返します。  
}
```

2行目。

Answer: A ([メッセージを残す](#))

最新問題: 9

与えられた:

```
public class Main {  
    class Student {  
        String classname;  
        Student(String classname) {  
            this.classname = classname;  
        }  
    }  
    public static void main(String[] args) {  
        var student = new Student("Biology"); // line 3  
    }  
}
```

Mainクラスをコンパイルする2つの独立した変更はどれですか？ 2つ選択してください。）

A. Studentクラス宣言全体を別のJavaファイルStudent.javaに移動します。

B. 2行目をpublic Student (String classname)に変更します。

C. 1行目をパブリッククラスのStudent{に変更します。

D. 3行目をStudent student = new Student ("Biology");に変更します。

E. 1行目を静的クラスStudent{に変更します。

Answer: B,D ([メッセージを残す](#))

```
ORACLE
1 import java.util.*;
2 import java.io.*;
3 import java.lang.Thread;
4 import java.util.ArrayList;
5 import java.util.LinkedList;
6 import java.util.List;
7 import java.util.function.Consumer;
8 import java.util.stream.Stream;
9 import java.util.stream.IntStream;
10 import java.util.Optional;
11
12
13 public class Main {
14     class Student {
15         String classname;
16         public Student (String classname) {
17             this.classname = classname;
18         }
19
20     }
21     public static void main (String[] args) {
22         var student = new Student ("Biology");
23     }
24 }
```

最新問題: 10

与えられた :

`jdeps -jdkinternals C:\worksheet4\SimpleSecurity\jar\classes.jar`

期待される出力を説明するのはどれですか？

- A. `jdeps`は、参照されているすべてのJDK内部APIのモジュール依存関係とパッケージ名を一覧表示します。見つかった場合は、提案された代替品がコンソールに出力されます。
- B. `jdeps`は、`-jdkinternals`オプションがコンソールに出力するために`-summary` または `--verbose` オプションのいずれかを必要とするというエラーメッセージを出力します。
- C. `-jdkinternals`オプションは、`.jar`内のすべてのクラスを分析し、すべてのクラスレベルの依存関係を出力します。
- D. `-jdkinternals`オプションは、`.jar`内のすべてのクラスを分析して、JDK内部APIへのクラスレベルの依存関係を確認します。見つかった場合は、推奨される置換の結果がコンソールに出力されます。

Answer: A (メッセージを残す)

説明

`-jdkinternals`オプションは、`.jar`内のすべてのクラスを分析して、JDK内部APIへのクラスレベルの依存関係を確認します。見つかった場合は、推奨される置換の結果がコンソールに出力されます。

最新問題: 11

与えられた :

```

public class FunctionalInterfaceTest {
    public static void main(String[] args) {
        List fruits = Arrays.asList("apple", "orange", "banana");
        Consumer<String> c = System.out::print;
        Consumer<String> output = c.andThen(x -> System.out.println(": " + x.toUpperCase()));
        fruits.forEach(output);
    }
}

```

出力は何ですか？

A. : APPLE :ORANGE :BANANA

アップルオレンジバナナ

B. : APPLE :ORANGE :BANANA

C. APPLE :apple ORANGE :orange BANANA :banana

D. appleorangebanana

:APPLE :ORANGE :BANANA

E. アップル :APPLEオレンジ :ORANGEバナナ :BANANA

Answer: E ([メッセージを残す](#))

The screenshot shows a Java IDE with the following code:

```

1  import java.util.*;
2  import java.io.*;
3  import java.lang.Thread;
4  import java.util.ArrayList;
5  import java.util.LinkedList;
6  import java.util.List;
7  import java.util.function.Consumer;
8
9  public class FunctionalInterfaceTest {
10     public static void main (String[] args) {
11         List fruits = Arrays.asList("apple", "orange", "banana");
12         Consumer<String> c = System.out::print;
13         Consumer<String> output = c.andThen(x -> System.out.println(": " + x.toUpperCase()));
14
15         fruits.forEach(output);
16     }
17 }
18 }

```

Below the code editor, the "Execute Mode, Version, Inputs & Arguments" section is visible. It shows "JDK 11.0.4" selected, "Interactive" mode, and "Stdin Inputs" set to "CommandLine Arguments". The "Execute" button is highlighted.

The "Result" section shows the output of the program:

```

apple:APPLE
orange:ORANGE
banana:BANANA

```

最新問題: 12

与えられた:

```
String[][] arr = {
    {"Red", "White"},
    {"Black"},
    {"Blue", "Yellow", "Green", "Violet"}
};
for(int row = 0; row < arr.length; row++) {
    int column = 0;
    for(; column < arr[row].length; column++) {
        System.out.println "[" + row + ", " + column + "] = " + arr[row][column]);
    }
}
```

結果はどうなりますか？

- A. [0,0]=赤[0,1]=白[1,0]=黒[1,1]=青[2,0]=黄[2,1]=緑[3,0]=バイオレット
- B. [0,0]=赤[1,0]=黒[2,0]=青
- C. java.lang.ArrayIndexOutOfBoundsExceptionがスローされました
- D. [0,0]=赤[0,1]=白[1,0]=黒[2,0]=青[2,1]=黄[2,2]=緑[2,3]=バイオレット

Answer: D ([メッセージを残す](#))



```
Console 1 Console 2 Console 3
[0,0] =Red
[0,1] =White
[1,0] =Black
[2,0] =Blue
[2,1] =Yellow
[2,2] =Green
[2,3] =Violet

Completed with exit code: 0
```

最新問題: 13

与えられた:

```

public class Price {
    private final double value;
    public Price(String value) {
        this(Double.parseDouble(value));
    }
    public Price(double value) {
        this.value = value;
    }
    public Price () {}
    public double getValue() { return value; }
    public static void main(String[] args) {
        Price p1 = new Price("1.99");
        Price p2 = new Price(2.99);
        Price p3 = new Price();
        System.out.println(p1.getValue()+" "+p2.getValue()+" "+p3.getValue());
    }
}

```

結果はどうなりますか？

- A. コンパイルは失敗します。
- B. 1.99,2.99,0
- C. 1.99,2.99,0.0
- D. 1.99,2.99

Answer: ([解答を表示する](#))

```

1 public class Price {
2     private final double value;
3     public Price(String value) {
4         this(Double.parseDouble(value));
5     }
6     public Price(double value) {
7         this.value = value;
8     }
9     public Price () {}
10    public double getValue() { return value; }
11    public static void main (String[] args) {
12        Price p1 = new Price("1.99");
13        Price p2 = new Price(2.99);
14        Price p3 = new Price();
15        System.out.println(p1.getValue()+" "+p2.getValue()+" "+p3.getValue());
16    }
17 }
18 }

```

variable value might not have been initialized

最新問題: 14

与えられた：

```

public class MyResource {
    public MyResource () {
    }
    // Resource methods
}

```

try-with-resourcesステートメントでmyResourceクラスを使用するとします。どの変更がこれを達成しますか？

- A. AutoCloseableを実装し、closeメソッドをオーバーライドします。
- B. AutoCloseableを拡張し、autoCloseメソッドをオーバーライドします。
- C. AutoCloseableを実装し、autoCloseメソッドをオーバーライドします。

D. AutoCloseableを拡張し、closeメソッドをオーバーライドします。

Answer: A ([メッセージを残す](#))

最新問題: 15

与えられた :

```
public class Plant { }

and

public class Tulip extends Plant { }

and

public class Garden {
    private static Plant plant;
    public static void main(String[] args) {
        plant = new Tulip();
        feed(plant);
        feed(plant);
    }
    public static void feed(Plant p) {
        if (p instanceof Tulip) {
            System.out.println("Take extra care");
        }
        p = null;
    }
}
```

結果はどうなりますか？

A. 特に注意してください

細心の注意を払ってください

B. プログラムは何も出力しません。

C. 実行時に例外がスローされます

D. 特に注意してください

Answer: C ([メッセージを残す](#))

最新問題: 16

与えられた :

```

public class ExSuper extends Exception {
    private final int eCode;
    public ExSuper(int eCode, Throwable cause) {
        super(cause);
        this.eCode = eCode;
    }

    public ExSuper(int eCode, String msg, Throwable cause) {
        super(msg, cause);
        this.eCode = eCode;
    }
    public String getMessage() {
        return this.eCode+": "+super.getMessage()+"-"+this.getCause().getMessage();
    }
}

public class ExSub extends ExSuper {
    public ExSub(int eCode, String msg, Throwable cause)
    { super(eCode, msg, cause); }
}

```

およびコードフラグメント：

```

try {
    String param1 = "Oracle";
    if (param1.equalsIgnoreCase("oracle")) {
        throw new ExSub(9001, "APPLICATION ERROR-9001", new FileNotFoundException ("MyFile.txt"));
    }
    throw new ExSuper(9001, new FileNotFoundException ("MyFile.txt")); // Line 1
} catch (ExSuper ex) {
    System.out.println(ex.getMessage());
}

```

結果はどうなりますか？

9001 java.io.FileNotFoundException :MyFile.txt-MyFile.txt

A. 9001 java.io.FileNotFoundException :MyFile.txt-MyFile.txt

9001 :アプリケーションエラー--9001-MyFile. txt

B. 9001 :アプリケーションエラー--9001-MyFile. txt

C. .コンパイルは1行目で失敗します。

Answer: ([解答を表示する](#))

有効な **1Z0-819** 問題集は GoShiken.com が提供された合格しやすい 1Z0-819 試験問題集！ GoShiken.com が最新の **1Z0-819** 試験問題集を提供しています。GoShiken.com 1Z0-819 試験問題は最新で、解答が正確でございます。最新の GoShiken.com 1Z0-819 問題集をゲットする人はこちら: <https://www.goshiken.com/Oracle/1Z0-819-mondaishu.html> (**29730%OFF**問題集溶と正解付きで **30%w** 特別割引コード: **Freepdfdumps**)

与えられた：

```
1. public class Main {
2.     public static void greet(String... args) {
3.         System.out.print("Hello ");
4.         for (String arg : args) {
5.             System.out.println(arg);
6.         }
7.     }
8.     public static void main(String[] args) {
9.         Main c = null;
10.        c.greet();
11.    }
12. }
```

結果はどうなりますか？

- A. NullPointerExceptionが4行目でスローされます。
- B. NullPointerExceptionが10行目でスローされます。
- C. コンパイルエラーが発生します。
- D. こんにちは

Answer: ([解答を表示する](#))



最新問題: 18

モジュールcom.exampleからメインクラスcom.acme.Mainを実行するコマンドラインはどれですか？

- A. java -classpath com.example.jar com.acme.Main
- B. java --module-path mods -m com.example / com.acme.Main
- C. java --module-path mods com.example / com.acme.Main
- D. java -classpath com.example.jar -m com.example / com.acme.Main

Answer: D ([メッセージを残す](#))

最新問題: 19

毎月の住宅ローンの支払いを計算する式が与えられた場合：

$$M = P \frac{r(1+r)^n}{(1+r)^n - 1}$$

およびこれらの宣言：

```
double m;           //monthly payment
double r = 0.05/12; //monthly interest rate
int p = 100_000;    //principal
int n = 180;        //number of payments
```

数式をどのようにコーディングできますか？

- A. `m = p * ((Math.pow(1 + r, n) / Math.pow(1 + r, n) - 1));`
- B. `m = p * r * Math.pow(1 + r, n) / Math.pow(1 + r, n) - 1;`
- C. `m = p * ((Math.pow(1 + r, n) / Math.pow(1 + r, n) - 1) )`
- D. `m = p * (r * Math.pow(1 + r, n) / Math.pow(1 + r, n) ) - 1 )`

Answer: C ([メッセージを残す](#))

最新問題: 20

与えられた：

```
public class Confidential implements Serializable{
    private String data;

    public Confidential(String data) {
        this.data = data;
    }
}
```

これらのオブジェクトの安全なシリアル化はどれですか？ 2つ選択してください。)

- A. `serialPersistentFields`配列フィールドを定義します。
- B. クラスを抽象化します。
- C. フィールドを一時的に宣言します。
- D. インスタンスをシリアルプロキシに置き換えるには`readResolve`のみを実装し、`writeReplace`は実装しません。
- E. インスタンスをシリアルプロキシに置き換えるために`writeReplace`のみを実装し、`readResolve`は実装しません。

Answer: A,D ([メッセージを残す](#))

最新問題: 21

与えられた：

```
int arr[][] = {{5,10},{8,12},{9,3}};
long count = Stream.of(arr)
    .flatMapToInt(IntStream::of)
    .map(n -> n + 1)
    .filter(n -> (n % 2 == 0))
    .peek(System.out::print)
    .count();
System.out.println(" " + count);
```

結果はどうなりますか？

- A. 6910 3
- B. 10126 3
- C. 3
- D. 6104 3

Answer: ([解答を表示する](#))

```

1 import java.util.*;
2 import java.io.*;
3 import java.lang.Thread;
4 import java.util.ArrayList;
5 import java.util.LinkedList;
6 import java.util.List;
7 import java.util.function.Consumer;
8 import java.util.stream.Stream;
9 import java.util.stream.IntStream;
10
11
12 public class Main {
13
14     public static void main(String[] args) {
15         int arr[][] = {{5,10}, {8,12}, {9,3}};
16         long count = Stream.of(arr)
17             .flatMapToInt(IntStream::of)
18             .map (n -> n + 1)
19             .filter(n -> (n % 2 == 0))
20             .peek(System.out::print)
21             .count();
22         System.out.println(" " + count);
23     }
24 }

```

Execute Mode, Version, Inputs & Arguments

JDK 11.0.4

CommandLine Arguments

Result

CPU Time: 0.32 sec(s), Memory: 34220 kilobyte(s)

5104 3

最新問題: 22

会社には、パッケージを含むJava8jarファイルを使用する既存の販売アプリケーションがあります。

```

com.company.customer;
com.company.customer.orders;
com.company.customer.info;
com.company.sales;
com.company.sales.leads;
com.company.sales.closed;
com.company.orders;
com.company.orders.pending;

```

com.company.orders.shipped。

このjarファイルをcustomer、sales、ordersの3つのモジュールにモジュール化するには、どのmodule-info.javaが正しいでしょうか。

A)

```
module com.company.customer {  
    opens com.company.customer;  
}  
module com.company.sales {  
    opens com.company.sales;  
}  
module com.company.orders {  
    opens com.company.orders;  
}
```

B)

```
module com.company.customer {  
    exports com.company.customer;  
}  
module com.company.sales {  
    exports com.company.sales;  
}  
module com.company.orders {  
    exports com.company.orders;  
}
```

C)

```
module com.company.customer {  
    requires com.company.customer;  
}  
module com.company.sales {  
    requires com.company.sales;  
}  
module com.company.orders {  
    requires com.company.orders;  
}
```

D)

```
module com.company.customer {  
    provides com.company.customer;  
}  
module com.company.sales {  
    provides com.company.sales;  
}  
module com.company.orders {  
    provides com.company.orders;  
}
```

A. オプションC

B. オプションD

C. オプションA

D. オプションB

Answer: A ([メッセージを残す](#))

最新問題: 23

PrintServiceAPIモジュールで定義された印刷サービスのサービスプロバイダーにとって正しいmodule-info.javaはどれですか？

モジュールPrintServiceProvider {

A. PrintServiceAPIが必要です。

org.printservice.spi.Printを提供します

com.provider.PrintService;

}

モジュールPrintServiceProvider {

B. PrintServiceAPIが必要です。

com.provider.PrintServiceを使用します。

}

モジュールPrintServiceProvider {

C. PrintServiceAPIが必要です。

org.printservice.spi.Printwithをエクスポートします

com.provider.PrintService;

}

D. PrintServiceAPIが必要です。

org.printservice.spiをエクスポートします。

}

モジュールPrintServiceProvider {

Answer: D ([メッセージを残す](#))

最新問題: 24

与えられた：

```
for(var i = 0; i < 10; i++)
    switch(i%5) {
        case 2:
            i *= i;
            break;
        case 3:
            i++;
            break;
        case 1:
        case 4:
            i++;
            continue;
        default:
            break;
    }
System.out.print(i + " ");
i++;
}
```

結果はどうなりますか？

A. 0 4 9

B. 0

C. 10

D. 何もない

Answer: D ([メッセージを残す](#))

最新問題: 25

どのコードフラグメントがコンパイルされますか？

```
A. Comparator comparator = new Comparator<?>() {  
    public int compare(Integer i, Integer j) {  
        return i.compareTo(j);  
    }  
};  
  
B. var comparator = new Comparator<>() {  
    public int compare(Integer i, Integer j) {  
        return i.compareTo(j);  
    }  
};  
  
C. Comparator<> comparator = new Comparator<Integer>() {  
    public int compare(Integer i, Integer j) {  
        return i.compareTo(j);  
    }  
};  
  
D. Comparator<Integer> comparator = new Comparator<>() {  
    public int compare(Integer i, Integer j) {  
        return i.compareTo(j);  
    }  
};
```

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

Answer: D ([メッセージを残す](#))

```
1 import java.io.*;  
2 import java.util.*;  
3 class abc {  
4     public static void main(String[] args) {  
5  
6         Comparator<Integer> comparator = new Comparator<>() {  
7             public int compare(Integer i, Integer j) {  
8                 return i.compareTo(j);  
9             }  
10        };  
11    }  
12 }  
13 }  
14 }
```

最新問題: 26

与えられた：

```

public class Tester {
    public static void main(String[] args) {
        int x = 4;
        int y = 2;
        System.out.println(x+y+"=(x+y)="+x+y);
    }
}

```

結果はどうなりますか？

A. 実行時に例外がスローされます。

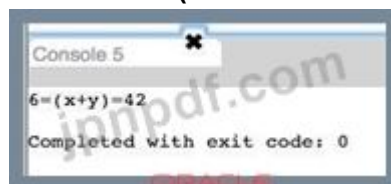
B. 42 = x + y) = 42

C. 42 = x + y) = 6

D. 6 = x + y) = 42

E. 6 = x + y) = 6

Answer: D (メッセージを残す)



最新問題: 27

```

public class X {
    protected void print(Object obj) {
        System.out.println(obj);
    }
    public final void print(Object... objects) {
        for(Object object : objects) {
            print(object);
        }
    }
    public void print(Collection collection) {
        collection.forEach(System.out::println);
    }
}

```

and

```

public class Y extends X {
    public void print(Object obj) {
        System.out.print "[" + obj + "]";
    }
    public void print(Object... objects) {
        for(Object object : objects) {
            System.out.println "[" + object + "]";
        }
    }
    public void print(Collection collection) {
        print(collection.toArray());
    }
}

```

このコンパイルが失敗するのはなぜですか？

- A. メソッドY. print (オブジェクト)はメソッドsuper.print (オブジェクト)を呼び出さない
- B. メソッドprint (オブジェクト)とメソッドprint (オブジェクト...)は互いに重複しています。
- C. メソッドx. 印刷 (オブジェクト)はYにアクセスできません。
- D. メソッドY. print (Object ...)は、最終メソッドx.print (Object ...)をオーバーライドできません。
- E. メソッドxで。印刷 (レクション)、システム。Out::printsは無効なJava識別子です。

Answer: ([解答を表示する](#))

最新問題: 28

与えられた :

```
import java.util.*;
public class Foo {
    public List<Number> foo(Set<CharSequence> m) { ... }
}
```

and

```
import java.util.*;
public class Bar extends Foo {
    //line 1
}
```

正常にコンパイルするために、Barの1行目に追加できる2つのステートメントはどれですか？ 2つ選択してください。)

- A. public List <Integer> foo (TreeSet <String> m){...}
- B. public ArrayList <Integer> foo \$Set <String> m){...}
- C. public List <Integer> foo \$Set <String> m){...}
- D. public ArrayList <Number> foo \$Set <CharSequence> m){...}
- E. public List <Object> foo \$Set <CharSequence> m){...}
- F. public List <Integer> foo \$Set <CharSequence> m){...}

Answer: ([解答を表示する](#))

最新問題: 29

与えられた :

