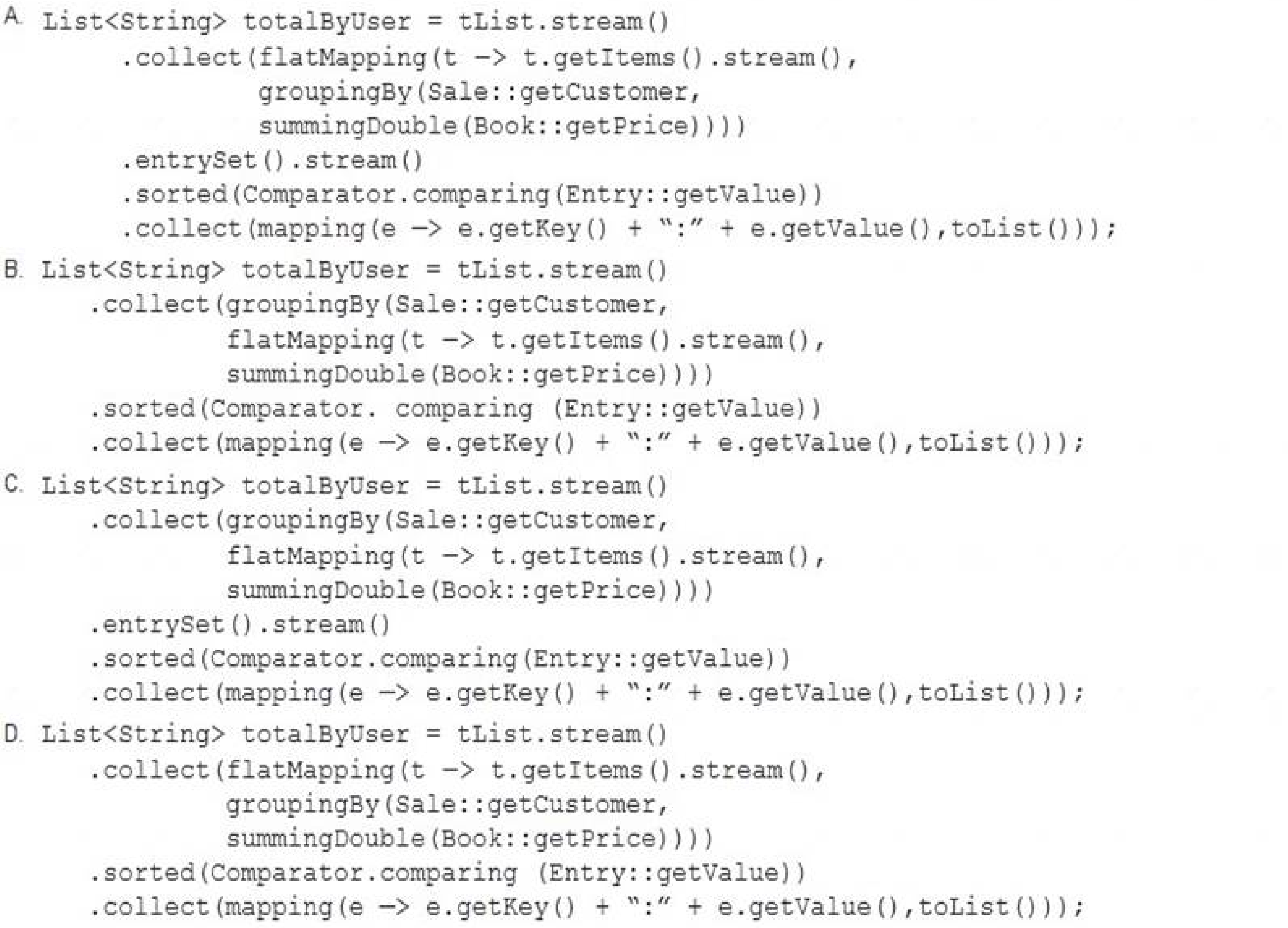
# Oracle

## Exam Questions 1Z0-819

Java SE 11 Developer

**NEW QUESTION 1**
A bookstore's sales are represented by a list of Sale objects populated with the name of the customer and the books they purchased.
public class Sale { private String customer;
private List<Book> items;
// constructor, setters and getters not shown
}
public class Book { private String name; private double price;
// constructor, setters and getters not shown
}
Given a list of Sale objects, tList, which code fragment creates a list of total sales for each customer in ascending order?



```
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. Option A
B. Option B
C. Option C
D. Option D

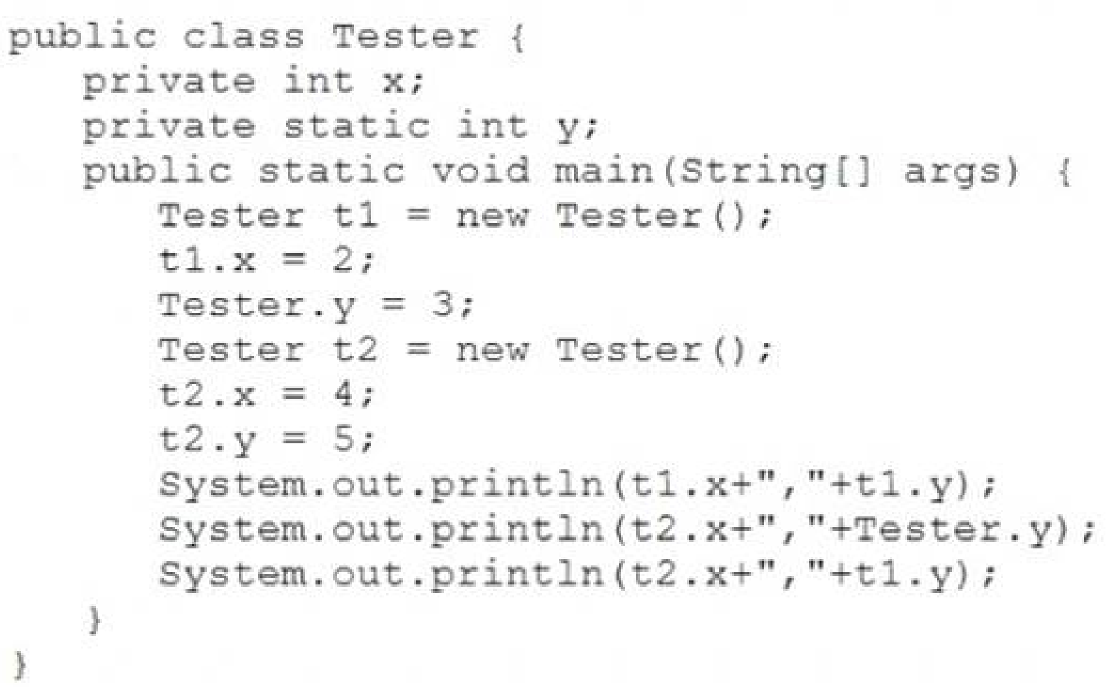
**Answer:** C

**NEW QUESTION 2**
Which two commands are used to identify class and module dependencies? (Choose two.)

A. jmod describe
B. java Hello.java
C. jdeps --list-deps
D. jar --show-module-resolution
E. java --show-module-resolution

**Answer:** CE

**NEW QUESTION 3**
Given:



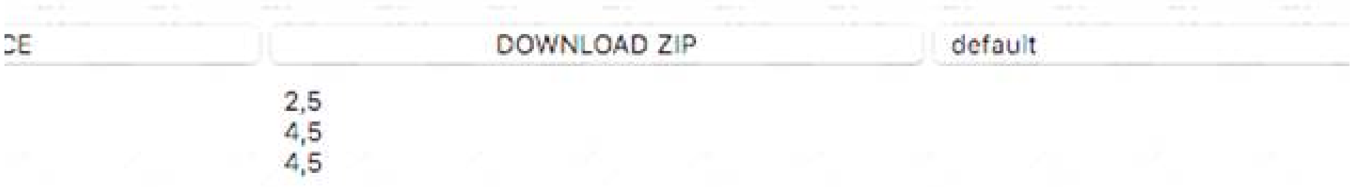
```
public class Tester {
    private int x;
    private static int y;
    public static void main(String[] args) {
        Tester t1 = new Tester();
        t1.x = 2;
        Tester.y = 3;
        Tester t2 = new Tester();
        t2.x = 4;
        t2.y = 5;
        System.out.println(t1.x+","+t1.y);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

What is the result?

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

**Answer:** C

**Explanation:**



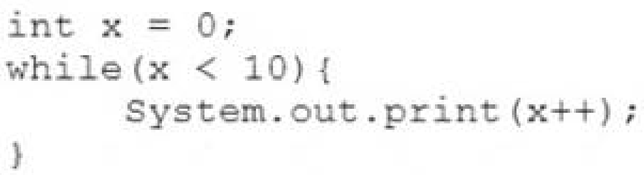
```
2,5
4,5
4,5
```

**NEW QUESTION 4**
Which two statements are true about the modular JDK? (Choose two.)

A. The foundational APIs of the Java SE Platform are found in the java.base module.
B. An application must be structured as modules in order to run on the modular JDK.
C. It is possible but undesirable to configure modules’ exports from the command line.
D. APIs are deprecated more aggressively because the JDK has been modularized.

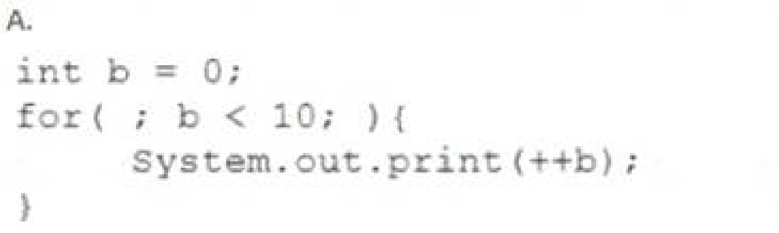
**Answer:** AC

**NEW QUESTION 5**
Given the code fragment:



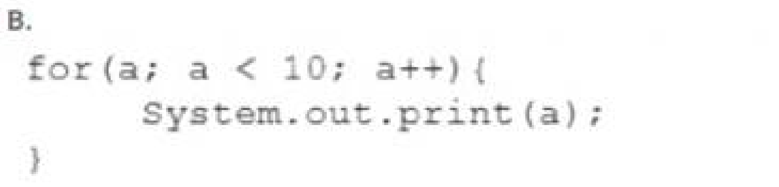
```
int x = 0;
while(x < 10){
      System.out.print(x++);
}
```

Which “for” loop produces the same output?



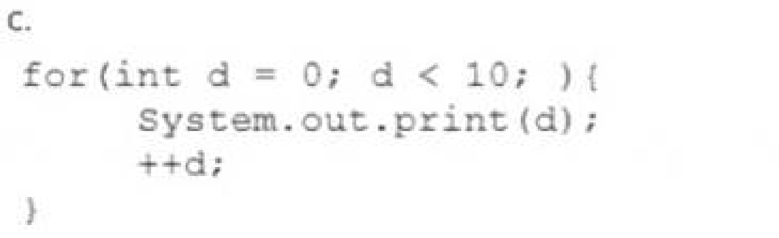
A.

```
int b = 0;
for( ; b < 10; ){
      System.out.print(++b);
}
```



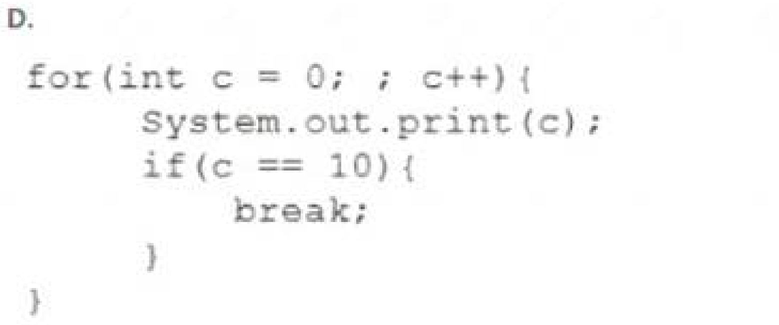
B.

```
for(a; a < 10; a++){
      System.out.print(a);
}
```



C.

```
for(int d = 0; d < 10; ){
      System.out.print(d);
      ++d;
}
```



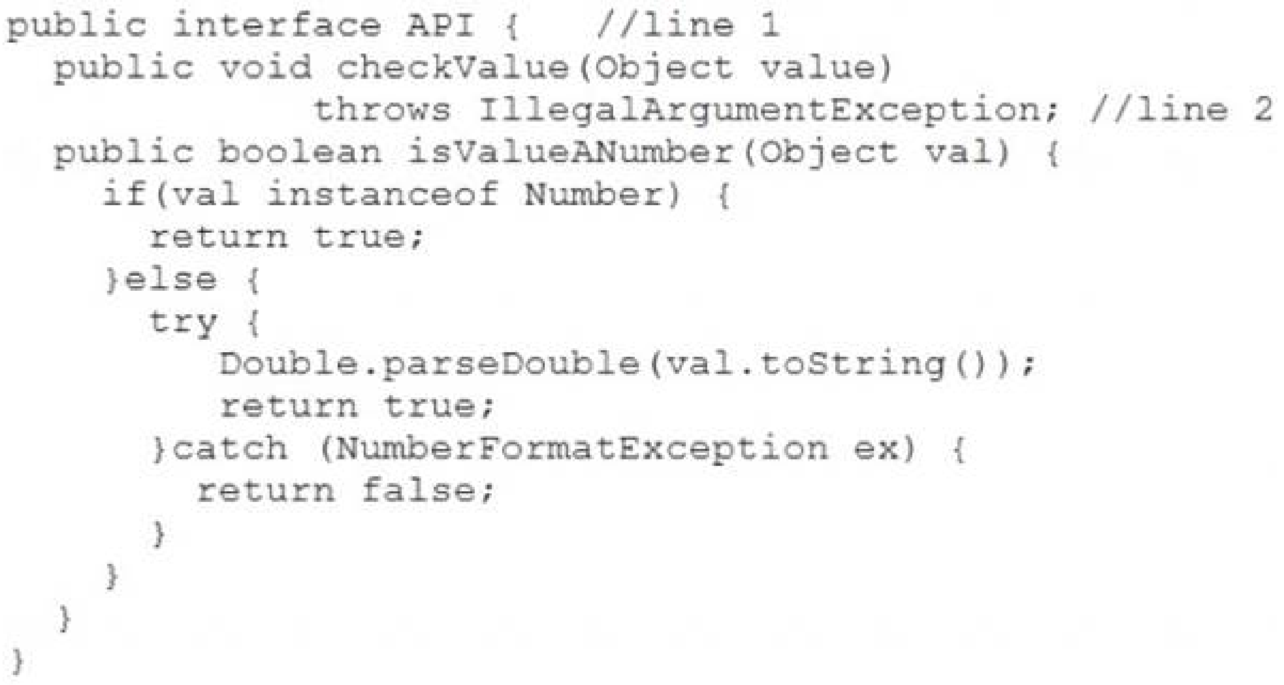
D.

```
for(int c = 0; ; c++){
      System.out.print(c);
      if(c == 10){
          break;
      }
}
```

