1.

class Bird {

String place, color; int weight;

Bird(){weight=1;}

Bird(String place, String color){

this.place = place;

this.color = color;

this.weight = 2;

}

Bird(String place, String color, int weight){

this.place = place;

this.color = color;

this.weight = weight;

}

@Override

public String toString() {

return ("("+place+", "+color+", "+weight+")");

}

}

public class Main {

public static void main(String[] args) {

Bird x,y,z;

x = new Bird();

y = new Bird("Hola", "blue");

z = new Bird("Hola", "blue", 5);

System.out.println(x);

System.out.println(y);

System.out.println(z);

}

}

Output:

**(null, null, 1)**

**(Hola, blue, 2)**

**(Hola, blue, 5)**

2.

public class Vase {

String color;

int price;

Vase(){

price = 5;

}

Vase(String color, int price) {

this.color = color;

this.price = price;

}

@Override

public String toString() {

return (color+","+price); //To change body of generated methods, choose Tools | Templates.

}

}

class SpecVase extends Vase{

int type;

SpecVase(){type = 5;}

SpecVase(String color, int price, int type) {

super(color, price);

this.type = type;

}

void display(){

String s = "("+super.toString()+","+type+")";

System.out.println(s);

}

public static void main(String[] args) {

SpecVase v = new SpecVase();

v.display();

}

}

Output:

(null, 5, 5)

3.

A programmer need to create a logging method that can accept an arbitrary number of argument. For example, it may be called in these ways:

loglt("log message 1");

loglt("log message 2","log message 3");

loglt("log message 4","log message 5","log message 6");

which declaration satisfies this requirement?

A. public void loglt(String... msgs)

B. public void loglt(String[] msgs)

C. public void loglt(String \* msgs)

D. public void loglt(String msgs1, String msgs2, String msgs3)

A

4.

what is the output when the following program is run?

class A {public int x;}

public class Main

{

static void fun(A t) {t.x += 2;}

public static void main(String args[])

{A t = new A();

t.x = 99;

System.out.print(t.x + " ");

t.x++; // tăng 1

System.out.print(t.x + " ");

fun(t); // tăng 2

System.out.print(t.x);

}

}

A. 98 99 101

B. 99 99 101

C. 99 100 100

D. 99 100 101

E. 99 100 102

E

5.

given the following class definition

public class Upton{

public static void main(String args[]){

}

public void amethod(int i){}

//here

}

Which of the following would be illegal to place after the comment //here?

A. private void anothermethod(){}

B. public int amethod(int z){}

C. public int amethod(int i,int j){return 99;}

D. protected void amethod(long l){}

B

6.

What will happen when you attempt to compile and run the following code?

import java.io.\*;

class Base{

public static void amethod()throws FileNotFoundException{}

}

public class ExcepDemo extends Base{

public static void main(String argv[]){

ExcepDemo e = new ExcepDemo();

}

public static void amethod(){}

protected ExcepDemo(){

try{

DataInputStream din = new DataInputStream(System.in);

System.out.println("Pausing");

din.readChar();

System.out.println("Continuing");

this.amethod();

}catch(IOException ioe) {}

}

}

1) Compile time error caused by protected constructor  
2) Compile time error caused by amethod not declaring Exception  
3) Runtime error caused by amethod not declaring Exception  
4) Compile and run with output of "Pausing" and "Continuing" after a key is hit

Answer: 4

7.

What will happen when you attempt to compile and run the following code?

import java.io.\*;

class Base{

public static void amethod()throws FileNotFoundException{}

}

public class ExcepDemo extends Base{

public static void main(String argv[]){

ExcepDemo e = new ExcepDemo();

}

public static void amethod(int i)throws IOException{}

private ExcepDemo(){

try{

DataInputStream din = new DataInputStream(System.in);

System.out.println("Pausing");

din.readChar();

System.out.println("Continuing");

this.amethod();

}catch(IOException ioe) {}

}

}

1) Compile error caused by private constructor  
2) Compile error caused by amethod declaring Exception not in base version  
3) Runtime error caused by amethod declaring Exception not in base version  
4) Compile and run with output of "Pausing" and "Continuing" after a key is hit

Ans: 4

Question 3)

What will happen when you attempt to compile and run this code?

import java.io.\*;

class Base{

public static void amethod()throws FileNotFoundException{}

}

public class ExcepDemo extends Base{

public static void main(String argv[]){

ExcepDemo e = new ExcepDemo();

}

public static void amethod(int i)throws IOException{}

private boolean ExcepDemo(){

try{

DataInputStream din = new DataInputStream(System.in);

System.out.println("Pausing");

din.readChar();

System.out.println("Continuing");

this.amethod();

return true;

}catch(IOException ioe) {}

finally{

System.out.println("finally");

}

return false;

}

}

1) Compilation and run with no output.  
2) Compilation and run with output of "Pausing", "Continuing" and "finally"  
3) Runtime error caused by amethod declaring Exception not in base version  
4) Compile and run with output of "Pausing" and "Continuing" after a key is hit

Ans: 1.

---

Will this following code compile?

try{

} catch (Exception e) {

} catch (ArithmeticException a) {

}

A. This code will compile

B. This code will not compile

B

Catch đầu tiên là Exception, nó có thể catch mọi Exception, bao gồm cả ArithmeticException, nên không thể catch được catch thứ hai. Do đó code không compile.

---

In order for objects in a List to be sorted, those objects must implement which interface method?

A. Comparable interface its compare method

B. Comparable interface its compareTo method

C. Comparable interface its equals method

D. Compare interface its compareTo method

B

Interface thường có đuôi able.

---

public class Person {

int age;

String name;

public Person() {

this("Peter");

System.out.print("first ");

}

public Person(String name) {

this(42, "Peter");

System.out.printf("second ");

}

public Person(int age, String name) {

this.age = age;

this.name = name;

System.out.printf("third ");

}

public static void main(String args[]){

Person b = new Person();

System.out.printf(b.name +" " + b.age);

}

}

What is the result?

A. Peter 42 third second first

B. third second first Peter 42

C. first second third Peter 42

D. third first second Peter 42

E. Peter 42 first second third

B

"this()" màu đỏ và xanh ở trên sẽ trỏ đến các constructor Person có màu tương ứng.

this("Peter"); sẽ trỏ đến constructor chỉ có tham số là String, đó là Person(String name);

this(42, "Peter"); sẽ trỏ đến constructor nào có có tham số là int, String, đó là Person(int age, String name);

Do đó thứ tự là third second first.

Ở đây "this()" (có ngoặc) được dùng để gọi constructor của lớp hiện tại. "this()" phải đặt ở đầu mỗi constructor.