```

class Test {
    public static void doThings() throws GeneralException
    {
        try {
            throw new RuntimeException("Something happened");
        } catch (Exception e) {
            throw new SpecificException(e.getMessage());
        }
    }
    public static void main(String args[]) {
        try {
            Test.doThings();
        } catch (Exception e) {
            System.out.println(e.getMessage());
        }
    }
}

class GeneralException /* line 1 */ {
    public GeneralException(String s) { super(s); }
}

class SpecificException /* line 2 */ {
    public SpecificException(String s) { super(s); }
}

```

コードが何かを印刷できるようにするには、どのオプションを選択する必要がありますか？

A. 1行目にextendsGeneralExceptionを追加します。

2行目にextendsExceptionを追加します。

B. 1行目にextendsSpecificExceptionを追加します。

2行目にGeneralExceptionを追加します。

C. 1行目にextendsExceptionを追加します。

2行目にextendsExceptionを追加します。

D. 1行目にextendsExceptionを追加します。

2行目にGeneralExceptionを追加します。

Answer: D ([メッセージを残す](#))

```
1 import java.util.*;
2 import java.io.*;
3 import java.lang.Thread;
4 import java.util.ArrayList;
5 import java.util.LinkedList;
6 import java.util.List;
7
8 public class Test {
9
10     public static void doThings() throws GeneralException {
11         try{
12             throw new RuntimeException("Something happened");
13         } catch (Exception e) {
14             throw new SpecificException (e.getMessage());
15         }
16     }
17 }
18
19 public static void main(String args[]) {
20     try{
21         Test.doThings();
22     } catch (Exception e) {
23         System.out.println(e.getMessage());
24     }
25 }
26 class GeneralException extends Exception {
27     public GeneralException(String s) { super(s); }
28 }
29 class SpecificException extends GeneralException {
30     public SpecificException(String s) { super(s);}
31 }
32 }
```

最新問題: 30

与えられた :

```

import java.time.LocalDate;
import static java.time.DayOfWeek.*;
public class Main {
    public static void main(String[] args) {
        var today = LocalDate.now().with(TUESDAY).getDayOfWeek();
        switch(today) {
            case SUNDAY:
            case SATURDAY:
                System.out.println("Weekend");
                break;
            case MONDAY: FRIDAY:
                System.out.println("Working");
            default:
                System.out.println("Unknown");
        }
    }
}

```

ORACLE®

結果はどうなりますか？

- A. WorkingUnknown
- B. 不明
- C. 火曜日不明
- D. コンパイルは失敗します。
- E. 火曜日
- F. 作業中

Answer: B ([メッセージを残す](#))



最新問題: 31

与えられた：

```

public class Employee {
    private String name;
    private String locality;
    /* the constructor, getter and setter methods code goes here */
}

```

と：

```

8. List<Employee> roster = new ArrayList<>();
9. long empCount = roster.stream()
10. /* insert code here */
11. System.out.print(empCount);

```

10行目に挿入されたときに、名簿リストから一意の地域の数を入力するコードはどれですか？

- A. .map {e} e.getLocality ())
.collect Collectors.toSet ())
。カウント ()
- B. .map {Employee :: getLocality}
。明確 ()
。カウント ()
- C. map {e} e.getLocality ())
。カウント ()
- D. .filter {Employee :: getLocality}
。明確 ()
。カウント ()
- Answer: ([解答を表示する](#))

有効な **1Z0-819** 問題集は GoShiken.com が提供された合格しやすい 1Z0-819 試験問題集！ GoShiken.com が最新の **1Z0-819** 試験問題集を提供しています。GoShiken.com 1Z0-819 試験問題は最新で、解答が正確でございます。最新の GoShiken.com 1Z0-819 問題集をゲットする人はこちら: <https://www.goshiken.com/Oracle/1Z0-819-mondaishu.html> (**29730%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

与えられた:

```
import java.util.ArrayList;
import java.util.Arrays;
public class NewMain {
    public static void main(String[] args) {
        String[] fruitNames = { "apple", "orange",
                                "grape", "lemon", "apricot", "watermelon" };
        var fruits = new ArrayList<>(Arrays.asList(fruitNames));
        fruits.sort((var a, var b) -> -a.compareTo(b));
        fruits.forEach(System.out::println);
    }
}
```

結果はどうなりますか？

- A. スイカorangelemongrapeapricotapple
- B. 何もない
- C. appleapricotgrapelemonorangewatermelon
- D. appleorangegrapelemonapricotwatermelon

Answer: ([解答を表示する](#))

```
Console 3
watermelon
orange
lemon
grape
apricot
apple

Completed with exit code: 0
```

最新問題: 33

JDKをモジュール化する理由を説明しているのはどれですか？ 2つ選択してください。）

- A. Java言語を理解しやすい
- B. アプリケーションの堅牢性を向上させます
- C. アプリケーションモジュールとJDKモジュールをリンクするカスタムランタイムを簡単に構築できます
- D. 実装の詳細を公開しやすく
- E. セキュリティと保守性を向上させます

Answer: B,E ([メッセージを残す](#))

最新問題: 34

与えられた：

```
パスp1=Paths.get ("/ strike / Exam / topsecret / Answers");
```

```
パスp2=Paths.get ("/ strike / Exam / Answers / temp.txt");
```

```
パスp3=Paths.get ("/ strike / Answers / topsecret");
```

..\..\..\Answers\topsecretを出力する2つのステートメントはどれですか？ 2つ選択してください。）

- A. System.out.print p2.relativeTo p1) ;
- B. System.out.print p1.relativeTo p2) ;
- C. System.out.print p2.relativeTo p3) ;
- D. System.out.print p3.relativeTo p2) ;
- E. System.out.print p3.relativeTo p1) ;
- F. System.out.print p1.relativeTo p3) ;

Answer: E,F ([メッセージを残す](#))

最新問題: 35

与えられた：

```

import java.util.*;

public class Main {
    static Map<String, String> map = new HashMap<>();
    static List<String> keys =
        new ArrayList<>(List.of("A", "B", "C", "D"));
    static String[] values =
        {"one", "two", "three", "four" };

    static {
        for(var i = 0; i < keys.size(); i++) {
            map.put(keys.get(i), values[i]);
        }
    }

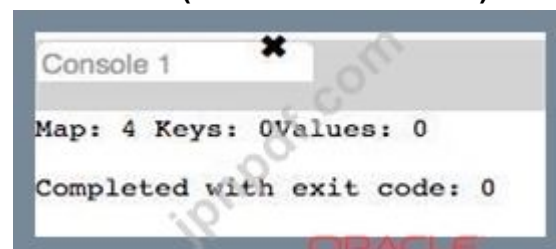
    public static void main(String[] args) {
        keys.clear();
        values = new String[0];
        System.out.println("Map: " + map.size() +
            " Keys: " + keys.size() +
            " Values: " + values.length);
    }
}

```

結果はどうなりますか？

- A. マップ :0キー :0値 :0
- B. コンパイルは失敗します。
- C. マップ :4キー :4値 :4
- D. マップ :4キー :0値 :0
- E. マップ :0キー :4値 :4

Answer: D ([メッセージを残す](#))



The screenshot shows a console window titled 'Console 1' with the following output:

```

Map: 4 Keys: 0Values: 0
Completed with exit code: 0

```

最新問題: 36

与えられた :

```

1 public class Secret {
2     String[] names;
3     public Secret(String[] names) {
4         this.names = names;
5     }
6     public String[] getNames() {
7         return names;
8     }
9 }

```

Java SEセキュリティガイドラインを実装する3つのアクションはどれですか？ 3つ選択してください。）

- A. getNames (メソッド名をgetNames()に変更します。
- B. 3行目をprivate Secret \$String [] names){に変更します。
- C. 6行目をpublicsynchronized String [] getNames ({}に変更します。
- D. 4行目をthis.names = names.clone ()に変更します。
- E. 2行目をprivate final String [] names;に変更します。
- F. 7行目を変更してnames.clone ()を返します。
- G. 2行目を保護された揮発性のString []名に変更します;。

Answer: B,E,G (メッセージを残す)

最新問題: 37

与えられた:

```

1 public class DNASynth {
2     int aCount;
3     int tCount;
4     int cCount;
5     int gCount;
6
7     DNASynth(int a, int tCount, int c, int g){
8         // line 1
9     }
10    int setCCount(int c){
11        return c;
12    }
13    void setGCount(int gCount){
14        this.gCount = gCount;
15    }
16 }

```

1行目に挿入されたときにインスタンス変数を正しく変更する2行のコードはどれですか？ 2つ選択してください。）

- A. tCount = tCount;
- B. cCount = setCCount (c);

C. setCCount (c)= cCount;

D. setGCount (g);

E. aCount = a;

Answer: A,E ([メッセージを残す](#))

最新問題: 38

与えられた :

```
public class Main {  
    class Student { // line 1  
        String classname;  
        Student(String classname) { // line 2  
            this.classname = classname;  
        }  
    }  
    public static void main(String[] args) {  
        var student = new Student("Biology"); // line 3  
    }  
}
```

Mainクラスをコンパイルする2つの独立した変更はどれですか？ 2つ選択してください。)

A. Studentクラス宣言全体を別のJavaファイルStudent.javaに移動します。

B. 2行目をpublic Student (String classname)に変更します。

C. 1行目をパブリッククラスのStudent{に変更します。

D. 3行目をStudent student = new Student ("Biology");に変更します。

E. 1行目を静的クラスStudent{に変更します。

Answer: B,D ([メッセージを残す](#))

ORACLE

```
1 import java.util.*;
2 import java.io.*;
3 import java.lang.Thread;
4 import java.util.ArrayList;
5 import java.util.LinkedList;
6 import java.util.List;
7 import java.util.function.Consumer;
8 import java.util.stream.Stream;
9 import java.util.stream.IntStream;
10 import java.util.Optional;
11
12
13 public class Main {
14     class Student {
15         String classname;
16         public Student (String classname) {
17             this.classname = classname;
18         }
19
20     }
21     public static void main (String[] args) {
22         var student = new Student ("Biology");
23     }
24 }
```

最新問題: 39

与えられた :

```

import java.io.*;
public class Tester {
    public static void main(String[] args) {
        try {
            doA();
            doB();
        } catch(IOException e) {
            System.out.print("c");
            return;
        } finally{
            System.out.print("d");
        }
        System.out.print("f");
    }
    private static void doA() {
        System.out.print("a");
        if (false) {
            throw new IndexOutOfBoundsException();
        }
    }
    private static void doB() throws FileNotFoundException {
        System.out.print("b");
        if (true) {
            throw new FileNotFoundException();
        }
    }
}

```

結果はどうなりますか？

- A. コンパイルは失敗します。
- B. adf
- C. abdf
- D. abcd
- E. abd

Answer: D ([メッセージを残す](#))

最新問題: 40

与えられた：

```

enum QUALITY {
    A(100), B(75), C(50);
    int percent;
    private QUALITY(int percent) {
        this.percent = percent;
    }
}

```

と

checkQuality (QUALITY.A);

と

```

void checkQuality(QUALITY q) {
    switch (q) {
        case /* Insert code here */ :
            System.out.println("Best");
            break;
        default :
            System.out.println("Not best");
            break;
    }
}

```

Bestを出力するためにswitchステートメントに挿入できるコードフラグメントはどれですか？

- A. A
- B. A.toString ()
- C. QUALITY.A
- D. QUALITY.A.ValueOf ()

Answer: ([解答を表示する](#))

最新問題: 41

与えられた：

```

1. public class Test {
2.     private static class Greet {
3.         private void print() {
4.             System.out.println("Hello World");
5.         }
6.     }
7.     public static void main(String[] args) {
8.         Test.Greet i = new Greet();
9.         i.print();
10.    }
11. }

```

結果はどうなりますか？

- A. コンパイルは9行目で失敗します。
- B. コンパイルは2行目で失敗します。
- C. Hello World
- D. コンパイルは8行目で失敗します。

Answer: ([解答を表示する](#))

```
1 public class Test {
2     private static class Greet {
3         private void print() {
4             System.out.println("Hello World");
5         }
6     }
7     public static void main(String[] args) {
8         Test.Greet i = new Greet();
9         i.print();
10    }
11 }
```

Execute Mode, Version, Inputs & Arguments

JDK 11.0.4

CommandLine Arguments

Result

CPU Time: 0.16 sec(s), Memory: 32504 kilobyte(s)

Hello World

最新問題: 42

与えられた :

```
public static void doThings() throws GeneralException
{
    try {
        throw new RuntimeException("Someting happened");
    } catch (Exception e) {
        throw new SpecificException(e.getMessage());
    }
}

public static void main(String args[]) {
    try{
        Test.doThings();
    } catch (Exception e) {
        System.out.println(e.getMessage());
    }
}

class GeneralException /* line 1 */ {
    public GeneralException(String s) { super(s); }
}

class SpecificException /* line 2 */ {
    public SpecificException(String s) { super(s); }
}
```

コードが何かを印刷できるようにするには、どのオプションを選択する必要がありますか？

A. 1行目にextendsGeneralExceptionを追加します。

2行目にextendsExceptionを追加します。

B. 1行目にextendsSpecificExceptionを追加します。

2行目にGeneralExceptionを追加します。

C. 1行目にextendsExceptionを追加します。

2行目にextendsExceptionを追加します。

D. 1行目にextendsExceptionを追加します。

2行目にGeneralExceptionを追加します。

Answer: D (メッセージを残す)

```
1 import java.util.*;
2 import java.io.*;
3 import java.lang.Thread;
4 import java.util.ArrayList;
5 import java.util.LinkedList;
6 import java.util.List;
7
8 public class Test {
9
10     public static void doThings() throws GeneralException {
11         try{
12             throw new RuntimeException("Something happened");
13         } catch (Exception e) {
14             throw new SpecificException (e.getMessage());
15         }
16     }
17 }
18
19 public static void main(String args[]) {
20     try{
21         Test.doThings();
22     } catch (Exception e) {
23         System.out.println(e.getMessage());
24     }
25 }
26 class GeneralException extends Exception {
27     public GeneralException(String s) { super(s); }
28 }
29 class SpecificException extends GeneralException {
30     public SpecificException(String s) { super(s); }
31 }
32 }
```

最新問題: 43

与えられた :

```
public class Foo {
    public <T> Collection<T> foo(Collection<T> arg) { ... }
}

and

public class Bar extends Foo { ... }
```

メソッドがBarに追加された場合、正しい2つのステートメントはどれですか。2つ選択してください。)

- A. `public <T> List <T> foo Collection <T> arg){...}`はFoo.fooをオーバーライドします。
- B. `public <T> Collection <T> foo Collection <T> arg){...}`はFoo.fooをオーバーロードします。
- C. `public <T> Collection <T> foo Stream <T> arg){...}`はFoo.fooをオーバーロードします。
- D. `public <T> Iterable <T> foo Collection <T> arg){...}`はFoo.fooをオーバーライドします。
- E. `public <T> Collection <T> bar Collection <T> arg){...}`はFoo.fooをオーバーロードします。
- F. `public Collection <String> foo Collection <String> arg){...}`はFoo.fooをオーバーライドします。

Answer: ([解答を表示する](#))

最新問題: 44

与えられた:

```
public class Test{
    private int num = 1;
    private int div = 0;
    public void divide() {
        try {
            num = num / div;
            System.out.print("Exception");
        }
        catch(ArithmeticException ae) { num = 100;
        catch(Exception e) { num = 200; }
        finally { num = 300; }
        System.out.print(num);
    }
    public static void main(String args[])
    {
        Test test = new Test();
        test.divide();
    }
}
```

出力は何ですか？

- A. 300
- B. 例外
- C. 200
- D. 100

Answer: A ([メッセージを残す](#))

```

1 public class Test{
2     private int num = 1;
3     private int div = 0;
4
5     public void divide() {
6         try {
7             num = num / div;
8             System.out.print("Exception");
9         }
10        catch(ArithmeticException ae) { num = 100; }
11        catch(Exception e) { num = 200; }
12        finally { num = 300; }
13        System.out.print(num);
14    }
15    public static void main(String args[])
16    {
17        Test test = new Test();
18        test.divide();
19    }
20 }

```

Execute Mode, Version, Inputs & Arguments

JDK 11.0.4

CommandLine Arguments

Result

CPU Time: 0.15 sec(s), Memory: 32484 kilobyte(s)

300

最新問題: 45

与えられた :

```

class CustomType<T> {
    public <T> int count(T[] anArray, T element) {
        int count = 0;
        for(T e : anArray){
            if (e.equals(element)) ++count;
        }
        return count;
    }
}

```

と

```

public class Test extends CustomType {
    public static void main(String[] args) {
        String[] words = {"banana", "orange", "apple", "lemon"};
        Integer[] numbers = {1, 2, 3, 4, 5};
        CustomType type = new CustomType();
        CustomType<String> stringType = new CustomType<>();
        System.out.println(stringType.count(words, "apple"));
        System.out.println(type.count(words, "apple"));
        System.out.println(type.count(numbers, 3));
    }
}

```

結果はどうなりますか？

A. 実行時にNullPointerExceptionがスローされます。

B. コンパイルは失敗します。

C. 1

ヌル

ヌル

D. 1

1

1

E. 実行時にClassCastExceptionがスローされます。

Answer: B (メッセージを残す)

```

Console 4
Error: Could not find or load main class CustomType
Caused by: java.lang.ClassNotFoundException: CustomType

```

最新問題: 46

与えられた：

```

public static void main(String[] args)
{
    char letter = 'b';
    int i = 0;
    switch(letter) {
        case 'a':
            i++;
            break;
        case 'b':
            i++;
        case 'c' | 'd': // line 1
            i++;
        case 'e':
            i++;
            break;
        case 'f':
            i++;
            break;
        default:
            System.out.print(letter);
    }
    System.out.println(i);
}

```

結果はどうなりますか？

- A. b1
- B. 2
- C. b2
- D. 1
- E. b3
- F. 3
- G. 1行目のエラーが原因でコンパイルが失敗します。

Answer: F ([メッセージを残す](#))

Result

CPU Time: 0.23 sec(s), Memory: 32708 kilobyte(s)

3

有効な **1Z0-819** 問題集は GoShiken.com が提供された合格しやすい 1Z0-819 試験問題集！ GoShiken.com が最新の **1Z0-819** 試験問題集を提供しています。GoShiken.com 1Z0-819 試験問題は最新で、解答が正確でございます。最新の GoShiken.com 1Z0-819 問題集をゲットする人はこちら: <https://www.goshiken.com/Oracle/1Z0-819-mondaishu.html> (**29730%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: 47

与えられた :

```
LocalDate d1 = LocalDate.of (1997,2,7);
```

```
DateTimeFormatter dtf
```

```
DateTimeFormatter.ofPattern (*ここにコードを挿入*);
```

```
System.out.println (dtf.format (d1) )
```

1997年2月7日金曜日の日付をフォーマットするパターンはどれですか？

A. "eeee dd +" th of "+ MMM yyyy"

B. MMMMyyyyのeeeeed'th」

C. "eeee d +" th of "+ MMMM yyyy"

D. MMMyyyyのeeeeed'th」

Answer: D ([メッセージを残す](#))

最新問題: 48

与えられた :

```
public class X {  
    private Collection collection;  
    public void set(Collection collection) {  
        this.collection = collection;  
    }  
}
```

and

```
public class Y extends X {  
    public void set(Map<String,String> map) {  
        super.set(map); // line 1  
    }  
}
```

Yクラスがコンパイルされるように1行目を置き換えることができる2行はどれですか？ 2つ選択してください。)

A. map.forEach (k(v)-> set (v));)

B. set (map)

C. super.set (map.values ());)

D. super.set (List <String> map)

E. set (map.values ());)

Answer: ([解答を表示する](#))

最新問題: 49

与えられた :

```

var numbers = List.of(1,2,3,4,5,6,7,8,9,10);
// line 1
StringBuilder sb = new StringBuilder();
for(int a: numbers) {
    sb.append(f.apply(a));
    sb.append(" ");
}
System.out.println(sb.toString());

```

1行目のどのステートメントがこのコードのコンパイルを可能にしますか？

- A. 関数<整数、整数> f = n> n * 2;
- B. 関数<整数> f = n> n * 2;
- C. Function <int> f = n> n * 2;
- D. Function <int、int> f = n> n * 2;
- E. 関数f = n> n * 2;

Answer: ([解答を表示する](#))

```

15
16 public class Main {
17     public static void main(String[] args) {
18         var numbers = List.of(1,2,3,4,5,6,7,8,9,10);
19         Function<Integer, Integer> f = n -> n * 2;
20         StringBuilder sb = new StringBuilder();
21         for(int a: numbers) {
22             sb.append(f.apply(a));
23             sb.append(" ");
24         }
25         System.out.println(sb.toString());
26     }
27 }
28
Result
CPU Time: 0.22 sec(s), Memory: 33056 kilobyte(s)
2 4 6 8 10 12 14 16 18 20

```

最新問題: 50

与えられた：

```

public class Test {
    private String[] strings;
}

```

クラスフィールド文字列をコンパイルして設定する2つのコンストラクターはどれですか？ 2つ選択してください。）

A.