A. Option A
B. Option B
C. Option C
D. Option D

**Answer:** C

**NEW QUESTION 6**
Given:



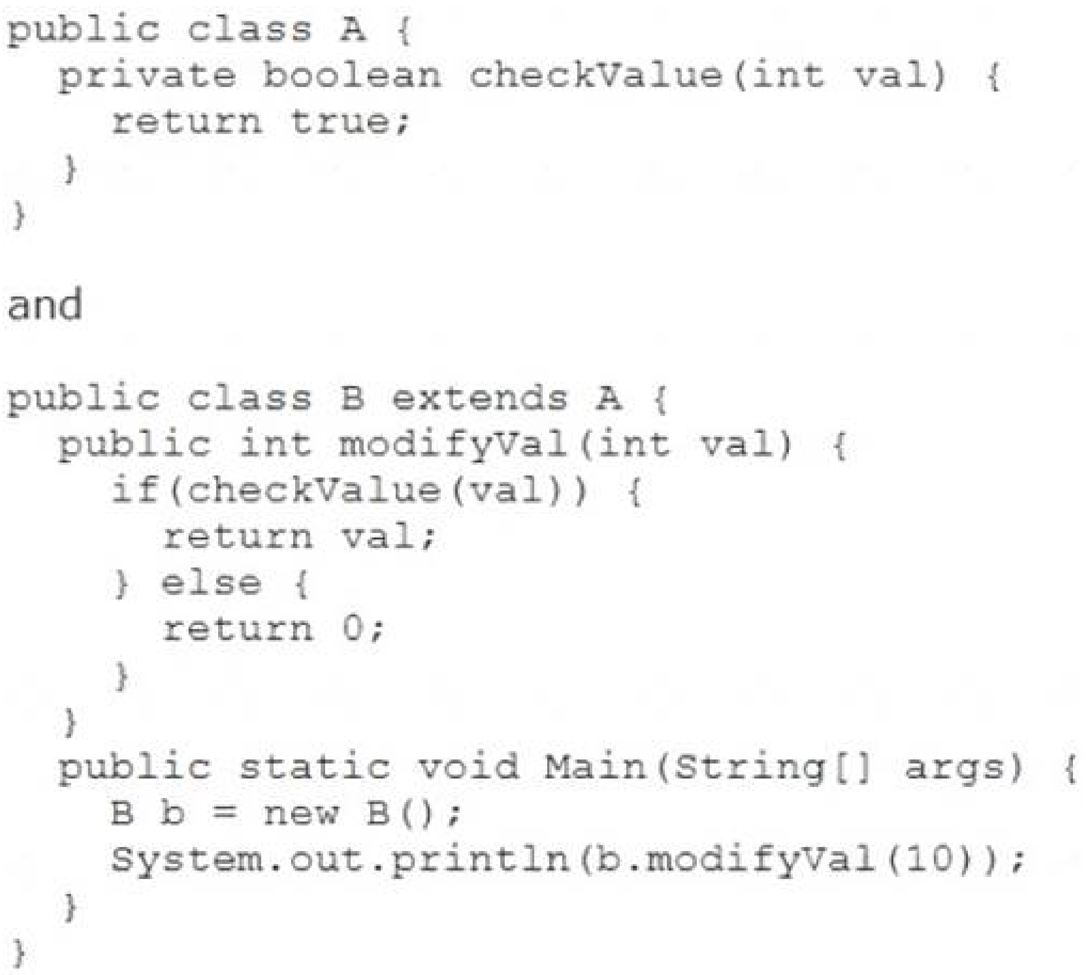
```
public interface API {   //line 1
  public void checkValue(Object value)
              throws IllegalArgumentException; //line 2
  public boolean isValueANumber(Object val) {
    if(val instanceof Number) {
      return true;
    }else {
      try {
         Double.parseDouble(val.toString());
 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

Which two changes need to be made to make this class compile? (Choose two.)

A. Change Line 1 to an abstract class:public abstract class API {
B. Change Line 2 access modifier to protected:protected void checkValue(Object value)throws IllegalArgumentException;
C. Change Line 1 to a class:public class API {
D. Change Line 1 to extend java.lang.AutoCloseable:public interface API extends AutoCloseable {
E. Change Line 2 to an abstract method:public abstract void checkValue(Object value)throws IllegalArgumentException;

**Answer:** CE

**NEW QUESTION 7**
Given:



```
public class A {
  private boolean checkValue(int val) {
    return true;
  }
}
```

and

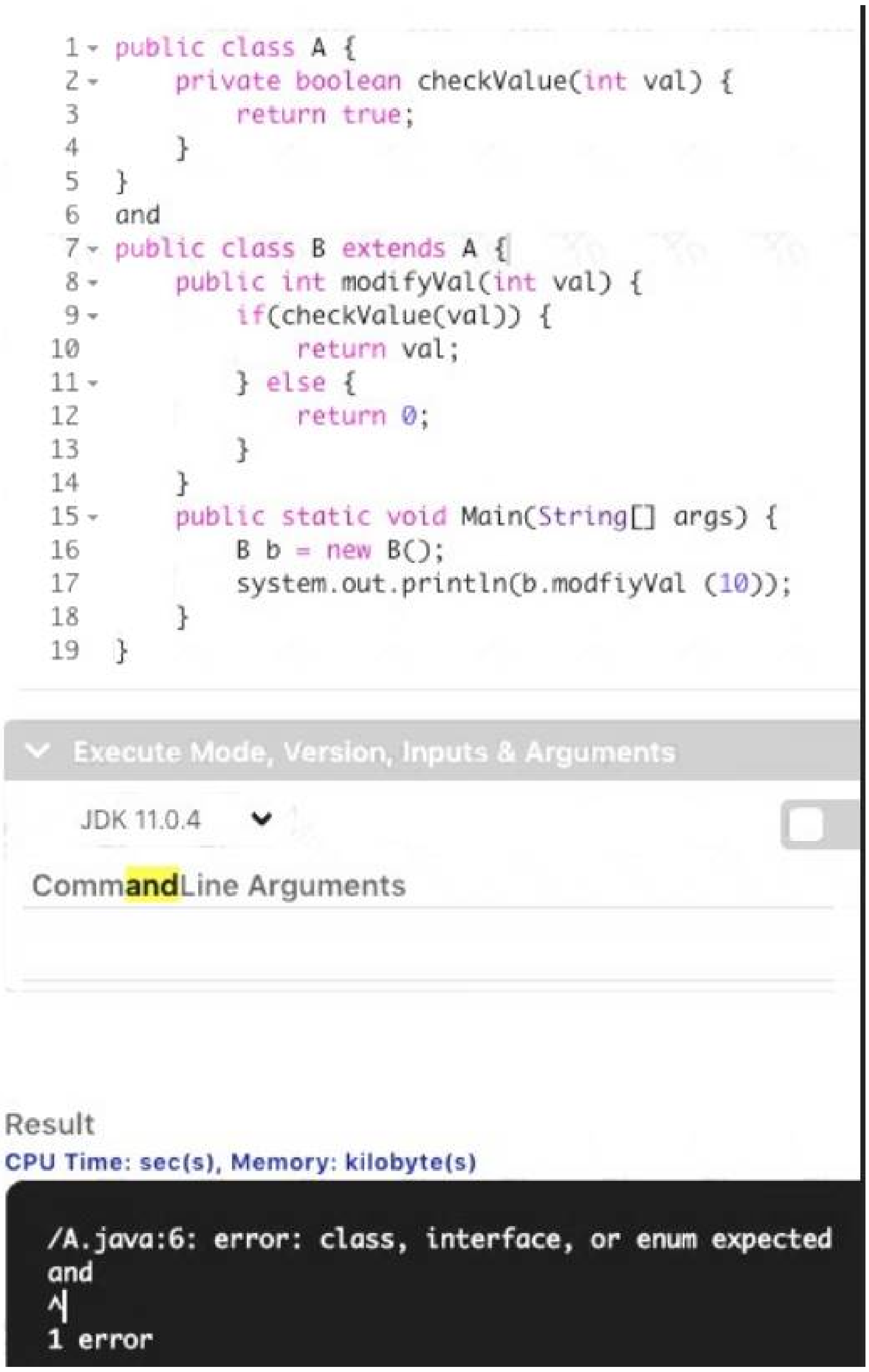
```
public class B extends A {
  public int modifyVal(int val) {
    if(checkValue(val)) {
      return val;
    } else {
      return 0;
    }
  }
  public static void Main(String[] args) {
    B b = new B();
    System.out.println(b.modifyVal(10));
  }
}
```

What is the result?

A. nothing
B. It fails to compile.
C. A java.lang.IllegalArgumentException is thrown.
D. 10

**Answer:** B

**Explanation:**



```
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 }
```

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
```

**NEW QUESTION 8**

Given the code fragment:

Path currentFile = Paths.get("/scratch/exam/temp.txt"); Path outputFile = Paths get("/scratch/exam/new.txt"); Path directory = Paths.get("/scratch/");

Files.copy(currentFile, outputFile); Files.copy(outputFile, directory);

Files.delete (outputFile);

The /scratch/exam/temp.txt file exists. The /scratch/exam/new.txt and /scratch/new.txt files do not exist. What is the result?

A. /scratch/exam/new.txt and /scratch/new.txt are deleted.

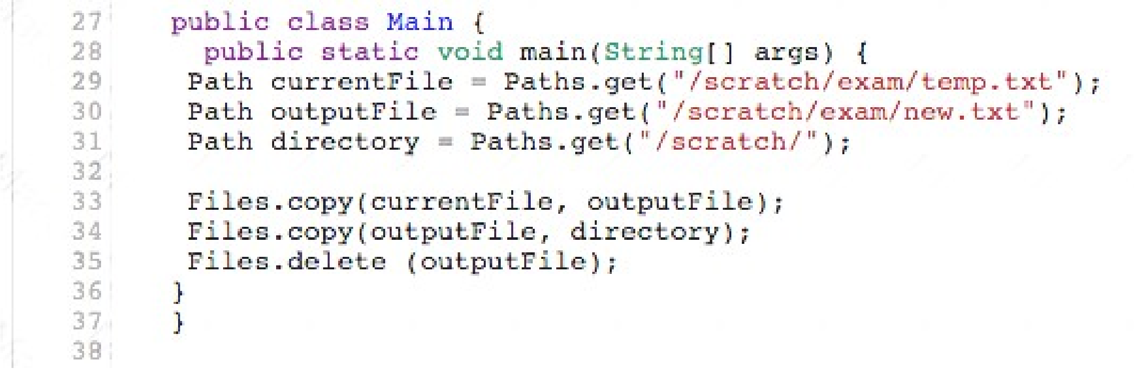
B. The program throws a FileaAlreadyExistsException.

C. The program throws a NoSuchFileException.

D. A copy of /scratch/exam/new.txt exists in the /scratch directory and /scratch/exam/new.txt is deleted.

**Answer:** C

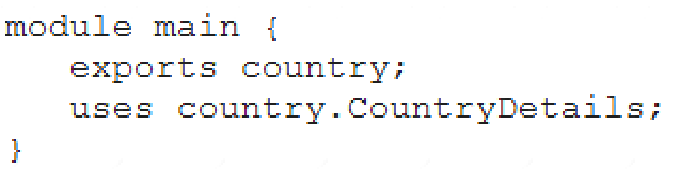
**Explanation:**



```
public class Main {
  public static void main(String[] args) {
 Path currentFile = Paths.get("/scratch/exam/temp.txt");
 Path outputFile = Paths.get("/scratch/exam/new.txt");
 Path directory = Paths.get("/scratch/");

 Files.copy(currentFile, outputFile);
 Files.copy(outputFile, directory);
 Files.delete (outputFile);
}
}
```

**NEW QUESTION 9**

Given an application with a main module that has this module-info.java file:



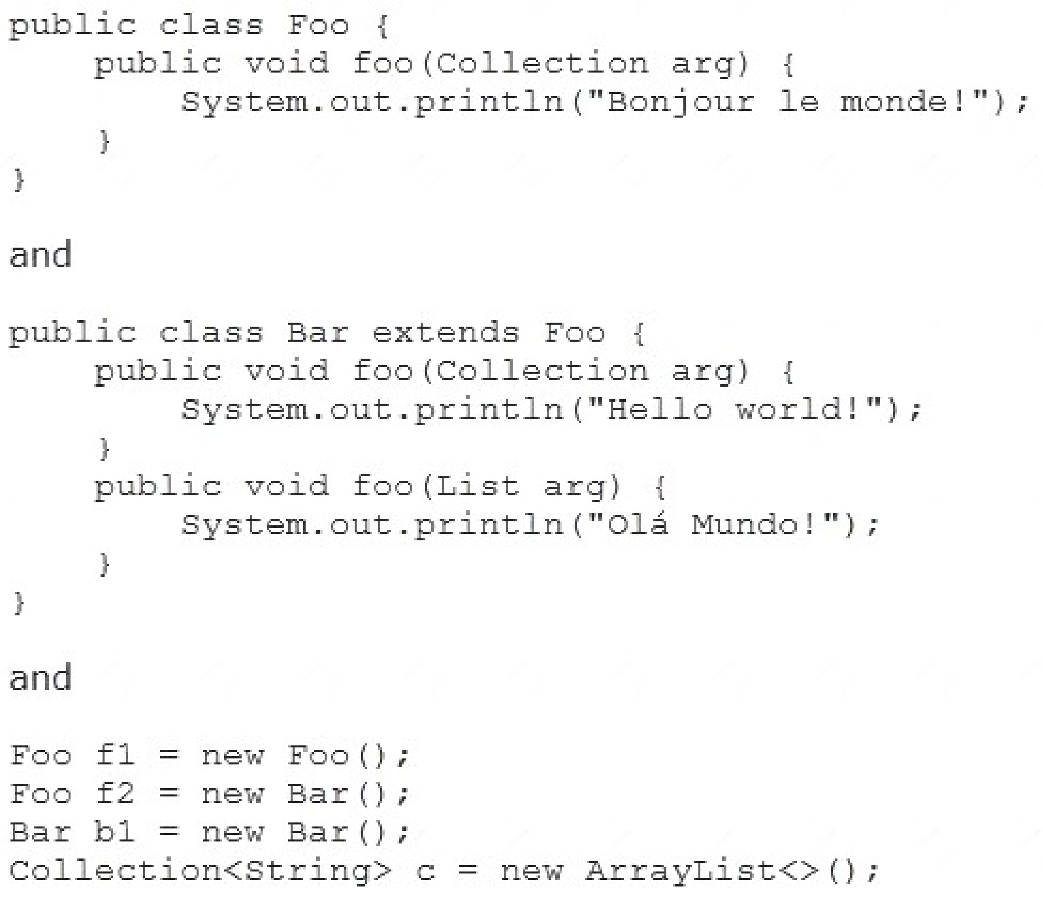
```
module main {
   exports country;
   uses country.CountryDetails;
}
```

Which two are true? (Choose two.)

A. A module providing an implementation of country.CountryDetails can be compiled and added without recompiling the main module.
B. A module providing an implementation of country.CountryDetails must have a requires main; directive in its module-info.java file.
C. An implementation of country.countryDetails can be added to the main module.
D. To compile without an error, the application must have at least one module in the module source path that provides an implementation of country.CountryDetails.
E. To run without an error, the application must have at least one module in the module path that provides an implementation of country.CountryDetails.

**Answer:** BD

**NEW QUESTION 10**
Given:



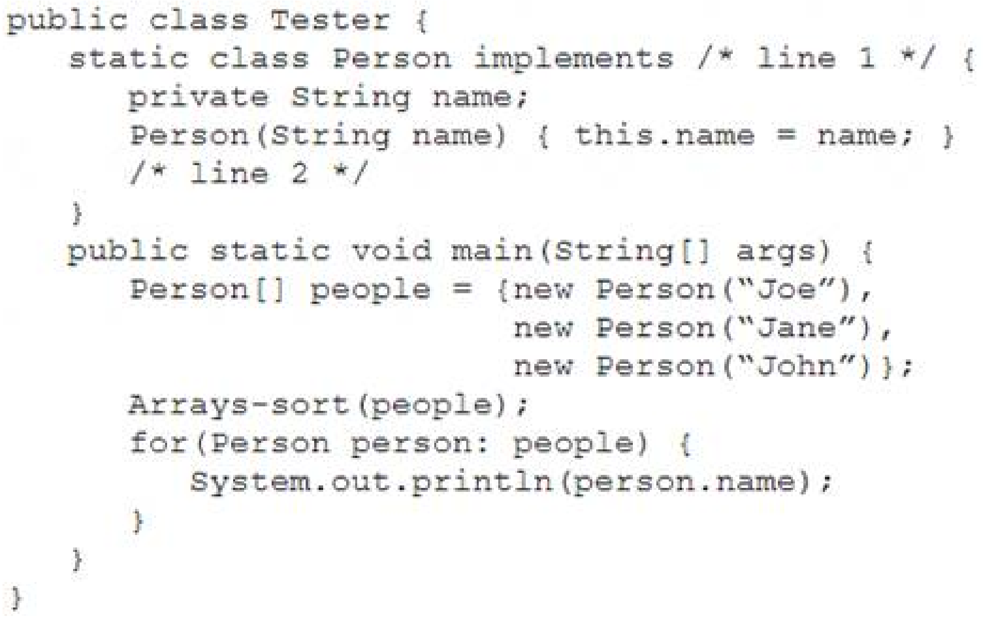
```
public class Foo {
    public void foo(Collection arg) {
        System.out.println("Bonjour le monde!");
    }
}

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<>();
```

Which three are true? (Choose three.)