--

What is the output when you try to compile and run the following program?

public class Main{

void f(String t) {System.out.printfln("String");}

void f(StringBuffer h) {System.out.printfln("StringBuffer");}

public static void main(String argv[]){

f("ABC");

System.out.printfln();

}

}

A. String

B. No output, compile-time error

C. The code runs with no output

D. StringBuffer

A

---

1 answer

what happens when you try to compile and run this application?

import.java.util.\*;

public class Apple {

public static void main(String[] a) {

Set<Apple> set = new TreeSet<Apple>();

set.add(new Apple()); //sai từ đây

set.add(new Apple());

set.add(new Apple());

}

}

A. No exception is thrown

B. An exception is thrown at line 7 (Apple cannot be cast to java.lang.Comparable)

C. Compiler error

D. An exception is thrown at line 8 (Apple cannot be cast to java.lang.Comparable)

E. An exception is thrown **at line 6** (Apple cannot be cast to java.lang.Comparable)

B

Apple cannot be cast to java.lang.Comparable

---

which of the following classes supports developers to get the **pointer if a file**?

A. java.io.FileStream

B. java.io.File

C**. java.io.RandomAccessFile**

D. java.io.FileInputStream

C

---

which of the following is the correct syntax for suggesting that the JVM performs garbage collection?

A. System.free();

B. System.setGarbageCollection();

C. System.gc();

D. System.out.gc();

C

Cú pháp dùng để gợi ý rằng JVM thực hiện thu gom rác: System.gc();

---

1 answer

by default, the java.lang package is imported into every Java program

T

---

when creating your own class and you want to make it directly **support sorting**, which interface must it implement?

A. **Comparable**

B. Sortator

C. Sortable

D. Comparator

A

---

what will happen when you attempt to compile and run the following program (please note that the Object class does not have the foo() method):

class A {

void foo() {System.out.printf("A");}

}

class B {

void foo() {System.out.printf("B");}

}

class C extends A {

void foo() {System.out.printf("C");}

}

class Main {

public static void main(String[] args) {

Object t = new A();

t.foo(); //trong lớp Object không thể tìm thấy hàm foo(), đúng: A t = new A();

t = new B();

t.foo();

t = new C();

t.foo();

}

}

A. ABC

B. **Compile-time error**

C. BCA

D. BAC

E. ACB

B

---

Which of the following statements is **true**?

A. A final object's data cannot be changed

B. A final class can be subclassed

C. A final method cannot be overloaded

D. A final object cannot ve reassigned a now address in memory

D

Một đối tượng final không thể gán lại một địa chỉ hiện tại trong bộ nhớ

---

int j;

for(int i = 0; i < 14; i++) {

if(i < 10) {

j = 2 + i;

}

System.out.printfln("j: " + j + "i: " +i);

}

What is WRONG with the above code?

A. You cannot print integer values without converting them to strings

B. You cannot declare integer i inside the for-loop declaration

C. Nothing

D. The syntax of the "if" statement is incorrect

E. Integer "j" is **not initialized**

E

---

To write object to an object file. The right order of object creations is:

A. **FileOutputStream- ObjectOutputStream**

B. FileReader - ObjectOutputStream

C. File - ObjectOutputStream - FileOutputStream

D. File - ObjectOutputStream - Writer

A

---

what will happen when you attempt to compile and run the following frogram:

class Box {

int a,b;

Box() {}

Box(int x, int y) {a=x, b=y;}

}

class Reg extends Box {

int c;

Reg() {}

Reg(int x, int y, int z) {

this.c = z;

super(x,y);

}

void display() {

System.out.printf(a+b+c);

}

}

class Main {

public static void main(String[] args) {

Reg t = new Reg(2, 3, 4);

t.display();

}

}

A. 6

B. 5

C. 9

D. **Compile time error**

D

---

which of the following is true about Wrapped classes?

A. Wrapper classes are: Boolean, Char, Byte, Short, Integer, Long, Float, and Double

B. Wrapper classes are: Boolean, Character, Byte, Integer, Long, Float, and Double

C. Wrapper classes are classes that **allow primitive types to be accessed as objects**

C

---

2 answer

you have been givenn a design document for a veterinay registration system for implementation in Java. It states:

" A pet has an owner, a registration date, and a vaccination-due date. A cat is a pet that has a flag indicating whether it has been neutered, and a textual description of its markings."

Given that the Pet class has already been defined, which of the following fields would be appropriate for inclusion in the Cat class as members?

(Select the most appropriate two declarations).

A. Pet thePet;

B. Date vaccinationDue;

C. **boolean neutered;**

D. Date registered;

E. Cat theCat;

F. **String markings**

C F

---

Given:

10. public class classA{

11. public void count(int i) {

12. count(++i);

13. }

14. }

And:

20. ClassA a = new ClassA();

21. a.count(3);

Which exception or error should be thrown by the virtual machine?

A. NumberFormarException

B. **StackOverflowError**

C. illegalArgumentException

D. ExceptionlnlnitializerError

E. NullPointerException

B

---

Which of the following methods of the java.io.File can be used to *create a new file*?

A. newFile()

B. There is no such method. Just do File f = new File ("filename.txt"); then the newfile, named filename.txt will be created

C. **createNewFile()**

D. makeNewFile()

C

---

which of the following modifiers does not allow a variable to be modified (sửa đổi) its value once it was initialized?

A. transient

B. final

C. private

D. static

B

---

which of the following most closely dexcribes (mô tả) the process of *overriding*?

A. A method with the same name but different parameters gives multiple uses for the same method name

B. A class is prevented from accessing methods in its immadiate ancestor

C. A class with the same name replaces the functionality if a class defined earlier in the hierarchy

D. A method with the same name completely replaces the functionality of a method earlier in the hierarchy

Một phương thức có cùng tên thay thế hoàn toàn chức năng của một phương thức trước đó trong hệ thống kế thừa

D

---

select the most correct statement:

A. A protected method may only be accessed by classes or interfaces of the same package

B. A protected method may only be accessed by classes of the same package or by subclasses of the class in which it is declared

C. A protected method may only be accessed by the class in which it is declared or by the subclasses of that class

D. A protected method may only be accessed by classes or interfaces of the same package or by subclasses of the class in which it is declared

D Một phương thức protected chỉ có thể được truy cập bởi các lớp hoặc interface của cùng một package hoặc bởi các lớp con của lớp mà nó được khai báo

---

what is the output when the following program is run?

public class Test{

public static void main(String[] args){

String a = "XYZ";

String b = new String("XYZ");

if(a==b)

System.out.print(" a == b ");

else

System.out.print( a # b, ");

String c = b.intern();

if(a==c)

System.out.println(" a == c");

else

System.out.println(" a # c");}

}

A. **a # b, a == c**

B. a == b, a # c

C. a # b, a # c

D. No output

E. a == b, a == c

A

---

A compound statement (câu lệnh ghép) is:

A. **A collection** of one or more statements enclosed in braces

B. a way of setting the value of a variable

C. a way of declaring variables

D. A statement involving if and else

A

Tập hợp một hoặc nhiều câu lệnh được đặt trong dấu ngoặc nhọn

---

which statement is true about the following method?

int seltXor(int i) {

return i ^ i;

}

A. it alway **returns 0.**

B. it alway returns 1.

C. it alway an int where every bit is 1

D. The returned value varies depending on the argument

A

---

**An overridden method can be in the same class**

A. F

B. Depends on the particular implementation of the Java Virtual Machine

C. T

**F**

---

Select correct statement

A. **String objects are constants. StringBuffer objects are not**

B. StringBuffer objects are constants. String objects are not

C. Both String and StringBuffer pbject are constants

D. Both String and StringBuffer pbject are not constants

A

---

what will be the result when you attempt to compile and run the following code?

public class Conv{

public static void main(String argv[]){

Conv c = new Conv();

String s = new String("ello");

c.amethod(s);

}

public void amethod(String s){

char c='H';

c += s;

System.out.println(c);

}

}

A. Compilation and output the string "Hello"

B. Compilation and output the string "ello"

C. Compilation and output the string elloH

D. **Compile time error**

D

---

the **ability** of a programming **language** to process objects differently depending on their type is

A, Inheritance

B. Overloading

C. Polymorphism

D. Astraction

E. Encapsulation

F. Overriding

khả năng của một ngôn ngữ lập trình để xử lý các đối tượng khác nhau tùy thuộc vào loại của chúng là

C

---

An instance of the java.ulti.Scanner class can read date from the keyboard (1), a file (2), a string of characters (3)

1 is ..., 2 is ..., 3 is ...

A. None of the others

B. T F F

C. T F T

D. T T F

E. T T T

E. **T T T**

---

what is the output of the following code?

1: String str = "Welcome";

2: str.concat( to Java!");

3: System.out.println(str);

A. Prints "Welcome to Java!"

B. Prints **"Welcome"**

C. Runtime exception at line 2

D. Compilation error at line 2

B

---

(Choose 1 answer)

Consider the following class definition:

1. public class Test extends Base{

2. public *Test(int )* {

3. }

4. public Test(int j, int k) {

5. super(i, k);

6. }

7 .}

Which of the following is legal calls to construct instances of the Test class?

A. Test t = new Test(1, 2, 3);

B. Test t = new Test;

C. Test t = new **Test(1);**

D. Test t = new Test() ;

C

---

Select a correct statement about interfaces

A. An interface is a class that has method implementations with no data

B. An interface is an abstract class with no date

C. In its most common form, an interface is a group of related methods with empty bodies

D. An interface is a class that has at least one abstract method

C

Ở dạng phổ biến nhất, interface là một nhóm các phương thức có liên quan không có body

---

Select the most correct statement

A. If a checked exception may be thrown within the body of a method, the method must catch the exception

B. If a checked exception may be thrown within the body of a method, the method must declare it in its throwns clause

C. If a checked exception may be thrown within the body of a method, **the method must either catch the exception or declare it in its throws clause**

D. If a checked exception may be thrown within the body of a method, the method mus tcatch the exception and eclare it in its throws clause

C

---

what will happen it you try to compile and run the following code:

public class MyClass{

static int i;

public static void main(String argv[]){

System.out.println(i);

}

}

A. null

B. 1

C. Error Variable i may not have been intialized

D. 0

D

Không khởi tạo giá trị => i = 0

---

given the following classes. which of the following statements will not compile?

interface IFace{}

class CFace implements IFace{}

clase base{}

public class ObRef extends Base{

public static void main(String argv[]){

ObRef ob = new ObRef();

Base b = new Base();

Object o1 = new Object();

IFace o2 = new CFace();

}

}

A. o1=b;

B. **ob=b;**

C. b=ob;

D. o1=o2

B

---

interfaces cannot extend ...... . but they can extend......

A. **classes, interfaces**

B. classes, objects

C. interfaces, classes

a

---

\_\_\_\_ allows an operation to have different **behavior** on different objects

A. **Polymorphism**

B. Encapsulation

C. Destruction

D. Construction

E. Inheritance

F. Persistence

A

---

2 Answer

which of the following are *legal*?

A. Vector <String> theVec = new Vector<String>**()**;

B. List<String> theList = new Vector<String>**()**;

C. List<String> theList = new Vector<String>;

D. Vector<String> theVec = new Vector<String>;

ab

---

2 Answer

given the following code

which of the following statements can be legally inserted in place of the comment //here?

class Base{}

public class MyCast extends Base{

static boolean b1=false;

static int i = -1;

static double d = 10.1;

public static void main(String argv[]){

MyCast m = new MyCast();

Base b = new Base();

//here

}

}

A. b1=i;

B. **b=m;**

C. **d=i;**

D. m=b;

B C

---

given the following code method definition in a class that otherwise compiles correctly:

1.public booleantestAns(String ans, int n){

2. boolean slt;

3. if( ans.equalsgnoreCase("YES")&& n > 5 ) rsit = true;

4. return rsit;

5.}

what will be the result of trying to compile the class and execute the testAns method with inputs of "no" and 5?

A. A compiler error will **prevent compilation**

B. A runtime exception will be thrown in the testAns method.

C. A result of false will be returned

D. A result of true will be returned

A

---

select the **correct syntax** for throwing an exception when declaring a method

A. [Modifier] {Return type] Identifier (Parameters) throws TypeOfException

B. [Modifier] {Return Type] Identifier (Parameters) *{*

***throws*** *TypeOfException;*

*}*

C. [Modifier] {Return type] Identifier (Parameters){

throw TypeOfException;

}

D. [Modifier] {Return type] Identifier (Parameters) throw TypeOfException

E. None of the others

B

---

2 answers

which of the following may be declared final (khai báo sau cùng)?

A. Methods

B. Classes

C. Interfaces

A B

---

what method of the java.io.File class can **create** a file on the hard drive?

A. makeNewFile()

B. newFile()

C. makerFile()

D. createFile()

E. createNewFile()

E

---

how do you use the File class to list the contents of a directory?

A. StringBuider [] contents = newFile.list();

B. The File class does not provide a way to list the contents of a directory

C. String [] contents = myFile.list();

D. File [] contents = myFile.list();

C

**String [] contents = myFile.list();**

---

consider the following cod:

1.**Dog rover, fido;**

2.Animal anim;

3.

4.rover = new Dog();

5.anim = rover;

6.fido = (Dog)anim;

where:

Mammal extends Animal

Dog extends Mammal

which of the following statements is true?