```
public Test(List<String> strings) {  
    this.strings = strings;  
}
```

B.

```
public Test(String... strings) {  
    strings = strings;  
}
```

C.

```
public Test(String... strings) {  
    this.strings = strings;  
}
```

D. **ORACLE**

```
public Test(String strings) {  
    strings = strings;  
}
```

E.

```
public Test(String[] strings) {  
    this.strings = strings;  
}
```

A. オプションA

B. オプションD

C. オプションE

D. オプションC

E. オプションB

Answer: C,D ([メッセージを残す](#))

最新問題: 51

与えられた :

```

List<Reader> dataFiles = new ArrayList<>();
File indexFile = new File("MyIndex.idx");
try (BufferedReader indexReader =
    new BufferedReader(new FileReader(indexFile))) {
    for(String file = indexReader.readLine(); file != null;
        file = indexReader.readLine()) {
        BufferedReader dataReader = new BufferedReader (
            new FileReader(new File(file))); // Line 1
        dataFiles.add(dataReader); // Line 2
        processData(dataReader); // Line 3
    }
} catch (IOException ex) {
    ...
} finally {
    for(Reader r : dataFiles) {
        try {
            r.close();
        } catch (IOException ex) {
            ...
        } // Line 4
    }
}

```

潜在的なサービス拒否状態からこのコードを保護するものは何ですか？

- A. 3行目の後に、dataReader.close（を追加します。
- B. 1行目の前で、dataFilesのサイズをチェックして、しきい値を超えていないことを確認します。
- C. 3行目で、processData（dataReader）をtrywithresourcesで囲みます。
- D. 1行目で、各dataReaderを開くときにリソースでtryを使用します。
- E. 4行目の後に、indexReader.close（を追加します。

Answer: C ([メッセージを残す](#))

最新問題: 52

有効な注釈の使用法はどれですか？ 3つ選択してください。）

- A. var obj = new @Interned MyObject（）
- B. Function <String, String> func = @NonNull var x> x.toUpperCase（）
- C. var v = "Hello" + @Interned）"World"
- D. Function <String, String> func = @NonNull x> x.toUpperCase（）
- E. Function <String, String> func = var @NonNull x> x.toUpperCase（）
- F. var myString = @NonNull String）str;

Answer: A,D,E ([メッセージを残す](#))

最新問題: 53

与えられた：

```
public class Foo {
    public <T> Collection<T> foo(Collection<T> arg) { ... }
}
```

and

```
public class Bar extends Foo { ...
```

メソッドがBarに追加された場合、正しい2つのステートメントはどれですか。2つ選択してください。）

- A. `public <T> Collection <T> foo Collection <T> arg){...}`はFoo.fooをオーバーロードします。
- B. `public Collection <String> foo Collection <String> arg){...}`はFoo.fooをオーバーライドします。
- C. `public <T> Collection <T> foo $Stream <T> arg){...}`はFoo.fooをオーバーロードします。
- D. `public <T> Iterable <T> foo Collection <T> arg){...}`はFoo.fooをオーバーライドします。
- E. `public <T> List <T> foo Collection <T> arg){...}`はFoo.fooをオーバーライドします。
- F. `public <T> Collection <T> bar Collection <T> arg){...}`はFoo.fooをオーバーロードします。

Answer: D,E ([メッセージを残す](#))

最新問題: 54

与えられた：

```
void myLambda(){
    int i = 25;
    Supplier<Integer> foo = () -> i;
    i++;
    System.out.println(foo.get());
}
```

どちらが正しいですか？

- A. コードは25を出力します。
- B. コードはコンパイルされません。
- C. コードはコンパイルされますが、結果は出力されません。
- D. コードは実行時に例外をスローします。

Answer: ([解答を表示する](#)**)**

最新問題: 55

与えられた：

```

public class Tester {
    public static void main(String[] args) {
        char letter = 'b';
        int i = 0;
        switch(letter) {
            case 'a':
                i++;
                break;
            case 'b':
                i++;
            case 'c' | 'd': // line 1
                i++;
            case 'e':
                i++;
                break;
            case 'f':
                i++;
                break;
            default:
                System.out.print(letter);
        }
        System.out.println(i);
    }
}

```

結果はどうなりますか？

- A. b1
- B. 2
- C. b2
- D. 1
- E. b3
- F. 3
- G. 1行目のエラーが原因でコンパイルが失敗します。

Answer: F ([メッセージを残す](#))

Result

CPU Time: 0.23 sec(s), Memory: 32708 kilobyte(s)

3

最新問題: 56

与えられた：

```

public class Main {

    public static void checkConfiguration(String filename) {
        File file = new File(filename);
        if(!file.exists()) {
            throw new Error("Fatal Error: Configuration File, "
                + filename + ", is missing.");
        }
    }

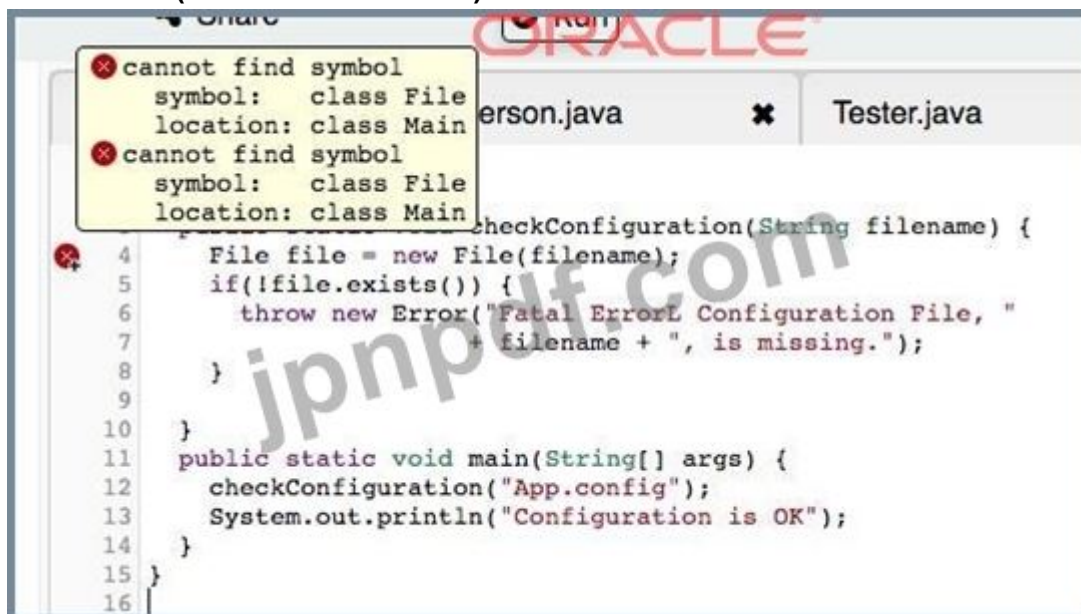
    public static void main(String[] args) {
        checkConfiguration("App.config");
        System.out.println("Configuration is OK");
    }
}

```

ファイル「App.config」が見つからない場合、結果はどうなりますか？

- A. 設定はOKです
- B. コンパイルは失敗します。
- C. スレッド "main" の例外 java.lang.Error : Fatalエラー：構成ファイル App.config がありません。
- D. 何もない

Answer: B ([メッセージを残す](#))



最新問題: 57

コードフラグメントが与えられた場合：

```
String s = "";
if (Double.parseDouble("11.00f") > 11) {
    s += 1;
}
if (1_7 == Integer.valueOf("17")) {
    s += 2;
}
if (1024 > 1023L) {
    s += 3;
}
System.out.print(s);
```

結果はどうなりますか？

- A. 23
- B. 12
- C. 123
- D. 13

Answer: A ([メッセージを残す](#))



最新問題: 58

与えられた：

```
public class Main {
    public static void main(String[] args) {
        Thread t1 = new Thread(new MyThread());
        Thread t2 = new Thread(new MyThread());
        Thread t3 = new Thread(new MyThread());

        t1.start();
        t2.run();
        t3.start();

        t1.start();
    }
}

class MyThread implements Runnable {
    public void run() {
        System.out.println("Running.");
    }
}
```

どちらが正しいですか？

- A. 実行時にIllegalThreadStateExceptionがスローされます。
- B. 3つのスレッドが作成されます。
- C. コンパイルは失敗します。
- D. 4つのスレッドが作成されます。

Answer: ([解答を表示する](#))



```
Running.
Running.
Running.

Exception in thread "main" java.lang.IllegalThreadStateException
    at java.base/java.lang.Thread.start(Thread.java:794)
    at Main.main(Main.java:12)
```

最新問題: 59

与えられた：

```
public class Main {
    private String[] strings = {"ABCDEFGHJKLMNOPQRSTUVWXYZ",
                                "abcdefghijklmnopqrstuvwxyz", "0123456789"};
    public void write(String filename) {
        // line 1
        for (String str: strings) {
            ByteBuffer buffer = ByteBuffer.wrap(str.getBytes());
            fileChannel.write(buffer);
        }
    } catch (IOException e) {
        e.printStackTrace();
    }
}
public static void main(String[] args) {
    Main test = new Main();
    test.write("file_to_path");
}
```

1行目でFilechannelオブジェクトを取得したいとします。

どのコードフラグメントがこれを達成しますか？

A)

```
try (FileChannel fineChannel = Channels.newChannel(new FileOutputStream(filename));) {
```

B)

```
try(FileChannel fileChannel = new FileOutputStream(filename).getChannel();) {
```

C)

```
try (FileChannel fileChannel = new FileOutputStream(new FileChannel(filename));) {
```

D)

```
try(FileChannel fileChannel = new FileChannel(new FileOutputStream(filename));) {
```

A. オプションD

B. オプションB

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: 60

与えられた :

```
1. public class Test {  
2.     private static class Greet {  
3.         private void print() {  
4.             System.out.println("Hello World");  
5.         }  
6.     }  
7.     public static void main(String[] args) {  
8.         Test.Greet i = new Greet();  
9.         i.print();  
10.    }  
11. }
```

結果はどうなりますか？

A. コンパイルは9行目で失敗します。

B. コンパイルは2行目で失敗します。

C. Hello World

D. コンパイルは8行目で失敗します。

Answer: ([解答を表示する](#))

```
1- public class Test {
2-     private static class Greet {
3-         private void print() {
4-             System.out.println("Hello World");
5-         }
6-     }
7-     public static void main(String[] args) {
8-         Test.Greet i = new Greet();
9-         i.print();
10-    }
11- }
```

ORACLE

Execute Mode, Version, Inputs & Arguments

JDK 11.0.4

CommandLine Arguments

Result

CPU Time: 0.16 sec(s), Memory: 32504 kilobyte(s)

Hello World

最新問題: 61

どのコードフラグメントがコンパイルされますか？

```

1 Comparator comparator = new Comparator<?>() {
2     public int compare(Integer i, Integer j) {
3         return i.compareTo(j);
4     }
5 };
6
7 var comparator = new Comparator<>() {
8     public int compare(Integer i, Integer j) {
9         return i.compareTo(j);
10    }
11 };
12
13 Comparator<> comparator = new Comparator<Integer>() {
14     public int compare(Integer i, Integer j) {
15         return i.compareTo(j);
16    }
17 };
18
19 Comparator<Integer> comparator = new Comparator<>() {
20     public int compare(Integer i, Integer j) {
21         return i.compareTo(j);
22    }
23 };

```

- 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

Answer: D (メッセージを残す)

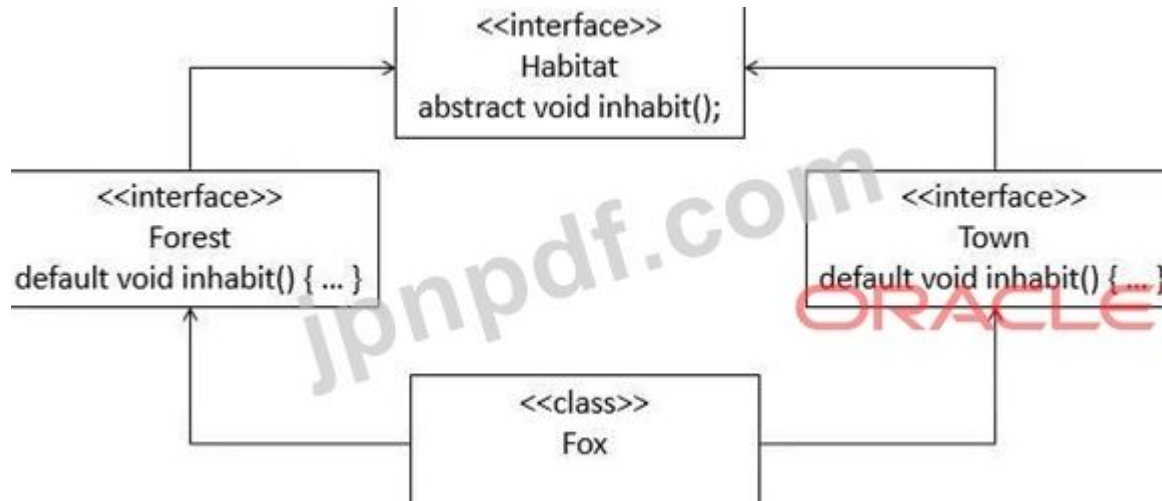
```

1 import java.io.*;
2 import java.util.*;
3 class abc {
4     public static void main(String[] args) {
5
6         Comparator<Integer> comparator = new Comparator<>() {
7             public int compare(Integer i, Integer j) {
8                 return i.compareTo(j);
9             }
10        };
11    }
12 }
13 }
14

```

有効な **1Z0-819** 問題集は GoShiken.com が提供された合格しやすい 1Z0-819 試験問題集！ GoShiken.com が最新の **1Z0-819** 試験問題集を提供しています。GoShiken.com 1Z0-819 試験問題は最新で、解答が正確でございます。最新の GoShiken.com 1Z0-819 問題集をゲットする人はこちら: <https://www.goshiken.com/Oracle/1Z0-819-mondaishu.html> (29730%OFF問題集溶と正解付きで 30%w 特別割引コード: **Freepdfdumps**)

与えられた：



Foxクラスについて正しい説明はどれですか。

- A. Foxクラスは、呼び出しを試みない限り、inhabitメソッドをオーバーライドする必要はありません。
- B. Foxクラスは、ForestまたはTownインターフェースのいずれかを実装する必要がありますが、両方を実装することはできません。
- C. ForestとTownが互換性のある実装を提供している場合、Foxクラスはinhabitメソッドをオーバーライドする必要はありません。
- D. Foxが実装する最初のインターフェースからのinhabitメソッドの実装が優先されます。
- E. Foxクラスは、inhabitメソッドの実装を提供する必要があります。

Answer: C ([メッセージを残す](#))

最新問題: 63

コードフラグメントが与えられた場合：

```
int x = 0;
do {
    x++;
    if (x == 1) {
        continue;
    }
    System.out.println(x);
} while(x < 1);
```

結果はどうなりますか？

- A. プログラムは何も出力しません。
- B. 0
- C. 無限ループで1を出力します。
- D. 1
- E. 01

Answer: ([解答を表示する](#)**)**

最新問題: 64

与えられた：

```

public class Main {
    public static void main(String[] args) {
        Thread t1 = new Thread(new MyThread());
        Thread t2 = new Thread(new MyThread());
        Thread t3 = new Thread(new MyThread());

        t1.start();
        t2.run();
        t3.start();

        t1.start();
    }
}

class MyThread implements Runnable {
    public void run() {
        System.out.println("Running.");
    }
}

```

どちらが正しいですか？

- A. 実行時にIllegalThreadStateExceptionがスローされます。
- B. 3つのスレッドが作成されます。
- C. コンパイルは失敗します。
- D. 4つのスレッドが作成されます。

Answer: A ([メッセージを残す](#))



```

Running.
Running.
Running.