A. b1.foo(c) prints Bonjour le monde!
B. f1.foo(c) prints Hello world!
C. f1.foo(c) prints Olá Mundo!
D. b1.foo(c) prints Hello world!
E. f2.foo(c) prints Olá Mundo!
F. b1.foo(c) prints Olá Mundo!
G. f2.foo(c) prints Bonjour le monde!
H. f2.foo(c) prints Hello world!
I. f1.foo(c) prints Bonjour le monde!

**Answer:** BFG

**NEW QUESTION 10**
Given:



```
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);
      }
   }
}
```

You want the code to produce this output:
John
Joe Jane

Which code fragment should be inserted on line 1 and line 2 to produce the output?

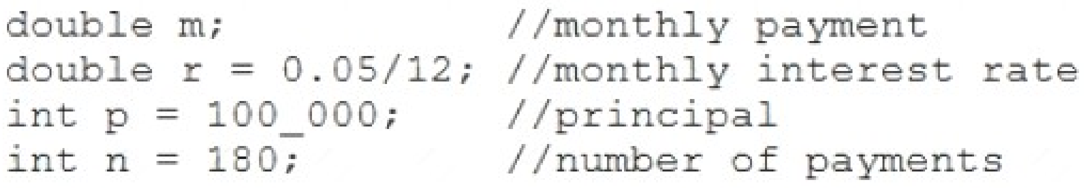
A. Insert Comparator<Person> on line 1.Insertpublic int compare(Person p1, Person p2) { return p1.name.compare(p2.name);}on line 2.
B. Insert Comparator<Person> on line 1.Insertpublic int compareTo(Person person) { return person.name.compareTo(this.name);}on line 2.
C. Insert Comparable<Person> on line 1.Insertpublic int compare(Person p1, Person p2) { return p1.name.compare(p2.name);}on line 2.
D. Insert Comparator<Person> on line 1.Insertpublic int compare(Person person) { return person.name.compare(this.name);}on line 2.

**Answer:** B

**NEW QUESTION 14**
Given the formula to calculate a monthly mortgage payment:

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

and these declarations:



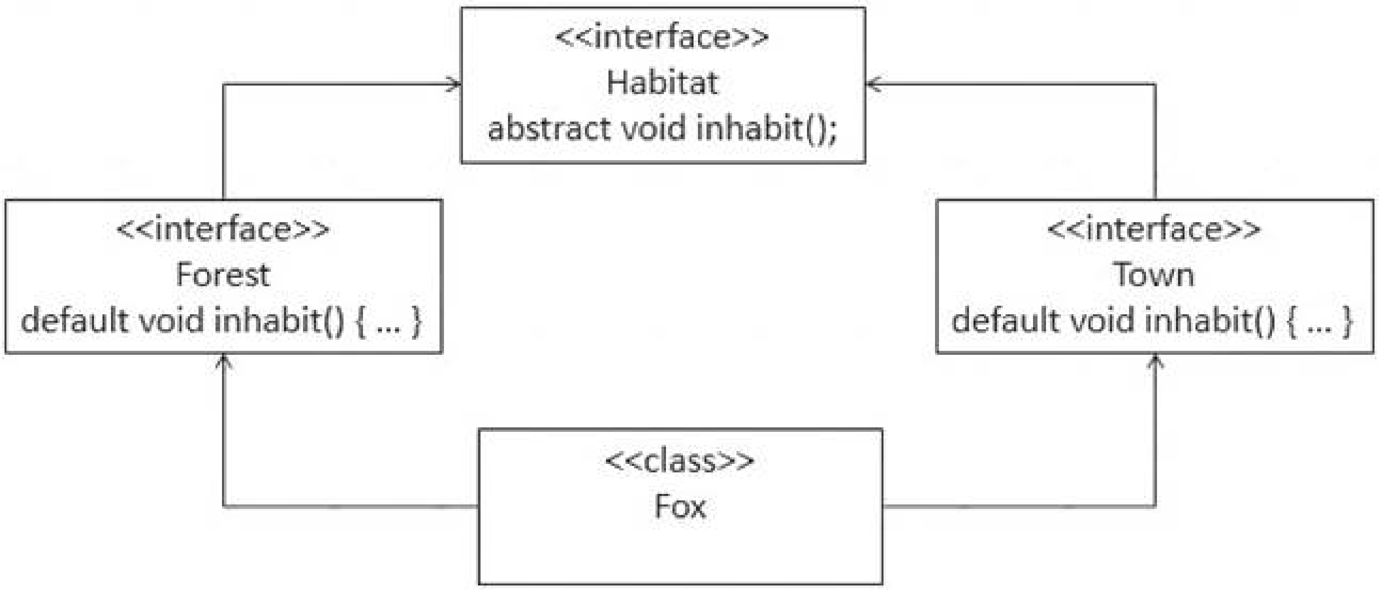
```
double m;             //monthly payment
double r = 0.05/12;   //monthly interest rate
int p = 100_000;      //principal
int n = 180;          //number of payments
```

How can you code the formula?

A. m = p * (r * 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 * r * 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:** A

**NEW QUESTION 19**
Given:

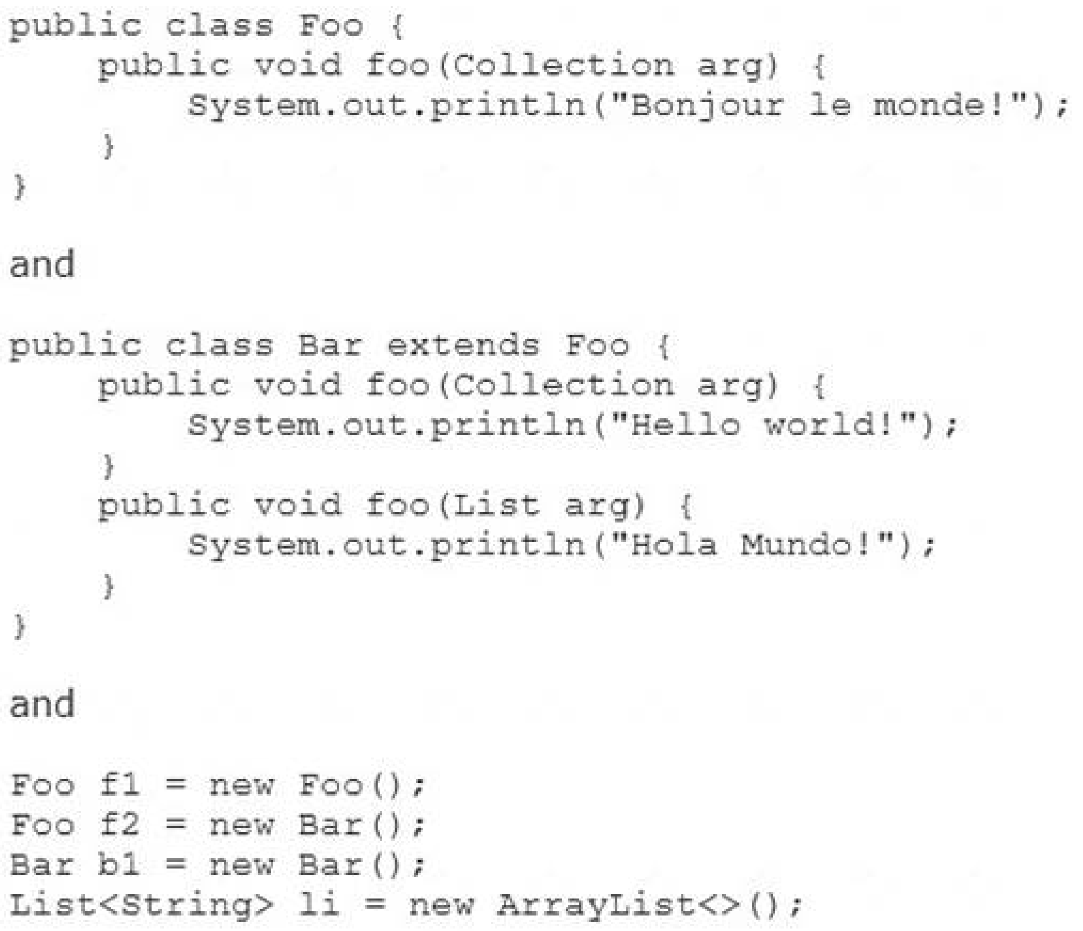


Which statement is true about the Fox class?

A. Fox class does not have to override inhabit method, so long as it does not try to call it.
B. Fox class does not have to override the inhabit method if Forest and Town provide compatible implementations.
C. Fox class must implement either Forest or Town interfaces, but not both.
D. The inhabit method implementation from the first interface that Fox implements will take precedence.
E. Fox class must provide implementation for the inhabit method.

**Answer:** B

**NEW QUESTION 21**
Given:



```
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<>();
```

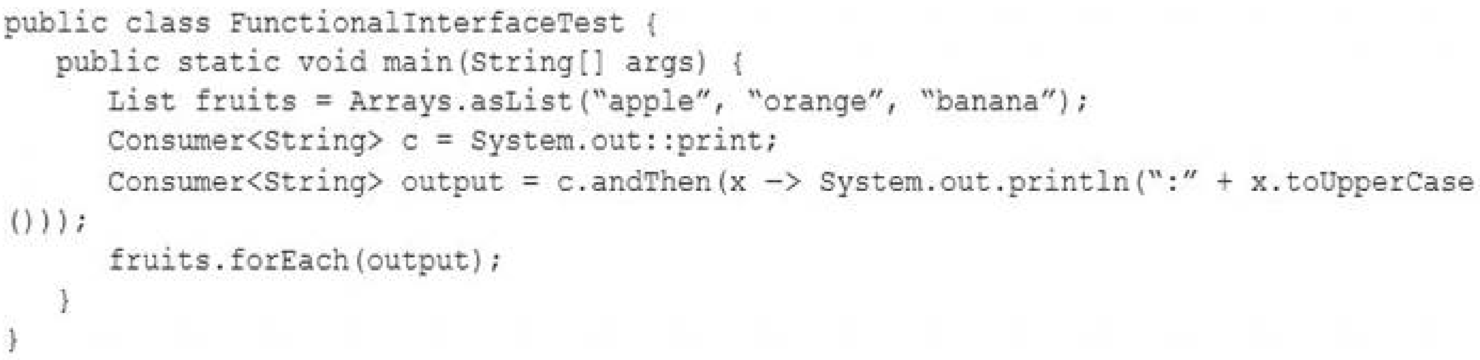
Which three are correct? (Choose three.)

A. b1.foo(li) prints Hello world!
B. f1.foo(li) prints Bonjour le monde!
C. f1.foo(li) prints Hello world!
D. f1.foo(li) prints Hola Mundo!
E. b1.foo(li) prints Bonjour le monde!
F. f2.foo(li) prints Hola Mundo!
G. f2.foo(li) prints Bonjour le monde!
H. b1.foo(li) prints Hola Mundo!
I. f2.foo(li) prints Hello world!

**Answer:** ABH

**NEW QUESTION 25**

Given:



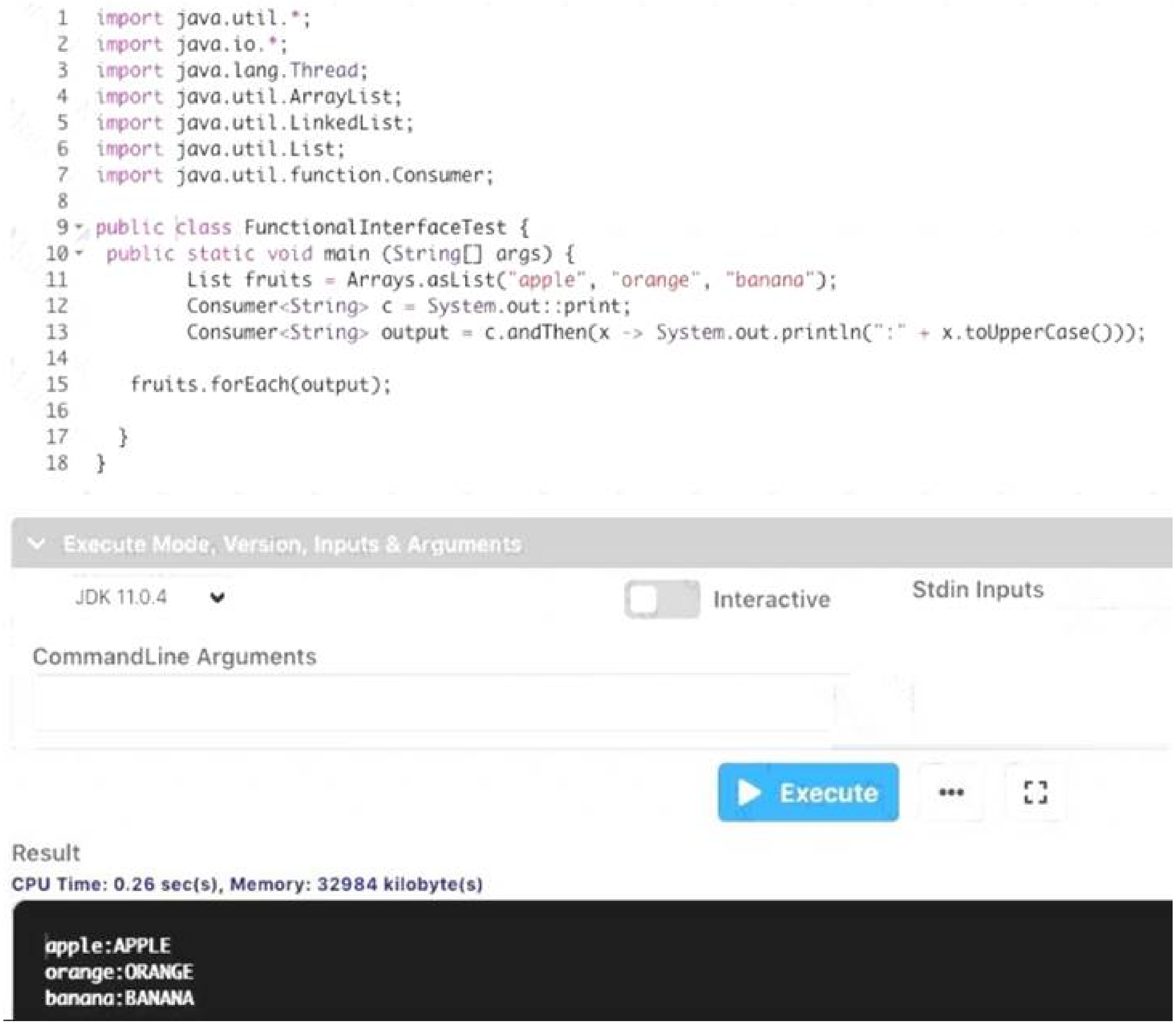
```
public class FunctionalInterfaceTest {
   public static void main(String[] args) {
      List fruits = Arrays.asList(“apple”, “orange”, “banana”);
      Consumer<String> c = System.out::print;
      Consumer<String> output = c.andThen(x -> System.out.println(“:” + x.toUpperCase
()));
      fruits.forEach(output);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

What is the output?

A. :APPLE:ORANGE:BANANAappleorangebanana
B. :APPLE:ORANGE:BANANA
C. APPLE:apple ORANGE:orange BANANA:banana
D. appleorangebanana:APPLE:ORANGE:BANANA
E. apple:APPLE orange:ORANGE banana:BANANA

**Answer:** E

**Explanation:**



