

Oracle.1z1-819.v2022-08-30.q110

試験コード:	1z1-819
試験名称:	Java SE 11 Developer
認定資格:	Oracle
無料問題数:	110
バージョン:	v2022-08-30
アクセス数:	1195
ページビュー数:	1100
https://www.jpnpdf.com/Oracle.1z1-819.v2022-08-30.q110-mondaishu.html	

最新問題: 1

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;  
            }  
        }  
    }  
}
```



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. 0)。1
- C. 無限ループ印刷を作成します :0)。one1)。two1)。2...
- D. java.lang.NullPointerExceptionがスローされます。
- E. 0)。one1)。two2)。三

Answer: C ([メッセージを残す](#))

最新問題: 2

コードフラグメントが与えられた場合：

```
Locale locale = Locale.US;  
// line 1  
double currency = 1_00.00;  
System.out.println(formatter.format(currency));
```

通貨の値を\$100.00として表示するとします。

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. NumberFormatフォーマッター= NumberFormat.getInstance (locale).getCurrency ();
- B. NumberFormatフォーマッター= NumberFormat.getInstance (locale);
- C. NumberFormatフォーマッター= NumberFormat.getCurrencyInstance (locale);
- D. NumberFormatフォーマッター= NumberFormat.getCurrency (locale);

Answer: ([解答を表示する](#)**)**

最新問題: 3

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

Answer: A,E ([メッセージを残す](#))

最新問題: 4

コードフラグメントが与えられた場合：

```

public class Main {
    public static void main(String[] args) {
        try {
            Path path = Paths.get("/u01/work");
            // line 1
            System.out.println(attributes.isDirectory());
        } catch (IOException e) {
            e.printStackTrace();
        }
    }
}

```

パスがディレクトリであるかどうかを調べたいとします。

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. BasicFileAttributes属性=ファイル、readAttributes (パス、FileAttributes、クラス);
- B. BasicFileAttributes属性= Files.readAttributes (path、BasicFileAttributes.class
- C. BasicFileAttributes属性= Files.getAttribute (path、" insdirectory ");
- D. BasicFileAttributes属性=ファイルisDirectory (パス);

Answer: A ([メッセージを残す](#))

最新問題: 5

与えられた :

```

public class Hello {
    class Greeting {
        void sayHi() {
            System.out.println("Hello world");
        }
    }
    public static void main(String... args) {
        // Line 1
    }
}

```

コードでHelloworldを印刷できるようにするには、1行目にどのコードを挿入する必要がありますか？

- A. Hello.Greeting myG = new Hello.Greeting (); myG.sayHi ();
- B. Hello myH = new Hello ();
Hello.Greeting myG = myH.new Hello.Greeting ();
myG.sayHi ();
- C. Hello myH = new Hello ();
Hello.Greeting myG = myH.new Greeting ();
myG.sayHi ();

D. Hello myH = new Hello () あいさつ myG=new Greeting () myG.sayHi ()

Answer: C ([メッセージを残す](#))

最新問題: 6

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](#))

```
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
```

Execute Mode, Version, Inputs & Arguments

JDK 11.0.4

Interactive

Stdin Inputs

CommandLine Arguments

Execute

Result

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)

最新問題: 7

与えられた:

```
ArrayList<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行目で、processData (dataReader) をtrywithresourcesで囲みます。
- B. 1行目で、各dataReaderを開くときにリソースでtryを使用します。
- C. 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。
- D. 4行目の後に、indexReader.close (を追加します。
- E. 3行目の後に、dataReader.close (を追加します。

Answer: A ([メッセージを残す](#))

最新問題: 8

与えられた:

```

public class Main {
    public static void main(String[] args) {
        int i = 1;
        for(String s : args) {
            System.out.println((i++) + " " + s);
        }
    }
}

```

このコマンドで実行：

javaメイン1、2、3

このクラスの出力は何ですか？

- A. 何もない
- B. コンパイルは失敗します。
- C. 1)1つ
- D. 1)1)2)23)3
- E. java.lang.ArrayIndexOutOfBoundsExceptionがスローされます。

Answer: D ([メッセージを残す](#))

最新問題: 9

与えられた：

```

// line 1
List<String> fruits = new ArrayList<>(List.of("apple", "orange", "banana"));
fruits.replaceAll(function);

```

1行目のどのステートメントがこのコードフラグメントのコンパイルを可能にしますか？

- A. 関数function = String :: toUpperCase;
- B. UnaryOperator関数= s> s.toUpperCase ()
- C. UnaryOperator <String> function = String :: toUpperCase;
- D. Function <String> function = m> m.toUpperCase ()

Answer: ([解答を表示する](#)**)**

```

1
2 import java.io.*;
3 import java.util.*;
4 import java.util.stream.Stream;
5 import java.util.function.Function;
6 import java.util.function.UnaryOperator;
7
8 class Hello {
9     public static void main(String[] args) {
10
11         UnaryOperator<String> function = String::toUpperCase;
12         List<String> fruits = new ArrayList<>(List.of("apple", "orange", "banana"));
13         fruits.replaceAll(function);
14
15     }
16 }

```

最新問題: 10

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

Answer: A ([メッセージを残す](#))

最新問題: 11

与えられた:

```
public class Test {  
    public static void main(String[] args) {  
        AnotherClass ac = new AnotherClass();  
        SomeClass sc = new AnotherClass();  
        ac = sc;  
        sc.methodA();  
        ac.methodA();  
    }  
}  
  
class SomeClass {  
    public void methodA() {  
        System.out.println("SomeClass#methodA()");  
    }  
}  
  
class AnotherClass extends SomeClass {  
    public void methodA() {  
        System.out.println("AnotherClass#methodA()");  
    }  
}
```

結果はどうなりますか？

- A. 実行時にClassCastExceptionがスローされます。

- B. AnotherClass#methodA (~~AnotherClass#methodA~~ ())
C. コンパイルは失敗します。
D. SomeClass#methodA (~~AnotherClass#methodA~~ ())
E. AnotherClass#methodA (~~SomeClass#methodA~~ ())
F. SomeClass#methodA (~~SomeClass#methodA~~ ())

Answer: C ([メッセージを残す](#))

```
1 public class Test {
2     public static void main (String[] args) {
3         AnotherClass ac = new AnotherClass();
4         × incompatible types: SomeClass cannot be converted to AnotherClass
5         ac = sc;
6         sc.methodA();
7         ac.methodA();
8     }
9 }
10 class SomeClass {
11     public void methodA() {
12         System.out.println("SomeClass#methodA()");
13     }
14 }
15
16 class AnotherClass extends SomeClass {
17     public void methodA() {
18         System.out.println("AnotherClass#methodA()");
19     }
20 }
```

最新問題: 12

与えられた :

```

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

    void setACount(int cCount){
        cCount = cCount;
    }
    void setTCount(){
        this.tCount = tCount;
    }
    int setCCount(){
        return cCount;
    }
    int setGCount(int g){
        gCount = g;
        return gCount;
    }
    void setAllCounts(int x){
        aCount = tCount = this.cCount = setGCount(x);
    }
}

```

フィールド値を変更する2つの方法はどれですか？ 2つ選択してください。）

- A. setTCount
- B. setACount
- C. setCCount
- D. setAllCounts
- E. setGCount

Answer: ([解答を表示する](#))

最新問題: 13

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: (解答を表示する)

```
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: 14

与えられた：

```
public class Main {
    public static void main(String[] args) {
        var numbers = List.of(1,2,3,4,5,6,7,8,9,10);
        Optional<Integer> result = numbers.stream().filter(x -> x % 3 != 0).reduce((i, j)
-> i + j);
        result.ifPresent(System.out::print); // line 1
    }
}
```

1行目について正しいのはどれですか？

- A. 値が存在しない場合、実行時にNoSuchElementExceptionがスローされます。
- B. 常にSystem.out :: printステートメントを実行します。
- C. 値が存在しない場合、実行時にNullPointerExceptionがスローされます。
- D. 値が存在しない場合、何も行われません。

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 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         var numbers = List.of(1,2,3,4,5,6,7,8,9,10);
16         Optional<Integer> result = numbers.stream().filter (x -> x % 3 != 0).reduce( (i, j) -> i + j);
17     }
18 }
19 }
```