A. Line 5 will not compile.

B. The code will compile and run

C. The code will compile and run, but the cast in line 6 is not required and can be eliminated

D. Line 6 will not compile

E. The code will compile but will throw an exception at line 6

B

**The code will compile and run**

---

a class defines an **entity**, while an object is the actual entity

a.true

b.false

a

---

is this code snippet incorrect?

List<String>myIntList = new LinkedList<String>();

myIntList.add(0);

a. false

b. true

B

---

a(n) \_\_\_\_ is a **characteristic** that describes an object

A. **attribute**

B. Method

C. Message

D. Operation

E. Event

A

---

select INCORRECT statement about serialization.

A. **when an Object Output Stream** serializes an object that contains references to another object, every references object is not serialized along with the original object

B. The process of writing an object is called serialization

C. When an object is serialized, it will probably be deserialized by a different JVM

D. To serialize an object, firts create an instance to java.io.ObjectOutputStream.

A

---

which of the following may appear on the **left-hand** side of an instanceof operator?

A. A class

B. An interface

C. A reference

D. A variable of primitive type

C

Right-hand => class và interface

---

all the method of the \_\_\_ class are static

A. String

B. Math

C. Svstem

D. Runtime

B

các hàm về toán học được sử dụng mà không cần phải tạo ra một đối tượng lớp **Math**

---

what will happen when you attempt to compile and run the following code

public class As{

int j = 10;

int j;

char z = 1;

boolean b;

public static void main(String argv[]){

As a = new As();

a.amethod();

}

public void amethod(){

System.out.println(j);

System.out.println(b);

}

}

A. Compilation succeeds and at run time an **output of 0 and false**

B. Compile time error b is not initizlised

C. Compile time error z must be assigned a char value

D. Compilation succeeds and at run time an output of 0 and true

A

---

given the following code, which of the results that follow would you expect?

1. package mail;

2.

3. interface Box {

4. protected void open();

5. void close();

6. public void empty();

7. }

A. The code will not compile **because of line 4.**

B. The code will compile

C. The code will not compile because of line 6.

C. The code will not compile because of line 5.

A

---

given:

public class Pass {

public static void main(String[] args) {

int x = 5;

Pass p = new Pass();

p.doStuff(x);

System.out.print(" main x = " + x);

}

void doStuff(int x) {

System.out.print("doStuff x = " + x++);

}

}

what is the result?

A, Compilation fails

B. doStuff x = 5 main x = 6

C. An exception is thrown at runtime.

D. doStuff x = 6 main x = 5

E. **doStuff x = 5 main x = 5**

F. doStuff x = 6 main x = 6

E

* Toán tử tăng trước ++x: tăng giá trị x trước khi thực hiện các phép toán khác trong cùng 1 câu lệnh
* Toán tử tăng sau x++: tăng giá trị x sau khi thực hiện các phép toán khác trong cùng 1 câu lệnh

---

An *object* is an **instance** of a *class*

A. F

B. T

T

---

import java.io.FileInputStream;

import java.io.IOException;

public class Ppvg {

public static void main(String[] args) {

Ppvg p = new Ppvg();

p.fliton();

}

public int fliton() {

try {

FileInputStream din = new FileInputStream("Ppvg.java");

din.read();

} catch (IOException ioe) {

System.out.println("flytwick");

return 99;

} finally {

System.out.println("fliton");

}

return -1;

}

}

Assuming the file Ppvg.java is avaliable to be read wich of the following statements is true if you try to compile and run the program?

A. The program will run and output both **"fliton" and "flytwick"**

B. The program will run and output only "fliton"

C. An error will occur at compile time because the method fliton attempts to return two values

D. The program will run and output only "flytwick"

A

Catch xong => finally luôn được thực thi cuối cùng

---

given that you have a method scale defined as follows, where scalex and scaley are constants

public Point scale( int x, int y ){

return new Point(

( int )(x / scalex ),

( int )( y / scaley ) ) ;

}

what will happen when you call this method with double primitives instead of int, as in the following fragment?

1. double px = 10.02 ;

2. double py = 20.34;

3. Point thePoint = **scale( px, py )** ;

A. The compiler objects to line 1.

B. The program compiles and runs

C. The program compiles but a runtime cast exception is thrown.

D. A compiler error occurs in **line 3**

D

---

what will be the output of the following code?

1. public class **integerequals**

2. {

3. public static void main (String args[])

4. {

5. Integer a = new Integer(0);

6. Integer b = new Integer(0);

7. System.out.println(a==b);

8. }

9. }

A. The program compiles but causes a runtime exception at line 7

B. The program compiles and prints true

C. The program **compiles and prints false**

D. The compiler will show an error at line 7

C

2 đối tượng a, b new không thể == nên in ra false

---

which of these class is used to **read** characters in a file?

A. **FileReader**

B. InputStreamReader

C. FileInputStream

D. FileWriter

A

---

which of the following may override a method whose signature is void xyz(float f)?

A. public int xyz(float f)

B. public void xyz(float f)

C. private int xyz(float f)

D. private void xyz(float f)

B

Override => cùng signature

---

consider the following class:

1. class Test{

2. void foo(int i) {

3. System.out.println("int version");

4. }

5. void foo(String s) {

6. System,out.println("String version");

7. }

8.

9. public static void main(String args[]) {

10. Test t = new Test();

11. char ch = 'p';

12. t.foo(ch);

13. }

14. }

which of the following statements is true?

A. The code will compile and produce the following output: String version.

B. Line 12 will not compile, because no version of foo() takes a char argument.

C. Line 5 will not compile, because void methods cannot be overridden.

D. The code will compile and produce the following output: **int version.**

D

---

in which stream, data unit is primitive data type or string?

A. Binary high-level stream

B. Binary low-level stream

C. Character stream

D. Object stream

A

---

\_\_\_\_ is the **process of identifying and grouping** attributes and actions related to a particular entity as relevant to the application at hand

A. Persistence

B. Construction

C. Polymorphism

D. Data abstraction

E. Inheritance

F. Encapsulation

B

là quá trình xác định và nhóm các thuộc tính và hành động liên quan đến một thực thể cụ thể có liên quan đến ứng dụng hiện có

---

given:

11. public static void main(String[] args) {

12. Object obj = new int[] {1, 2, 3};

13. int[] someArray = (int[])obj;

14. for (int i: someArray) System.out.print(i +" ");

15. }

what is the result?

A. Compilation fails because of an error in line 14.

B. A ClassCastException is thrown at runtime.

C. Compilation fails because of an error in line 13

D. **1 2 3**

E. Compilation fails because of an error in line 12/

D

---

what is the output of the following code?