```
import java.util.*;
import java.io.*;
import java.lang.Thread;
import java.util.ArrayList;
import java.util.LinkedList;
import java.util.List;
import java.util.function.Consumer;

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);

    }
 }
```

JDK 11.0.4

Result

CPU Time: 0.26 sec(s), Memory: 32984 kilobyte(s)

```
apple:APPLE
orange:ORANGE
banana:BANANA
```

**NEW QUESTION 27**

Given:

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

What code must you insert on Line 1 to enable the code to print Hello world?

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

**Answer:** B

**NEW QUESTION 32**

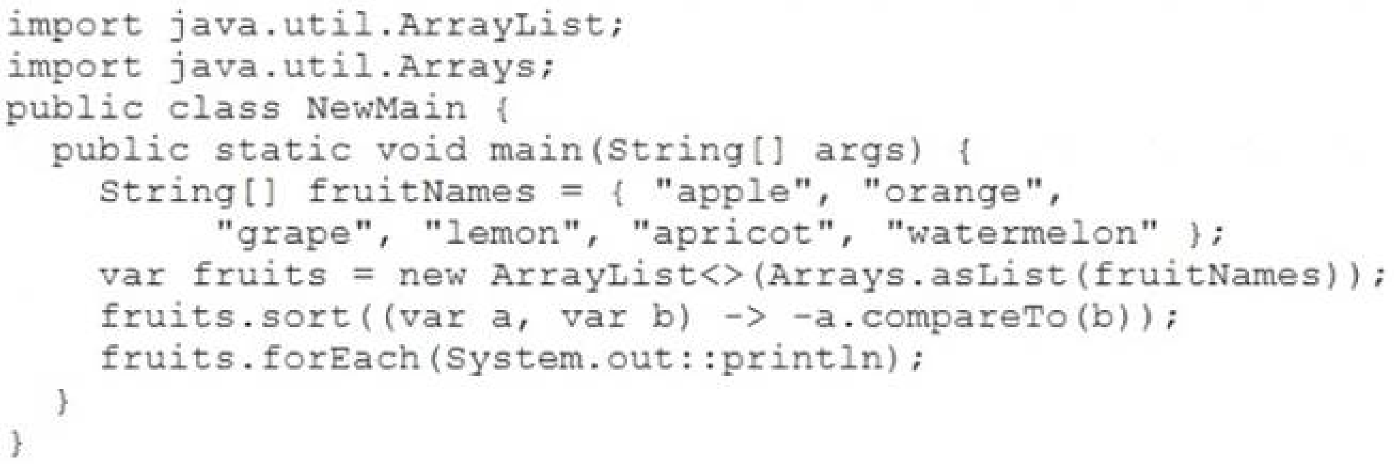
Which three guidelines are used to protect confidential information? (Choose three.)

A. Limit access to objects holding confidential information.
B. Clearly identify and label confidential information.
C. Manage confidential and other information uniformly.
D. Transparently handle information to improve diagnostics.
E. Treat user input as normal information.
F. Validate input before storing confidential information.
G. Encapsulate confidential information.

**Answer:** ADF

**NEW QUESTION 34**

Given:



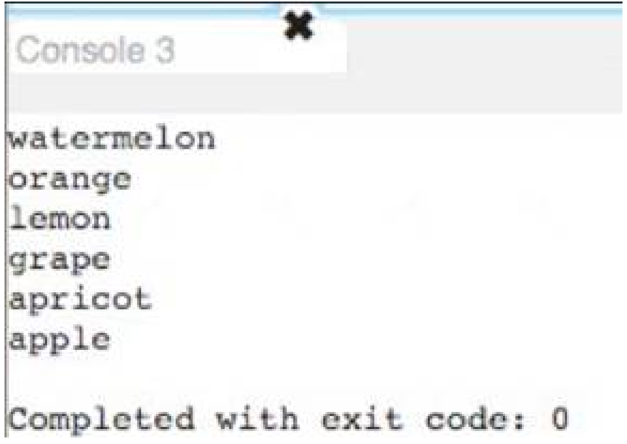
```
import java.util.ArrayList;
import java.util.Arrays;
public class NewMain {
  public static void main(String[] args) {
    String[] fruitNames = { "apple", "orange",
         "grape", "lemon", "apricot", "watermelon" };
    var fruits = new ArrayList<>(Arrays.asList(fruitNames));
    fruits.sort((var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

What is the result?

A. watermelonorangelemongrapeapricotapple
B. nothing
C. appleapricotgrapelemonorangewatermelon
D. appleorangegrapelemonapricotwatermelon

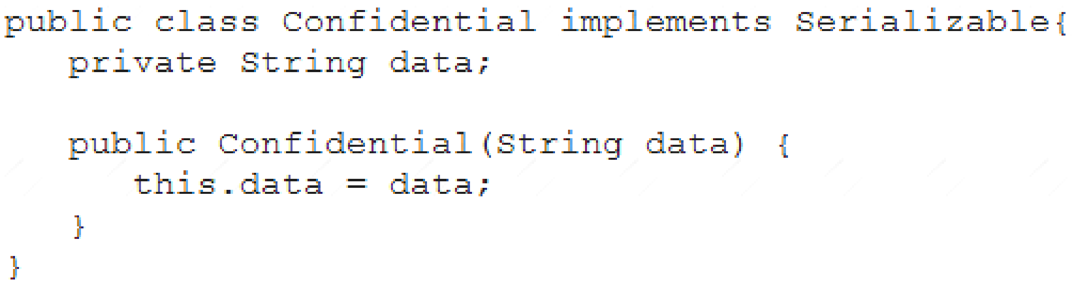
**Answer:** A

**Explanation:**



```
watermelon
orange
lemon
grape
apricot
apple

Completed with exit code: 0
```

**NEW QUESTION 38**
Given:



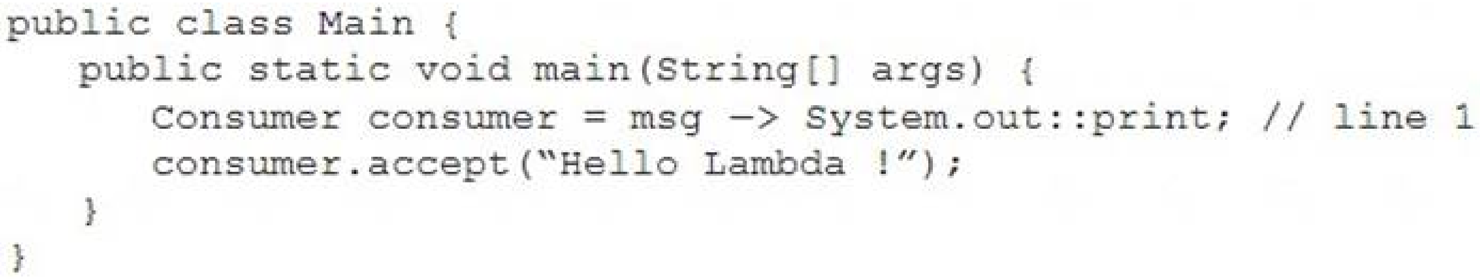
```
public class Confidential implements Serializable{
   private String data;

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

Which two are secure serialization of these objects? (Choose two.)

A. Define the serialPersistentFields array field.
B. Declare fields transient.
C. Implement only readResolve to replace the instance with a serial proxy and not writeReplace.
D. Make the class abstract.
E. Implement only writeReplace to replace the instance with a serial proxy and not readResolve.

**Answer:** AC

**NEW QUESTION 43**
Given:



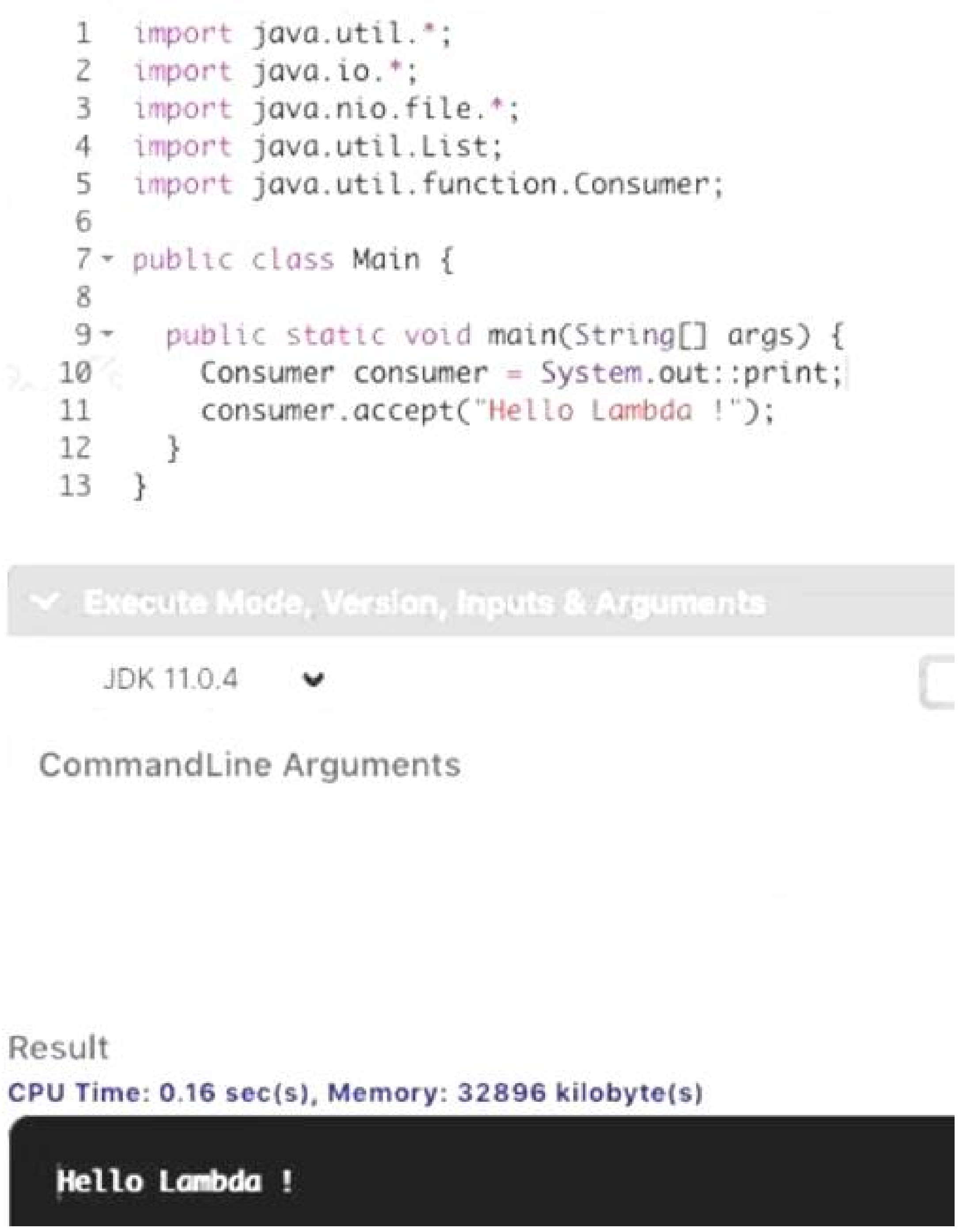
```
public class Main {
   public static void main(String[] args) {
      Consumer consumer = msg —> System.out::print; // line 1
      consumer.accept("Hello Lambda !");
   }
}
```

This code results in a compilation error.
Which code should be inserted on line 1 for a successful compilation?

A. Consumer consumer = msg -> { return System.out.print(msg); };
B. Consumer consumer = var arg > {System.out.print(arg);};
C. Consumer consumer = (String args) > System.out.print(args);
D. Consumer consumer = System.out::print;

**Answer:** D

**Explanation:**



```
import java.util.*;
import java.io.*;
import java.nio.file.*;
import java.util.List;
import java.util.function.Consumer;

public class Main {

  public static void main(String[] args) {
    Consumer consumer = System.out::print;
    consumer.accept("Hello Lambda !");
  }
}
```

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

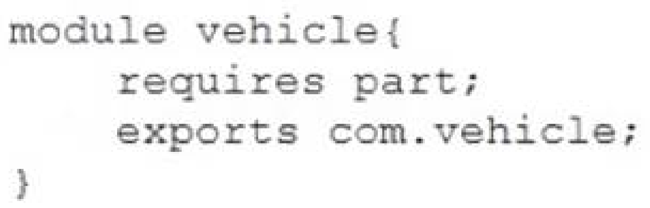
```
Hello Lambda !
```

**NEW QUESTION 46**

Given this requirement:

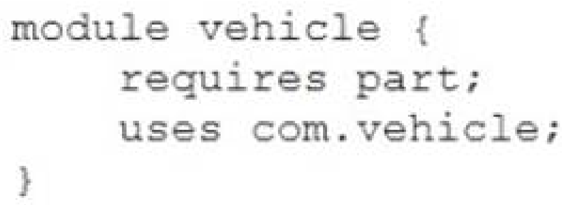
Module vehicle depends on module part and makes its com.vehicle package available for all other modules. Which module-info.java declaration meets the requirement?

A



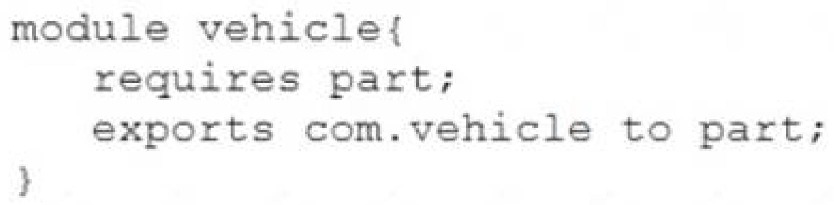
```
module vehicle{
     requires part;
     exports com.vehicle;
}
```

B



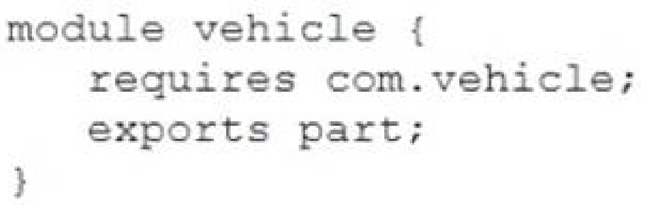
```
module vehicle {
     requires part;
     uses com.vehicle;
}
```

C



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

D

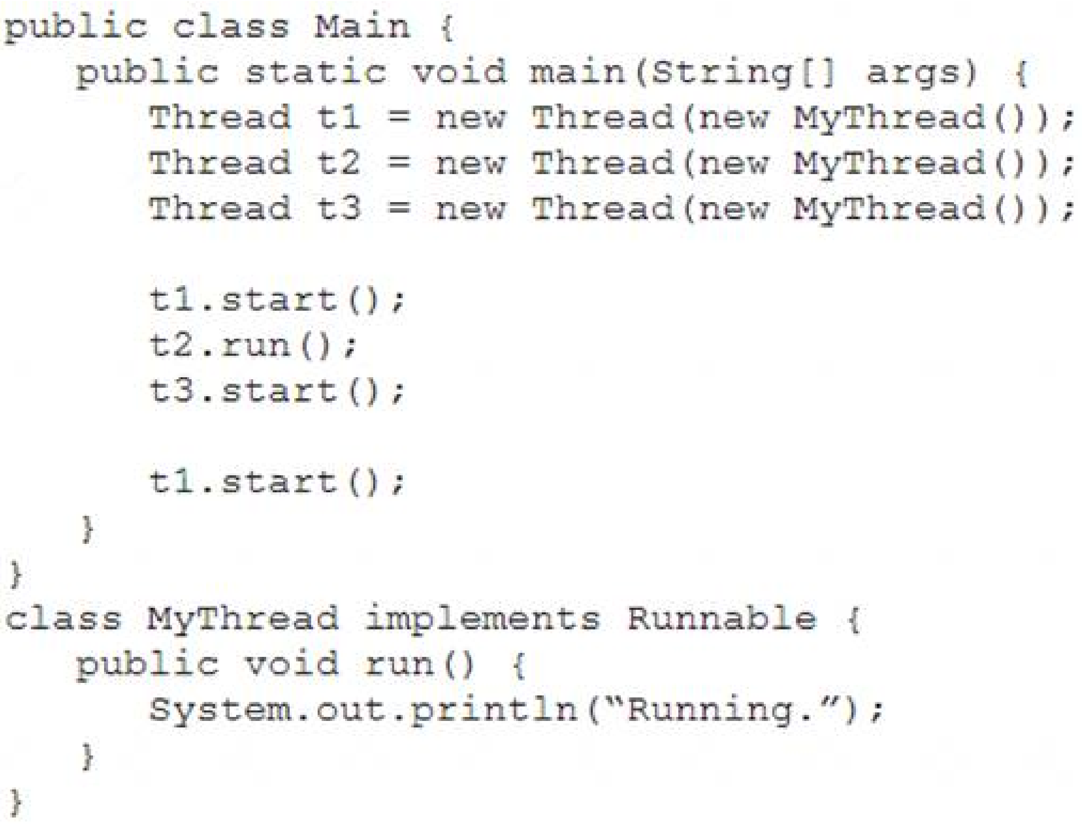