Result

CPU Time: 0.18 sec(s), Memory: 33380 kilobyte(s)

JDoodle in Action... Running the program...
ORACLE

最新問題: 15

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をエクスポートします。

}

モジュールPrintServiceProvider{

B. PrintServiceAPIが必要です。

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

com.provider.PrintService;

}

C. PrintServiceAPIが必要です。

com.provider.PrintServiceを使用します。

}

モジュールPrintServiceProvider{

D. 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{

Answer: A ([メッセージを残す](#))

最新問題: 16

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

Answer: B ([メッセージを残す](#))

有効な **1z1-819** 問題集は GoShiken.com が提供された合格しやすい 1z1-819 試験問題集 !
GoShiken.com が最新の **1z1-819** 試験問題集を提供しています。GoShiken.com 1z1-819 試験問題は最新で、解答が正確でございます。最新の GoShiken.com 1z1-819 問題集をゲットする人はこちら: <https://www.goshiken.com/Oracle/1z1-819-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: 17

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 * (r * Math.pow (1 + r, n) / Math.pow (1 + r, n) - 1) ;`
- B. `m = p * ( * 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: ([解答を表示する](#)**)**

最新問題: 18

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. set (map)
- B. super.set (List<String> map)
- C. map.forEach (k,(v)-> set (v));)
- D. super.set (map.values ());)
- E. set (map.values ());)

Answer: ([解答を表示する](#))

最新問題: 19

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？

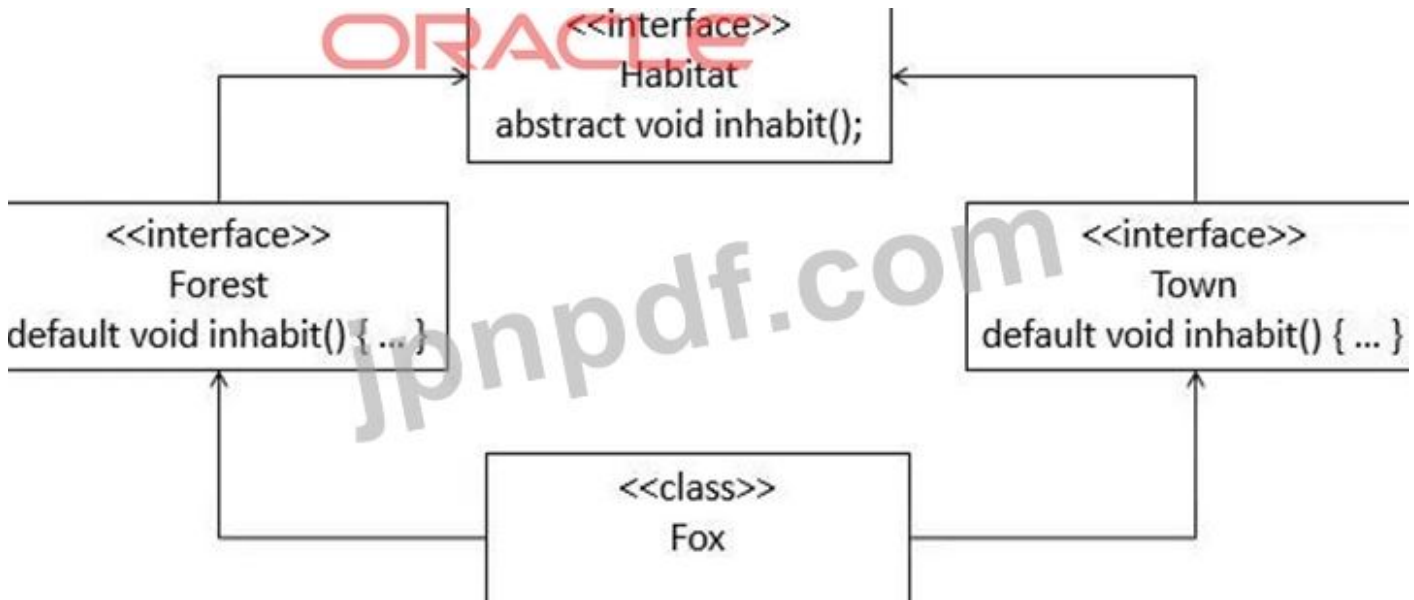
- A. そろばんaba = v、w-> x * y;
- B. そろばんaba = (int i,j)-> (return i * j);;
- C. そろばんaba = (int m、int n)-> {m * n};
- D. そろばんaba = (int e、int f)-> {return e * f};;

E. そろばん $a \times b \rightarrow a * b$;

Answer: B,E ([メッセージを残す](#))

最新問題: 20

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、inhabitメソッドの実装を提供する必要があります。
- B. 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。
- C. 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。
- D. 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。
- E. 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。

Answer: ([解答を表示する](#))

最新問題: 21

与えられた:

```
public class Main {
    public static void main(String[] args) {
        try (BufferedReader br = new BufferedReader(new InputStreamReader(System.in));) {
            String input = br.readLine();
            System.out.println("Input String was: " + input);
        } catch (IOException e) {
            e.printStackTrace();
        }
    }
}
```

どちらが正しいですか？

- A. System.outは標準出力ストリームです。ストリームは、System.outが呼び出されたときにのみ開かれます。

- B. System.inは他のストリームを再割り当てできません。
- C. System.outは、デフォルトではjava.io.OutputStreamのインスタンスです。
- D. System.inは標準の入カストリームです。ストリームはすでに開いています。

Answer: ([解答を表示する](#))

最新問題: 22

この列挙型宣言を考えると：

```
1. enum Letter {  
2.   ALPHA(100), BETA(200), GAMMA(300);  
3.   int v;  
4.   Letter(int v) { this.v = v; }  
5.   /* Insert code here */  
6. }
```

このコードを調べてください：

System.out.println (Letter.values () [1]);

このコードで200を出力するには、5行目にどのコードを記述する必要がありますか？

- A. public String toString () {return String.valueOf ALPHA.v);}
- B. public String toString () {return String.valueOf Letter.values () [1];}
- C. public String toString () {return String.valueOf v);}
- D. String toString () {return "200";}

Answer: C ([メッセージを残す](#))

```
14 - enum Letter {  
15     ALPHA(100), BETA(200), GAMMA(300);  
16     int v;  
17     Letter(int v) { this.v = v; }  
18     public String toString() { return String.valueOf(v);  
19  
20  
21  
22 }  
23 - public static void main (String[] args) {  
24     System.out.println(Letter.values() [1]);  
25 }  
26 }  
27  
28
```

Result

Compiled and executed in 1.099 sec(s)

200

ORACLE

最新問題: 23

与えられた：

```

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

Answer: A ([メッセージを残す](#))

最新問題: 24

与えられた :

```

public 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 }
```

ORACLE®

最新問題: 25

与えられた :

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

と

```
public class Tester {  
    public static void main(String[] args) {  
        Person p = new Person("Joe");  
        checkPerson(p);  
        System.out.println(p);  
        p = null;  
        checkPerson(p);  
        System.out.println(p);  
    }  
    public static Person checkPerson(Person p) {  
        if (p == null) {  
            p = new Person("Mary");  
        } else {  
            p = null;  
        }  
        return p;  
    }  
}
```

結果はどうなりますか？

ジョー

A. null

ヌル

B. メアリー

ジョー

C. 結婚する

ヌル

D. null

Answer: A ([メッセージを残す](#))



最新問題: 26

与えられた :

```

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

```

and

```

public class Tester {
    public static void main(String[] args) {
        Person p = null;
        checkPerson(p);
        System.out.println(p);
        p = new Person("Mary");
        checkPerson(p);
        System.out.println(p);
    }
    public static Person checkPerson(Person p) {
        if (p == null) {
            p = new Person("Joe");
        } else {
            p = null;
        }
        return p;
    }
}

```

結果はどうなりますか？

- A. ジョーマリー
- B. ジョーヌル
- C. nullnull
- D. nullMary

Answer: D ([メッセージを残す](#))

Console 1 Console 2 Console 3
null
Mary
Completed with exit code: 0

最新問題: 27

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

Answer: A ([メッセージを残す](#))

最新問題: 28

与えられた :