Exception in thread "main" java.lang.IllegalThreadStateException
    at java.base/java.lang.Thread.start(Thread.java:794)
    at Main.main(Main.java:12)

```

最新問題: 65

TripleThis.javaが与えられた場合：

```

6. import java.util.function.*;
7. public class TripleThis {
8.     public static void main(String[] args) {
9.         Function tripler = x -> { return (Integer) x * 3; };
10.        TripleThis.printValue(tripler, 4);
11.    }
12.    public static <T> void printValue(Function f, T num) {
13.        System.out.println(f.apply(num));
14.    }
15. }

```

TripleThis.javaをコンパイルすると、このコンパイラに警告が表示されます。

注 :TripleThis.javaは、チェックされていない、または安全でない操作を使用します。
 一緒に行われた2つの置換のうち、このコンパイラの警告を削除するのはどれですか？

A. 9行目をfunction <Integer> tripler = x-> {return Integer.parseInt(x)*3;}に置き換えます。}

B. 9行目をfunction <Integer> tripler = x-> {return Integer.parseInt(x)*3;}に置き換えます。}

C. 12行目をpublic static <T> void printValue Function <T, T> f, T num){、

D. 12行目をpublic static void printValue function <Integer> f, int num){に置き換えます。}

E. 12行目をpublic static int printValue function <Integer, Integer> f, T num){に置き換えます。}

Answer: B,E ([メッセージを残す](#))

最新問題: 66

与えられた：

```
class Super {
    final int num; // line n1
    public Super(int num) {
        this.num = num;
    }
    final void method() {
        System.out.println("Output from Super");
    }
}
class Sub extends Super {
    int num; // line n2
    Sub(short num) { // line n3
        super(num);
    }
    protected void method() { // line n4
        System.out.println("Output from Sub");
    }
}
```

コードのどの行でコンパイルエラーが発生しますか？

- A. 行n1
- B. 行n3
- C. 行n2
- D. 行n4

Answer: D ([メッセージを残す](#))

```
1
2 class Sub extends Super {
3     int num; //line n2
4     Sub(short num) { //line n3
5         super(num);
6     }
7     protected void method() { //line n4
8         System.out.println("Output from Sub");
9     }
10 }
11
```

最新問題: 67

与えられた：

```
import java.util.List;
import java.util.Optional;
public class Test {
    public static void main(String[] args) {
        var items = List.of(new Item("A", 10), new Item("B", 2),
                             new Item("C", 12), new Item("D", 5),
                             new Item("E", 6));
        double avg = items.stream().mapToInt(i -> i.amount).average().orElse(0.0);
        Optional<Item> item = items.parallelStream()
                                   .filter(i -> i.amount < avg).findAny();
        System.out.println(item.orElseThrow());
    }
}
class Item {
    public String name; public int amount;
    public Item(String name, int amount) {
        this.name = name; this.amount = amount;
    }
    @Override
    public String toString() { return "Name: " + name + ", Amount: " + amount; }
}
```

何が本当？

- A. 実行時にNoSuchElementExceptionがスローされます。
- B. コンパイルは失敗します。
- C. これは、プログラムが実行されるたびに同じ結果を出力するはずです。
- D. プログラムを実行するたびに同じ結果が出力されない場合があります。

Answer: D ([メッセージを残す](#))

```

16
17- public class Test {
18-     public static void main (String[] args) {
19-         var items = List.of(new Item("A", 10), new Item("B", 2),
20-                             new Item("C", 12), new Item("D", 5),
21-                             new Item("E", 6));
22-         double avg = items.stream().mapToInt(i -> i.amount). average().orElse(0.0);
23-         Optional<Item> item = items.parallelStream()
24-                                 .filter(i -> i.amount < avg).findAny();
25-         System.out.println(item.orElseThrow());
26-     }
27- }
28
29- class Item {
30-     public String name; public int amount;
31-     public Item(String name, int amount) {
32-         this.name = name; this.amount = amount;
33-     }
34-     @Override
35-     public String toString() { return "Name: " + name + ", Amount: " + amount; }
36- }

```

Execute Mode, Version, Inputs & Arguments

JDK 11.0.4



Interactive

CommandLine Arguments



Execute

Result

CPU Time: 0.26 sec(s), Memory: 39112 kilobyte(s)

Name: B, Amount: 2

ORACLE

```

17 public class Test {
18     public static void main (String[] args) {
19         var items = List.of(new Item("A", 10), new Item("B", 2),
20                             new Item("C", 12), new Item("D", 5),
21                             new Item("E", 6));
22         double avg = items.stream().mapToInt(i -> i.amount). average().orElse(0.0);
23         Optional<Item> item = items.parallelStream()
24             .filter(i -> i.amount < avg).findAny();
25         System.out.println(item.orElseThrow());
26     }
27 }
28
29 class Item {
30     public String name; public int amount;
31     public Item(String name, int amount) {
32         this.name = name; this.amount = amount;
33     }
34     @Override
35     public String toString() { return "Name: " + name + ", Amount: " + amount; }
36 }

```

Execute Mode, Version, Inputs & Arguments

JDK 11.0.4
Interactive

CommandLine Arguments

Execute

Result

CPU Time: 0.33 sec(s), Memory: 38888 kilobyte(s)

Name: E, Amount: 6

最新問題: 68

書店の売り上げは、顧客の名前と購入した本が入力された販売オブジェクトのリストで表されます。

パブリッククラスセール{

 プライベート文字列の顧客。

 プライベートリスト<Book>アイテム;

//コンストラクター、セッター、ゲッターは表示されません

}

パブリッククラスブック{

 プライベート文字列名。

 プライベートダブル価格;

//コンストラクター、セッター、ゲッターは表示されません

}

SaleオブジェクトのリストtListが与えられた場合、どのコードフラグメントが各顧客の総売上のリストを昇順で作成しますか？

```

A. List<String> totalByUser = tList.stream()
    .collect(flatMapping(t -> t.getItems().stream(),
        groupingBy(Sale::getCustomer,
            summingDouble(Book::getPrice))))
    .entrySet().stream()
    .sorted(Comparator.comparing(Entry::getValue))
    .collect(mapping(e -> e.getKey() + ":" + e.getValue(),toList()));

B. List<String> totalByUser = tList.stream()
    .collect(groupingBy(Sale::getCustomer,
        flatMapping(t -> t.getItems().stream(),
            summingDouble(Book::getPrice))))
    .sorted(Comparator.comparing(Entry::getValue))
    .collect(mapping(e -> e.getKey() + ":" + e.getValue(),toList()));

C. List<String> totalByUser = tList.stream()
    .collect(groupingBy(Sale::getCustomer,
        flatMapping(t -> t.getItems().stream(),
            summingDouble(Book::getPrice))))
    .entrySet().stream()
    .sorted(Comparator.comparing(Entry::getValue))
    .collect(mapping(e -> e.getKey() + ":" + e.getValue(),toList()));

D. List<String> totalByUser = tList.stream()
    .collect(flatMapping(t -> t.getItems().stream(),
        groupingBy(Sale::getCustomer,
            summingDouble(Book::getPrice))))
    .sorted(Comparator.comparing(Entry::getValue))
    .collect(mapping(e -> e.getKey() + ":" + e.getValue(),toList()));

```

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

Answer: D ([メッセージを残す](#))

最新問題: 69

内容を見ると :

```
MessagesBundle.properties file:  
username = Username  
password = Password
```

ORACLE®

and

```
MessagesBundle_fr_FR.properties file:  
username = Utilisateur  
password = Le passe
```

and

```
MessagesBundle_ru.properties file:  
username = Пользователь  
password = Пароль
```

and the code fragment:

```
public class Test {  
    public static void main(String[] args) {  
        Locale.setDefault(Locale.FRANCE);  
        ResourceBundle msg = ResourceBundle.getBundle("MessageBundle", new Locale("ru"))  
        System.out.println("User " + msg.getString("username"));  
        System.out.println("Pass " + msg.getString("password"));  
    }  
}
```

結果はどうなりますか？

A)

```
User = Пользователь  
Pass = Пароль
```

B)

The compilation fails.

C)

A `MissingResourceException` is thrown at runtime.

D)

```
User = Utilisateur  
Pass = Le passe
```

E)

ユーザーユーザー名

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

E. オプション

Answer: B,E ([メッセージを残す](#))

最新問題: 70

与えられた :

```

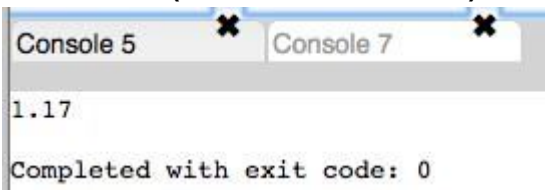
public class Tester {
    public static void main(String[] args) {
        byte x = 7, y = 6;
        // line 1
        System.out.println(z);
    }
}

```

1行目に追加したときに1.17の出力を生成する式はどれですか？

- A. float z = (float) Math.round ((float)x / y * 100) / 100);
- B. float z = Math.round ((int) x / y, 2);
- C. float z = Math.round ((float)x / y, 2);
- D. float z = Math.round ((float)x / y * 100) / (float) 100;

Answer: D (メッセージを残す)



Console 5 Console 7

1.17

Completed with exit code: 0

最新問題: 71

与えられた：

```

public class Foo {
    public void foo(Collection arg) {
        System.out.println("Bonjour le monde!");
    }
}

and

public class Bar extends Foo {
    public void foo(Collection arg) {
        System.out.println("Hello world!");
    }
    public void foo(List arg) {
        System.out.println("Olá Mundo!");
    }
}

and

Foo f1 = new Foo();
Foo f2 = new Bar();
Bar b1 = new Bar();
Collection<String> c = new ArrayList<>();

```

正しいのはどれですか？ 3つ選択してください。）

- A. b1.foo ()はBonjour le mondeを出力します！

- B. b1.foo ()はHello world！を出力します
C. f2.foo ()はBonjour le mondeを出力します！
D. f2.foo ()はHello world！を出力します
E. f1.foo ()はOla Mundoを出力します！
F. b1.foo ()はOla Mundoを出力します！
G. f1.foo ()はBonjour le mondeを出力します！
H. f1.foo ()はHello world！を出力します。
I. f2.foo ()はOla Mundoを出力します！

Answer: C,F,H (メッセージを残す)

最新問題: 72

機能的なインターフェイスと見なされる2つのインターフェイスはどれですか？ 2つ選択してください。)

@FunctionalInterface

A. インターフェイスInterfaceC {
public boolean equals (Object o);
intbreed (int x);
intcalculate (int x, int y);
}

@FunctionalInterface

B. インターフェイスInterfaceD {
intbreed (int x);
}

@FunctionalInterface

C. インターフェイスInterfaceE {
public boolean equals (int i);
intbreed (int x);
}

インターフェイスInterfaceA{

D. int GERM = 13;
public default int getGERM () {return GERM; }
}

インターフェイスInterfaceB{

E. int GERM = 13;
public default int getGERM () {return get ();}
private int get () {return GERM; }
public boolean equals (Object o);
intbreed (int x);
}

Answer: A,C (メッセージを残す)

説明/参照 :

最新問題: 73

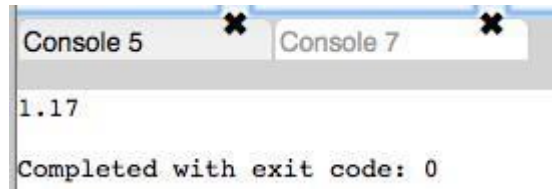
与えられた :

```
public class Tester {  
    public static void main(String[] args) {  
        byte x = 7, y = 6;  
        // line 1  
        System.out.println(z);  
    }  
}
```

1行目に追加したときに1.17の出力を生成する式はどれですか？

- A. float z = (float) Math.round ((float)x / y * 100) / 100);
- B. float z = Math.round ((int) x / y, 2);
- C. float z = Math.round ((float)x / y, 2);
- D. float z = Math.round ((float)x / y * 100) / (float) 100;

Answer: D (メッセージを残す)



The screenshot shows a Java IDE console window with two tabs: 'Console 5' and 'Console 7'. The 'Console 7' tab is active and displays the output '1.17'. Below the output, it says 'Completed with exit code: 0'.

最新問題: 74

与えられた :

```

import java.io.*;
public class Tester {
    public static void main(String[] args) {
        try {
            doA();
            doB();
        } catch (IOException e) {
            System.out.print("c");
            return;
        } finally{
            System.out.print("d");
        }
        System.out.print("f");
    }
    private static void doA() {
        System.out.print("a");
        if (false) {
            throw new IndexOutOfBoundsException();
        }
    }
    private static void doB() throws FileNotFoundException {
        System.out.print("b");
        if (true) {
            throw new FileNotFoundException();
        }
    }
}

```

結果はどうなりますか？

- A. abcd
- B. コンパイルは失敗します。
- C. adf
- D. abd
- E. abdf

Answer: A ([メッセージを残す](#))

最新問題: 75

与えられた :

```
import java.io.*;
public class Tester {
    public static void main(String[] args) {
        try {
            doA();
            doB();
        } catch(IOException e) {
            System.out.print("c");
            return;
        } finally{
            System.out.print("d");
        }
        System.out.print("f");
    }
    private static void doA() {
        System.out.print("a");
        if (false) {
            throw new IndexOutOfBoundsException();
        }
    }
    private static void doB() throws FileNotFoundException
        System.out.print("b");
        if (true) {
            throw new FileNotFoundException();
        }
    }
}
```

ORACLE®

結果はどうなりますか？

- A. adf
- B. abcd
- C. abdf
- D. コンパイルは失敗します。
- E. abd

Answer: B ([メッセージを残す](#))

最新問題: 76

これらのモジュール宣言を調べてください。

```

module ServiceAPI {
    exports com.example.api;
}

module ServiceProvider {
    requires ServiceAPI;
    provides com.example.api with com.myimpl.Impl
}

module Consumer {
    requires ServiceAPI;
    uses com.example.api;
}

```

正しい2つのステートメントはどれですか？ 2つ選択してください。）

- A. コンシューマーモジュールにはServiceProviderモジュールが必要です。
- B. ServiceProviderモジュールは、実行時にcom.example.api APIを提供できる唯一のモジュールです。
- C. ServiceProviderモジュールはcom.myimplパッケージをエクスポートする必要があります。
- D. ServiceProviderモジュールは、com.example.api APIを使用するモジュール (Consumerなど)のIDを認識していません。
- E. com.example.api APIを別のモジュールServiceAPIに配置すると、複数のプロバイダーモジュールを簡単にインストールできます。

Answer: A,B (メッセージを残す)

有効な **1Z0-819** 問題集は GoShiken.com が提供された合格しやすい 1Z0-819 試験問題集！ GoShiken.com が最新の **1Z0-819** 試験問題集を提供しています。GoShiken.com 1Z0-819 試験問題は最新で、解答が正確でございます。最新の GoShiken.com 1Z0-819 問題集をゲットする人はこちら: <https://www.goshiken.com/Oracle/1Z0-819-mondaishu.html> (297**30%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

与えられた：

```

パブリッククラスX{

```

```

}

```

と

```

public final class Y extends X {

```

```

}

```

これら2つのクラスをコンパイルした結果はどうなりますか？

- A. クラスXにゼロ引数コンストラクターが定義されていないため、コンパイルは失敗します。
- B. クラスXまたはクラスYのいずれかがtoString (メソッドを実装する必要があるため、コンパイルは失敗します。
- C. 最終クラスが別のクラスを拡張できないため、コンパイルは失敗します。
- D. コンパイルは成功します。

Answer: B (メッセージを残す)

```

14 public class Main {
15     public static void main (String[] args) {
16         public class X {
17             }
18         }
19     }
20     public final class Y extends X {
21     }
22 }
23 }
24

```

最新問題: 78

正しい3つの初期化ステートメントはどれですか？ 3つ選択してください。）

- A. String contact# = " (+2) 999) 232)";
- B. バイトb = 10; char c = b;
- C. ブール値true = 4 == 4);
- D. short sh = (short) 'A';
- E. int [] [] e = {{1,1},{2,2}};
- F. float x = 1.99;
- G. int x = 12_34;

Answer: D,E,G ([メッセージを残す](#))

最新問題: 79

与えられた：

```
StringBuilder s = new StringBuilder("ABCD");
```

sがAQCDになる原因はどれですか？

- A. s.replace s.indexOf ("B"), s.indexOf ("C"), "Q");
- B. s.replace s.indexOf ("A"), s.indexOf ("C"), "Q");
- C. s.replace s.indexOf ("A"), s.indexOf ("B"), "Q");
- D. s.replace s.indexOf ("B"), s.indexOf ("B"), "Q");

Answer: ([解答を表示する](#))

最新問題: 80

java.util.functionパッケージのどのインターフェースがプリミティブ型を返すことができますか？

- A. サプライヤー
- B. LongConsumer
- C. ToDoubleFunction
- D. BiFunction

Answer: ([解答を表示する](#))

最新問題: 81

コードを分析します。

```

public class Test {
    static String prefix = "Global.";
    private String name = "namespace";
    public static String getName() {
        return new Test().name;
    }
    public static void main(String[] args) {
        Test t = new Test();
        System.out.println(/* Insert code here */);
    }
}

```

Global namespaceを生成するためにprintlnメソッド内に挿入できる2つのオプションはどれですか？ 2つ選択してください。）

- A. Test.getName + prefix
- B. new Test ().prefix + new Test ().name
- C. プレフィックス+名前
- D. Test.prefix + Test.getName ()
- E. Test.prefix + Test.name
- F. prefix + Test.name

Answer: B,D (メッセージを残す)

最新問題: 82

与えられた：

```

public interface Builder {
    public A build(String str);
}

and

public class BuilderImpl implements Builder {
    @Override
    public B build(String str) {
        return new B(str);
    }
}

```

このコードが正しくコンパイルされると仮定すると、正しい3つのステートメントはどれですか。 3つ選択してください。）

- A. Bをファイナルにすることはできません。
- B. BはAのサブタイプです。
- C. Aを抽象化することはできません。
- D. AはBのサブタイプです。
- E. Aをファイナルにすることはできません。
- F. Bを抽象化することはできません。

Answer: B,E,F (メッセージを残す)

最新問題: 83

与えられた：

```

public class Main {
    public static void main(String[] args) {
        List l = new ArrayList();
        l.add("hello");
        l.add("world");
        print(l);
    }
    private static void print(List<String>... args) {
        for (List<String> str : args) {
            System.out.println (str);
        }
    }
}

```

コンパイルから警告を削除するには、どのアノテーションを使用する必要がありますか？

- A. mainメソッドとprintメソッドの@SuppressWarnings
- B. mainの@SuppressWarnings ("unchecked")およびprintメソッドの@SafeVarargs
- C. mainの@SuppressWarnings ("rawtypes")とprintメソッドの@SafeVarargs
- D. mainメソッドとprintメソッドの@SuppressWarnings ("all")

Answer: B ([メッセージを残す](#))

```

13 @SuppressWarnings("unchecked")
14 public class Main {
15
16     public static void main(String[] args) {
17
18         List l = new ArrayList();
19         l.add("Hello");
20         l.add("world");
21         print(l);
22     }
23
24     private static void print(List<String>... args) {
25         for (List<String> str : args) {
26             System.out.println (str);
27         }
28     }
29
30 }
31 @SafeVarargs
32 }

```

最新問題: 84

与えられた：

var Fruits = List.of ("apple", "orange", "banana", "lemon");

文字nを含む最初の要素を調べたいと思います。どのステートメントがこれを達成しますか？

- A. 文字列の結果= Fruits.stream () filter f-> f.contains ("n") } findAny ()
- B. fruits.stream () filter f-> f.contains ("n") } forEachOrdered \$System.out :: print);
- C. オプション<String> result = Fruits.stream () filter f-> f.contains ("n") } findFirst ()
- D. オプション<String> result = Fruits.stream () anyMatch f-> f.contains ("n") }

Answer: B ([メッセージを残す](#))

ORACLE

```
1 import java.io.*;
2 import java.util.*;
3 public class abc {
4     public static void main(String[] args) {
5
6         var fruits = List.of("apple", "orange", "banana", "lemon");
7
8         fruits.stream().filter(f -> f.contains("n")).forEachOrdered(System.out::print);
9
10    }
11 }
12
```

Execute Mode, Version, Inputs & Arguments

JDK 11.0.4

☐ Interactive

Stdin In

CommandLine Arguments

Execute

Result

CPU Time: 0.19 sec(s), Memory: 33200 kilobyte(s)

orangebanana lemon

最新問題: 85

与えられた :

```
import java.sql.Timestamp;
public class Test {
    public static void main(String[] args) {
        Timestamp test = new Timestamp(1);
    }
}
```

およびコマンド :

```
javac Test.java
jdeps -summary Test.class
```

これらのコマンドを実行するとどうなりますか？

Test.class-> java.sql-> java.base

A.

B. Test.class-> java.sql

java.sql-> java.base

C. 実行時に、jdepsコマンドはエラーを表示します。

Test.class-> java.base

D. Test.class-> java.sql

Test.class-> java.base

Answer: ([解答を表示する](#))

最新問題: 86

```
public class Thing {
    private String name;
    public Thing(String name) {
        this.name = name;
    }
    public String toString() {
        return name;
    }
}

and

public class Tester {
    public static void main(String[] args) {
        Thing[] things = processThings();
        /* line 1 */
        for (Thing t: things) {
            System.out.println(t);
        }
    }
    public static Thing[] processThings(){
        Thing[] things = new Thing[3];
        things[0] = new Thing("Hat");
        things[1] = new Thing("Rat");
        things[2] = things[0];
        things[0] = new Thing("Cat");
        things[1] = things[2];
        return things;
    }
}
```

1行目のガベージコレクションに適格なThingオブジェクトはいくつありますか？

- A. 3
- B. 4
- C. 0
- D. 2
- E. 1

Answer: E ([メッセージを残す](#))

最新問題: 87

コードフラグメントが与えられた場合：