```
module vehicle {
    requires com.vehicle;
    exports part;
}
```

A. Option A
B. Option B
C. Option C
D. Option D

**Answer:** A

**NEW QUESTION 47**
Given:



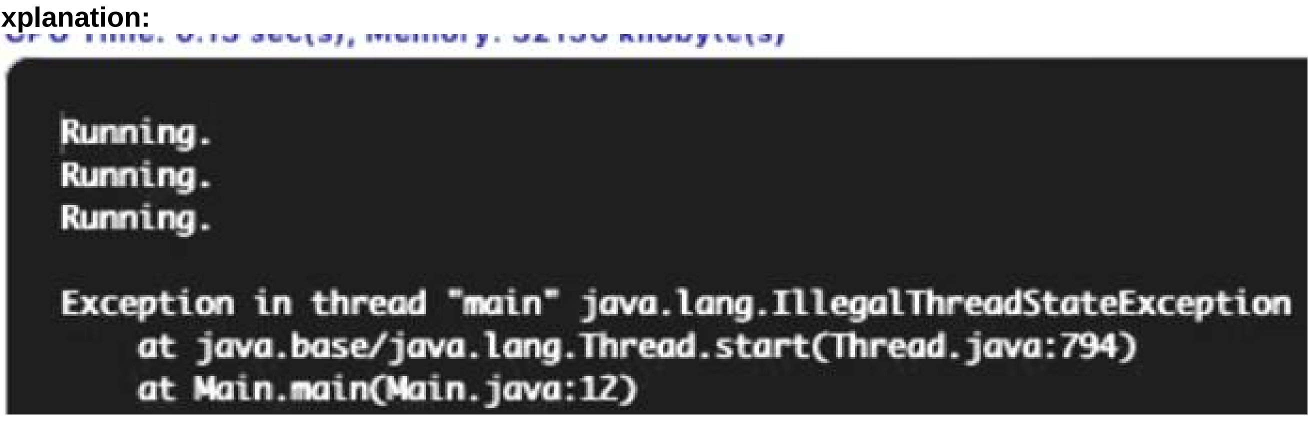
```
public class Main {
   public static void main(String[] args) {
      Thread t1 = new Thread(new MyThread());
      Thread t2 = new Thread(new MyThread());
      Thread t3 = new Thread(new MyThread());

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

      t1.start();
   }
}
class MyThread implements Runnable {
   public void run() {
      System.out.println("Running.");
   }
}
```

Which one is correct?

A. An IllegalThreadStateException is thrown at run time.
B. Three threads are created.
C. The compilation fails.
D. Four threads are created.

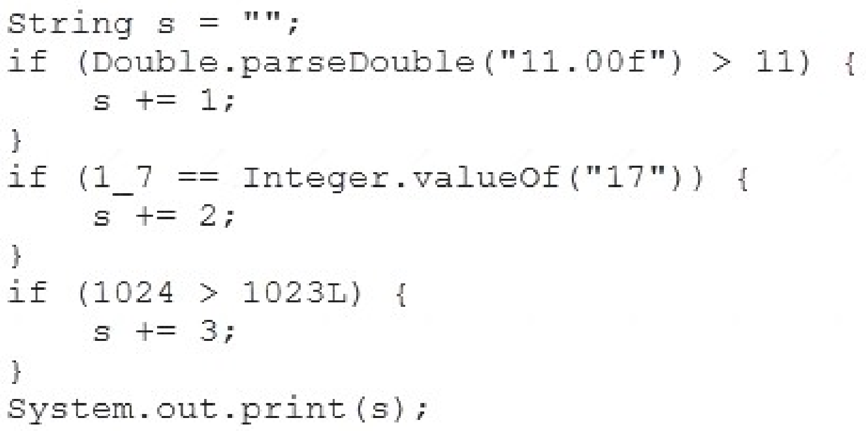
**Answer:** A

**Explanation:**



```
CPU Time: 0.15 sec(s), Memory: 32136 kilobyte(s)

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)
```

**NEW QUESTION 49**
Given the code fragment:



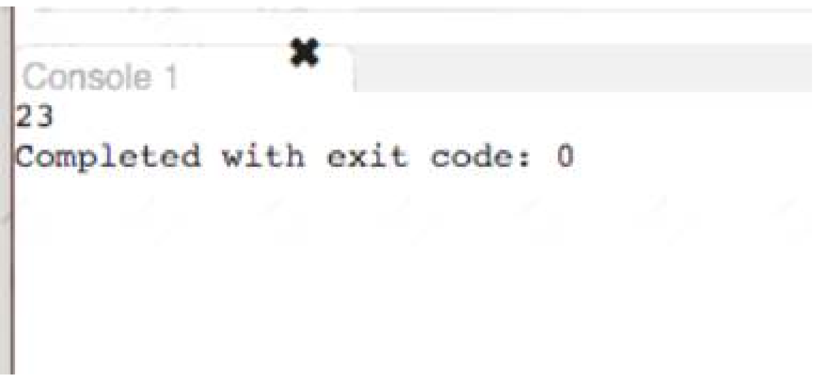
```
String s = "";
if (Double.parseDouble("11.00f") > 11) {
    s += 1;
}
if (1_7 == Integer.valueOf("17")) {
    s += 2;
}
if (1024 > 1023L) {
    s += 3;
}
System.out.print(s);
```

What is the result?

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

**Answer:** A

**Explanation:**



```
23
Completed with exit code: 0
```

**NEW QUESTION 52**
Given:

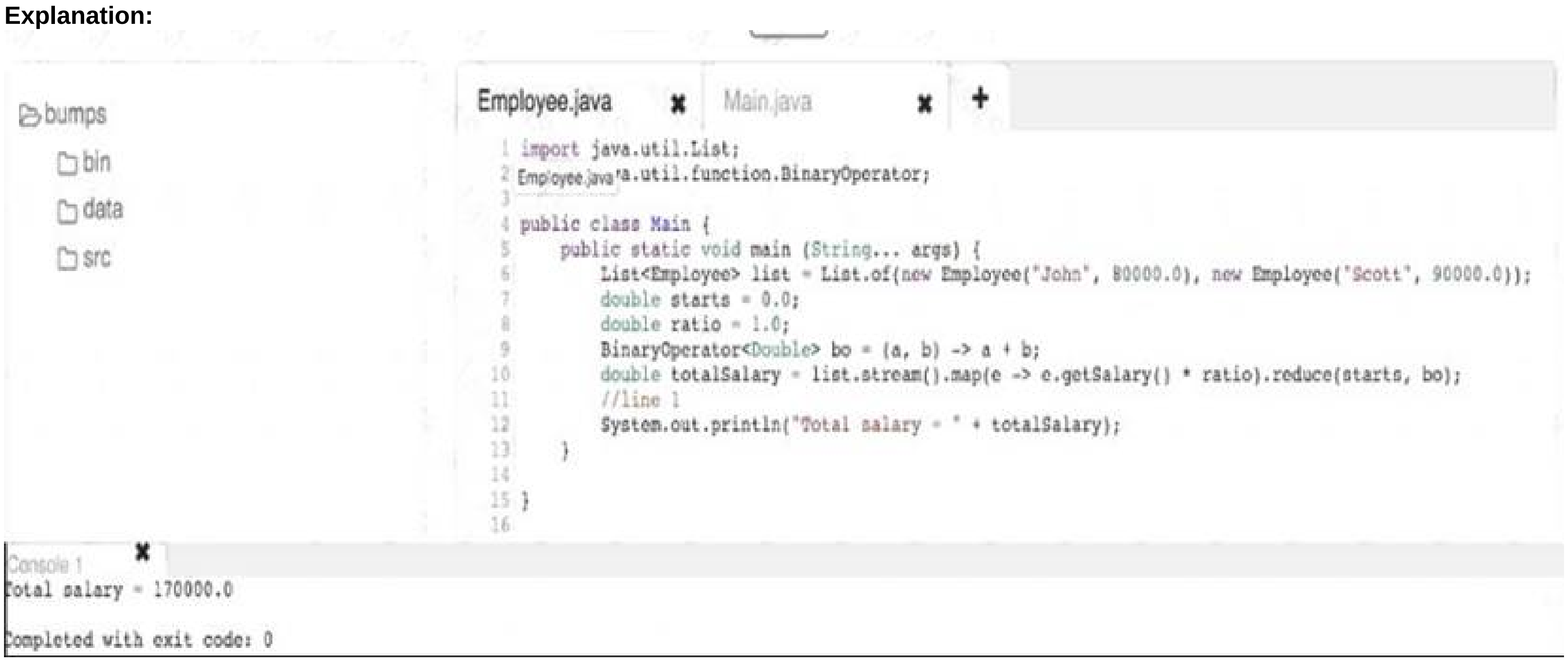
```
import java.util.List;
import java.util.function.BinaryOperator;
public class Main {
   public static void main(String... args) {
      List<Employee> list = List.of(new Employee("John", 80000.0), new Employee("Scott",
90000.0));
      double starts = 0.0;
      double ratio = 1.0;
      BinaryOperator<Double> bo = (a, b) -> a + b;
double totalSalary = list.stream().map(e -> e.getSalary() * ratio).reduce(starts, bo);
// line 1
      System.out.println("Total salary = " + totalSalary);
   }
}

class Employee {
   String name;
   double salary;
   public Employee(String name, double salary) {
      this.name = name;
      this.salary = salary;
   }
   public String getName() { return name; }
   public double getSalary() { return salary; }
}
```

Which statement is equivalent to line 1?

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

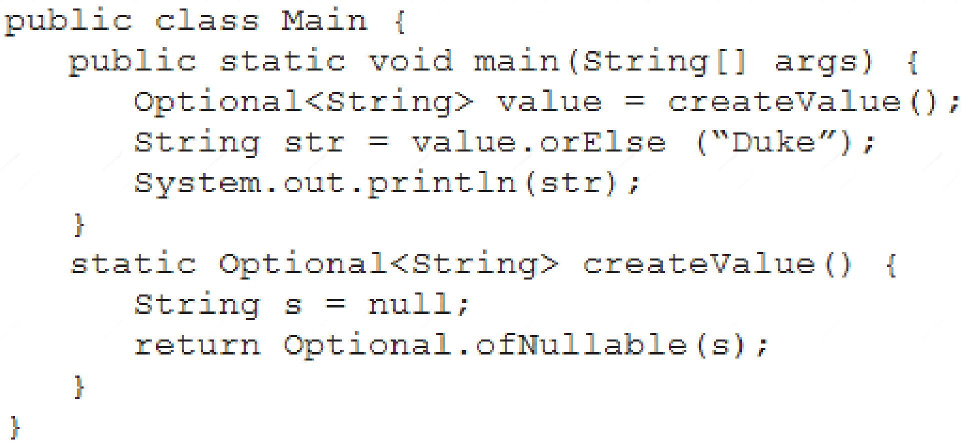
**Explanation:**



```
import java.util.List;
Employee.java'a.util.function.BinaryOperator;

public class Main {
    public static void main (String... args) {
        List<Employee> list = List.of(new Employee('John', 80000.0), new Employee('Scott', 90000.0));
        double starts = 0.0;
        double ratio = 1.0;
        BinaryOperator<Double> bo = (a, b) -> a + b;
        double totalSalary = list.stream().map(e -> e.getSalary() * ratio).reduce(starts, bo);
        //line 1
        System.out.println("Total salary = " + totalSalary);
    }

}
```

```
Total salary = 170000.0

Completed with exit code: 0
```

**NEW QUESTION 53**
Given:



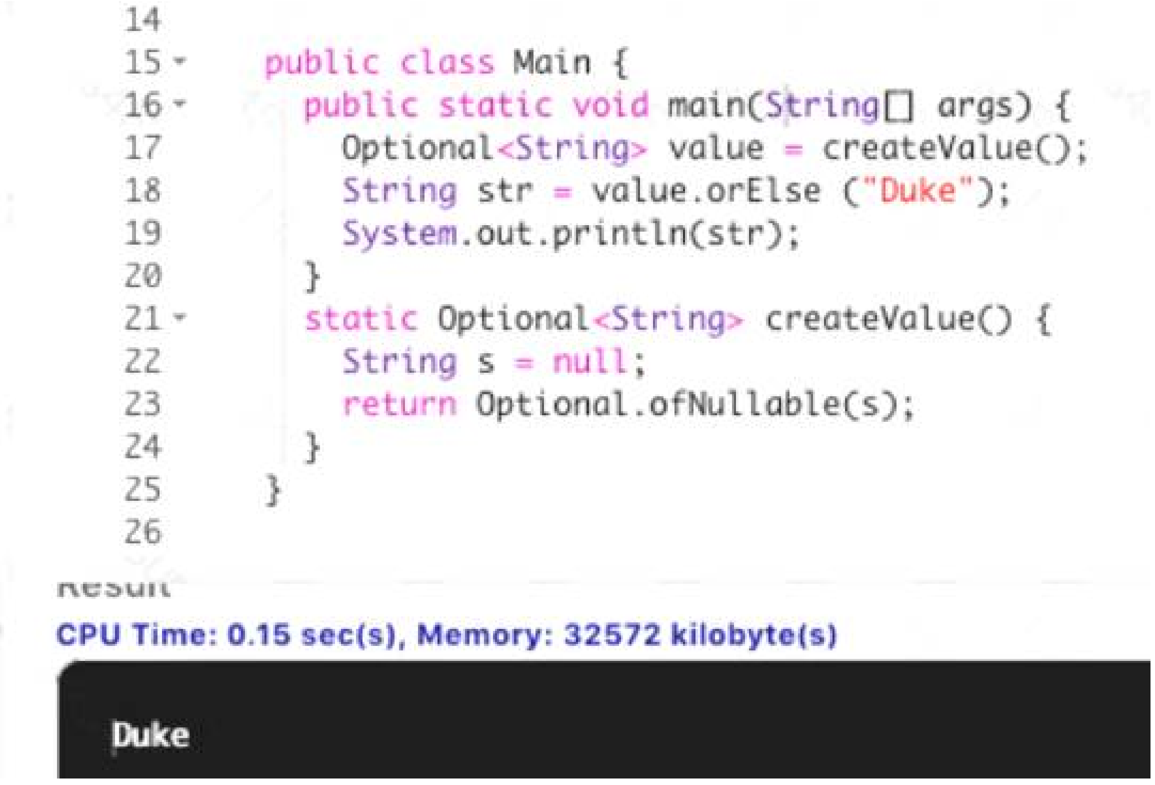
```
public class Main {
    public static void main(String[] args) {
        Optional<String> value = createValue();
        String str = value.orElse ("Duke");
        System.out.println(str);
    }
    static Optional<String> createValue() {
        String s = null;
        return Optional.ofNullable(s);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

What is the output?