```
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。

C. 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。

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: B ([メッセージを残す](#))

最新問題: 29

適切なJDBCURLはどれですか？

A. http // localhost

mysql.jdbc :3306 / database

B. http://localhost.mysql.com:3306/database

C. jdbc:mysql.com // localhost :3306 / database

D. jdbc :mysql // localhost :3306 / database

Answer: D ([メッセージを残す](#))

最新問題: 30

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 a Java IDE with the following code in the editor:

```
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 IDE shows the execution mode as "JDK 11.0.4". The "Execute" button is highlighted. The "Result" section shows the following output:

```
Result  
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)
```

最新問題: 31

与えられた :

```
5. List<String> list1 = new ArrayList<>();
6. list1.add("A");
7. list1.add("B");
8. List<String> list2 = Collections.unmodifiableList(list1);
9. list1.add("C");
10. System.out.println(list1);
11. System.out.println(list2);
```

結果はどうなりますか？

A. [A、B、C]

11行目に例外がスローされます。

B. [A、B、C]

[A、B]

C. [A、B、C]

[A、B、C]

D. 9行目では、実行時に例外がスローされます。

Answer: C ([メッセージを残す](#))

説明

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

```
1 import java.io.*;
2 import java.util.*;
3 public class Tester {
4     public static void main(String[] args) {
5         List<String> list1 = new ArrayList<>();
6         list1.add("A");
7         list1.add("B");
8         List<String> list2 = Collections.unmodifiableList(list1);
9         list1.add("C");
10        System.out.println(list1);
11        System.out.println(list2);
12    }
13 }
14 }
```

Execute Mode, Version, Inputs & Arguments

JDK 11.0.4

CommandLine Arguments

ORACLE®

Result

CPU Time: 0.19 sec(s), Memory: 32452 kilobyte(s)

```
[A, B, C]
[A, B, C]
```

有効な 1z1-819 問題集は GoShiken.com が提供された合格しやすい 1z1-819 試験問題集！

GoShiken.com が最新の 1z1-819 試験問題集を提供しています。GoShiken.com 1z1-819 試験問題は最新で、解答が正確でございます。最新の GoShiken.com 1z1-819 問題集をゲットする

人はこちら: <https://www.goshiken.com/Oracle/1z1-819-mondaishu.html> (29730%OFF問題集
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:

```
public static void main(String[] args) {  
    try (Reader reader1 = new FileReader("File1.txt");  
        Reader reader2 = new FileReader("File2.txt");  
        Reader reader3 = new FileReader("File3.txt")) {  
  
        } catch (IOException ex) {  
            Logger.getLogger(Main.class.getName()).log(Level.SEVERE, null, ex);  
        }  
    // Line 1  
    System.out.println("Done");  
}
```

実行して3つのファイルがすべて存在する場合、1行目の各リーダーの状態はどうなりますか？

- A. 3つのリーダーすべてが閉じられました。
- B. コンパイルは失敗します。
- C. 3つのリーダーはすべて開いたままです。
- D. reader1のみが閉じられました。

Answer: B ([メッセージを残す](#))

最新問題: 33

与えられた:

```
public class Foo {  
    private final ReentrantLock lock = new ReentrantLock();  
    private State state;  
    public void foo() throws Exception {  
        try {  
            lock.lock();  
            state.mutate();  
        }  
        finally {  
            lock.unlock();  
        }  
    }  
}
```

Fooクラスのスレッドを安全にするために何が必要ですか？

- A. 変更は必要ありません。
- B. ロックの宣言を静的にします。
- C. ロックコンストラクター呼び出しを新しいReentrantLock (true)に置き換えます。
- D. ロックの宣言をfooメソッド内に移動します。

Answer: C ([メッセージを残す](#))

参照：

/how-to-make-java-class-thread-safe

最新問題: 34

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

Answer: E ([メッセージを残す](#))

最新問題: 35

与えられた：

```
public class Person {
    private String name = "Joe Bloggs";
    public Person(String name) {
        this.name = name;
    }
    public String toString() {
        return name;
    }
}

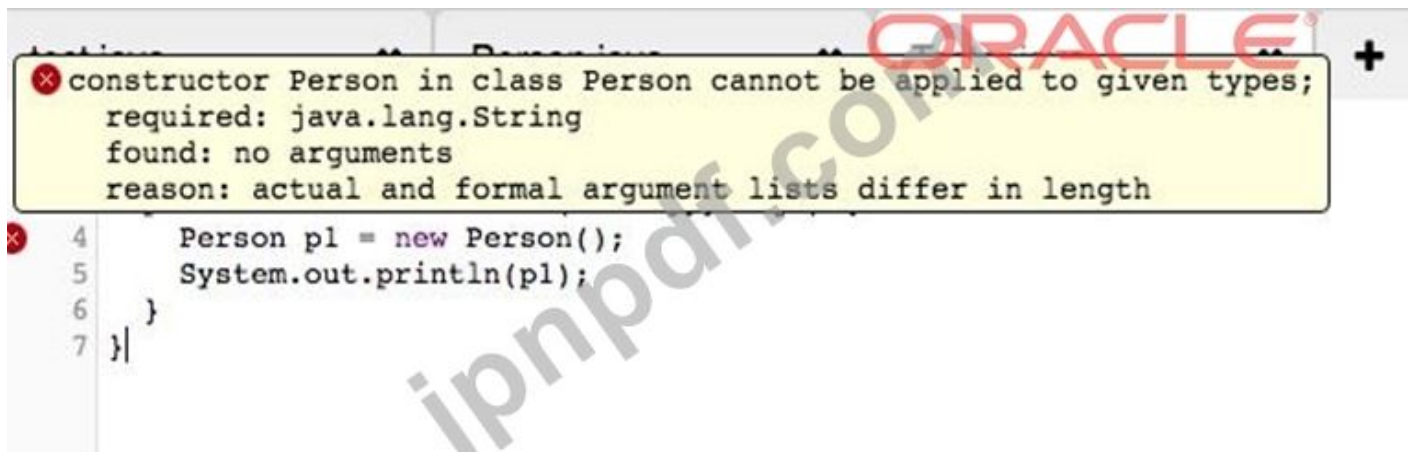
and

public class Tester {
    public static void main(String[] args) {
        Person p1 = new Person(); // line 1
        System.out.println(p1);
    }
}
```

結果はどうなりますか？

- A. null
- B. ジョーブログ
- C. 1行目のエラーが原因でコンパイルが失敗します。
- D. p1

Answer: C ([メッセージを残す](#))



最新問題: 36

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

Answer: ([解答を表示する](#))

最新問題: 37

コードフラグメントが与えられた場合 :

```
public class Main {  
    public static void main(String... args) {  
        List<String> list1 = new ArrayList<>(  
            List.of("Plane", "Automobile", "Motorcycle"));  
        List<String> list2 = new ArrayList<>(List.copyOf(list1));  
  
        list1.sort((String item1, String item2) -> item1.compareTo(item2));  
        list2.sort((String item1, String item2) -> item1.compareTo(item2));  
        System.out.println(list1.equals(list2));  
    }  
}
```

結果はどうなりますか？

- A. java.lang、UnsupportedOperationExceptionがスローされます。
- B. 偽
- C. 真
- D. java.lang.NullPointerExceptionがスローされます。

Answer: A ([メッセージを残す](#))

最新問題: 38

与えられた :

```
public interface EulerInterface {
    double getEulerValue();
}

public class EulerLambda {
    public static void main(String[] args) {
        EulerInterface myEulerInterface;
        myEulerInterface = () -> "2.71828";
        System.out.println("Value of Euler = " + myEulerInterface.getEulerValue());
    }
}
```

結果はどうなりますか？

- A. オイラーの値= 2.71828
- B. ランタイム例外をスローします。
- C. コードはコンパイルされません。
- D. オイラーの値= "2.71828"

Answer: C ([メッセージを残す](#))

最新問題: 39

与えられた :