```
int[] secA = { 2, 4, 6, 8, 10 };
int[] secB = { 2, 4, 8, 6, 10 };
int res1 = Arrays.mismatch(secA, secB);
int res2 = Arrays.compare(secA, secB);
System.out.print(res1 + " : " + res2);
```

結果はどうなりますか？

- A. 3 0
- B. 2 3
- C. 2 -1
- D. -1 2

Answer: C ([メッセージを残す](#))

最新問題: 88

与えられた :

Automobile.java

```
public abstract class Automobile { //line 1
    abstract void wheels();
}
```

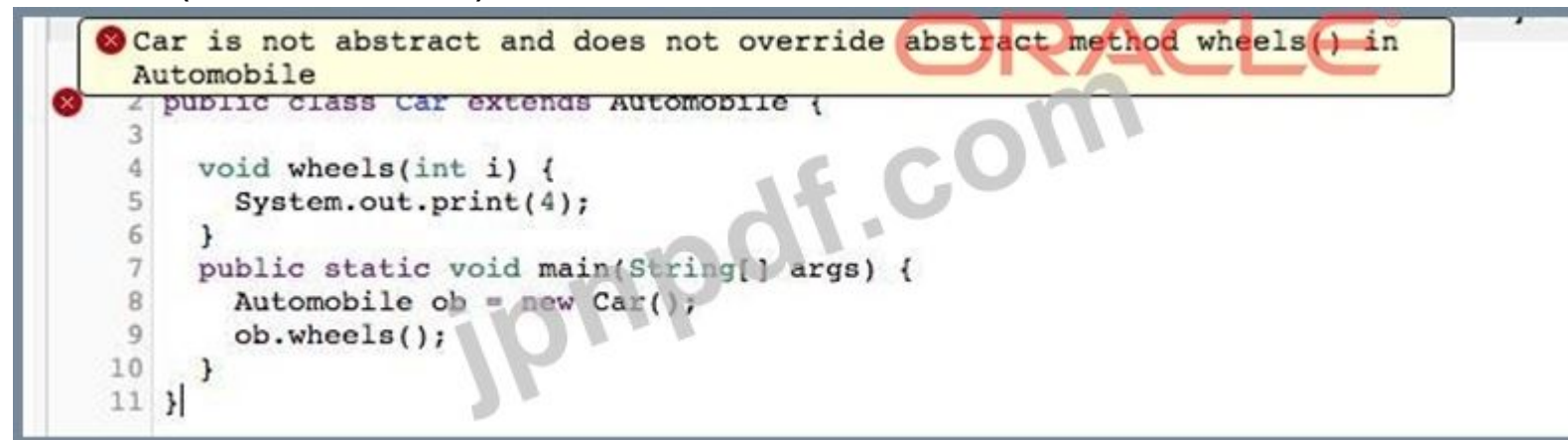
Car.java

```
public class Car extends Automobile {
    // line 2
    void wheels(int i) { // line 3
        System.out.print(4);
    }
    public static void main(String[] args) {
        Automobile ob = new Car(); // line 4
        ob.wheels();
    }
}
```

コードが4を出力するには、何をする必要がありますか？

- A. 3行目のwheelsメソッドからパラメーターを削除します。
- B. 2行目に@Overrideアノテーションを追加します。
- C. 2行目のコードをCar ob = new Car ()に置き換えます。
- D. 1行目の抽象キーワードを削除します。

Answer: B ([メッセージを残す](#))



最新問題: 89

与えられた :

```

public class Person {
    private String name;
    public void setName(String name) {
        String title = "Dr. ";
        name = title+name;
    }
    public String toString() {
        return name;
    }
}

and

public class Test {
    public static void main(String args[]) {
        Person p = new Person();
        p.setName("Who");
        System.out.println(p);
    }
}

```

結果はどうなりますか？

- A. ドクター .フォー
- B. スル博士
- C. 実行時に例外がスローされます。
- D. null

Answer: ([解答を表示する](#))



```

Console 1
null
Completed with exit code: 0

```

最新問題: 90

与えられた：

```

class CustomType<T> {
    public <T> int count(T[] anArray, T element) {
        int count = 0;
        for(T e : anArray) {
            if (e.equals(element)) ++count;
        }
        return count;
    }
}

```

と

```

public class Test extends CustomType {
    public static void main(String[] args) {
        String[] words = {"banana", "orange", "apple", "lemon"};
        Integer[] numbers = {1, 2, 3, 4, 5};
        CustomType type = new CustomType();
        CustomType<String> stringType = new CustomType<>();
        System.out.println(stringType.count(words, "apple"));
        System.out.println(type.count(words, "apple"));
        System.out.println(type.count(numbers, 3));
    }
}

```

結果はどうなりますか？

A. 実行時にNullPointerExceptionがスローされます。

B. コンパイルは失敗します。

C. 1

ヌル

ヌル

D. 1

1

1

E. 実行時にClassCastExceptionがスローされます。

Answer: B ([メッセージを残す](#))

```

Console 4 ✖
Error: Could not find or load main class CustomType
Caused by: java.lang.ClassNotFoundException: CustomType

```

最新問題: 91

与えられた：

```

public interface InterfaceOne {
    void printOne();
}

```

printOne（を正常にオーバーライドする3つのクラスはどれですか？ 3つ選択してください。）

ORACLE

A.

```
public abstract class TestClass implements InterfaceOne {  
    public abstract void printOne();  
}
```

B.

```
public class TestClass implements InterfaceOne {  
    private void printOne() {  
        System.out.println("one");  
    }  
}
```

C.

```
public class TestClass implements InterfaceOne {  
    public void printOne() {  
        System.out.println("one");  
    }  
}
```

D.

```
public abstract class TestClass implements InterfaceOne {  
    public void printOne() {  
        System.out.println("one");  
    }  
}
```

E.

```
public abstract class TestClass implements InterfaceOne {  
    public String printOne() {  
        return "one";  
    }  
}
```

F.

```
public class TestClass {  
    public void printOne() {  
        System.out.println("one");  
    }  
}
```

A. オプションB

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A

F. オプションC

Answer: D,E,F ([メッセージを残す](#))

有効な 1Z0-819 問題集は GoShiken.com が提供された合格しやすい 1Z0-819 試験問題集！ GoShiken.com が最新の 1Z0-819 試験問題集を提供しています。GoShiken.com 1Z0-819 試験問題は最新で、解答が正確でございます。最新の GoShiken.com 1Z0-819 問題集をゲットする人はこちら: <https://www.goshiken.com/Oracle/1Z0-819-mondaishu.html> (29730%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

与えられた:

/code/a/Test.java

含む:

```
package a;
import b.Best;
public class Test {
    public static void main(String[] args) {
        Best b = new Best();
    }
}
```

と

/code/b/Best.java

含む:

パッケージb;

パブリッククラスベスト{}

すべてのクラスのバイトコードを生成する有効な方法はどれですか？

A. javac -d / code /code/a/Test.java

B. java /code/a/Test.java /code/b/Best.java

C. java -cp / code a.Test

D. java /code/a/Test.java

E. javac -d / code /code/a/Test.java /code/b/Best.java

F. javac -d / code / code / a / Test

Answer: E ([メッセージを残す](#))

最新問題: 93

与えられた:

```

class ConSuper {
    protected ConSuper() {
        this(2);
        System.out.print("1");
    }
    protected ConSuper(int a) {
        System.out.print(a);
    }
}

```

and

```

public class ConSub extends ConSuper{
    ConSub() {
        this(4);
        System.out.print("3");
    }
    ConSub(int a) {
        System.out.print(a);
    }
    public static void main (String[] args)
        new ConSub(4);
}

```

結果はどうなりますか？

- A. 2134
- B. 2143
- C. 214
- D. 234

Answer: C ([メッセージを残す](#))



最新問題: 94

コードフラグメントが与えられた場合：

```

char[][] arrays = {{'a', 'd'}, {'b', 'e'}, {'c', 'f'}};
for (char[] xx : arrays) {
    for (char yy : xx) {
        System.out.print(yy);
    }
    System.out.print(" ");
}

```

結果はどうなりますか？

- A. ab cd ef

- B. abc def
- C. コンパイルは失敗します。
- D. ad be cf
- E. 実行時にArrayIndexOutOfBoundsExceptionがスローされます。

Answer: ([解答を表示する](#))

最新問題: 95

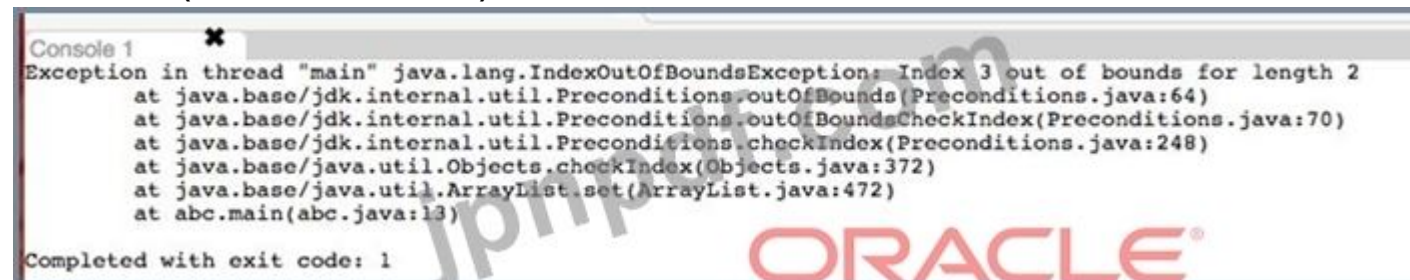
与えられた :

```
var data = new ArrayList <> ()
data.add ("Peter");
data.add 30);
data.add ("Market Road");
data.set 1, 25);
data.remove 2);
data.set 3, 1000L);
System.out.print 4data);
```

出力は何ですか？

- A. [Market Road、1000]
- B. [ピーター、30、マーケットロード]
- C. [ピーター、25、ヌル、1000]
- 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](#))



最新問題: 96

与えられた :

```
public class Sportscar extends Automobile{
    private float turbo;
    ....
    public void setTurbo (float turbo){
        this.turbo = turbo;
    }
}
```

スポーツカークラスについて何が知られていますか？

- A. Sportscarサブクラスは、スーパークラスAutomobileのsetTurboメソッドをオーバーライドできません。
- B. Sportscarクラスは、スーパークラスAutomobileからsetTurboメソッドを継承します。

C. Sportscarクラスは、Automobileクラスよりも多くの機能を備えたスーパークラスです。

D. SportscarクラスはAutomobileのサブクラスであり、そのメソッドを継承します。

Answer: D ([メッセージを残す](#))

最新問題: 97

コードフラグメントが与えられた場合：

```
public class FizzBuzz {
    public static String convert(int x) {
        if (x % 15 == 0) return "FizzBuzz";
        else if (x % 3 == 0) return "Fizz";
        else if (x % 5 == 0) return "Buzz";
        else return Integer.toString(x);
    }

    public static void main(String[] args) {
        for (int i = 1; i < 101; i++) {
            System.out.println(convert(i));
        }
    }
}
```

forステートメントを置き換えるコードフラグメントはどれですか？

A. `intstream.rangeclosed (1, 100).mapToObj {FizzBuzz :: convert}.forEach $System.out :: println`;

B. `IntStream.rangeClosed (1, 100).map FizzBuzz :: convert).forEach $System.out :: println`;

C. `IntStream.range (1, 100).mapToObj FizzBuzz :: convert).forEach $System.out :: println`;

D. `IntStream.ranged (1, 100).map FizzBuzz :: convert).forEach $System.out :: println`;

Answer: ([解答を表示する](#)**)**

最新問題: 98

与えられた：

```
1. public class Test {
2.     private static class Greet {
3.         private void print() {
4.             System.out.println("Hello World");
5.         }
6.     }
7.     public static void main(String[] args) {
8.         Test.Greet i = new Greet();
9.         i.print();
10.    }
11. }
```

結果はどうなりますか？

A. コンパイルは9行目で失敗します。

B. コンパイルは2行目で失敗します。

C. Hello World

D. コンパイルは8行目で失敗します。

Answer: ([解答を表示する](#))

```
1 public class Test {  
2     private static class Greet {  
3         private void print() {  
4             System.out.println("Hello World")  
5         }  
6     }  
7     public static void main(String[] args) {  
8         Test.Greet i = new Greet();  
9         i.print();  
10    }  
11 }
```

Execute Mode, Version, Inputs & Arguments

JDK 11.0.4

CommandLine Arguments

Result

CPU Time: 0.16 sec(s), Memory: 32504 kilobyte(s)

Hello World

ORACLE

最新問題: 99

与えられた :

```
class Mycar {  
}  
  
and  
  
javac C:\workspace4\Mycar.java
```

javacの期待される結果は何ですか？

- A. javacは、エラーや警告なしでMycar.javaをコンパイルします。
- B. javacはクラスのコンパイルに失敗し、エラーメッセージC \ worksheet4 \ Mycar.java :1 :error :expectedimportjava.langを出力します
- C. javacはクラスのコンパイルに失敗し、エラーメッセージを出力します。エラー :メインクラスMycar.classが見つからないかロードできませんでした
- D. javacはクラスのコンパイルに失敗し、エラーメッセージを出力します。C \ worksheet4 \ Mycar.java :1 :error :パッケージjavaは存在しません

Answer: A ([メッセージを残す](#))

最新問題: 100

コードフラグメントが与えられた場合：

```
9. Integer[] ints = {1,2,3,4,5,6,7};
10. var list = Arrays.asList(ints);
11. UnaryOperator<Integer> uo = x -> x * 3;
12. list.replaceAll(uo);
```

11行目を置き換えることができるのはどれですか？

A. UnaryOperator <Interger> uo = var x-> {return x * 3; };

B. UnaryOperator <Interger> uo = (int x)-> x * 3;

C. UnaryOperator <Interger> uo = x-> {return x * 3; };

D. UnaryOperator <Interger> uo = (var x)-> x * 3);

Answer: D ([メッセージを残す](#))

最新問題: 101

コードフラグメントが与えられた場合：

```
9. Integer[] ints = {1,2,3,4,5,6,7};
10. var list = Arrays.asList(ints);
11. UnaryOperator<Integer> uo = x -> x * 3;
12. list.replaceAll(uo);
```

11行目を置き換えることができるのはどれですか？

A. UnaryOperator <Interger> uo = (int x)-> x * 3;

B. UnaryOperator <Interger> uo = x-> {return x * 3; };

C. UnaryOperator <Interger> uo = (var x)-> x * 3);

D. UnaryOperator <Interger> uo = var x-> {return x * 3; };

Answer: C ([メッセージを残す](#))

最新問題: 102

与えられた：

```

public class Tester {
    public static void main(String[] args) {
        char letter = 'b';
        int i = 0;
        switch(letter) {
            case 'a':
                i++;
                break;
            case 'b':
                i++;
            case 'c' | 'd': // line 1
                i++;
            case 'e':
                i++;
                break;
            case 'f':
                i++;
                break;
            default:
                System.out.print(letter);
        }
        System.out.println(i);
    }
}

```

結果はどうなりますか？

- A. b1
- B. 2
- C. b2
- D. 1
- E. b3
- F. 3
- G. 1行目のエラーが原因でコンパイルが失敗します。

Answer: ([解答を表示する](#))

Result

CPU Time: 0.23 sec(s), Memory: 32708 kilobyte(s)

3

最新問題: 103

getterメソッドとsetterメソッド、およびこのコードフラグメントとともに、年齢と名前を持つPersonクラスが与えられます。

```
List<Person> persons = new ArrayList(List.of(new Person(44,"Tom"),
                                              new Person(40,"Aman"),
                                              new Person(40,"Peter")));
persons.sort(Comparator.comparing((Person::getAge))
              .thenComparing(Person::getName)
              .reversed());
persons.forEach(p1->System.out.print(" "+p1.getName()));
```

結果はどうなりますか？

- A. トム・ピーター・アマン
- B. トムアマンピーター
- C. アマン・ピーター・トム
- D. アマン・トム・ピーター

Answer: C ([メッセージを残す](#))

最新問題: 104

オートボクシングの成功例はどれですか？ 2つ選択してください。)

- A. 長いc = 23L;
- B. ダブルd = 4;
- C. Float g = Float.valueOf (null);
- D. 整数e = 5;
- E. フロートf = 6.0;
- F. 文字列a = "A";

Answer: D,F ([メッセージを残す](#))

最新問題: 105

与えられた：

```
public class Employee {
    private String name;
    private LocalDate birthday;
    // the constructors, getters, and setters methods go here
}
```

と

```
List<Employee> roster = new ArrayList<>();
// ...
Predicate<Employee> y = (Employee e) -> e.getBirthday()
    .isBefore(IsoChronology.INSTANCE.date(1989, 1, 1));
Set<String> s1 = roster.stream()
// Line 1
```

1行目のどのコードフラグメントがs1セットに、1月1日より前に生まれたすべての従業員の名前を含むようにします。
1989年？

```

A. .collect(Collectors.partitioningBy(y))
   .get(true)
   .stream()
   .map(Employee::getName)
   .collect(Collectors.toCollection(TreeSet::new));

B. .collect(Collectors.partitioningBy(y))
   .get(true)
   .map(Employee::getName)
   .collect(Collectors.toSet());

C. .collect(Collectors.partitioningBy(y, Collectors.mapping(
    Employee::getName, Collectors.toSet())));

D. .collect(Collectors.partitioningBy(y, Collectors.groupingBy(
    Employee::getName, Collectors.toCollection(TreeSet::new))));

```

A. オプションD

B. オプションA

C. オプションB

D. オプションC

Answer: ([解答を表示する](#))

最新問題: 106

与えられた :

```

package test.t1;
public class A {
    public int x = 42;
    protected A() {} // line 1
}

and

package test.t2;
import test.t1.*;
public class B extends A {
    int x = 17; // line 2
    public B() { super(); } // line 3
}

and

package test;
import test.t1.*;
import test.t2.*;
public class Tester {
    public static void main(String[] args) {
        A obj = new B(); // line 4
        System.out.println(obj.x); // line 5
    }
}

```

結果はどうなりますか？

A. 42

- B. 2行目のエラーが原因でコンパイルが失敗します。
- C. 1行目のエラーが原因でコンパイルが失敗します。
- D. 4行目のエラーが原因でコンパイルが失敗します。
- E. 3行目のエラーが原因でコンパイルが失敗します。
- F. 5行目のエラーが原因でコンパイルが失敗します。
- G. 17

Answer: A ([メッセージを残す](#))

有効な **1Z0-819** 問題集は GoShiken.com が提供された合格しやすい 1Z0-819 試験問題集！ GoShiken.com が最新の **1Z0-819** 試験問題集を提供しています。GoShiken.com 1Z0-819 試験問題は最新で、解答が正確でございます。最新の GoShiken.com 1Z0-819 問題集をゲットする人はこちら: <https://www.goshiken.com/Oracle/1Z0-819-mondaishu.html> (**29730%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: 107

コードフラグメントが与えられた場合：

```
public class City {  
    public static void main(String[] args) {  
        String[] towns = {"boston", "paris", "bangkok", "oman"};  
        Comparator<String> ms = (a, b) -> b.compareTo(a);  
        Arrays.sort(towns, ms);  
        System.out.println(Arrays.binarySearch(towns, "oman", ms));  
    }  
}
```

結果はどうなりますか？

- A. 1
- B. 2
- C. -3
- D. -1

Answer: B ([メッセージを残す](#))

最新問題: 108

与えられた：