A. null
B. A NoSuchElementException is thrown at run time.
C. Duke
D. A NullPointerException is thrown at run time.

**Answer:** C

**Explanation:**



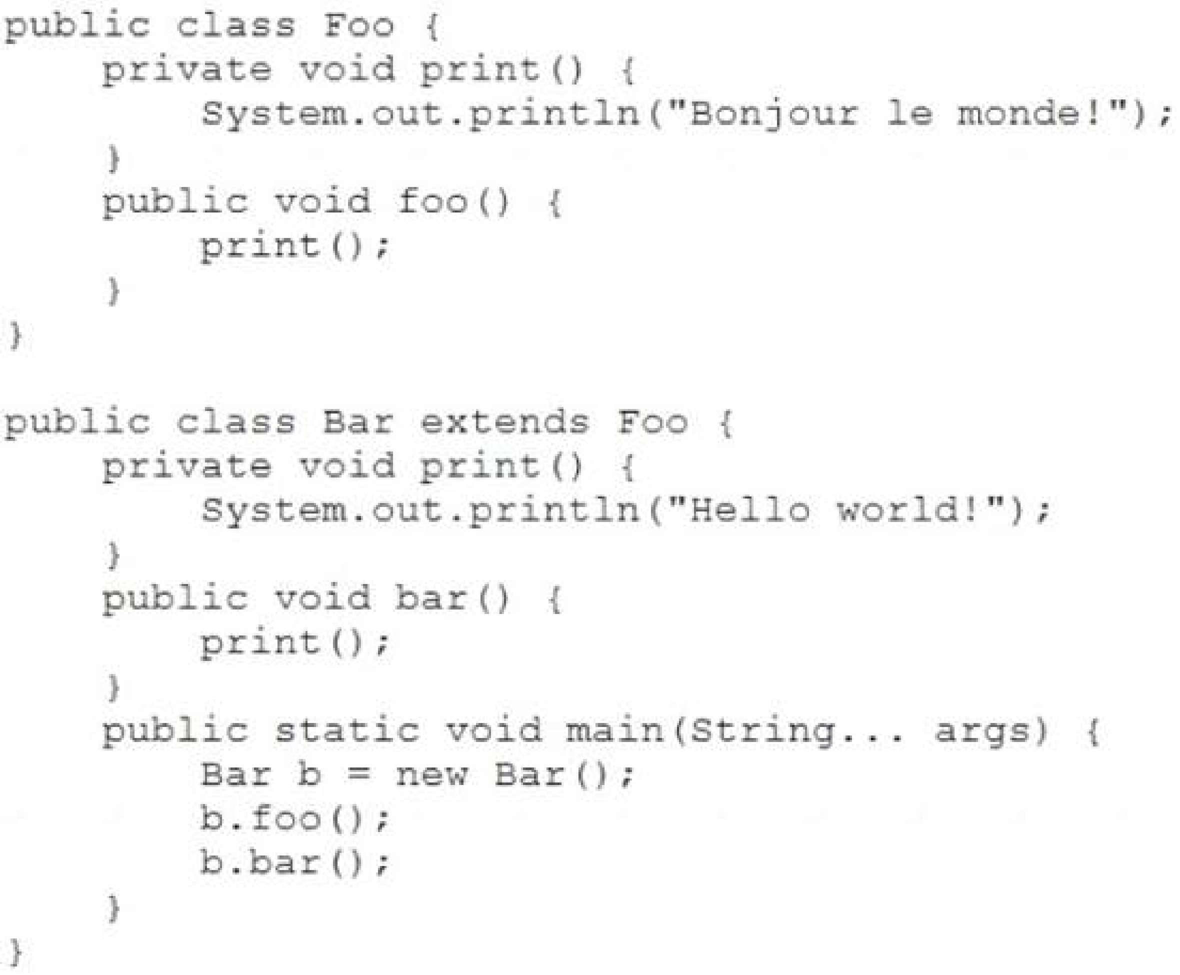
```
public class Main {
  public static void main(String[] args) {
    Optional<String> value = createValue();
    String str = value.orElse ("Duke");
    System.out.println(str);
  }
  static Optional<String> createValue() {
    String s = null;
    return Optional.ofNullable(s);
  }
}
```

Result

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

```
Duke
```

**NEW QUESTION 54**
Given:



```
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();
    }
}
```

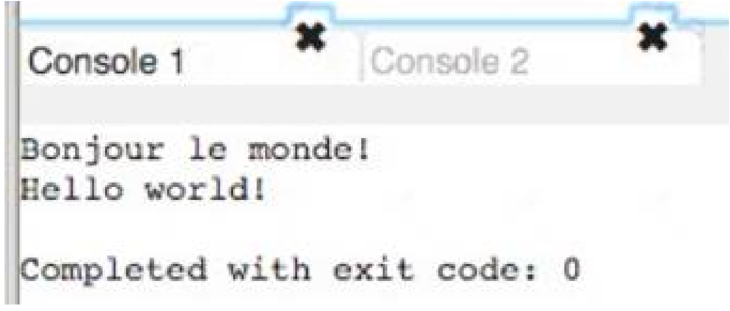
What is the output?

A. Hello world!Bonjour le monde!
B. Hello world!Hello world!

C. Bonjour le monde!Hello world!
D. Bonjour le monde!Bonjour le monde!

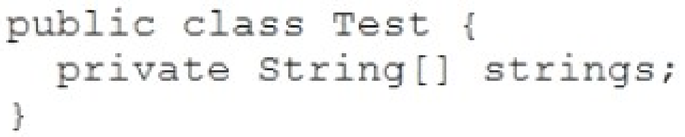
**Answer:** C

**Explanation:**



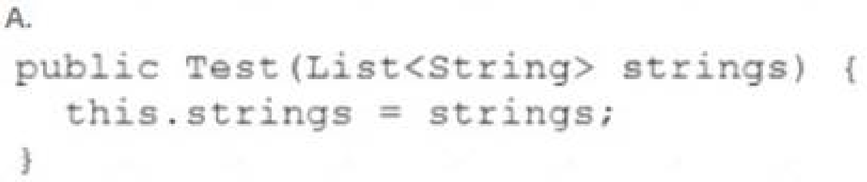
```
Bonjour le monde!
Hello world!

Completed with exit code: 0
```

**NEW QUESTION 58**
Given:



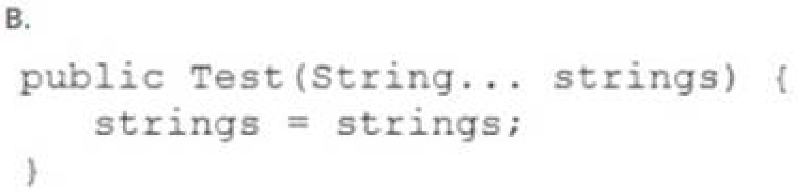
```
public class Test {
   private String[] strings;
}
```

Which two constructors will compile and set the class field strings? (Choose two.)



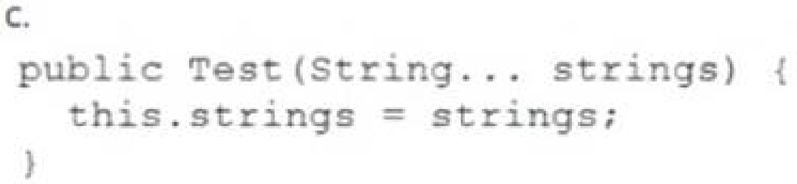
A.

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



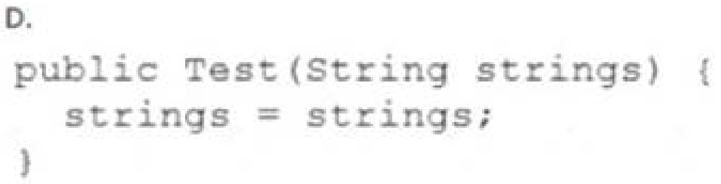
B.

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



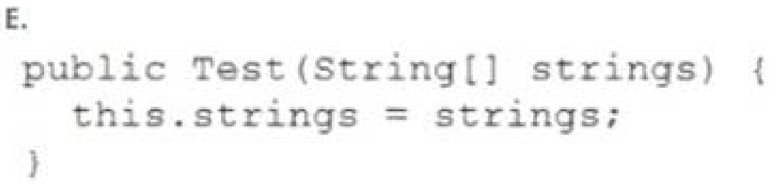
C.

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



D.

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



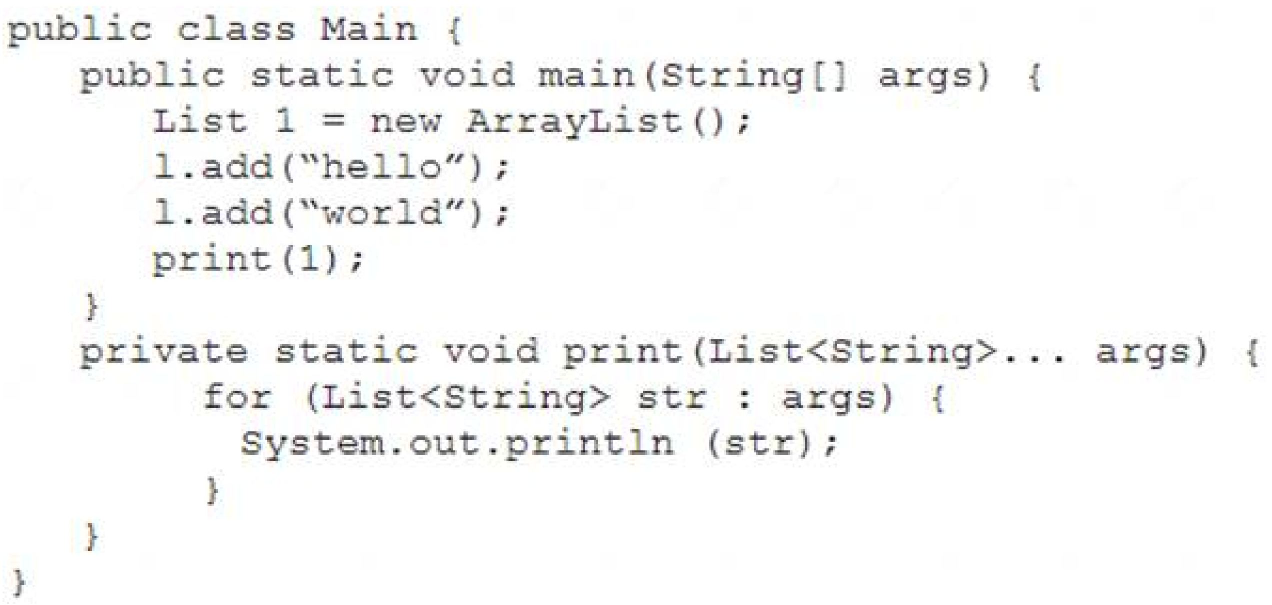
E.

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

A. Option A
B. Option B
C. Option C
D. Option D
E. Option E

**Answer:** CE

**NEW QUESTION 60**
Given:



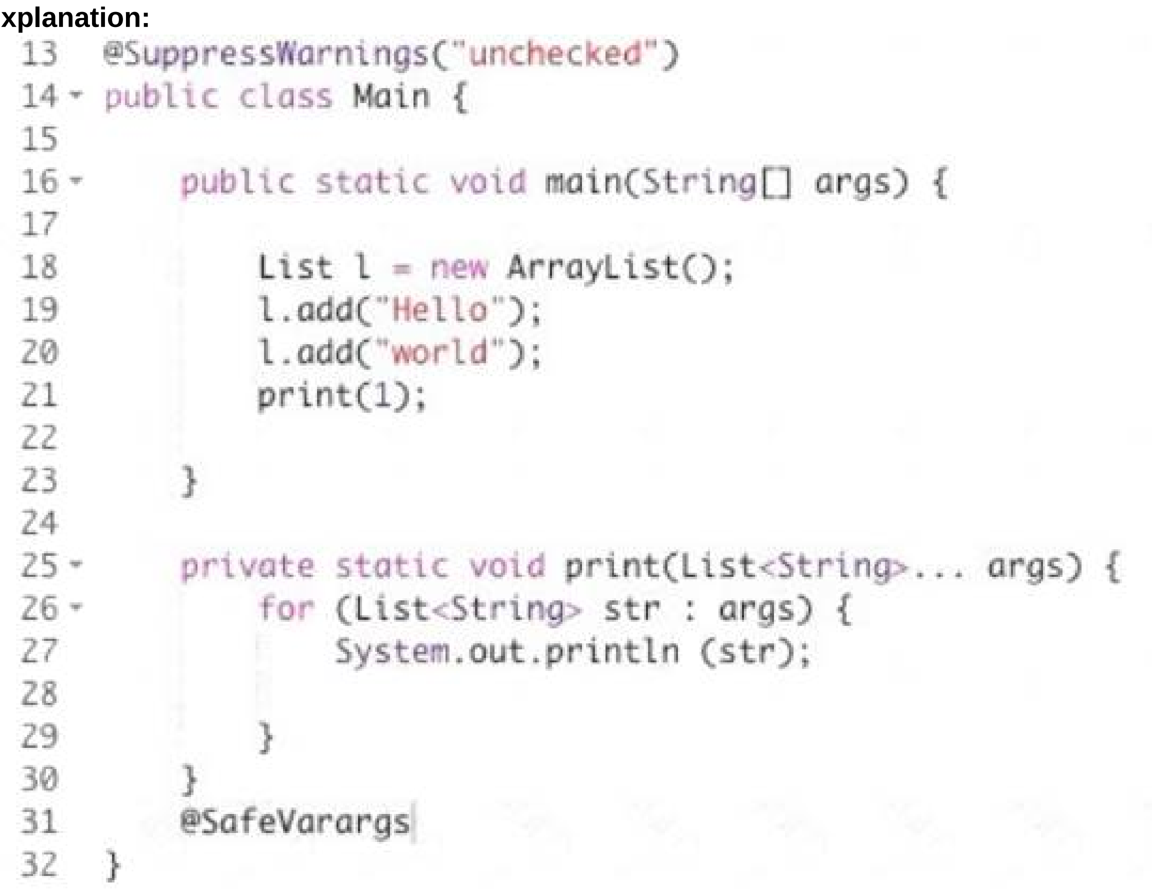
```
public class Main {
   public static void main(String[] args) {
      List 1 = new ArrayList();
      l.add("hello");
      l.add("world");
      print(1);
   }
   private static void print(List<String>... args) {
        for (List<String> str : args) {
         System.out.println (str);
        }
   }
}
```

Which annotation should be used to remove warnings from compilation?