1: int i = 16;

2: int i = 17;

3:

4: System.out.println("i >> 1 = " + (i >> 1));

5. System.out.println("j >> 1 = " + (j >> 1));

A. **i>>1 = 8**

**j>> 1 = 8**

B. i>>1 = 7

j>> 1 = 8

C. i>>1 = 8

j>> 1 = 9

D. Depends on the particular implementation of the Java Vitual Machine

E. i>> 1 = 7

j>> 1 = 7

A

---

what will happen when you attempt to compile and run the following code?

public class Agg{

static public **long** i=10;

public static void main(String argv[]){

switch(i){

default:

System.out.println("no value given");

case 1:

System.out.println("one");

case 10:

System.out.println("ten");

case 5:

System.out.println("five");

}

}

}

A. **Compile time error**

B. Output of "ten" followed by "five"

C. Output of "ten"

D. Compilation and run time error because of location of default

A

Switch(i) thì i không thể là long, i chỉ có thể là int, char, byte, short, enum

---

suppose prim is an int and wrapped is an Interger. Which of the following are legal Java statements?

A. prim = new Integer(9);

B. **All the others**

C. prim = wrapped;

D. wrapped = prim;

E. wrapped = 9;

B

---

2 answers

which of the following keywords can be applied to the variables or methods of an **interface** ?

A. public

B. static

C. Depends on the particular implementation of the Java Virtual Machine

D. private

E. protected

F. inner

A B

từ khóa nào sau đây có thể được áp dụng cho các biến hoặc phương thức của **giao diện**?

=> public, static

---

given

20. public class CreditCard {

22. private String cardID;

23. private Integer limit;

24. public String ownerName;

26. public void setCardInformation(String cardID,

27. String ownerName,

28. Integer limit) {

29. this.cardID = cardID;

30. this.ownerName = ownerName;

31. this.limit = limit;

32. }

33. }

which is true?

A. The **ownerName** variable breaks encapsulation.

B. The cardID and limit variables break polymorphism.

C. The class is fully encapsulated.

D. The code demonstrates polymorphism

A

---

what will happen when you attempt to compile and run the following code?

public class Inc{

public static void main(String argv[]){

Inc inc = new Inc();

int i = 0;

inc.fermin(i);

System.out.println(i++);

}

void fermin(int i){

i++;

}

}

A. Compile time error

B. Output of 2

C. Output of 1

D. **Output of 0**

D

Hàm fermin không có tác dụng thay đổi giá trị i.

---

a public member of a class can be accessed from anywhere; within the package, outside the package, within a subclass, as well as within a non-subclass.

A. T

B. F

A

một thành viên công khai của một lớp có thể được truy cập từ bất kỳ đâu; trong gói, bên ngoài gói, trong một lớp con, cũng như trong một lớp không phải là lớp con.

---

when you compile a program written in the Java programming language, the compiler converts the human-readable source file info platform-independent code that a Java Virtual Machine can understand. What is this platform-independent code called?

A. binary code

B. machine code

C. **bytecode**

D. cpu instruction

C

---

given:

11. public static void foo(String str) {

12. try {

13. float x = Float.parseFloat(str);

14. } catch (NumberFormatException e) {

15. x = 0;

16. } finally {

17. System.out.println(x);

18. }

19. }

20. public static void main(String[] args) {

21. foo("invalid");

22. }

what is the result?

A. Compilation fails at **line 15** with message about **variable x not found**

B. 0.0

C. A ParseException is thrown by the too method at runtime

D. A NumberFormatException is thrown by the foo method at runtime

A

---

which of the following statements is INCORRECT?

A, If a class has any abstract methods it must be declared abstract itself

B. When applied to a class, the final modifier means it cannot be sub-classed

C. None of others

D. All methods in an abstract class must be declared as abstract

D

All methods in an abstract class must be declared as abstract

---

which of the statements below is true?

A. To change the current working directory, call the cd() method of the File class

B. To change the current working directory, call the changeWorkingDirectory() method of the File class

C. To change the current working directory, call the setWorkingDirectory() method of the File class

D. To check whether the file denoted by the abstract pathname is a directory or not, call the **isDirectory()** method of the File class

D

---

Consider the following code. Which line will not compile?

1. Object ob = new Object();

2. String[] stringarr = new String[50];

3. Float floater = new Float(3.14f);

4. ob = stringarr;

5. ob = stringarr[5];

6. floater = ob; //float không thể gán bằng stringarr

7. ob = floater;

A. Line 6

B. Line 5

C. Line 4

D. Line 7

A

---

Given a string constructed by calling s = new String("xyzzy"), which of the calls modifies the string?

A. s.substring(3);

B. s.trim();

C. s.replace('z', 'a');

D. s.concat(s);

E. None of the others

F. s.append("aaa");

E

=> tất cả đều không thay đổi s

---

given the following declarations

String s1=new String("Hello")

String s2=new String("there");

String s3=new String();

Which of the following is legal operation?

A. s3=s1-s2;

B. s3=s1 && s2

C. s3=s1 && s2

D. **s3= s1 + s2**

D

---

which of the statement below is true?

A. UTF characters are all 16 bits

B. UTF characters are all 24 bits

C. Bytecode characters are all 16 bits

D. **Unicode** characters are all 16 bits

E. UTF characters are all 8 bits

D

---

The pack provides some of the **most useful** Java class that are frequently needed in all types of applications

A. java.lang

B. java.util

C. java.pack

D. java.io

A

---

is that declaration is correct?

MyList<Point> list = new MyList()<Point>

A. F

B. T

A

Thiếu (); ở cuối

---

which of the following statements is INCORRECT?

A. All of the methods in an interface are **implicitly abstract**

B. A method in an interface can access class level varriables

C. All of the variables in an interface are implicitly static

D. All of the variables in an interface are impicitly final

A

Tất cả các phương thức trong một interface là hoàn toàn trừu tượng

---

what will happen when you attempt to compile and run the following code

class Base{

private void amethod(int iBase){

System.out.println("Base.amethod");

}

}

class Over extends Base{

public static void main(String argv[]){

Over o = new Over();

int Base=0'

o.amethod(iBase);

}

public void amethod(int iOver){

System.out.println("Over.amethod");

}

}

A. output of "Over.amethod"

B. Runtime error complaining that Base.amethod is private

C. Output of "Base.amethod"

D. Compile time error complaining that Base.amethod is private

A

Chỉ chạy được hàm public.

---

The process of bringing an object into **existence** is called construction

A. T

B. F

A

---

All objects belonging to the same class have the same characteristics and possible actions

A. T

B. F

B

Tất cả các đối tượng thuộc cùng một lớp có cùng đặc điểm và các hành động có thể xảy ra

=> sai

---

2 answer

which of the following can you perform using the File class?

A. Return the name of the parent directory

B. Delete a file

C. **Find of a file constains text or binary information**

D. **Change the current directory**

cd

---

given the following code, what will be the outcome?