```

public class DNASynth {
    int aCount;
    int tCount;
    int cCount;
    int gCount;

    int getACount(int aCount){
        return aCount;
    }
    int getTCount(int tCount){
        return this.tCount;
    }
    int getCCount(){
        return getTotalCount() - this.aCount - getTCount(0) - gCount;
    }
    int getGCount(){
        return getGCount();
    }
    int getTotalCount(){
        return aCount + getTCount(0) + this.cCount + this.gCount;
    }
}

```

インスタンスフィールドを読み取る有効な方法を容易にする2つの方法はどれですか？ 2つ選択してください。)

- A. getTotalCount
- B. getCCount
- C. getACount
- D. getTCount
- E. getGCount

Answer: A,B ([メッセージを残す](#))

最新問題: 109

コードフラグメントが与えられた場合：

```

int x = 0;
do {
    x++;
    if (x == 1) {
        continue;
    }
    System.out.println(x);
} while(x < 1);

```

結果はどうなりますか？

- A. 0
- B. 01
- C. プログラムは何も出力しません。
- D. 無限ループで1を出力します。
- E. 1

Answer: C ([メッセージを残す](#))

最新問題: 110

与えられた：

```
public interface Builder {
    public A build(String str);
}

and

public class BuilderImpl implements Builder {
    @Override
    public B build(String str) {
        return new B(str);
    }
}
```

このコードが正しくコンパイルされると仮定すると、正しい3つのステートメントはどれですか。3つ選択してください。)

- A. Bを抽象化することはできません。
- B. Bをファイナルにすることはできません。
- C. AはBのサブタイプです。
- D. Aを抽象化することはできません。
- E. BはAのサブタイプです。
- F. Aをファイナルにすることはできません。

Answer: A,E,F ([メッセージを残す](#))

最新問題: 111

与えられた：

```
public class Main {
    class Student {
        String classname;
        Student(String classname) {
            this.classname = classname;
        }
    }
    public static void main(String[] args) {
        var student = new Student("Biology");
    }
}
```

Mainクラスをコンパイルする2つの独立した変更はどれですか？ 2つ選択してください。)

- A. Studentクラス宣言全体を別のJavaファイルStudent.javaに移動します。
- B. 2行目をpublic Student \$tring classname)に変更します。
- C. 1行目をパブリッククラスのStudent{に変更します。
- D. 3行目をStudent student = new Student ("Biology");に変更します。
- E. 1行目を静的クラスStudent{に変更します。

Answer: ([解答を表示する](#))

ORACLE

```
1 import java.util.*;
2 import java.io.*;
3 import java.lang.Thread;
4 import java.util.ArrayList;
5 import java.util.LinkedList;
6 import java.util.List;
7 import java.util.function.Consumer;
8 import java.util.stream.Stream;
9 import java.util.stream.IntStream;
10 import java.util.Optional;
11
12
13 public class Main {
14     class Student {
15         String classname;
16         public Student (String classname) {
17             this.classname = classname;
18         }
19
20     }
21     public static void main (String[] args) {
22         var student = new Student ("Biology");
23     }
24 }
```

最新問題: 112

どのコードフラグメントがコンパイルされますか？

```

A. Comparator comparator = new Comparator<?>() {
    public int compare(Integer i, Integer j) {
        return i.compareTo(j);
    }
};

B. var comparator = new Comparator<>() {
    public int compare(Integer i, Integer j) {
        return i.compareTo(j);
    }
};

C. Comparator<> comparator = new Comparator<Integer>() {
    public int compare(Integer i, Integer j) {
        return i.compareTo(j);
    }
};

D. Comparator<Integer> comparator = new Comparator<>() {
    public int compare(Integer i, Integer j) {
        return i.compareTo(j);
    }
};

```

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

Answer: ([解答を表示する](#))

```

1 import java.io.*;
2 import java.util.*;
3 class abc {
4     public static void main(String[] args) {
5
6         Comparator<Integer> comparator = new Comparator<>() {
7             public int compare(Integer i, Integer j) {
8                 return i.compareTo(j);
9             }
10        };
11    }
12 }
13 }
14

```

最新問題: 113

Java SE仕様にAPIが含まれている2つのモジュールはどれですか？ 2つ選択してください。）

A. jdk.httpserver

B. java.desktop

C. javafx

D. java.logging

E. jdk.jartool

Answer: A,D ([メッセージを残す](#))

最新問題: 114

与えられた :

```
class MyClass {  
    public static void main(String [] args) {  
        System.out.println(arg[1] + "--" + arg[3] + "--" + arg[0]);  
    }  
}
```

このコマンドを使用して実行 :

java MyClassMyCarは赤です

このクラスの出力は何ですか？

- A. My--Car--is
- B. java--Myclass--My
- C. MyClass--Car--red
- D. 車-赤私の
- E. My--is--java

Answer: D ([メッセージを残す](#))

最新問題: 115

与えられた :

```
public class Foo {  
    private void print() {  
        System.out.println("Bonjour le monde!");  
    }  
    public void foo() {  
        print();  
    }  
}  
  
public class Bar extends Foo {  
    private void print() {  
        System.out.println("Hello world!");  
    }  
    public void bar() {  
        print();  
    }  
    public static void main(String... args) {  
        Bar b = new Bar();  
        b.foo();  
        b.bar();  
    }  
}
```

出力は何ですか？

- A. Hello world ! Bonjour le monde !
- B. Hello world ! Hello world !
- C. Bonjour le monde ! Hello world !
- D. Bonjour le monde ! Bonjour le monde !

Answer: ([解答を表示する](#))



最新問題: 116

与えられた :

```
public class SerializedMessage implements Serializable {
    String message;
    LocalDateTime createdTime;
    transient LocalDateTime updatedDateTime;;
    SerializedMessage(String message) {
        this.message = message;
        this.createdTime = LocalDateTime.now();
    }
    private void readObject (ObjectInputStream in) {
        try {
            in.defaultReadObject();
            this.updatedDateTime = LocalDateTime.now();
        } catch (IOException | ClassNotFoundException e) {
            e.printStackTrace();
        }
    }
}
```

readObjectメソッドはいつ呼び出されますか？

- A. このオブジェクトがシリアル化される前
- B. このオブジェクトが逆シリアル化される前
- C. メソッドが呼び出されることはありません。
- D. このオブジェクトがシリアル化された後
- E. このオブジェクトが逆シリアル化された後

Answer: ([解答を表示する](#))

最新問題: 117

与えられた :

```
import java.util.*;

public class Main {
    static Map<String, String> map = new HashMap<>();
    static List<String> keys =
        new ArrayList<>(List.of("A", "B", "C", "D"));
    static String[] values =
        {"one", "two", "three", "four"};

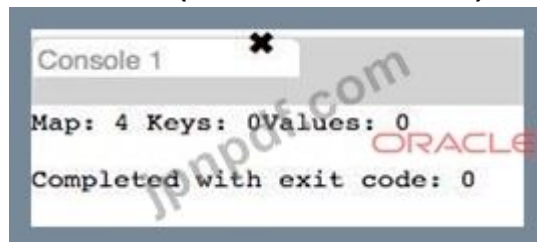
    static {
        for(var i = 0; i < keys.size(); i++) {
            map.put(keys.get(i), values[i]);
        }
    }

    public static void main(String[] args) {
        keys.clear();
        values = new String[0];
        System.out.println("Map: " + map.size() +
            " Keys: " + keys.size() +
            " Values: " + values.length);
    }
}
```

結果はどうなりますか？

- A. マップ :0キー :0値 :0
- B. コンパイルは失敗します。
- C. マップ :4キー :4値 :4
- D. マップ :4キー :0値 :0
- E. マップ :0キー :4値 :4

Answer: D ([メッセージを残す](#))



```
Console 1
Map: 4 Keys: 0Values: 0
Completed with exit code: 0
```

最新問題: 118

与えられた :

```
package a;
public abstract class Animal {
    protected abstract void walk();
}
package b;
public abstract class Human extends Animal {
    // line 1
}
```

1行目に挿入された2行のうち、このコードをコンパイルできるのはどれですか？ 2つ選択してください。）

- A. void walk ({})
- B. 保護されたvoid walk ({})
- C. public abstract void walk ({})
- D. private void walk ({})
- E. 抽象void walk ({})

Answer: B,C ([メッセージを残す](#))

最新問題: 119

この要件を考えると：

モジュール車両はモジュール部品に依存し、そのcom.vehicleパッケージを他のすべてのモジュールで使えるようにします。

どのmodule-info.java宣言が要件を満たしていますか？

A

```
module vehicle{  
    requires part;  
    exports com.vehicle;  
}
```

B

```
module vehicle {  
    requires part;  
    uses com.vehicle;  
}
```

C

```
module vehicle{  
    requires part;  
    exports com.vehicle to part;  
}
```

D

```
module vehicle {  
    requires com.vehicle;  
    exports part;  
}
```

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: B ([メッセージを残す](#))

最新問題: 120

与えられた :

```
import java.util.*;

public class Main {
    static Map<String, String> map = new HashMap<>();
    static List<String> keys =
        new ArrayList<>(List.of("A", "B", "C", "D"));
    static String[] values =
        {"one", "two", "three", "four" };

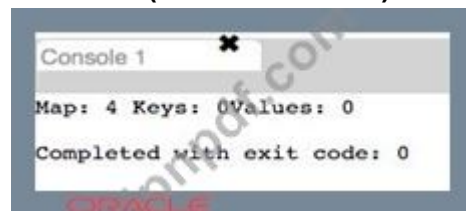
    static {
        for(var i = 0; i < keys.size(); i++) {
            map.put(keys.get(i), values[i]);
        }
    }

    public static void main(String[] args) {
        keys.clear();
        values = new String[0];
        System.out.println("Map: " + map.size() +
            " Keys: " + keys.size() +
            " Values: " + values.length);
    }
}
```

結果はどうなりますか？

- A. マップ :0キー :0値 :0
- B. コンパイルは失敗します。
- C. マップ :4キー :4値 :4
- D. マップ :4キー :0値 :0
- E. マップ :0キー :4値 :4

Answer: ([解答を表示する](#))



```
Console 1
Map: 4 Keys: 0Values: 0
Completed with exit code: 0
```

最新問題: 121

与えられた：

```
public class Hello {
    public static void main(String[] args) {
        System.out.println(args[0]+args[1]+args[2]);
    }
}
```

コマンドを使用して実行：

```
java Hello "Hello World" Hello World
```

出力は何ですか？

- A. こんにちはWorldHelloWorld
- B. ハローワールドハローワールド

- C. Hello World Hello World
- D. HelloHello WorldHelloWorld
- E. 実行時に例外がスローされます。

Answer: ([解答を表示する](#))

有効な **1Z0-819** 問題集は GoShiken.com が提供された合格しやすい 1Z0-819 試験問題集！ GoShiken.com が最新の **1Z0-819** 試験問題集を提供しています。GoShiken.com 1Z0-819 試験問題は最新で、解答が正確でございます。最新の GoShiken.com 1Z0-819 問題集をゲットする人はこちら: <https://www.goshiken.com/Oracle/1Z0-819-mondaishu.html> (**29730%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

与えられた：

```
1. void insertionSort(int values[]) {
2.     int n = values.length;
3.     for (int j = 1; j < n; j++) {
4.         int tmp = values[j];
5.         int i = j - 1;
6.         while ( (i > -1) && (values[i] > tmp) ) {
7.             values[i + 1] = values[i];
8.             i--;
9.         }
10.        values[i + 1] = tmp;
11.    }
12. }
```

その後、assert i<0||を挿入できます。values [i] <= values [i + 1]; 値の配列が部分的に並べ替えられていることを確認するには？

- A. 8行目以降
- B. 6行目以降
- C. 5行目以降
- D. 10行目以降

Answer: B ([メッセージを残す](#))

```

1 import java.util.*;
2 import java.io.*;
3 import java.lang.Thread;
4 import java.util.ArrayList;
5 import java.util.LinkedList;
6 import java.util.List;
7 import java.util.function.Consumer;
8 import java.util.stream.Stream;
9 import java.util.stream.IntStream;
10
11
12 public class Main {
13
14
15     void insertionSort(int values[]) {
16         int n = values.length;
17         for (int j = 1; j < n; j++) {
18             int tmp = values[j];
19
20             int i = j - 1;
21             assert i < 0 || values[i] <= values[i + 1];
22             while ((i > 1) && (values[i] > tmp)) {
23                 values[i + 1] = values[i];
24                 i--;
25             }
26             values[i + 1] = tmp;
27         }
28     }
29 }
30
31 }

```

最新問題: 123

getterメソッドとsetterメソッド、およびこのコードフラグメントとともに、年齢と名前を持つPersonクラスが与えられます。

```

List<Person> persons = new ArrayList(List.of(new Person(44, "Tom"),
                                                new Person(40, "Aman"),
                                                new Person(40, "Peter")));
persons.sort(Comparator.comparing(Person::getAge)
                        .thenComparing(Person::getName)
                        .reversed());
persons.forEach(p1->System.out.print(" "+p1.getName()));

```

結果はどうなりますか？

- A. アマン・トム・ピーター
- B. トムアマンピーター
- C. アマン・ピーター・トム
- D. トム・ピーター・アマン

Answer: C ([メッセージを残す](#))

最新問題: 124

与えられた：

```

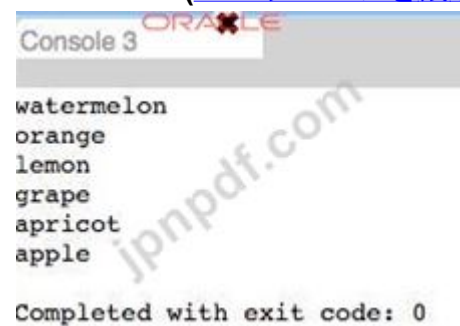
import java.util.ArrayList;
import java.util.Arrays;
public class NewMain {
    public static void main(String[] args) {
        String[] fruitNames = { "apple", "orange",
            "grape", "lemon", "apricot", "watermelon" };
        var fruits = new ArrayList<>(Arrays.asList(fruitNames));
        fruits.sort((var a, var b) -> -a.compareTo(b));
        fruits.forEach(System.out::println);
    }
}

```

結果はどうなりますか？

- A. スイカorangelemongrapeapricotapple
- B. 何もない
- C. appleapricotgrapelemonorangewatermelon
- D. appleorangegrapelemonapricotwatermelon

Answer: A (メッセージを残す)



```

watermelon
orange
lemon
grape
apricot
apple
Completed with exit code: 0

```

最新問題: 125

与えられた：

```

public class Point {
    @JsonField(type=JsonField.Type.STRING, name="name")
    private String _name;

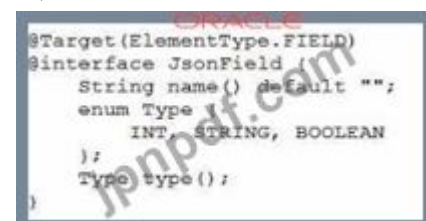
    @JsonField(type=JsonField.Type.INT)
    private int x;

    @JsonField(type=JsonField.Type.INT)
    private int y;
}

```

PointクラスをコンパイルするJsonFieldアノテーションの正しい定義は何ですか？

A)



```

@Target(ElementType.FIELD)
@interface JsonField {
    String name() default "";
    enum Type {
        INT, STRING, BOOLEAN
    };
    Type type();
}

```

B)

```

@interface JsonField {
    String name();
    enum Type {
        INT, STRING, BOOLEAN
    };
    Type type();
}

```

C)

```

@Retention(RetentionPolicy.RUNTIME)
@Target(ElementType.METHOD)
@interface JsonField {
    String name() default "";
    enum Type {
        INT, STRING, BOOLEAN
    };
    Type type();
}

```

A. オプションA

B. オプションC

C. オプションB

Answer: A ([メッセージを残す](#))

最新問題: 126

与えられた：

```

public class Confidential implements Serializable {
    private String data;

    public Confidential(String data) {
        this.data = data;
    }
}

```

これらのオブジェクトの安全なシリアル化はどれですか？ 2つ選択してください。）

A. インスタンスをシリアルプロキシに置き換えるためにwriteReplaceのみを実装し、readResolveは実装しません。

B. クラスを抽象化します。

C. インスタンスをシリアルプロキシに置き換えるにはreadResolveのみを実装し、writeReplaceは実装しません。

D. フィールドを一時的に宣言します。

E. serialPersistentFields配列フィールドを定義します。

Answer: C,E ([メッセージを残す](#))

最新問題: 127

与えられた：

```

public class Tester {
    static class Person implements /* line 1 */ {
        private String name;
        Person(String name) { this.name = name; }
        /* line 2 */
    }
    public static void main(String[] args) {
        Person[] people = {new Person("Joe"),
                           new Person("Jane"),
                           new Person("John")};

        Arrays.sort(people);
        for(Person person: people) {
            System.out.println(person.name);
        }
    }
}

```

コードで次の出力を生成する必要があります。

ジョン

ジョー

ジェーン

出力を生成するには、1行目と2行目にどのコードフラグメントを挿入する必要がありますか？

A. 1行目にComparator <Person>を挿入します。

入れる

```

public int compare (Person p1, Person p2){
    p1.name.compare (p2.name);を返します。
}

```

2行目。

B. 1行目にComparable <Person>を挿入します。

入れる

```

public int compare (Person p1, Person p2){
    p1.name.compare (p2.name);を返します。
}

```

2行目。

C. 1行目にComparator <Person>を挿入します。

入れる

```

public int compareTo (Person person){
    person.name.compareTo (this.name);を返します。
}

```

2行目。

D. 1行目にComparator <Person>を挿入します。

入れる

```
public int compare (Person person){
    person.name.compare (this.name);を返します。
}
```

2行目。

Answer: C ([メッセージを残す](#))

最新問題: 128

与えられた：

```
enum Color implements Serializable {
    R(1), G(2), B(3);
    int c;
    public Color(int c) {
        this.c = c;
    }
}
```

コンパイルを成功させるためのアクションは何ですか？

- A. public Color (int c)をprivate Color (int c)に置き換えます。
- B. intcを置き換えます。private intc;を使用します。
- C. intcを置き換えます。プライベートfinalintc;を使用します。
- D. 列挙型カラーの実装をパブリック列挙型カラーに置き換えます。
- E. enumColorをpublicenumColorに置き換えます。

Answer: ([解答を表示する](#)**)**

```
1
2 import java.io.*;
3 import java.util.*;
4 class Hello {
5
6
7     enum Color implements Serializable {
8         R(1), G(2), B(3);
9         int c;
10        private Color (int c) {
11            this.c = c;
12        }
13    }
14 }
```

最新問題: 129

与えられた：

jdeps -jdkinternals C:\worksheet4\SimpleSecurity\jar\classes.jar

期待される出力を説明するのはどれですか？

- A. jdepsは、参照されているすべてのJDK内部APIのモジュール依存関係とパッケージ名を一覧表示します。見つかった場合は、提案された代替品がコンソールに出力されます。
- B. jdepsは、-jdkinternalsオプションがコンソールに出力するために-summary または--verbose オプションのいずれかを必要とするというエラーメッセージを出力します。

C. -jdkinternalsオプションは、.jar内のすべてのクラスを分析し、すべてのクラスレベルの依存関係を出力します。

D. -jdkinternalsオプションは、.jar内のすべてのクラスを分析して、JDK内部APIへのクラスレベルの依存関係を確認します。見つかった場合は、推奨される置換の結果がコンソールに出力されます。

Answer: ([解答を表示する](#))

説明

-jdkinternalsオプションは、.jar内のすべてのクラスを分析して、JDK内部APIへのクラスレベルの依存関係を確認します。見つかった場合は、推奨される置換の結果がコンソールに出力されます。

最新問題: 130

```
class Worker {  
    private boolean finished = false;  
    public void consumeResource(Resource resource){  
        while(!resource.isReady()){  
            System.out.println("waiting for a resource");  
            try {  
                Thread.sleep(1000);  
            } catch (InterruptedException e) {  
                e.printStackTrace();  
            }  
        }  
        setFinished(true);  
    }  
    public boolean isFinished() {  
        return finished;  
    }  
    private void setFinished(boolean finished) {  
        this.finished = finished;  
    }  
}
```

そしてコードフラグメント：

```
Resource resource = new Resource();  
Worker worker = new Worker();  
Thread t1 = new Thread() -> resource.processWork(worker);  
Thread t2 = new Thread() -> worker.consumeResource(resource);  
  
t1.start();  
t2.start();
```

コードフラグメントの実行ではどのような状況が発生しますか？

A. デッドロック

B. 競合状態

C. 飢餓

D. Livelock

Answer: ([解答を表示する](#))

最新問題: 131

コードフラグメントが与えられた場合：

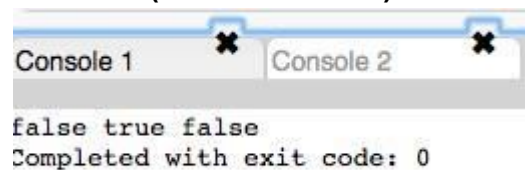
```
String s1 = new String("ORACLE");
String s2 = "ORACLE";
String s3 = s1.intern();

System.out.print((s1==s2) + " ");
System.out.print((s2==s3) + " ");
System.out.println(s1==s3);
```

結果はどうなりますか？

- A. false true true
- B. true false false
- C. false false true
- D. false true false

Answer: (解答を表示する)



```
false true false
Completed with exit code: 0
```

最新問題: 132

与えられた：

```
public class Main {
    public static void main(String[] args) {
        Thread t1 = new Thread(new MyThread())
        Thread t2 = new Thread(new MyThread())
        Thread t3 = new Thread(new MyThread())

        t1.start();
        t2.run();
        t3.start();

        t1.start();
    }
}

class MyThread implements Runnable {
    public void run() {
        System.out.println("Running.");
    }
}
```

どちらが正しいですか？

- A. 実行時にIllegalThreadStateExceptionがスローされます。
- B. 3つのスレッドが作成されます。
- C. コンパイルは失敗します。
- D. 4つのスレッドが作成されます。

Answer: A (メッセージを残す)

```
Running.  
Running.  
Running.  
  