A. @SuppressWarnings on the main and print methods
B. @SuppressWarnings("unchecked") on main and @SafeVarargs on the print method
C. @SuppressWarnings("rawtypes") on main and @SafeVarargs on the print method
D. @SuppressWarnings("all") on the main and print methods

**Answer:** B

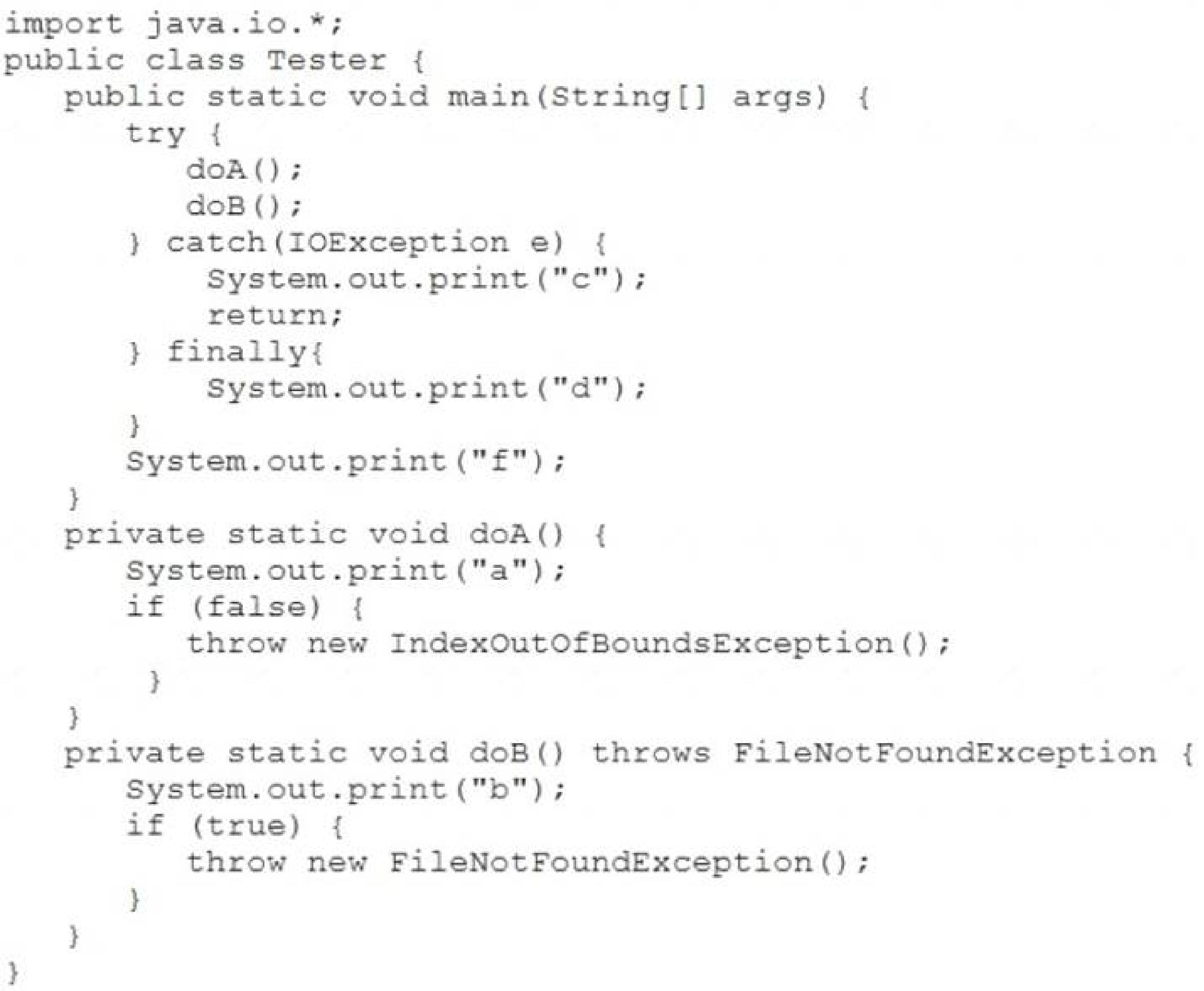
**Explanation:**



```
@SuppressWarnings("unchecked")
public class Main {

    public static void main(String[] args) {

        List l = new ArrayList();
        l.add("Hello");
        l.add("world");
        print(1);

    }

    private static void print(List<String>... args) {
        for (List<String> str : args) {
            System.out.println (str);

        }
    }
    @SafeVarargs
}
```

**NEW QUESTION 65**

Given:



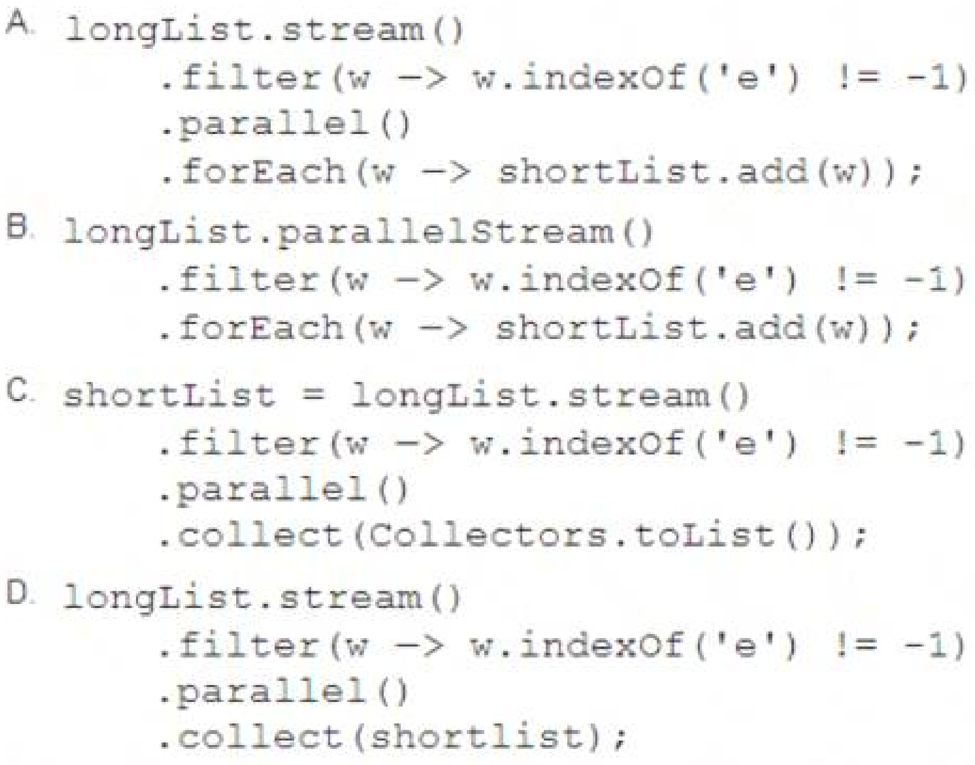
```
import java.io.*;
public class Tester {
   public static void main(String[] args) {
      try {
         doA();
         doB();
      } catch(IOException e) {
          System.out.print("c");
          return;
      } finally{
          System.out.print("d");
      }
      System.out.print("f");
   }
   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();
      }
   }
}
```

What is the result?

A. The compilation fails.
B. abdf
C. abd
D. adf
E. abcd

**Answer:** E

**NEW QUESTION 69**
Given:
List<String> longlist = List.of("Hello","World","Beat"); List<String> shortlist = new ArrayList<>();
Which code fragment correctly forms a short list of words containing the letter "e"?

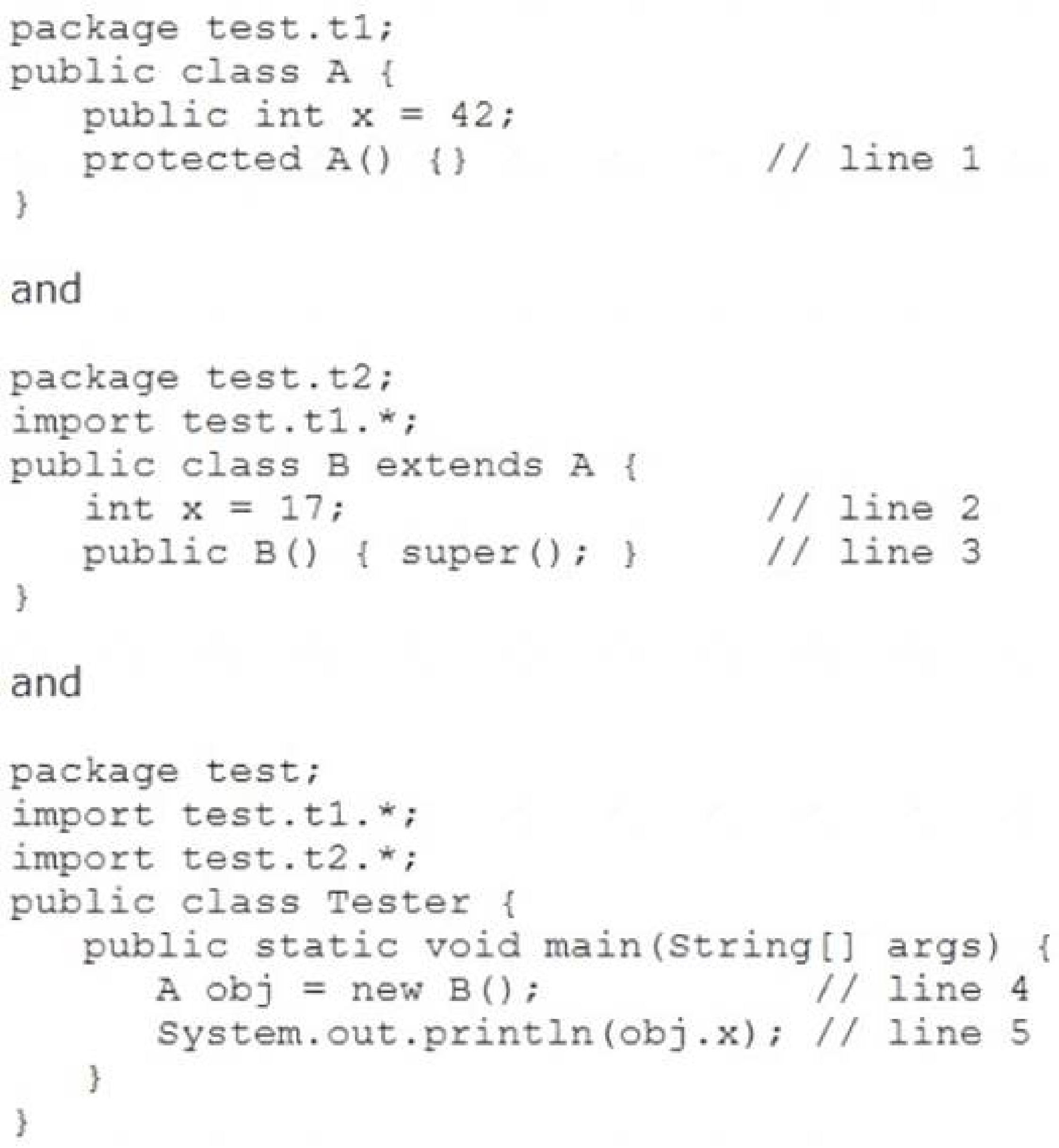


```
A. longList.stream()
        .filter(w -> w.indexOf('e') != -1)
        .parallel()
        .forEach(w -> shortList.add(w));
B. longList.parallelStream()
        .filter(w -> w.indexOf('e') != -1)
        .forEach(w -> shortList.add(w));
C. shortList = longList.stream()
        .filter(w -> w.indexOf('e') != -1)
        .parallel()
        .collect(Collectors.toList());
D. longList.stream()
        .filter(w -> w.indexOf('e') != -1)
        .parallel()
        .collect(shortlist);
```

A. Option A
B. Option B
C. Option C
D. Option D

**Answer:** C

**NEW QUESTION 70**
Given:



```
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
   }
}
```

What is the result?

A. 42
B. The compilation fails due to an error in line 4.
C. 17
D. The compilation fails due to an error in line 3.
E. The compilation fails due to an error in line 2.
F. The compilation fails due to an error in line 1.
G. The compilation fails due to an error in line 5.

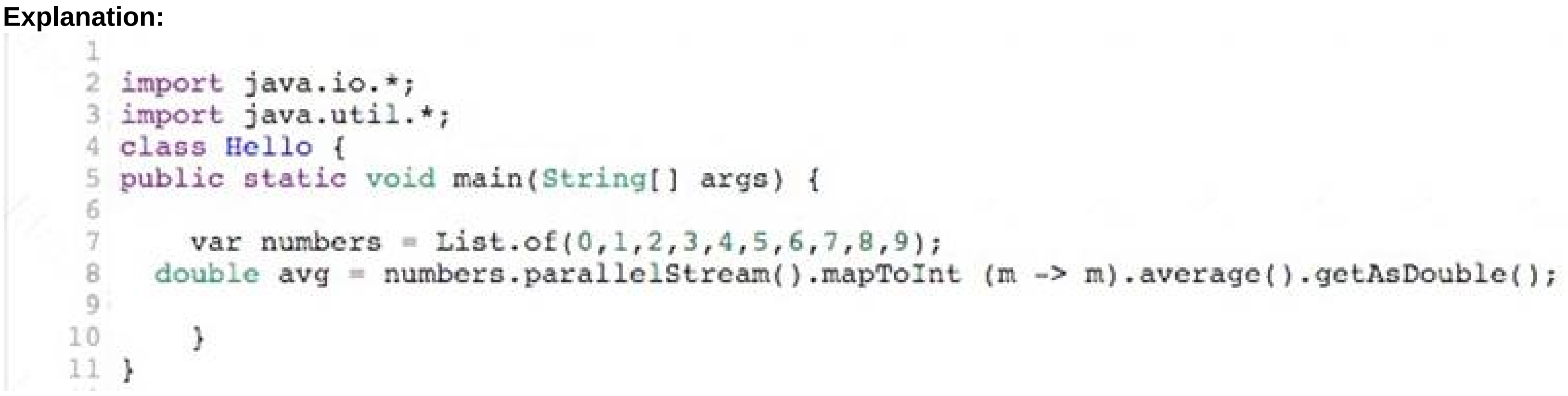
**Answer:** A

**NEW QUESTION 72**
var numbers = List.of(0,1,2,3,4,5,6,7,8,9);
You want to calculate the average of numbers. Which two codes will accomplish this? (Choose two.)