```
interface MyInterface1 {
    public int method() throws Exception;
    private void pMethod() { /* an implementation of pMethod */ }
}

interface MyInterface2 {
    public static void sMethod() { /* an implementation of sMethod */ }
    private boolean equals();
}

interface MyInterface3 {
    public void method();
    public void method(String str);
}

interface MyInterface4 {
    public void dMethod() { /* an implementation of dMethod */ }
    public void method();
}

interface MyInterface5 {
    public static void sMethod();
    public void method(String str);
}
```

ラムダ式で利用できる2つの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. MyInterface5
- B. MyInterface3
- C. MyInterface2
- D. MyInterface1
- E. MyInterface4

Answer: B,C ([メッセージを残す](#))

最新問題: 40

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> Iterable <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> List <T> foo (Collection <T> arg){...}`はFoo.fooをオーバーライドします。
- D. `public <T> Collection <T> foo (Stream <T> arg){...}`はFoo.fooをオーバーロードします。
- E. `public Collection <String> foo (Collection <String> 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: A,C ([メッセージを残す](#))

最新問題: 41

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。)

- ORACLE**
- 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D

B. オプションE

C. オプションA

D. オプションC

E. オプションB

Answer: B,D ([メッセージを残す](#))

最新問題: 42

与えられた:

ORACLE

```
public class Foo {  
    public static void main(String... args) {  
        for (var x : args) {  
            System.out.println(x);  
        }  
    }  
}
```

ローカル変数xのタイプは何ですか？

- A. 文字列[]
- B. キャラクター
- C. char
- D. 文字列

Answer: D ([メッセージを残す](#))

最新問題: 43

与えられた :

```
public interface AdaptorFirst {  
    void showFirst();  
}
```

showFirst (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？

A)

```
public abstract class MainClass implements AdaptorFirst {  
    public String showFirst(){  
        return "first";  
    }  
}
```

B)

```
public abstract class MainClass implements AdaptorFirst {  
    public void showFirst(){  
        System.out.println("first");  
    }  
}
```

C)

```
public class MainClass implements AdaptorFirst {  
    void showFirst();  
}
```

D)

```
public class MainClass implements AdaptorFirst {  
    private void showFirst(){  
        System.out.println("first");  
    }  
}
```

E)

```
public abstract class MainClass implements AdaptorFirst {  
    public abstract void showFirst();  
}
```

F)

```
public class MainClass implements AdaptorFirst {  
    public void showFirst(){  
        System.out.println("first");  
    }  
}
```

- A. オプションF
- B. オプションA
- C. オプションC

- D. オプションB
- E. オプションE
- F. オプションD

Answer: ([解答を表示する](#))

最新問題: 44

コードフラグメントが与えられた場合：

パスcurrentFile=Paths.get ("/ strike / Exam / temp.txt");

パスoutputFile=パスget ("/ strike / Exam / new.txt");

パスディレクトリ=Paths.get ("/ strike /");

Files.copy (currentFile、outputFile);

Files.copy (outputFile、directory);

Files.delete (outputFile);

/scratch/exam/temp.txtファイルが存在します。/scratch/exam/new.txtファイルと/scratch/new.txtファイルは存在しません。

結果はどうなりますか？

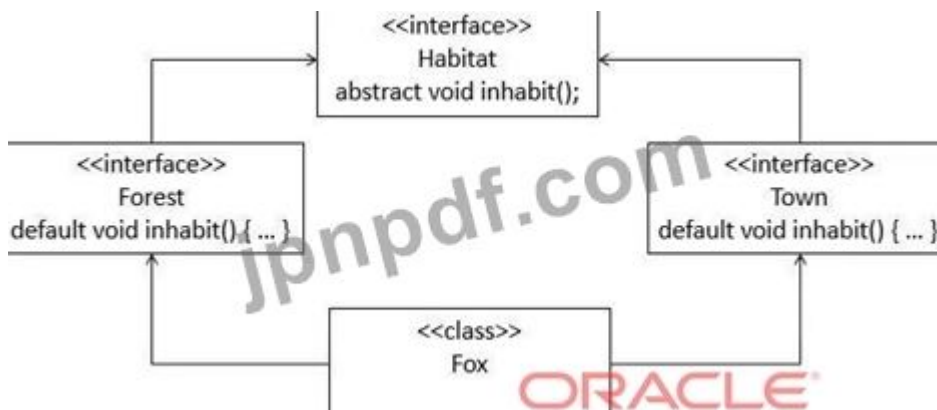
- A. /scratch/exam/new.txtと/scratch/new.txtが削除されます。
- B. プログラムはFileAlreadyExistsExceptionをスローします。
- C. プログラムはNoSuchFileExceptionをスローします。
- D. /scratch/exam/new.txtのコピーが/scratchディレクトリに存在し、/scratch/exam/new.txtが削除されます。

Answer: C ([メッセージを残す](#))

```
27 public class Main {
28     public static void main(String[] args) {
29         Path currentFile = Paths.get("/scratch/exam/temp.txt");
30         Path outputFile = Paths.get("/scratch/exam/new.txt");
31         Path directory = Paths.get("/scratch/");
32
33         Files.copy(currentFile, outputFile);
34         Files.copy(outputFile, directory);
35         Files.delete (outputFile);
36     }
37 }
38
```

最新問題: 45

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. 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。
- B. 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。
- C. 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。
- D. 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。
- 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: A ([メッセージを残す](#))

最新問題: 46

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。)

A. **ORACLE**
 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");
 }
 }

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

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

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

Answer: A,B,D ([メッセージを残す](#))

有効な 1z1-819 問題集は GoShiken.com が提供された合格しやすい 1z1-819 試験問題集！
GoShiken.com が最新の 1z1-819 試験問題集を提供しています。GoShiken.com 1z1-819 試験問題は最新で、解答が正確でございます。最新の GoShiken.com 1z1-819 問題集をゲットする人はこちら: <https://www.goshiken.com/Oracle/1z1-819-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

開発組織で使用されているツールの機能的なバグに取り組んでいます。調査の結果、この許可を含むセキュリティポリシーファイルを使用してツールが実行されていることがわかりました。

```
grant codebase "file:${klib.home}/j2se/home/klib.jar" {  
    permission java.security.AllPermission;  
};
```

あなたはどのような行動を取るべきですか？

- A. 内部ツールであり、一般に公開されていないため、何もありません。
- B. 必要な権限だけをリストすることは、継続的なメンテナンスの課題になるため、何もありません。
- C. 調査しているバグとは関係がないため、何もありません。
- D. 助成金が多すぎるため、助成金を削除します。
- E. 付与された過剰なアクセス許可を参照するツールに対してセキュリティバグを報告します。

Answer: E ([メッセージを残す](#))

最新問題: 48

与えられた：

```
public class Main {  
    public static void main(String[] args) {  
        Optional<String> value = createValue();  
        String str = value.orElse("Duke");  
        System.out.println(str);  
    }  
    static Optional<String> createValue() {  
        String s = null;  
        return Optional.ofNullable(s);  
    }  
}
```

出力は何ですか？

- A. null
- B. 実行時にNoSuchElementExceptionがスローされます。
- C. デューク
- D. 実行時にNullPointerExceptionがスローされます。

Answer: C ([メッセージを残す](#))

```

14
15 public class Main {
16     public static void main(String[] args) {
17         Optional<String> value = createValue();
18         String str = value.orElse ("Duke");
19         System.out.println(str);
20     }
21     static Optional<String> createValue() {
22         String s = null;
23         return Optional.ofNullable(s);
24     }
25 }
26

```

result

CPU Time: 0.15 sec(s), Memory: 32572 kilobyte(s)

Duke

最新問題: 49

与えられた :