Exception in thread "main" java.lang.IllegalThreadStateException  
at java.base/java.lang.Thread.start(Thread.java:794)  
at Main.main(Main.java:12)
```

最新問題: 133

与えられた :

```
int i = 10;  
do {  
    for(int j = i/2; j > 0; j--) {  
        System.out.print(j + " ");  
    }  
    i-=2;  
} while (i > 0);
```

結果はどうなりますか？

A. 5 4 3 2 1

B. 5

C. 何もない

D. 5 4 3 2 1 4 3 2 1 3 2 1 2 1 1

Answer: ([解答を表示する](#))

説明

グラフィカルユーザーインターフェイス、アプリケーションの説明が自動的に生成されます

```
1 public class Tester {
2     public static void main(String[] args) {
3         int i = 10;
4         do {
5             for (int j = i/2; j > 0; j--); {
6                 }
7             i-=2;
8         } while (i > 0);
9     }
10 }
```

Execute Mode, Version, Inputs & Arguments

JDK 11.0.4

CommandLine Arguments

Result

CPU Time: 0.14 sec(s), Memory: 29952 kilobyte(s)

最新問題: 134

与えられた :

```

public class Foo {
    public void foo(Collection arg) {
        System.out.println("Bonjour le monde!");
    }
}

and

public class Bar extends Foo {
    public void foo(Collection arg) {
        System.out.println("Hello world!");
    }
    public void foo(List arg) {
        System.out.println("Hola Mundo!");
    }
}

and

Foo f1 = new Foo();
Foo f2 = new Bar();
Bar b1 = new Bar();
List<String> li = new ArrayList<>();

```

正しいのはどれですか？ 3つ選択してください。）

- A. f1.foo ()はHola Mundoを出力します！
- B. f2.foo ()はBonjour le mondeを出力します！
- C. b1.foo ()はBonjour le mondeを出力します！
- D. f1.foo ()はBonjour le mondeを出力します！
- E. f2.foo ()はHello world！を出力します。
- F. f1.foo ()はHello world！を出力します。
- G. b1.foo ()はHello world！を出力します。
- H. f2.foo ()はHola Mundoを出力します！
- I. b1.foo ()はHola Mundoを出力します！

Answer: ([解答を表示する](#))

最新問題: 135

与えられた：

```
public class A {  
    private boolean checkValue(int val) {  
        return true;  
    }  
}  
  
and  
  
public class B extends A {  
    public int modifyVal(int val) {  
        if(checkValue(val)) {  
            return val;  
        } else {  
            return 0;  
        }  
    }  
    public static void Main(String[] args) {  
        B b = new B();  
        System.out.println(b.modifyVal(10));  
    }  
}
```

結果はどうなりますか？

- A. 何もない
- B. コンパイルに失敗します。
- C. 0
- D. java.lang.IllegalArgumentExceptionがスローされます。
- E. 10

Answer: B ([メッセージを残す](#))

```

1- public class A {
2-     private boolean checkValue(int val) {
3-         return true;
4-     }
5- }
6- and
7- public class B extends A {
8-     public int modifyVal(int val) {
9-         if(checkValue(val)) {
10-             return val;
11-         } else {
12-             return 0;
13-         }
14-     }
15-     public static void Main(String[] args) {
16-         B b = new B();
17-         system.out.println(b.modfiyVal (10));
18-     }
19- }

```

Execute Mode, Version, Inputs & Arguments

JDK 11.0.4

CommandLine Arguments

Result

CPU Time: sec(s), Memory: kilobyte(s)

```

/A.java:6: error: class, interface, or enum expected
and
^
1 error

```

最新問題: 136

与えられた :

```

public class Employee {
    private String name;
    private LocalDate birthday;
    // the constructors, getters, and setters methods go here
}

```

と

```

List<Employee> roster = new ArrayList<>();
// ...
Predicate<Employee> y = (Employee e) -> e.getBirthday()
    .isBefore(IsoChronology.INSTANCE.date(1989, 1, 1));
Set<String> s1 = roster.stream()
// Line 1

```

1行目のどのコードフラグメントがs1セットに、1月1日より前に生まれたすべての従業員の名前を含むようにします。
1989年？

```
A. .collect(Collectors.partitioningBy(y))
   .get(true)
   .stream()
   .map(Employee::getName)
   .collect(Collectors.toCollection(TreeSet::new));

B. .collect(Collectors.partitioningBy(y))
   .get(true)
   .map(Employee::getName)
   .collect(Collectors.toSet());

C. .collect(Collectors.partitioningBy(y, Collectors.mapping(
   Employee::getName, Collectors.toSet())));

D. .collect(Collectors.partitioningBy(y, Collectors.groupingBy(
   Employee::getName, Collectors.toCollection(TreeSet::new))));
```

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: B ([メッセージを残す](#))

有効な **1Z0-819** 問題集は GoShiken.com が提供された合格しやすい 1Z0-819 試験問題集！ GoShiken.com が最新の **1Z0-819** 試験問題集を提供しています。GoShiken.com 1Z0-819 試験問題は最新で、解答が正確でございます。最新の GoShiken.com 1Z0-819 問題集をゲットする人はこちら: <https://www.goshiken.com/Oracle/1Z0-819-mondaishu.html> (**29730%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

与えられた：

1行目に相当するステートメントはどれですか？

A. double totalSalary = list.stream ().map (e> e.getSalary () * ratio).reduce (bo).ifPresent (p> p.doubleValue ());)

B. double totalSalary = list.stream ().mapToDouble (e> e.getSalary () * ratio).sum;

C. double totalSalary = list.stream ().map (Employee::getSalary * ratio).reduce (bo).orElse (0.0);

D. double totalSalary = list.stream ().mapToDouble (e> e.getSalary () * ratio).reduce (starts、bo);

Answer: C ([メッセージを残す](#))

```
Employee.java * Main.java
1 import java.util.List;
2 import java.util.function.BinaryOperator;
3
4 public class Main {
5     public static void main(String... args) {
6         List<Employee> list = List.of(new Employee("John", 80000.0), new Employee("Scott", 90000.0));
7         double starts = 0.0;
8         double ratio = 1.0;
9         BinaryOperator<Double> bo = (a, b) -> a + b;
10        double totalSalary = list.stream().map(e -> e.getSalary() * ratio).reduce(starts, bo);
11        //line 1
12        System.out.println("Total salary = " + totalSalary);
13    }
14
15 }
16
console 1
Total salary = 170000.0
```

最新問題: 138

与えられた :

```
@Target(ElementType.METHOD)
@Retention(RetentionPolicy.RUNTIME)
public @interface AuthorInfo {
    String author() default "";
    String date();
    String[] comments() default {};
}
```

どちらが正しいですか？ 2つ選択してください。)

```
A. @AuthorInfo(date="1-1-2020", comments={ null })
   public class Hello {
       public void func() {}
   }

B. public class Hello {
   @AuthorInfo (date="1-1-2020. comments="Hello")
       public void func() {}
   }

C. public class Hello {
   @AuthorInfo
       public void func() {}
   }

D. @AuthorInfo(date="1-1-2020")
   public class Hello {
       public void func() {}
   }

E. public class Hello {
   @AuthorInfo(date="1-1-2020", author="Gandhi", comments={ "world" })
       public void func () {}
   }
```

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: 139

与えられた：

```
class Myclass {  
    public static void main(String [] args) {  
        System.out.println(arg[1] + "--" + arg[3] + "--" + arg[0]);  
    }  
}
```

このコマンドを使用して実行：

java Myclass MyCarは赤です

このクラスの出力は何ですか？

- A. 車-赤私の
- B. My--Car--is
- C. Myclass--Car--red
- D. My--is--java
- E. java--Myclass--My

Answer: ([解答を表示する](#))

最新問題: 140

与えられた：

```
public class Tester {  
    public static void main(String[] args) {  
        String s = "hat at store";  
        int x = s.indexOf("at");  
        s.substring(x + 3);  
        x = s.indexOf("at");  
        System.out.println(s + " " + x);  
    }  
}
```

結果はどうなりますか？

- A. 一度に1
- B. 実行時にindexOutOfBoundsExceptionがスローされます。
- C. 一度に0
- D. 4号店の帽子
- E. 店舗1の帽子

Answer: E ([メッセージを残す](#))

最新問題: 141

与えられた：

```
public method foo() throws FooException {  
    ...  
}
```

また、throws FooException句を省略すると、コンパイルエラーが発生します。

FooExceptionについて正しい説明はどれですか。

- A. FooExceptionはチェックされていません。
- B. fooの本体はFooExceptionのみをスローできます。
- C. FooExceptionはRuntimeErrorのサブクラスです。
- D. fooの本体は、FooExceptionまたはそのサブクラスの1つをスローできます。

Answer: D ([メッセージを残す](#))

最新問題: 142

機能的なインターフェースはどれですか？ 2つ選択してください。)

```
A. @FunctionalInterface
interface MyRunnable {
    public void run();
}

B. @FunctionalInterface
interface MyRunnable {
    public void run();
    public void call();
}

C. interface MyRunnable {
    public default void run() {}
    public void run(String s);
}

D. @FunctionalInterface
interface MyRunnable {
}

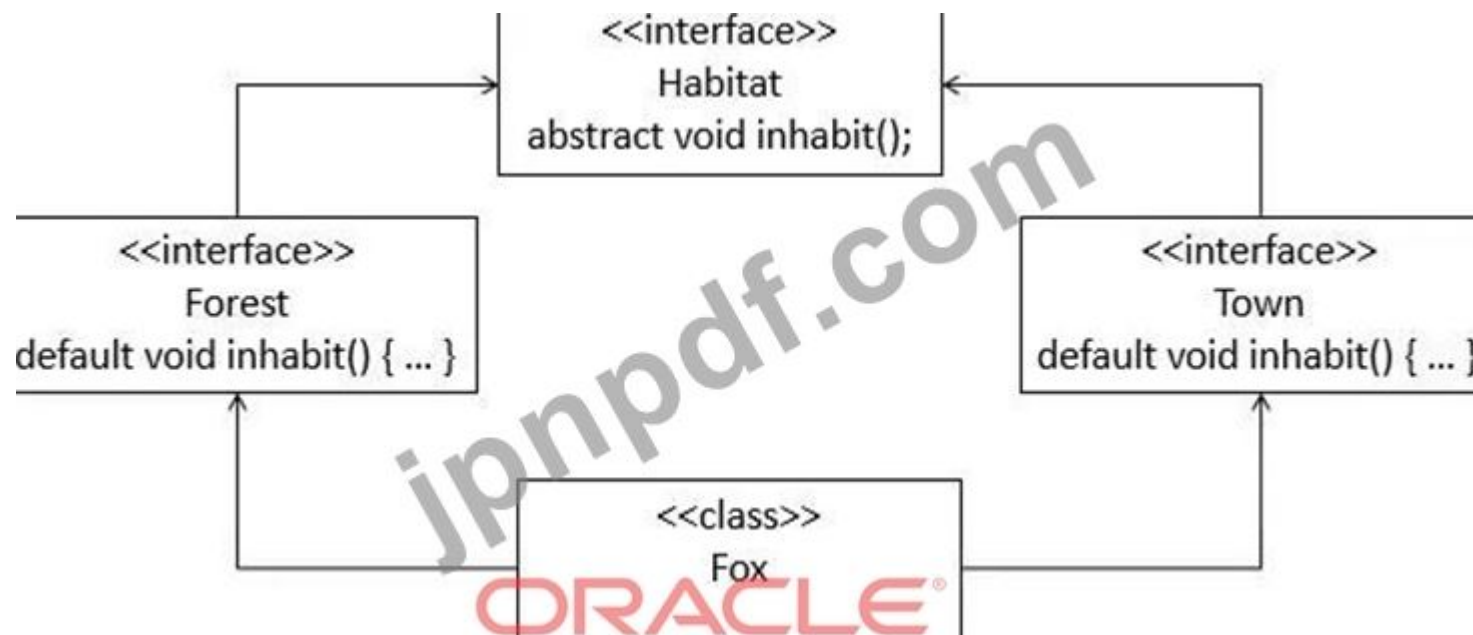
E. interface MyRunnable {
    @FunctionalInterface
    public void run();
}
```

- 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E
- E. オプションD

Answer: A,D ([メッセージを残す](#))

最新問題: 143

与えられた：



Foxクラスについて正しい説明はどれですか。

- A. Foxクラスは、呼び出しを試みない限り、inhabitメソッドをオーバーライドする必要はありません。
- B. Foxクラスは、ForestまたはTownインターフェースのいずれかを実装する必要がありますが、両方を実装することはできません。
- C. Foxが実装する最初のインターフェースからのinhabitメソッドの実装が優先されます。
- D. Foxクラスは、inhabitメソッドの実装を提供する必要があります。
- E. ForestとTownが互換性のある実装を提供している場合、Foxクラスはinhabitメソッドをオーバーライドする必要はありません。

Answer: ([解答を表示する](#))

最新問題: 144

与えられた：

```

public class Employee {
    private String name;
    private String neighborhood;
    private LocalDate birthday;
    private int salary;
}

and

List<Employee> roster = new ArrayList<>(...);
Map<String, Optional<Employee>> m = roster.stream()
// Line 1
  
```

1行目のどのコードフラグメントがmマップに、各近隣の給与が最も高い従業員を含むようにしますか？

- A)


```

.collect(Collectors.maxBy(Employee::getSalary,
    Collectors.groupingBy(Comparator.comparing(e -> e.getNeighborhood()))));
      
```
- B)


```

.collect(Collectors.groupingBy(Employee::getNeighborhood,
    Collectors.maxBy(Comparator.comparing(Employee::getSalary))));
      
```
- C)


```

.collect(Collectors.groupingBy(e -> e.getNeighborhood(),
    Collectors.maxBy((x, y) -> y.getSalary() - x.getSalary())));
      
```
- D)

```
.collect(Collectors.maxBy((x, y) -> y.getSalary() - x.getSalary(),  
Collectors.groupingBy(Employee::getNeighborhood)));
```

- 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: B ([メッセージを残す](#))

最新問題: 145

```
public class Employee {  
    private String name;  
    private String neighborhood;  
    // the constructors, setters, and getter methods go here  
}  
  
and  
  
List<Employee> roster = List.of(new Employee("John", "West town"),  
                                new Employee("Ray", "South town"),  
                                new Employee("Tom"),  
                                new Employee("Kenny", "West town"));
```

A)

```
Map<String, List<Employee>> e3 =  
    roster.stream()  
        .collect(Collectors.groupingBy(  
            e -> Optional.ofNullable(e.getNeighborhood())  
                .get()  
        ));
```

B)

```
Map<String, List<Employee>> e3 =  
    roster.stream()  
        .collect(Collectors.groupingBy(  
            e -> Optional.ofNullable(e.getNeighborhood())  
                .get()  
        ));
```

C)

```
Map<String, List<Employee>> e1 =  
    roster.stream()  
        .collect(Collectors.groupingBy(  
            (e -> Optional.ofNullable(e.getNeighborhood()))  
        ));
```

D)

```
Map<Object, List<Employee>> e2 =  
    roster.stream()  
        .collect(Collectors.groupingBy(  
            e -> Optional.ofNullable(e.getNeighborhood())  
        ));
```

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

Answer: D ([メッセージを残す](#))

最新問題: 146

コードフラグメントが与えられた場合：

```
int[] secA = { 2, 4, 6, 8, 10 };
int[] secB = { 2, 4, 8, 6, 10 };
int res1 = Arrays.mismatch(secA, secB);
int res2 = Arrays.compare(secA, secB);
System.out.print(res1 + " : " + res2);
```

結果はどうなりますか？

A. 2 :3

B. 3 :0

C. -1 :2

D. 2 :-1

Answer: ([解答を表示する](#))

最新問題: 147

ある会社には、2つのJava8jarファイルsales-3.10を含む既存のJavaアプリがあります。jarおよびclients-10.2.jar。

jarファイル、sales -8、10、jarはclients -10.2 jarのパッケージを参照しますが、clients-10.2jarはsales-8.10、jarのパッケージを参照しません。

彼らはclients-10.2.jarをモジュール化することを決定しました。

どのモジュール情報。Javaファイルは新しいライブラリバージョンのクライアントで機能します-10.3.jar?

A)

```
module com.company.clients{
    uses com.company.clients;
}
```

B)

```
module com.company.clients{
    requires com.company.clients;
}
```

C)

```
module com.company.clients {
    exports com.company.clients.Client;
}
```

D)

```
module com.company.clients {
    exports com.company.clients;
}
```

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: 148

与えられた：

```

public class Tester {
    private int x;
    private static int y;
    public static void main(String[] args) {
        Tester t1 = new Tester();
        t1.x = 2;
        Tester.y = 3;
        Tester t2 = new Tester();
        t2.x = 4;
        t2.y = 5;
        System.out.println(t1.x+", "+t1.y);
        System.out.println(t2.x+", "+Tester.y);
        System.out.println(t2.x+", "+t1.y);
    }
}

```

結果はどうなりますか？

- A. 2,34,34,5
- B. 2,34,54,5
- C. 2,54,54,5
- D. 2,34,54,3

Answer: C ([メッセージを残す](#))



最新問題: 149

この列挙型宣言を考えると：

```

1. enum Alphabet {
2.     A, B, C
3.
4. }

```

このコードを調べてください：

System.out.println Alphabet.getFirstLetter ();)

このコードをAで出力するには、3行目にどのコードを記述する必要がありますか？

- A. String getFirstLetter ({return A.toString ();})
- B. final String getFirstLetter ({return A.toString ();})
- C. static String getFirstLetter ({return Alphabet.values ([1] .toString ();})
- D. static String getFirstLetter ({return A.toString ();})

Answer: ([解答を表示する](#))

最新問題: 150

与えられた :