10. public class Funcs extends java.lang.Math {

11. public int add(int x, int y) {

12. return x + y;

13. }

14. public int sub(int x, int y) {

15. return x - y;

16. }

17. public static void main(String [] a) {

18. Funcs f = new Funcs();

19. System.out.println("" + f.add(1, 2) + " " + f.sub(3, 4));

20. }

21. }

A. "3-1" is printed out to the console

B. The code compiles but does not output anything

C. "3 1" is printed out to the console

D. **The line 10 causes compile-time error**

d

---

Suppose the declared type of x is a class, and the declared type of y is an interface. When is the assignment x = y; legal?

A. When the tyoe if x is an array

B. Always

C. **When** the type of **x is Object**

D. Never

C

Giả sử kiểu được khai báo của x là một lớp và kiểu được khai báo của y là một giao diện. Khi nào thì giao x = y; hợp pháp?

---

Which of the following signtures is valid for the main() method entry point of an application?

A. public static void main()

B. **public static void main(String[] args)**

C. public void main(String [] arg)

D. Public static int main(String [] arg)

B

---

All of the **numeric** wrapper classes in the java.lang package are subclasses of the abstract class .....

A. java.lang.Integer

B. java.lang.Wrapper

C. java.lang.Object

D. java**.lang.Number**

D

---

public class Test{

public static void main(String[] args){

String s1 = "xyz";

String s2 = new String("xyz");

if (s1 == s2) System.out.println("Line 4");

if (s1.equals(s2)) System.out.println("Line 6");

}

}

What is the output?

a. Line 4

Line 6

b. Line 4

c. Line 6

d. No output, compile error

e. No output

**Line 6**

---

public class Test{

public static void main(String[] args){

String s1 = "xyz";

String s2 = "xyz";

if (s1 == s2)

System.out.println("Line 4");

if (s1.equals(s2))

System.out.println("Line 6");

}

}

what is the output?

A. **Line 4**

**Line 6**

B. No output, compile error

C. Line 4

D. Line 6

E. No output

**A**

Biến tham trị bao gồm các kiểu nguyên thủy của JAVA như: int, long, double…

Biến kiểu tham chiếu bao gồm: String, array, kiểu đối tượng…

Khi sử dụng biến kiểu tham trị, JAVA chỉ cho phép bạn sử dụng toán tử so sánh “==”.

Khi sử dụng biến kiểu tham chiếu, JAVA cho phép sử dụng cả toán tử “==” và equals().

Khi sử dụng toán tử “==”, bộ xử lý của JAVA sẽ so sánh xem 2 biến tham chiếu này có trỏ đến cùng một đối tượng hay không, còn nếu bạn sử dụng phương thức equals(), bộ xử lý JAVA sẽ so sánh giá trị của 2 biến tham chiếu đó.

Ở trên đều là kiểu String nên cả 2 dòng đều in.

VD:

* Khi bạn thực hiện **Integer a = 3** thì lúc này bộ nhớ sẽ kiểm tra xem có địa chỉ nào đã có giá trị là **3** chưa, nếu chưa thì bộ nhớ sẽ lưu giá trị **3** vào một địa chỉ mới của bộ nhớ và trỏ **Integer a** đến địa chỉ chứa giá trị **3**, khi bạn tiếp tục thực hiện **Integer b = 3** thì lúc này bộ nhớ lại tiếp tục trỏ **Integer b** đến địa chỉ chứa giá trị **3** lúc nãy nên khi sử dụng toán tử "==" thì nhận được giá trị **true**.
* Khi bạn thực hiện **Integer a = new Integer(3)** và **Integer b = new Integer(3)** thì bộ nhớ sẽ lưu 2 giá trị **3** vào 2 địa chỉ khác nhau trên bộ nhớ và trỏ **Integer a** vào địa chỉ chứa giá trị **3** thứ nhất và **Integer b** đến địa chỉ chứa giá trị **3** thứ hai, nên khi sử dụng toán tử "==" sẽ nhận được kết quả false vì a và b được trỏ đến 2 địa chỉ khác nhau hoặc có thể hiểu là 2 đối tượng khác nhau.

Mặt khác toán tử equals chỉ quan tâm đến GIÁ TRỊ có bằng nhau không => vẫn dùng được với new.

---

2 answers

if(**check4Biz**(storeNum) != null) {}

Referring to the above, what data type could be returned by method check4Biz()?

A. **Boolean**

B. **String**

C. char

D. int

E. Byte

A B

---

which of the following statements can be used to call a constructor of the **super** class from its sub-class?

A. makeSuper();

B. super();

C. call();

D. invokeSuper();

B

**super();**

---

given

11. public abstract class Shape {

12. int x;

13. int y;

14. public abstract void draw();

15. public void setAnchor(int x, int y) {

16. this.x = x;

17. this.y = y;

18. }

19. }

and a class Circle that extends and fully inplements the Shape class.

Which is correct?

A. Circle c = new Circle(); c.Shape.setAnchor(10,10); c.Shape.draw();

B. Circle c = new Circle(); c.setAnchor(10,10); c.draw();

C. **Shape** s = new **Circle**(); s.setAnchor(10,10); s.draw();

D. Shape s = new Shape(); s.setAnchor(10,10); s.draw();

C

---

what will be the result of attemting to compile and run the following code?

abstract class MineBase {

abstract void amethod();

static int i;

}

public class Mine extends MineBase {

public static void main(String argv[]){

int[] ar = new int[5];

for(int i=0; i> ar.length; i++)

System.out.println(ar[i]);

}

}

A. IndexOutOfBoundes Error

B. **Error: Mine must be** **declared abstract**

C. Error: ar is used before it is initialized

D. a sequence of 5 0's will be printed

B

Trong class Mine chưa khai báo abstract method (amethod)

---

what will happen when you attempt to compile and run the following code

public class Hope{

public static void main(String argv[]){

Hope h = new Hope();

}

protected Hope(){

for(int i = 0; i < 10; i++){

System.out.println(i);

}

}

}

A. run time error: Constructors cannot be declared protected

B. Compilation error: Constructors cannot be declared protected

C. Compilation and running with output 0 to 9

D. Compilation and running with output 0 to 10

C

**Compilation and running with output 0 to 9**

---

Consider this class:

1. public class Test1 {

2. public float aMethod(float a, float b) {

3. }

4.

5. }

which of the following methods would be illegal if added (individually) at line 4?

A. public int aMethod(int a, int b) { }

B. public float aMethod(**float x, float y**) { }

C. private float aMethod(int a, int b, int c) { }

D. public float aMethod(float a, float b, int c) throws Exception { }

B

Hai phương thức trong cùng 1 class không thể cùng signature (overloading)

---

if you wanted to find out where the position of the letter v (ire return 2) int the string s containing "Java", which of the following could you use?