```

public class Test {
    public static void main(String[] args) {
        AnotherClass ac = new AnotherClass();
        SomeClass sc = new AnotherClass();
        ac = sc;
        sc.methodA();
        ac.methodA();
    }
}
class SomeClass {
    public void methodA() {
        System.out.println("SomeClass#methodA()");
    }
}
class AnotherClass extends SomeClass {
    public void methodA() {
        System.out.println("AnotherClass#methodA()");
    }
}

```

結果はどうなりますか？

- A. 実行時にClassCastExceptionがスローされます。
- B. AnotherClass#methodA (AnotherClass#methodA ())
- C. コンパイルは失敗します。
- D. SomeClass#methodA (AnotherClass#methodA ())
- E. AnotherClass#methodA (\$SomeClass#methodA ())
- F. SomeClass#methodA (\$SomeClass#methodA ())

Answer: C ([メッセージを残す](#))

```
1 public class Test {
2     public static void main (String[] args) {
3         AnotherClass ac = new AnotherClass();
4         // incompatible types: SomeClass cannot be converted to AnotherClass
5         ac = sc;
6         sc.methodA();
7         ac.methodA();
8     }
9 }
10 class SomeClass {
11     public void methodA() {
12         System.out.println("SomeClass#methodA()");
13     }
14 }
15 }
16 class AnotherClass extends SomeClass {
17     public void methodA() {
18         System.out.println("AnotherClass#methodA()");
19     }
20 }
```

ORACLE

最新問題: 50

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 (0).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 (0).orElse (0.0);`
- D. `double totalSalary = list.stream () mapToDouble e-> e.getSalary () * ratio).reduce (starts, 0);`

Answer: C ([メッセージを残す](#))

```
Employee.java x Main.java x +
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 x
Total salary = 170000.0
Completed with exit code: 0

最新問題: 51

与えられた :

```

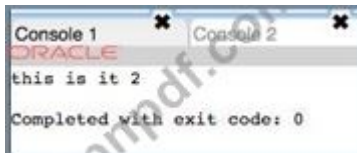
public class Tester {
    public static void main(String[] args) {
        String s = "this is it";
        int x = s.indexOf("is");
        s.substring(x+3);
        x = s.indexOf("is");
        System.out.println(s+" "+x);
    }
}

```

結果はどうなりますか？

- A. それは1ですか
- B. 実行時にIndexOutOfBoundsExceptionがスローされます。
- C. それは0ですか
- D. これは2です
- E. これは3です

Answer: ([解答を表示する](#))



最新問題: 52

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](#))

```

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

```

ORACLE
Execute Mode, Version, Inputs & Arguments

JDK 11.0.4 ☐ Interactive Stdin Input

CommandLine Arguments

Execute ...

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

orangebanana lemon

最新問題: 53

組織は、クラウドの顧客がmlib.jarを利用できるようにします。mlib.jarのコードクリーンアッププロジェクトに取り組んでいる間、顧客はこのメソッドを確認します。

```

public void enableService(String hostName, String portNumber) throws IOException {
    this.transportSocket = new Socket(hostName, portNumber);
}

```

顧客がアクセスできる方法の要件を満たすために、この方法にどのようなセキュリティ対策を追加する必要がありますか？

Insert this code before the call to new Socket:

```

    hostName = new String(hostName);
    portNumber = new String(portNumber);

```

- A. ソケットを開く前にhostNameパラメーターとportNumberパラメーターを検証するメソッドを作成します。
- B. enableServiceをプライベートにします。
- C.
- D. 新しいSocketへの呼び出しをAccessController.doPrivilegedブロックで囲みます。

Answer: D ([メッセージを残す](#))

最新問題: 54

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.jartool
- B. java.logging
- C. java.desktop

- D. javafx
- E. jdk.httpserver

Answer: B,E ([メッセージを残す](#))

最新問題: 55

内容を考えると :

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

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D

B. オプションB

C. オプション

D. オプションA

E. オプションC

Answer: A,C ([メッセージを残す](#))

最新問題: 56

与えられた :

```
public interface API { //line 1
    public void checkValue(Object value)
        throws IllegalArgumentException; //line 2
    public boolean isValueANumber(Object val) {
        if(val instanceof Number) {
            return true;
        }else {
            try {
                Double.parseDouble(val.toString());
                return true;
            }catch (NumberFormatException ex) {
                return false;
            }
        }
    }
}
```

このクラスをコンパイルするには、どの2つの変更を行う必要がありますか？ 2つ選択してください。)

- A. 行1を変更してjava.lang.AutoCloseableを拡張します public interface API extends AutoCloseable {
- B. 2行目のアクセス修飾子をprotected protected void checkValue (Object value) throws IllegalArgumentExceptionに変更します。
- C. 1行目をclass public class API{に変更します。
- D. 1行目を抽象クラスに変更します public abstract class API {
- E. 2行目を抽象メソッドに変更します public abstract void checkValue (Object value) throws IllegalArgumentException;

Answer: ([解答を表示する](#))

最新問題: 57

与えられた :

```

public class Menu {
    enum Machine{
        AUTO("Truck"), MEDICAL("Scanner");
        private String type;
        private Machine(String type) {
            this.type = type;
        }
        private void setType(String type) {
            this.type = type;
        }
        private String getType() {
            return type;
        }
    }
    public static void main(String[] args) {
        Machine.AUTO.setType("Sedan");
        for (Machine p : Machine.values()) {
            System.out.println(p + ": " + p.getType());
        }
    }
}

```

A) 実行時に例外がスローされます。

B)

```

AUTO: Sedan
MEDICAL: Scanner

```

C) 2行目のエラーが原因でコンパイルが失敗します。

D) 1行目のエラーが原因で、コンパイルが失敗します。

E)

```

AUTO: Truck
MEDICAL: Scanner

```

F)

3行目のエラーが原因で、コンパイルが失敗します。

A. オプションA

B. オプションF

C. オプションB

D. オプションD

E. オプションC

F. オプションE

Answer: A ([メッセージを残す](#))

最新問題: 58

与えられた :

```

class Super {
    static String greeting() { return "Good Night"; }
    String name() { return "Harry"; }
}

and

class Sub extends Super {
    static String greeting() { return "Good Morning"; }
    String name() { return "Potter"; }
}

and

class Test {
    public static void main(String[] args) {
        Super s = new Sub();
        System.out.println(s.greeting() + ", " + s.name());
    }
}

```

結果はどうなりますか？

- A. おはよう、ポッター
- B. おやすみなさい、ポッター
- C. おはよう、ハリー
- D. おやすみなさい、ハリー

Answer: B ([メッセージを残す](#))



最新問題: 59

与えられた :

```

public class Main {
    public static void main(String[] args) {
        Consumer consumer = msg -> System.out::print; // line 1
        consumer.accept("Hello Lambda !");
    }
}

```

このコードにより、コンパイルエラーが発生します。

コンパイルを成功させるには、1行目にどのコードを挿入する必要がありますか？

- A. コンシューマーコンシューマー= msg-> {return System.out.print (msg);};
- B. コンシューマーコンシューマー= var arg> {System.out.print (arg)};
- C. コンシューマーコンシューマー= \$string args> System.out.print (args);
- D. コンシューマーコンシューマー= System.out :: print;

Answer: D ([メッセージを残す](#))

```

1 import java.util.*;
2 import java.io.*;
3 import java.nio.file.*;
4 import java.util.List;
5 import java.util.function.Consumer;
6
7 public class Main {
8
9     public static void main(String[] args) {
10         Consumer consumer = System.out::print;
11         consumer.accept("Hello Lambda !");
12     }
13 }

```

Execute Mode, Version, Inputs & Arguments

JDK 11.0.4

CommandLine Arguments

Result

CPU Time: 0.16 sec(s), Memory: 32896 kilobyte(s)

Hello Lambda !