```
public class Foo {  
    private void print() {  
        System.out.println("Bonjour le monde!");  
    }  
    public void foo() {  
        print();  
    }  
}  
  
public class Bar extends Foo {  
    private void print() {  
        System.out.println("Hello world!");  
    }  
    public void bar() {  
        print();  
    }  
    public static void main(String... args) {  
        Bar b = new Bar();  
        b.foo();  
        b.bar();  
    }  
}
```

出力は何ですか？

- A. Hello world ! Bonjour le monde !
- B. Hello world ! Hello world !
- C. Bonjour le monde ! Hello world !
- D. Bonjour le monde ! Bonjour le monde !

Answer: ([解答を表示する](#))



最新問題: 151

与えられた :

```
List<String> list1 = new ArrayList<> ();
list1.add ("A");
list1.add ("B");
リストlist2 = List.copyOf (list1);
list2.add ("C");
List<List<String>> list3 = List.of (list1, list2);
System.out.println (list3);
結果はどうなりますか？
```

A. [[A、B]、[A、B]]

B. 実行時に例外がスローされます。

C. [[A、B]、[A、B、C]]

D. [[A、B、C]、[A、B、C]]

Answer: B ([メッセージを残す](#))

The screenshot shows the Oracle IDE interface. The code editor displays the following Java code:

```
11
12 public class Main {
13     public static void main(String[] args) {
14
15         List<String> list1 = new ArrayList<>();
16         list1.add("A");
17         list1.add("B");
18         List list2 = List.copyOf(list1);
19         list2.add("C");
20         List<List<String>> list3 = List.of(list1, list2);
21         System.out.println(list3);
22     }
23 }
24 }
25
```

Below the code editor, the "Execute Mode, Version, Inputs & Arguments" section shows the JDK version as "JDK 11.0.4". The "Interactive" checkbox is checked. The "Stdin Inputs" field is empty. The "CommandLine Arguments" field is also empty. The "Execute" button is visible.

The "Result" section shows the following output:

```
CPU Time: 0.16 sec(s), Memory: 32128 kilobyte(s)

Exception in thread "main" java.lang.UnsupportedOperationException
    at java.base/java.util.ImmutableCollections.uoe(ImmutableCollections.java:71)
    at java.base/java.util.ImmutableCollections$AbstractImmutableCollection.add(ImmutableCollections.java:75)
    at Main.main(Main.java:19)
```

有効な **1Z0-819** 問題集は GoShiken.com が提供された合格しやすい 1Z0-819 試験問題集！ GoShiken.com が最新の **1Z0-819** 試験問題集を提供しています。GoShiken.com 1Z0-819 試験問題は最新で、解答が正確でございます。最新の GoShiken.com 1Z0-819 問題集をゲットする人はこちら: <https://www.goshiken.com/Oracle/1Z0-819-mondaishu.html> (297**30%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: **152**

与えられた：

```
public static void main(String[] args) {
    final List<String> fruits =
        List.of("Orange", "Apple", "Lemmon", "Raspberry");
    final List<String> types =
        List.of("Juice", "Pie", "Ice", "Tart");
    final var stream =
        IntStream.range(0, Math.min(fruits.size(), types.size()))
            .mapToObj((i) -> fruits.get(i) + " " + types.get(i) );
    stream. forEach(System.out::println);
}
```

結果はどうなりますか？

- A. オレンジジュース
- B. コンパイルは失敗します。
- C. オレンジジュースアップルパイレモンアイスラズベリータルト
- D. プログラムは何も出力しません。

Answer: C (メッセージを残す)

```
12 public class Person {
13     public static void main (String[] args) {
14         final List<String> fruits =
15             List.of("Orange", "Apple", "Lemmon", "raspberry");
16         final List<String> types =
17             List.of("Juice", "Pie", "Ice", "Tart");
18         final var stream =
19             IntStream.range(0, Math.min(fruits.size(), types.size()))
20             .mapToObj ((i) -> fruits.get(i) + " " + types.get(i) );
21         stream. forEach(System.out::println);
22     }
23 }
24 }
```

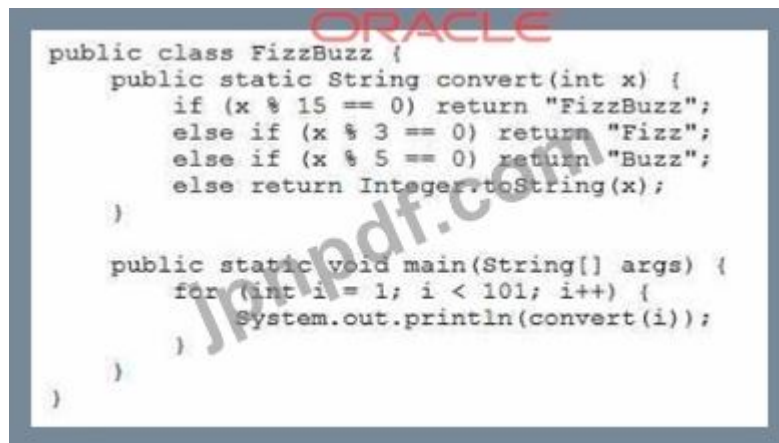
Result

compiled and executed in 1.227 sec(s)

```
Orange Juice
Apple Pie
Lemmon Ice
raspberry Tart
```

最新問題: 153

コードフラグメントが与えられた場合：



```
public class FizzBuzz {
    public static String convert(int x) {
        if (x % 15 == 0) return "FizzBuzz";
        else if (x % 3 == 0) return "Fizz";
        else if (x % 5 == 0) return "Buzz";
        else return Integer.toString(x);
    }

    public static void main(String[] args) {
        for (int i = 1; i < 101; i++) {
            System.out.println(convert(i));
        }
    }
}
```

forステートメントを置き換えるコードフラグメントはどれですか？

- A. `intstream.rangeclosed 1, 100).mapToObj {FizzBuzz :: convert}.forEach {System.out :: println};`
- B. `IntStream.range 1, 100).mapToObj {FizzBuzz :: convert}.forEach {System.out :: println};`
- C. `IntStream.rangeClosed 1, 100).map {FizzBuzz :: convert}.forEach {System.out :: println};`
- D. `IntStream.ranged, 100).map {FizzBuzz :: convert}.forEach {System.out :: println};`

Answer: C ([メッセージを残す](#))

最新問題: 154

この要件を考えると：

モジュール車両はモジュール部品に依存し、そのcom.vehicleパッケージを他のすべてのモジュールで使えるようにします。
どのmodule-info.java宣言が要件を満たしていますか？

```
module vehicle{
    requires part;
    exports com.vehicle;
}
```

```
module vehicle {
    requires part;
    uses com.vehicle;
}
```

```
module vehicle{
    requires part;
    exports com.vehicle to part;
```

ORACLE

```
module vehicle {
    requires com.vehicle;
    exports part;
}
```

- 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: 155

与えられた :

```
interface Abacus{
    public int calc (int a, int b);
}

public class Main {
    public static void main (String[] args) {
        int result = 0;
        // line 1
        result = aba.calc(10, 20);
        System.out.println(result);
    }
}
```

1つのコンパイルに独立して1行に挿入できる2つのコードはどれですか？

- A. そろばんaba = (int m, int n)-> {m * n};

B. そろばんaba = (int e, int f) -> {return e * f;};

C. そろばんaba = (int i, j) -> {return i * j};

D. そろばんaba = v, w -> x * y;

E. そろばんaba = a, b -> a * b;

Answer: C,E ([メッセージを残す](#))

最新問題: 156

与えられた :

```
1. interface Pastry {  
2.     void getIngredients();  
3. }  
4. abstract class Cookie implements Pastry {}  
5.  
6. class ChocolateCookie implements Cookie {  
7.     public void getIngredients() {}  
8. }  
9. class CoconutChocolateCookie extends ChocolateCookie {  
10.     void getIngredients(int x) {}  
11. }
```

どちらが正しいですか？

A. 4行目のエラーが原因でコンパイルが失敗します。

B. 2行目のエラーが原因でコンパイルが失敗します。

C. 10行目のエラーが原因でコンパイルが失敗します。

D. 7行目のエラーが原因でコンパイルが失敗します。

E. 9行目のエラーが原因でコンパイルが失敗します。

F. 6行目のエラーが原因でコンパイルが失敗します。

G. コンパイルは成功します。

Answer: F ([メッセージを残す](#))

最新問題: 157

与えられた :

```
class Item {
    public String name; public int count;
    public Item(String name, int count) {
        this.name = name; this.count = count;
    }
}
```

and the code fragment:

```
public class Test {
    public static void main(String[] args) {
        var items = List.of(new Item("A", 10), new Item("B", 2),
                             new Item("C", 12), new Item("D", 5), new Item("E", 6));
        // line 1
        System.out.println("There is an item for which the variable count is below zero.");
    }
}
```

変数カウントがゼロ未満のアイテムが含まれているアイテムリストを調べたいとします。

1行目のどのコードフラグメントがこれを達成していますか？

- A. `if (items.stream().filter(i -> count < 0).findFirst()) {}`
- B. `if (items.stream().allMatch(i -> count < 0) < 0) {}`
- C. `if (items.stream().anyMatch(i -> count < 0) < 0) {}`
- D. `if (items.stream().filter(i -> count < 0).findAny()) {}`

Answer: A ([メッセージを残す](#))

最新問題: 158

与えられた：

```
public class Test {
    public static void main(String[] args) {
        int x;
        int y = 5;
        if (y > 2) {
            x = ++y;
            y = x + 7;
        } else {
            y++;
        }
        System.out.print(x + " " + y);
    }
}
```

結果はどうなりますか？

- A. コンパイルエラー
- B. 0 5
- C. 6 13
- D. 5 12

Answer: A ([メッセージを残す](#))

```
1 public class Test {
2     public static void main (String[] args) {
3         int x;
4         int y = 5;
5         if (y > 2) {
6             x = ++y;
7             y = x + 7;
8         } else {
9             y++;
10        }
11        System.out.print(x + " "+y);
12    }
13 }
```

✖ variable x might not have been initialized

最新問題: 159

1行目のガベージコレクションに適格なThingオブジェクトはいくつありますか？

- A. 0
- B. 4
- C. 1
- D. 3
- E. 2

Answer: C ([メッセージを残す](#))

最新問題: 160

与えられた：

```
public class Main {
    public static void main(String[] args) {
        try {
            Path path = Paths.get("/u01/work/filestore.txt");
            boolean result = Files.deleteIfExists(path);
            if(result) System.out.println(path + "is deleted.");
            else System.out.println(path + "is not deleted.");
        } catch(IOException e) {
            System.out.println("Exception");
        }
    }
}
```

パス上のファイルが存在しないと想定します。結果はどうなりますか？

- A. コンパイルは失敗します。
- B. / u01 / work / filestore.txtは削除されません。
- C. 例外
- D. / u01 / work / filestore.txtが削除されます。

Answer: A ([メッセージを残す](#))



最新問題: 161

与えられた：

```
// line 1
lines.map(l -> l.toUpperCase())
    .forEach (line --> {
        try {
            Files.write(Paths.get("outputFile_to_path"),
line.getBytes(), StandardOpenOption.CREATE);
        } catch (IOException e) {
            e.printStackTrace();
        }
    });
catch (IOException e) {
    e.printStackTrace();
}
```

ファイルの読み取り時にStreamオブジェクトを取得する必要があります。1行目に挿入されたどのコードがこれを達成しますか？

- A. var lines = Files.lines Paths.get (INPUT_FILE_NAME) ;
- B. ストリームライン= Files.readAllLines Paths.get (INPUT_FILE_NAME) ;
- C. var lines = Files.readAllLines Paths.get (INPUT_FILE_NAME) ;
- D. Stream <String> lines = Files.lines (INPUT_FILE_NAME);

Answer: C (メッセージを残す)

最新問題: 162

与えられた：

```

ORACLE
interface Abacus{
    public int calc (int a, int b);
}

public class Main {
    public static void main (String[] args) {
        int result = 0;
        // line 1
        result = aba.calc(10, 20);
        System.out.println(result);
    }
}

```

1つのコンパイルに独立して1行に挿入できる2つのコードはどれですか？

- A. そろばんaba = (int e, int f)-> {return e * f;};
- B. そろばんaba = (int m, int n)-> {m * n};
- C. そろばんaba = (int i, j)-> {return i * j};
- D. そろばんaba = v, w-> x * y;
- E. そろばんaba = a, b)-> a * b;

Answer: C,E ([メッセージを残す](#))

最新問題: 163

これらのモジュール宣言を調べてください。

```

module ServiceAPI {
    exports com.example.api;
}

module ServiceProvider {
    requires ServiceAPI;
    provides com.example.api with com.myimpl.Impl;
}

module Consumer {
    requires ServiceAPI;
    uses com.example.api;
}

```

正しい2つのステートメントはどれですか？ (2つ選択してください。)

- A. コンシューマーモジュールにはServiceProviderモジュールが必要です。
- B. com.example.api APIを別のモジュールServiceAPIに配置すると、複数のプロバイダーモジュールを簡単にインストールできます。
- C. ServiceProviderモジュールは、実行時にcom.example.apiAPIを提供できる唯一のモジュールです。
- D. ServiceProviderモジュールは、com.example.api APIを使用するモジュール (Consumerなど)のIDを認識していません。
- E. ServiceProviderモジュールはcom.myimplパッケージをエクスポートする必要があります。

Answer: ([解答を表示する](#)**)**

最新問題: 164

与えられた：

```
public class Main {  
    public static void main(String[] args) {  
        for(int i = 0; i < args.length; i++) {  
            System.out.println(i + ")。" + args[i]);  
            switch(args[i]) {  
                case "one":  
                    continue;  
                case "two":  
                    i--;  
                    continue;  
                default:  
                    break;  
            }  
        }  
    }  
}
```

ORACLE®

このコマンドで実行：

javaメイン1、2、3

結果はどうなりますか？

- A. コンパイルは失敗します。
- B. 無限ループ印刷を作成します :0)。one1)。two1)。2...
- C. java.lang.NullPointerExceptionがスローされます。
- D. 0)。one1)。two2)。三
- E. 0)。1

Answer: B ([メッセージを残す](#))

最新問題: 165

与えられた：

```

package b;
public class Person {
    protected Person() { //line 1
    }
}

and

package a;
import b.Person;
public class Main { //line 2
    public static void main(String[] args) {
        Person person = new Person(); //line 3
    }
}

```

a.Mainが新しい人を割り当てることができるのはどちらですか？ 2つ選択してください。)

- A. 1行目で、アクセスmodifierPerson ({}を削除します
- B. 1行目で、アクセス修飾子をprivateprivate Person ({}に変更します。
- C. 1行目で、アクセス修飾子をpublicpublic Person ({}に変更します。
- D. 2行目で、アクセス修飾子をprotectedprotected classMain{}に変更します。
- E. 2行目でextendsPersonをMainclasspublic class Main extends Person {}に追加し、3行目を変更して新しいMain objectPerson person = new Main ()を作成します。

Answer: ([解答を表示する](#))

最新問題: 166

与えられた :

```

import java.util.*;

public class Main {
    static Map<String, String> map = new HashMap<>();
    static List<String> keys =
        new ArrayList<>(List.of("S", "P", "Q", "R"));
    static String[] values =
        {"senate", "people", "of", "rome"};

    static {
        for(var i = 0; i < keys.size(); i++) {
            map.put(keys.get(i), values[i]);
        }
    }

    public static void main(String[] args) {
        keys.clear();
        values = new String[0];
        System.out.println("Keys: " + keys.size() +
            " Values: " + values.length +
            " Map: " + map.size());
    }
}

```

結果はどうなりますか？

- A. コンパイルは失敗します。
- B. キー :0値 :0マップ :
- C. キー :4値 :4マップ :0
- D. キー :0値 :0マップ :0
- E. キー :4値 :4マップ :4

Answer: E ([メッセージを残す](#))

有効な **1Z0-819** 問題集は GoShiken.com が提供された合格しやすい 1Z0-819 試験問題集！ GoShiken.com が最新の **1Z0-819** 試験問題集を提供しています。GoShiken.com 1Z0-819 試験問題は最新で、解答が正確でございます。最新の GoShiken.com 1Z0-819 問題集をゲットする人はこちら: <https://www.goshiken.com/Oracle/1Z0-819-mondaishu.html> (**29730%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: 167

与えられた :

```
public class Plant { }

and

public class Tulip extends Plant { }

and

public class Garden {
    private static Plant plant;
    public static void main(String[] args) {
        plant = new Tulip();
        feed(plant);
        feed(plant);
    }
    public static void feed(Plant p) {
        if (p instanceof Tulip) {
            System.out.println("Take extra care");
        }
        p = null;
    }
}
```

結果はどうなりますか？

- A. 特に注意してください
細心の注意を払ってください
- B. 実行時に例外がスローされます
- C. 特に注意してください
- D. プログラムは何も出力しません。

Answer: ([解答を表示する](#))

最新問題: 168

与えられた :

```
1.  public class Secret {
2.      String[] names;
3.      public Secret(String[] names) {
4.          this.names = names;
5.      }
6.      public String[] getNames() {
7.          return names;
8.      }
9.  }
```

Java SEセキュリティガイドラインを実装する3つのアクションはどれですか？ 3つ選択してください。)

- A. 2行目を保護された揮発性のString[]名に変更します。
- B. 2行目をprivate final String[]names;に変更します。
- C. 3行目をprivate Secret \$String [] names){に変更します。
- D. 4行目をthis.names = names.clone ()に変更します。
- E. 7行目を変更してnames.clone ()を返します。
- F. getNames ()メソッド名をget \$ Names ()に変更します。
- G. 6行目をpublicsynchronized String [] getNames ()に変更します。

Answer: ([解答を表示する](#))

最新問題: 169

与えられた :

```
public class DNASynth {
    int aCount;
    int tCount;
    int cCount;
    int gCount;

    DNASynth(int a, int tCount, int c, int g){
        // line 1
    }
    int setCCount(int c){
        return c;
    }
    void setGCount(int gCount){
        this.gCount = gCount;
    }
}
```

1行目に挿入されたときにインスタンス変数を正しく変更する2行のコードはどれですか？ 2つ選択してください。)

- A. setCCount (c)= cCount;

- B. setGCount g);
- C. aCount = a;
- D. tCount = tCount;
- E. cCount = setCCount c);

Answer: C,D (メッセージを残す)

最新問題: 170

与えられた :

```
public interface A {  
    abstract void x();  
}  
  
and  
  
public abstract class B /* position 1 */ {  
    /* position 2 */  
    public void x() { }  
    public abstract void z();  
}  
  
and  
  
public class C extends B implements A {  
    /* position 3 */  
}
```

1つ以上のマークされた位置に挿入されたときに、クラスBとCをコンパイルできるコードはどれですか？

- A. public void z ({}//位置3
- B. @Override//位置2publicvoid z ({}//位置3
- C. @Override//位置3voidx ({}//位置3@Override//位置3publicvoidz ({}//位置3
- D. Aを実装します//位置1 @Override//位置2

Answer: (解答を表示する)

Valid 1Z0-819 Dumps shared by GoShiken.com for Helping Passing 1Z0-819 Exam! GoShiken.com now offer the **newest 1Z0-819 exam dumps**, the GoShiken.com 1Z0-819 exam **questions have been updated** and **answers have been corrected** get the **newest** GoShiken.com 1Z0-819 dumps with Test Engine here: <https://www.goshiken.com/Oracle/1Z0-819-mondaishu.html> (297 Q&As Dumps, **30%OFF** Special Discount: **Freepdfdumps**)