A. charAt(2);

B. **s.indexOf('v');**

C. indexOf(s,'v');

D. mid(2, s);

B

nếu bạn muốn tìm vị trí của ký tự v (ire return 2) trong chuỗi s chứa "Java", bạn có thể sử dụng cách nào sau đây?

---

select the correct statement:

A. An object reference cannot be cast to an interface reference

B. An object reference can cast to an interface reference when the object implements all methods of the referenced interface

C. An object reference can cast to an interface reference when the object implements the reference interface

D. An object reference can always be cast to an interface reference

B

Một tham chiếu đối tượng có thể truyền đến một tham chiếu giao diện khi đối tượng triển khai tất cả các phương thức của giao diện được tham chiếu

---

what will happen when you attempt to compile and run the following code?

class Base{

Base(){

System.out.println("Base");

}

}

public class Checket extends Base{

public class void main(String argv[]){

Checket c = new Checket();

super();

}

**Checket()**{

System.out.println("Checket");

}

}

A. Compile time error

B. Base followed by Checket

C. **runtime error**

D. Checket followed by Base

C

---

which of the following statement(s) is(are) true?

1)An abstract class cannot have any final methods

2)A final class may not have any abstract methods

A. Both statement 1 and 2

B. Only statement 1

C. Only statement 2

D. None of them

C

1) Một lớp trừu tượng không thể có bất kỳ phương thức final nào

**2) Một lớp final có thể không có bất kỳ phương thức trừu tượng nào**

---

The process of identifying **common features** of objects and methods is

A. Abstraction

B. Inheritance

C. Overloading

D. Polymorphism

E. Overriding

F. Encapsulation

A

---

which of the following code will compile without error?

A. package MyPackage;

class MyClass{}

import java.util.\*;

B. /\*This is an commnet \*/

package MyPackage;

import java.util.\*;

class MyClass{}

C. import java.util.\*;

package MyPackage;

class MyClass{}

B

Thứ tự package, import, class

---

which of the following collections supports accessing elements through **keys and values**?

A. HashMap

B. HashTree

C. HashList

D. **HashSet**

D

|  |  |  |  |
| --- | --- | --- | --- |
|  | List<E> | Set<E> | Map<K,V> |
| class | ArrayList, Vector | HashSet, TreeSet | HashMap, TreeMap |
| Đặc điểm | - index  - unsorted  - can duplicate | - no index  - can not duplicate | <Set, List>  - no index  - can not duplicate |
| Khác biệt | Vector (Threadsafe)  ArrayList(not) | HashSet: unsorted.  TreeSet: sorted. | HashMap: unsorted with key  TreeMap: sorted with key |
| Yêu cầu | Không có yêu cầu đặc biệt | HashSet, không yêu cầu đặc biệt, trong một số trường hợp có thể yêu cầu method equals.  Đối với TreeSet:  - Nếu TreeSet lưu trữ một loại đối tượng thì 1 trong 2 điều kiện sau phải thỏa mãn:  + Đối tượng mà ta cần lưu trữ phải được định thứ tự giữa các phần tử của chúng như thế nào, bằng cách class khởi tạo ra chúng phải được implement từ interface Comparable bắt buộc override lên method compareTo(o);  + Trong TreeSet phải định nghĩa thứ tự giữa các phần tử như thế nào bằng cách khi khởi tạo TreeSet phải truyền cho nó một đối tượng thuộc một class mà class đó được implement từ interface Comparator và bắt buộc override lên method compare(o1, o2);  ~ Nếu TreeSet lưu trữ hơn 1 loại đối tượng thì phải dùng cách 2 ở trên. | Mang đặc điểm của set tương ứng. |

---

10. interface Foo {

11. int bar();

12. }

13.

14. public class Beta {

15.

16. class A implements Foo {

17. public int Bar() { return 1; }

18. }

19.

20. public int fubar( Foo foo) { return foo bar(); }

21.

22. public void testFoo() {

23.

24. class A implements Foo {

25. public int bar() { return 2; }

26. }

27.

28. System.out.println( fubar( new A()));

29. }

30.

31. public static void main( String[] argv) {

32. new Beta().testFoo();

33. }

34. }

which statement is true?

A. If lines 24, 25 and 26 were removed, compilation would fail

B. The code compiles and the output is 1

C. The code compiles and the output is 2

D. **If lines 16, 17 and 18** were removed, compilation would fail

D

---

which of the following is true?

A. A class inherits constructors from its superclasses

B. A class inherits constructors from its direct superclass only

C. A class inherits constructors from its supperclasses when the program points out that

D. A class does not inherit constructors from any of ots superclasses

C

Một lớp kế thừa các hàm tạo từ các lớp cha của nó khi chương trình chỉ ra rằng

---

**How many bytes** does the following code write to file dest?

1. try {

2. FileOutputStream fos = newFileOutputStream("dest");

3. DataOutputStream dos = new DataOutputStream(fos);

4. dos.writelnt(3);

5. dos.writeFloat(0.0001f);

6. dos.close();

7. fos.close();

8. }

9. catch (IOException e) { }

**A. 8**

B. 2

C. 16

D. The number of bytes depends on the underlying system

E. 12

A

---

5. How many bytes does the following code write to file dest?

1. try {

2. FileOutputStream fos = newFileOutputStream("dest");

3. DataOutputStream dos = new DataOutputStream(fos);

4. dos.writeInt(3);

5. dos.writeDouble(0.0001); //không có f

6. dos.close();

7. fos.close();

8. }

9. catch (IOException e) { }

A. 2

B. 8

C. 12

D. 16

E. The number of bytes depends on the underlying system.

12

---

2 answers

which of the following class implement java.util.**List**?

A. java.util.HashMap

B. java.util.**LinkedList**

C. java.util.TreeSet

D. java.util**.ArrayList**

BD

---

Given:

11. public static void main(String[] args) {

12. try {

13. args=null;

14. args[0] = "test";

15. System.out.println(args[0]);

16. }catch (Exception ex) {

17. System.out.println("Exception");

18. }catch (NullPointerException npe) {

19. System.out.println("NullPointerException");

20. }

21. }

What is the result? (Choose one.)

a. Test

b. Exception

c. Compilation fails.

d. NullPointerException

C

Không thể compile từ trong try.

---

if all three top-level elements occur in a source file, they must appear in which orfer?

A. Imports, package declarations, classes/interfaces/enums

B. Classes/interfaces/enums, Imports, package declarations

C. Imports must come first; order for package declarations and Class/interface/enum definitions is not significant

D. **Package declarations, imports, Class/interface/enum definitions**

E. Package declaration must come first, order fir imports and class/interfaces/enum definitions is not significant

D

---

which of the following methods of the java.io.File can be used to check whether a file exists or not?