最新問題: 60

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:
            case FRIDAY:
                System.out.println("Working");
            default:
                System.out.println("Unknown");
        }
    }
}

```

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: 61

与えられた：

```
class MyPersistenceData {  
    String str;  
    private void methodA() {  
        System.out.println("methodA");  
    }  
}
```

あなたはjavを実装したい

a。Io、MyPersisteneDataクラスへのシリアル化可能なインターフェイス。
どのメソッドをオーバーライドする必要がありますか？

- A. writeExternalメソッド
- B. 何もない
- C. readExternalおよびwriteExternalメソッド
- D. readExternalメソッド

Answer: C ([メッセージを残す](#))

有効な 1z1-819 問題集は GoShiken.com が提供された合格しやすい 1z1-819 試験問題集！
GoShiken.com が最新の 1z1-819 試験問題集を提供しています。GoShiken.com 1z1-819 試験問題は最新で、解答が正確でございます。最新の GoShiken.com 1z1-819 問題集をゲットする人はこちら: <https://www.goshiken.com/Oracle/1z1-819-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: 62

数値、通貨、およびパーセンテージのフォーマットに使用されるデフォルトのロケールを設定する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. `Locale.setDefault ("en_CA");`
- B. `Locale.setDefault (Locale.Category.FORMAT, "zh-CN");`
- C. `Locale.setDefault ("es", Locale.US);`
- D. `Locale.setDefault (Locale.SIMPLIFIED_CHINESE);`
- E. `Locale.setDefault (Locale.Category.FORMAT, Locale.CANADA_FRENCH);`

Answer: A,E ([メッセージを残す](#))

最新問題: 63

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 `RuntimeError` のサブクラスです。
- B. `FooException` はチェックされていません。
- C. `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。
- D. `foo` の本体は `FooException` のみをスローできます。

Answer: C ([メッセージを残す](#))

最新問題: 64

このメソッド宣言について考えてみましょう。

```
void setSessionUser(Connection conn, String user) throws SQLException {  
    Statement stmt = conn.createStatement();  
    String sql = <EXPRESSION>;  
    stmt.execute();  
}
```

- A) `SETSESSIONAUTHORIZATION`」+ユーザー
- B) `SETSESSIONAUTHORIZATION`」+`stmt.quoteIdentifier (user)`

A または B は `<EXPRESSION>` の正しい代替品ですか？ その理由は何ですか？

- A. B. 呼び出し元のコードによって提供された値を引用すると、SQL インジェクションが防止されるためです。
- B. A と B は機能的に同等です。
- C. A、呼び出し元のコードによって提供されたユーザーの値を正確に送信するため。
- D. A、識別子を引用符で囲む必要がないため。
- E. B、呼び出し元のコードによって提供されるすべての値を引用符で囲む必要があるため。

Answer: C ([メッセージを残す](#))

最新問題: 65

与えられた：

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。privateintc;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: A ([メッセージを残す](#))

```
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: 66

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

Answer: B ([メッセージを残す](#))

最新問題: 67

与えられた :

```
List<String> list = ... ;  
list.forEach( x -> { System.out.println(x); } );
```

xのタイプは何ですか？

- A. 文字列
- B. リスト<文字列>
- C. リスト<キャラクター>
- D. char

Answer: A ([メッセージを残す](#))

最新問題: 68

与えられた :

```
class ConSuper {  
    protected ConSuper() {  
        this(2);  
        System.out.print("3");  
    }  
    protected ConSuper(int a) {  
        System.out.print(a);  
    }  
}  
  
and  
  
public class ConSub extends ConSuper {  
    ConSub() {  
        this(4);  
        System.out.print("1");  
    }  
    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. 2341
- B. 214
- C. 2134
- D. 234

Answer: C ([メッセージを残す](#))

最新問題: 69

与えられた :

```
class Employee {  
    String office;  
}
```



and the code fragment:

```
5. public class HRAApp {  
6.     var employee = new ArrayList<Employee>();  
7.     public var display() {  
8.         var employee = new Employee();  
9.         var offices = new ArrayList<>();  
10.        offices.add("Chicago");  
11.        offices.add("Bangalore");  
12.        for (var office : offices) {  
13.            System.out.print("Employee Location"+ office);  
14.        }  
15.    }  
16. }
```

コンパイルエラーの原因となる2行はどれですか？ 2つ選択してください。)

- A. 12行目
- B. 7行目
- C. 9行目
- D. 8行目
- E. 6行目

Answer: ([解答を表示する](#))

最新問題: 70

コードフラグメントが与えられた場合：

パスソース=Paths.get ("/ repo / a / a.txt");

パスの宛先=Paths.get ("/ repo");

Files.move (ソース、宛先); // ライン1

Files.delete (ソース); // 2行目

ソースファイルと宛先フォルダーが存在すると仮定すると、結果はどうなりますか？

- A. a.txtの名前がrepoに変更されました。
- B. /repo/a/a.txtのコピーが/ repoディレクトリに移動され、/ repo / a/a.txtが削除されます。
- C. java.nio.file.NoSuchFileExceptionが2行目にスローされます。
- D. java.nio.file.FileAlreadyExistsExceptionが1行目にスローされます。

Answer: B ([メッセージを残す](#))

最新問題: 71

入力を安全に検証する2つはどれですか？ 2つ選択してください。)

- A. 入力がすでに検証されていると仮定します。

- B. 値の数値範囲チェックをデータベースに委任します。
- C. 信頼できるドメイン固有のライブラリを使用して入力を検証します。
- D. 検証に合格するために、必要に応じて入力値を変更します。
- E. 有効な文字と入力値のみを受け入れます。

Answer: B,E ([メッセージを残す](#))

最新問題: 72

与えられた :

```
public class Tester {
    public static void main(String[] args) {
        StringBuilder sb = new StringBuilder(5);
        sb.append("HOWDY");
        sb.insert(0, ' ');
        sb.replace(3, 5, "LL");
        sb.insert(6, "COW");
        sb.delete(2, 7);
        System.out.println(sb.length());
    }
}
```

結果はどうなりますか？

- A. 4
- B. 3
- C. 実行時に例外がスローされます。
- D. 5

Answer: D ([メッセージを残す](#))

```
6 public class Tester {
7     public static void main(String[] args) {
8         StringBuilder sb = new StringBuilder (5);
9         sb.append ("HOWDY");
10        sb.insert (0, ' ');
11        sb.replace(3, 5, "LL");
12        sb.insert (6, "COW");
13        sb.delete(2, 7);
14        System.out.println(sb.length());
15    }
16 }
```

(command line arguments)

COMPILE & EXECUTE

PASTE SOURCE

Successfully compiled /tmp/java_82Tlan/Tester.java <-- main method

5

最新問題: 73

与えられた :

```

import java.io.FileNotFoundException;
import java.io.IOException;

public class Tester {
    public static void main(String[] args) {
        try {
            doA();
        } //line 1
    }
    private static void doA() throws IOException, IndexOutOfBoundsException {
        if (false) {
            throw new FileNotFoundException();
        } else {
            throw new IndexOutOfBoundsException();
        }
    }
}

```

このクラスをコンパイルするには、1行目に何を追加する必要がありますか？

- A. catch (FileNotFoundException | IndexOutOfBoundsException e){}
- B. catch (IOException e){}
- C. catch (FileNotFoundException e){} catch (IndexOutOfBoundsException e){}
- D. catch (IndexOutOfBoundsException e){} catch (FileNotFoundException e){}
- E. catch (FileNotFoundException | IOException e){}

Answer: B ([メッセージを残す](#))

最新問題: 74

与えられた：

```

package test;
import java.time.*;
public class Diary {
    private LocalDate now = LocalDate.now();
    public LocalDate getDate() {
        return now;
    }
}

```

ORACLE®
and

```

package test;
public class Tester {
    public static void main(String[] args) {
        Diary d = new Diary();
        System.out.println(d.getDate());
    }
}

```

正しい説明はどれですか？

- A. パッケージjava.timeのすべてのクラス。クラスDiaryにロードされます。
- B. クラステスターはjava.time.LocalDateをインポートする必要はありません。これは、パッケージテストのメンバーにすでに表示されているためです。
- C. テスターはコンパイルするためにjava.time.LocalDateをインポートする必要があります。

D. java.timeパッケージのLocalDateクラスのみがロードされます。

Answer: B ([メッセージを残す](#))

最新問題: 75

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D

C. オプションA

D. オプションB

Answer: D ([メッセージを残す](#))

最新問題: 76

与えられた:

```

public class Test {
    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
27 class GeneralException extends Exception {
28     public GeneralException(String s) { super(s); }
29 }
30
31 class SpecificException extends GeneralException {
32     public SpecificException(String s) { super(s); }
33 }

```

ORACLE®

有効な **1z1-819** 問題集は GoShiken.com が提供された合格しやすい 1z1-819 試験問題集！
 GoShiken.com が最新の **1z1-819** 試験問題集を提供しています。GoShiken.com 1z1-819 試験
 問題は最新で、解答が正確でございます。最新の GoShiken.com 1z1-819 問題集をゲットする
 人はこちら: <https://www.goshiken.com/Oracle/1z1-819-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: 77

dsがデータソースであり、EMPテーブルが適切に定義されていると仮定します。

```
try (Connection conn = ds.getConnection();
    PreparedStatement ps = conn.prepareStatement("INSERT INTO EMP VALUES(?, ?, ?)")) {
    ps.setObject(1, 101, JDBCType.INTEGER);
    ps.setObject(2, "SMITH", JDBCType.VARCHAR);
    ps.setObject(3, "HR", JDBCType.VARCHAR);
    ps.executeUpdate();
    ps.setInt(1, 102);
    ps.setString(2, "JONES");
    ps.executeUpdate();
}
```

このコードフラグメントの実行は何をしますか？

- A. 2つの行 (101, 'SMITH', 'HR') と (102, 'JONES', 'HR') を挿入します
- B. 2つの行 (101, 'SMITH', 'HR') と (102, 'JONES', NULL) を挿入します
- C. SQLException をスローします
- D. 1行を挿入します (101, 'SMITH', 'HR')

Answer: D ([メッセージを残す](#))

最新問題: 78

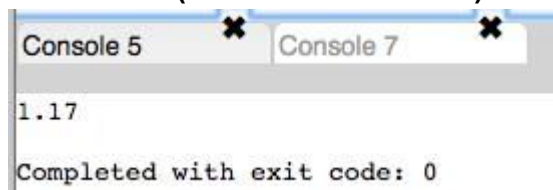
与えられた :

```
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 console window with two tabs: 'Console 5' and 'Console 7'. The output in the console is '1.17' followed by 'Completed with exit code: 0'.

最新問題: 79

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. }

```

ORACLE®

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: D ([メッセージを残す](#))



最新問題: 80

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

Answer: ([解答を表示する](#))

最新問題: 81

与えられた：

```

public class Test {
    private int sum;
    public int compute() {
        int x = 0;
        while(x < 3) {
            sum += x++;
        }
        return sum;
    }
    public static void main(String[] args) {
        Test t = new Test();
        int sum = t.compute();
        sum = t.compute();
        t.compute();
        System.out.println(sum);
    }
}

```

結果はどうなりますか？

- A. 9
- B. 実行時に例外がスローされます。
- C. 3
- D. 6

Answer: D ([メッセージを残す](#))



最新問題: 82

与えられた：

```

List<String> list = ... ;
list.forEach( x -> { System.out.println(x); } );

```

xのタイプは何ですか？

- A. char
- B. リスト<文字列>

C. リスト<キャラクター>

D. 文字列

Answer: [\(解答を表示する\)](#)

最新問題: 83

与えられた :

```
public class Foo {  
    public static void main(String... args) {  
        for (var x : args) {  
            System.out.println(x);  
        }  
    }  
}
```

ローカル変数xのタイプは何ですか？

A. char

B. キャラクター

C. 文字列[]

D. 文字列

Answer: D ([メッセージを残す](#))

最新問題: 84

組織は、クラウドの顧客がmlib.jarを利用できるようにします。mlib.jarのコードクリーンアッププロジェクトに取り組んでいる間、顧客はこのメソッドを確認します。

```
public void enableService(String hostName, String portNumber) throws IOException {  
    this.transportSocket = new Socket(hostName, portNumber);  
}
```

顧客がアクセスできる方法の要件を満たすために、この方法にどのようなセキュリティ対策を追加する必要がありますか？

A. enableServiceをプライベートにします。

B. 新しいSocketへの呼び出しをAccessController.doPrivilegedブロックで囲みます。

C. ソケットを開く前にhostNameパラメーターとportNumberパラメーターを検証するメソッドを作成します。

Insert this code before the call to new Socket:

```
hostName = new String(hostName);
```

D.

```
portNumber = new String(portNumber);
```

Answer: B ([メッセージを残す](#))

最新問題: 85

与えられた :

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

and

```
public class Tester {  
    public static void main(String[] args) {  
        Person p = null;  
        checkPerson(p);  
        System.out.println(p);  
        p = new Person("Mary");  
        checkPerson(p);  
        System.out.println(p);  
    }  
    public static Person checkPerson(Person p) {  
        if (p == null) {  
            p = new Person("Joe");  
        }else{  
            p = null;  
        }  
        return p;  
    }  
}
```

結果はどうなりますか？

A. ジョー

ヌル

B. null

メアリー

C. ジョー

結婚する

D. null

ヌル

Answer: A ([メッセージを残す](#))

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



最新問題: 86

コードフラグメントが与えられた場合：

```
public class Main {  
    private int count = 0; // line 1  
    public static void main(String[] args) { // line 2  
        Main test = new Main();  
        ExecutorService service = Executors.newFixedThreadPool(10);  
        for (int i = 0; i < 10; i++) {  
            service.submit(() -> {  
                for (int j = 0; j < 10000; j++) {  
                    test.count++; // line 3  
                }  
            });  
        }  
        service.shutdown();  
    }  
}
```

countvariableスレッドを安全にする必要があります。

要件を満たす2つの変更はどれですか？ 2つ選択してください。）

A. 2行目をpublic static synchronized void main \$String [] args){に置き換えます

B. 1行目をprivate volatile int count = 0に置き換えます。

C. 3行目を次のように置き換えます

```
同期テスト){  
test.count ++;  
}
```

D. 1行目をprivate AtomicInteger count = new AtomicInteger (0);に置き換え、3行目を
test.count.incrementAndGet ()に置き換えます。

E. 3行目を次のように置き換えます

```
同期 test.count){  
test.count ++;  
}
```

Answer: (解答を表示する)

説明/参照 <https://stackoverflow.com/questions/15852123/java-thread-safe-counter/15852164>

最新問題: 87

モジュラーJDKについて正しい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. JDKがモジュール化されているため、APIはより積極的に非推奨になっています。
- B. モジュール式JDKで実行するには、アプリケーションをモジュールとして構造化する必要があります。
- C. コマンドラインからモジュールのエクスポートを構成することは可能ですが、望ましくありません。
- D. Java SEプラットフォームの基本的なAPIは、java.baseモジュールにあります。

Answer: C,D ([メッセージを残す](#))

最新問題: 88

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

Answer: ([解答を表示する](#)**)**

最新問題: 89

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: ([解答を表示する](#)**)**

```

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: 90

Javaモジュールについて正しい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. パッケージが名前付きモジュールと名前なしモジュールの両方で定義されている場合、名前なしモジュール内のパッケージは無視されます。
- B. 任意の名前付きモジュールは、自動モジュール内のすべてのクラスに直接アクセスできます。
- C. -classpathからロードされたモジュラーjarは自動モジュールです。
- D. -module-pathからロードされたモジュラーjarは自動モジュールです。
- E. -classpathで見つかったクラスは、名前のないモジュールの一部です。

Answer: D,E ([メッセージを残す](#))

最新問題: 91

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. キー :0値 :0マップ :0
- B. キー :0値 :0マップ :
- C. コンパイルは失敗します。
- D. キー :4値 :4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 1z1-819 問題集は GoShiken.com が提供された合格しやすい 1z1-819 試験問題集！
 GoShiken.com が最新の 1z1-819 試験問題集を提供しています。GoShiken.com 1z1-819 試験問題は最新で、解答が正確でございます。最新の GoShiken.com 1z1-819 問題集をゲットする人はこちら: <https://www.goshiken.com/Oracle/1z1-819-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 :

```

// line 1
List<String> fruits = new ArrayList<>(List.of("apple", "orange", "banana"));
fruits.replaceAll(function);