A. double avg = numbers.stream().parallel().averagingDouble(a > a);
B. double avg = numbers.parallelStream().mapToInt (m > m).average().getAsDouble ();
C. double avg = numbers.stream().mapToInt (i > i).average().parallel();
D. double avg = numbers.stream().average().getAsDouble();
E. double avg = numbers.stream().collect(Collectors.averagingDouble(n > n));

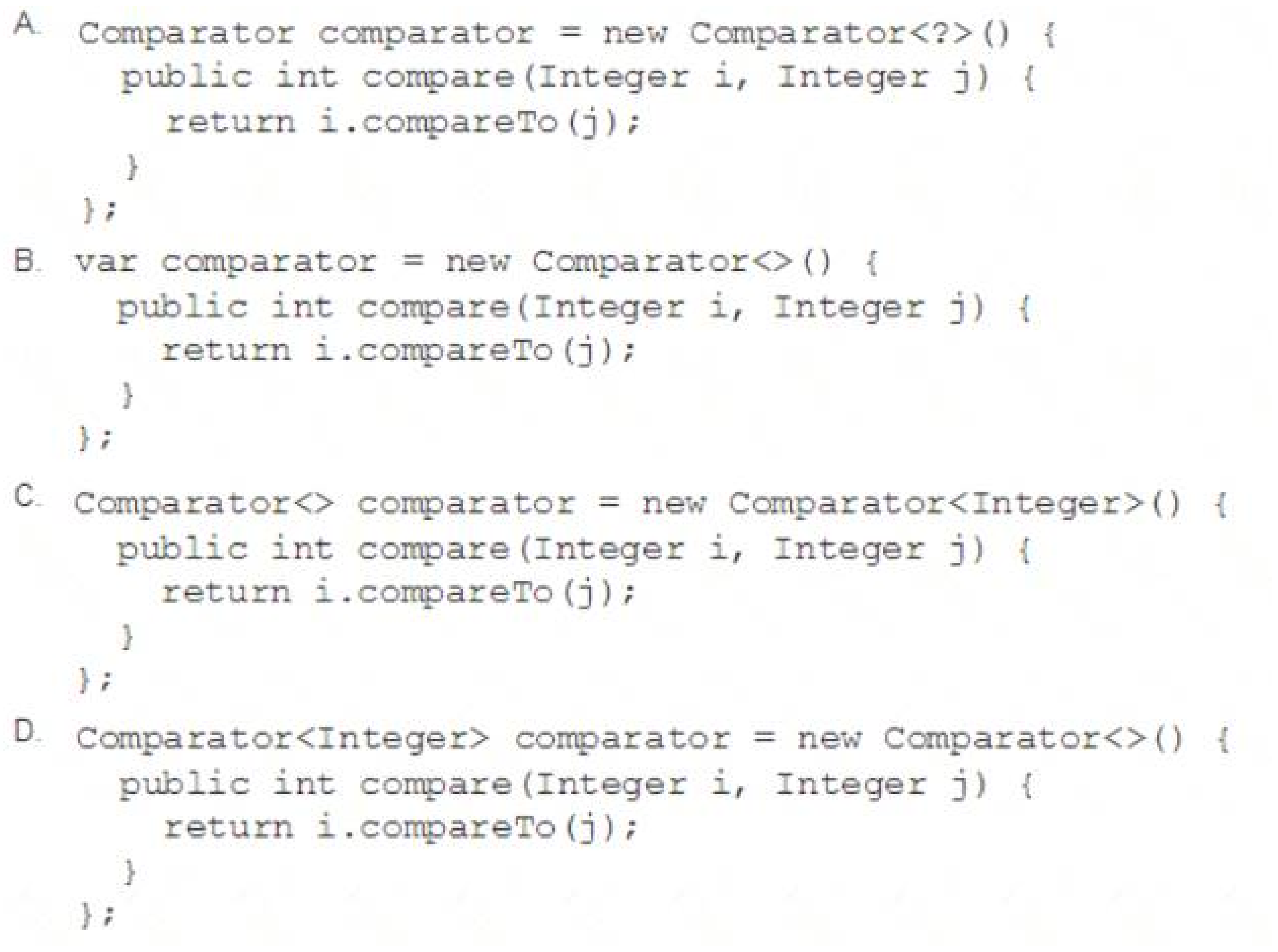
**Answer:** BD

**Explanation:**



```

import java.io.*;
import java.util.*;
class Hello {
public static void main(String[] args) {

    var numbers = List.of(0,1,2,3,4,5,6,7,8,9);
  double avg = numbers.parallelStream().mapToInt (m -> m).average().getAsDouble();

    }
}
```

**NEW QUESTION 76**
Which code fragment compiles?

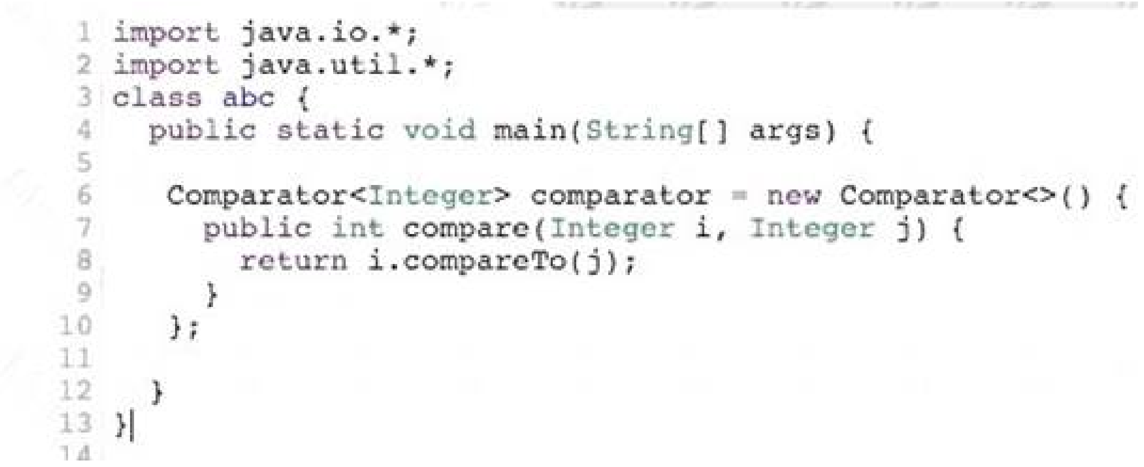


```
A. Comparator comparator = new Comparator<?>() {
     public int compare(Integer i, Integer j) {
       return i.compareTo(j);
     }
   };
B. var comparator = new Comparator<>() {
     public int compare(Integer i, Integer j) {
       return i.compareTo(j);
     }
   };
C. Comparator<> comparator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

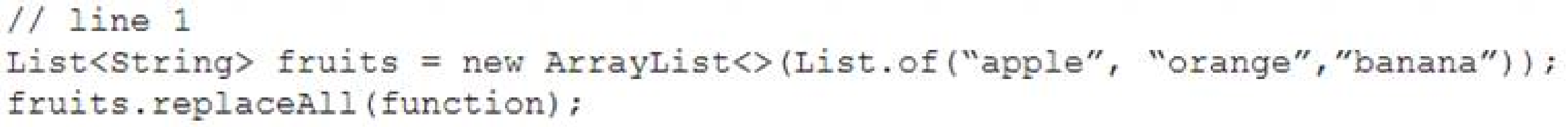
**Answer:** D

**Explanation:**



```
import java.io.*;
import java.util.*;
class abc {
  public static void main(String[] args) {

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

  }
}
```

**NEW QUESTION 78**
Given:



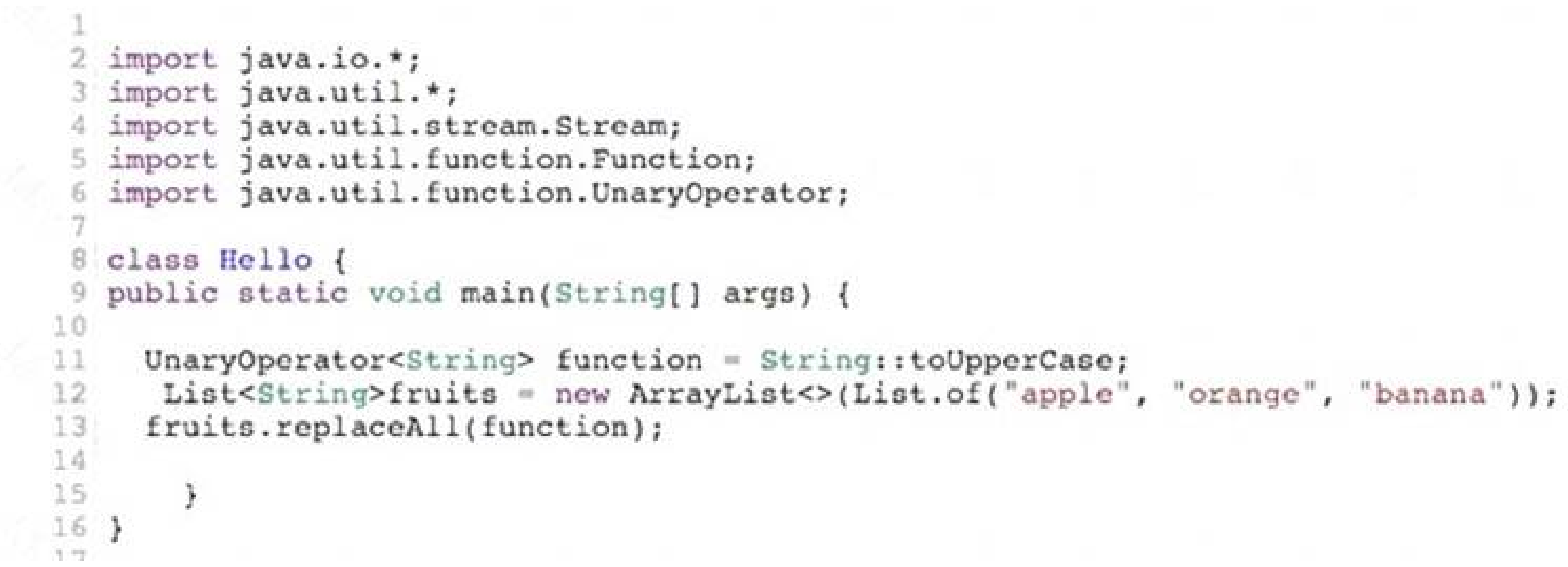
```
// line 1
List<String> fruits = new ArrayList<>(List.of("apple", "orange","banana"));
fruits.replaceAll(function);
```

Which statement on line 1 enables this code fragment to compile?

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

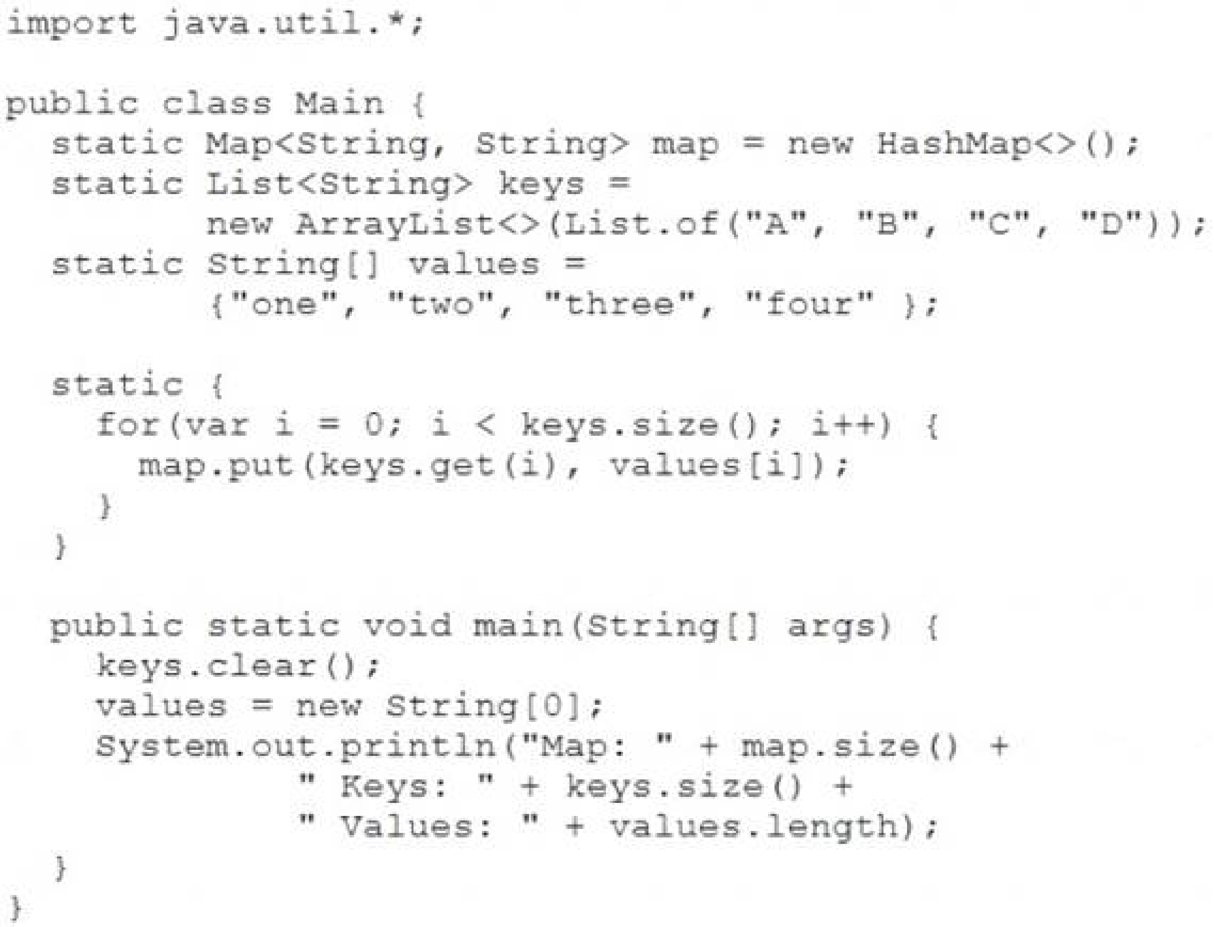
**Answer:** C

**Explanation:**



```
import java.io.*;
import java.util.*;
import java.util.stream.Stream;
import java.util.function.Function;
import java.util.function.UnaryOperator;

class Hello {
public static void main(String[] args) {

  UnaryOperator<String> function = String::toUpperCase;
   List<String>fruits = new ArrayList<>(List.of("apple", "orange", "banana"));
  fruits.replaceAll(function);

    }
}
```

**NEW QUESTION 80**
Given:



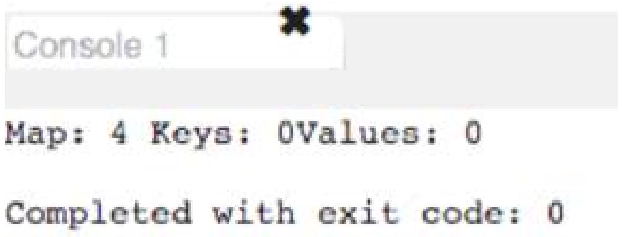
```
import java.util.*;

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);
  }
}
```

What is the result?

A. Map: 0 Keys: 0 Values: 0
B. The compilation fails.
C. Map: 4 Keys: 4 Values: 4
D. Map: 4 Keys: 0 Values: 0
E. Map: 0 Keys: 4 Values: 4

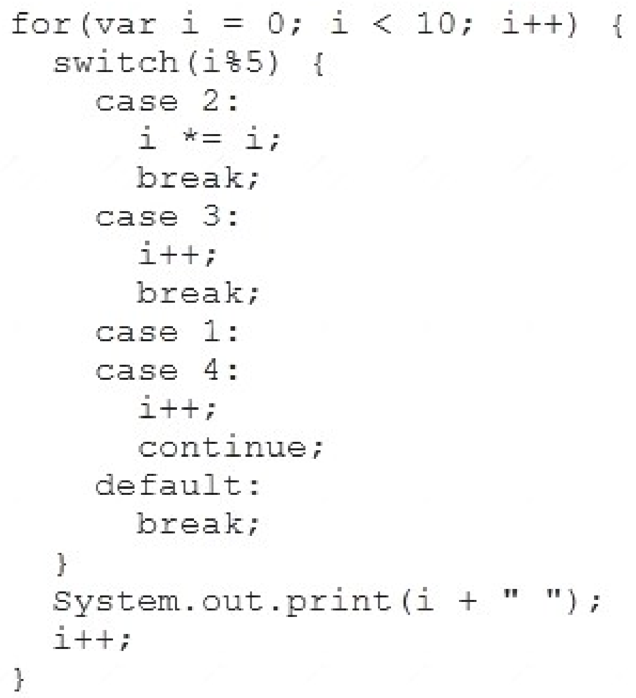
**Answer:** D

**Explanation:**



```
Map: 4 Keys: 0Values: 0

Completed with exit code: 0
```

**NEW QUESTION 82**
Given:



```
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++;
}
```

What is the result?

A. nothing
B. 10
C. 0 4 9

**Answer:** A

**NEW QUESTION 84**
Which code is correct?

A. Runnable r = “Message” > System.out.println();
B. Runnable r = () > System.out::print;
C. Runnable r = () -> {System.out.println(“Message”);};
D. Runnable r = > System.out.println(“Message”);
E. Runnable r = {System.out.println(“Message”)};

**Answer:** C

**NEW QUESTION 87**
......