```

1行目のどのステートメントがこのコードフラグメントのコンパイルを可能にしますか？

- A. 関数function = String :: toUpperCase;

B. UnaryOperator関数= s-> s.toUpperCase ()

C. UnaryOperator <String> function = String :: toUpperCase;

D. Function <String> function = m-> m.toUpperCase ()

Answer: (解答を表示する)

```
1
2 import java.io.*;
3 import java.util.*;
4 import java.util.stream.Stream;
5 import java.util.function.Function;
6 import java.util.function.UnaryOperator;
7
8 class Hello {
9     public static void main(String[] args) {
10
11         UnaryOperator<String> function = String::toUpperCase;
12         List<String> fruits = new ArrayList<>(List.of("apple", "orange", "banana"));
13         fruits.replaceAll(function);
14
15     }
16 }
17
```



最新問題: 93

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。)

```
1 @FunctionalInterface
2 interface MyRunnable {
3     public void run();
4 }
5
6 @FunctionalInterface
7 interface MyRunnable {
8     public void run();
9     public void call();
10 }
11
12 interface MyRunnable {
13     public default void run() {}
14     public void run(String s);
15 }
16
17 @FunctionalInterface
18 interface MyRunnable {
19 }
20
21 interface MyRunnable {
22     @FunctionalInterface
23     public void run();
24 }
```



A. オプションA

B. オプションB

C. オプションE

D. オプションD

E. オプションC

Answer: C,E ([メッセージを残す](#))

最新問題: 94

与えられた :

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

結果はどうなりますか？

A. それは1ですか

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

C. それは0ですか

D. これは2です

E. これは3です

Answer: D ([メッセージを残す](#))



```
Console 1 | Console 2  
this is it 2  
Completed with exit code: 0
```

最新問題: 95

与えられた :

```
public class SerializedMessage implements Serializable {  
    String message;  
    LocalDateTime createdAt;  
    transient LocalDateTime updatedAt;  
    SerializedMessage(String message) {  
        this.message = message;  
        this.createdAt = LocalDateTime.now();  
    }  
    private void readObject (ObjectInputStream in) {  
        try {  
            in.defaultReadObject();  
            this.updatedAt = 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シリアル化された後

- 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逆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る前

Answer: D ([メッセージを残す](#))

最新問題: 96

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

Answer: A ([メッセージを残す](#))

最新問題: 97

どのコードが正しいですか？

- A. 実行可能r = {System.out.println ("Message")};
- B. 実行可能r ="メッセージ">System.out.println ()
- C. 実行可能r = (→ System.out :: print;
- D. 実行可能r => System.out.println ("Message");
- E. 実行可能r = ()> {System.out.println ("Message");};

Answer: ([解答を表示する](#)**)**

最新問題: 98

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 ("B"), "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 ("C"), "Q");

Answer: ([解答を表示する](#))

最新問題: 99

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. 2行目のエラーが原因でコンパイルが失敗します。
- B. 7行目のエラーが原因でコンパイルが失敗します。
- C. 6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. コンパイルは成功します。
- F. 9行目のエラーが原因でコンパイルが失敗します。
- G. 10行目のエラーが原因でコンパイルが失敗します。

Answer: C ([メッセージを残す](#))

最新問題: 100

高性能に寄与するJavaの側面を説明しているのはどれですか？

- A. Javaは、頻繁に実行されるコードを監視および最適化します。
- B. Javaはガベージコレクションを優先します。
- C. Javaには、パイプラインバースト実行を有効にするために使用できる組み込み関数のライブラリがあります。
- D. Javaはコード実行を自動的に並列化します。

Answer: A ([メッセージを残す](#))

最新問題: 101

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 }
```

ORACLE®

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: 102

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)

Console 1
14
Completed with exit code: 0

ORACLE

最新問題: 103

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. }
```

1. ORACLE®
2. }

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: ([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
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: 104

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

Answer: C,D ([メッセージを残す](#))

最新問題: 105

このmodule-info.javaファイルを持つメインモジュールを持つアプリケーションがあるとします。

```

module main {
    exports country;
    uses country.CountryDetails;
}

```

どちらが正しいですか？ 2つ選択してください。）

- A. country.CountryDetailsの実装を提供するモジュールにはrequiresmainが必要です。module-info.javaファイルのディレクティブ。
- B. エラーなしで実行するには、アプリケーションは、country.CountryDetailsの実装を提供するモジュールパスに少なくとも1つのモジュールを持っている必要があります。
- C. エラーなしでコンパイルするには、アプリケーションは、country.CountryDetailsの実装を提供するモジュールソースパスに少なくとも1つのモジュールを持っている必要があります。
- D. country.countryDetailsの実装をメインモジュールに追加できます。
- E. country.CountryDetailsの実装を提供するモジュールは、メインモジュールを再コンパイルせずにコンパイルおよび追加できます。

Answer: ([解答を表示する](#)**)**

最新問題: 106

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](#))



有効な **1z1-819** 問題集は GoShiken.com が提供された合格しやすい 1z1-819 試験問題集！
GoShiken.com が最新の **1z1-819** 試験問題集を提供しています。GoShiken.com 1z1-819 試験問題は最新で、解答が正確でございます。最新の GoShiken.com 1z1-819 問題集をゲットする人はこちら: <https://www.goshiken.com/Oracle/1z1-819-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

与えられた :

```

1. {
2.   Iterator iter = List.of(1,2,3).iterator();
3.   while (iter.hasNext()) {
4.     foo(iter.next());
5.   }
6.   Iterator iter2 = List.of(1,2,3).iterator();
7.   while (iter.hasNext()) {
8.     bar(iter2.next());
9.   }
10. }
11. for (Iterator iter = List.of(1,2,3).iterator(); iter.hasNext(); ) {
12.   foo(iter.next());
13. }
14. for (Iterator iter2 = List.of(1,2,3).iterator(); iter.hasNext(); ) {
15.   bar(iter2.next());
16. }

```

コンパイル時エラーが発生するループはどれですか？

- A. ループ開始行14
- B. ループ開始行3
- C. ループ開始行7
- D. ループ開始行11

Answer: A ([メッセージを残す](#))

最新問題: 108

注釈タイプのどの宣言が有効ですか？

@interface作成者{

A. 文字列name (デフォルトはnullです。

文字列date (

})

@interface作成者{

B. 文字列名 (

文字列の日付;

})

@interface作成者{

C. 文字列名 (デフォルト "";

文字列date (

})

@interface作成者{

D. 文字列名 (

文字列の日付のデフォルト "";

})

E. 文字列名 (デフォルト "";

文字列date (

})

@interface Author extends Serializable {

Answer: E ([メッセージを残す](#))

最新問題: 109

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");
        }
    }
}
```

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](#))



Unkown

Completed with exit code: 0

最新問題: 110

Javaを動的にするものは何ですか？

- A. Javaコンパイラは、特定のターゲットプラットフォームで実行するようにクラスを前処理します。
- B. ランタイムは、機械語ソースと、さまざまな言語コンパイラからの実行可能ファイルを処理できます。
- C. 実行時に、クラスは必要に応じてロードされ、新しいコードモジュールはオンデマンドでロードできます。

D. Javaコンパイラは、リフレクションを使用して、クラスメソッドがターゲットプラットフォームのリソースでサポートされているかどうかをテストします。

Answer: C ([メッセージを残す](#))

Valid 1z1-819 Dumps shared by GoShiken.com for Helping Passing 1z1-819 Exam!

GoShiken.com now offer the **newest 1z1-819 exam dumps**, the GoShiken.com 1z1-819 exam **questions have been updated** and **answers have been corrected** get the **newest**

GoShiken.com 1z1-819 dumps with Test Engine here:

<https://www.goshiken.com/Oracle/1z1-819-mondaishu.html> (297 Q&As Dumps, **30%OFF**

Special Discount: **Freepdfdumps**)