A. isExists()

B. **exists()**

C. canExists

D. doExists()

B

---

An object is an instance of a class

T

---

what does the following code do?

Integer i = null;

if (i != null & i.intValue() == 5)

System.out.println("**Value is 5"**);

A. Prints nothing

B. Compile error

C. **Throws an exception**

D. Prints "Value is 5"

C

---

which of the following **should** always be caught?

A. Errors other than asserition errors

B. Asserition errors

C. **Checked exceptions**

D. Runtime exceptions

C

---

what is the value of k after the following code fragment?

int k = 0;

int n = 12

while (k < n)

{ k = k + 1;

}

A. 11

B. **12**

C. 0

D. unknown

B

---

Which of the following methods of the java.io.File can be used to check whether a file can be read or not?

A. checkRead()

B. **canRead()**

C. read()

D. isRead()

B

---

given:

public class Bar {

public static void main(String [] args) {

int x = 5;

boolean b1 = true;

boolean b2 = false;

if((x==4) & !b2)

System.out.print("1 "); // điều kiện sai nên không in 1

System.out.print("2 "); // dòng này luôn được in ra

if((b2 = true) & b1)

System.out.print("3"); //điều kiện đúng nên in 3

}

}

What is the result?

A. **2 3**

B. 2

C. 3

D. 1 2 3

A

Nếu if không có ngoặc thì chỉ thực thi 1 dòng lệnh kế tiếp.

---

what will be printed out if toy attempt to compile and run the following code?

int i = 9;

switch(i) {

default:

System.out.println("default");

case 0:

System.out.println("zero");

**break;**

case 1:

System.out.println("one");

case 2:

System.out.println("two");

}

A. default, zero

B. error default clause not defined

C. default

D. no output displayed

A

Do case 1 và case 2 đều không thỏa mãn i = 9 nên lệnh default sẽ được thực thi, kết thúc lệnh default không có break nên lệnh case 0 bị tràn, kết thúc lệnh case 0 là break, do đó in cả **default** và **zero**.

Ví dụ nếu đặt break ở cuối case 1 thì sẽ in default, zero, one.

---

When is x & y an int?

A, when neither x nor y is a float or a double

B. Always

C. Whenever x and y are byte, shorts, chars, ints or longs

D. **Whenever x and y are bytes, shorts, chars or ints**

D

---

2 answers

Which two code fragments **correctly** create and initialize a static array of int elements?

A. static final int[] a = new int[2] { 100,200 };

B. static final int[] a; static void init() {a=new int[3];a[0]=100; a[1]=200:}

C. **static final int[] a = { 100,200 };**

D. **static final int[] a; static { a=new int[2]; a[0]=100; a[1]=200 }**

cd

---

given:

1. public class TestString3 {

2. public static void main(String[] args) {

3. //insert code here

5. System.out.println(s);

6. }

7. }

which code fragment, inserted at line, generate the output 424789?

A. StringBuffer s = new StringBuffer("123456789"); s.substring(3,6).delete(1,3),insert(1, "24");

B. StringBuider s = new StringBuider("123456789"); s.substring(3,6).delete(1,2),insert(1, "24");

C. StringBuffer s = new StringBuffer("123456789"); s.delete(0,3).replace(1,3,"24");

D. String s ="123456789"; s = (s-"123").replace(1,3,"24) - "89"

C

s.delete(0,3): s = 456789

replace(1,3,"24"): s = 424789 //replace "24" với các kí tự có index từ 1 đến 3

---

you want to find out the value of the last element of an array. You write the following code. What will happen when you compile and run it?

public class MyAr{

public static void main(String argv[]){

int[] i = new int[5];

System.out.println(i[5]);

}

}

A. The string "null" will be output

B. The value 0 will be output

C. **An error at run time**

D. An error at compile time

C

Không tồn tại i[5] (runtime, compile xong đang chạy thì lỗi).

---

what interface can be implemented in order to create a class that can be serialized?

A. No interfaces need to be implemented. All class can be serialized

B. Have the class declare that it implements java.io.Externalizable, which defines two methods: readObject and writeObject

C. Have the class declare that it implements jave.io.Serializable, which defines two methods: readObject and writeObject

D. **Have the class declare that it implements java.io.Serializable. There are no methods in the interface**

D

java.io.Serializable: no methods

java.io.Externalizable, two methods: readObject and writeObject

---

what is the output of the following program?

class test

{

public void main(String[] args)

{

int i; //chưa khởi tạo giá trị

do

{

i++;

}while(i<0);

System.out.println(i);

}

}

A. Compile OK but display nothing

B. 0

C. Cannot compile and display error **"variable i might not have been initialized"**

D. 1

C

---

Which of the following statements is true?

A. Under no circumstances can a class be defined with the private modifier

**B. A inner class may under some circumstances be defined with the protected modifier**

**C. An interface cannot be instantiated**

D. Adding more classes via import statements will cause a performance overhead, only import classes you actually use

B C

B. Trong một số trường hợp, một lớp bên trong có thể được định nghĩa bằng private modifier

C. **Không thể** khởi tạo interface

---

The \_\_\_\_\_ method is used to **replace a character in a StringBuffer**, with another at a specified position

A. replaceCharAt()

B. setStringAt()

C. replace()

D. setCharAt

D

setCharAt()

---

1. public class Test {

2. public String foo(int x, int y) {

3. return "AA";

4. }

5.

6. public String foo(int... vals) {

7. return "BB";

8. }

9. }

given:

25. Test a=new Test();

26. System.out.println(a.foo(**4, 5**));

what is the result?

A. Line 26 prints "BB" to System.out

B. Compilation of class A will fail due to an error in line 6

C. An exception is thrown at line 26 at runtime

D. **Line 26 prints "AA" to System.out**

D

Số lượng tham số là 2 nên in "AA" trước.

Nếu số lượng tham số khác 2 thì sẽ in "BB".

Ví dụ System.out.println(a.foo(**4, 5, 6, 7**));

---

which of the File class description statements below is WRONG?

A. File class is a class which could be found in java.io package

B. File class helps accessing file/dictionary information only

C. File class has only one constructor with a file path parameter

D. File class doesn't have any method to access data in a file

C

File class chỉ có một hàm tạo với tham số đường dẫn tệp => SAI

---

which of the following is illegal statement?

A. int i=1/3;

B. double d=999d;

C. float f=1.01;

D. float f=1/3;

C

Đúng: float f = 1.01f hoặc thay f thành double.

---

2 answers

which two of the following interfaces are at the top of the hierarchies in the **Java Collections Framwork?**

A. Queue

B. SortedMap

C. Collection

D. Map

E. **List**

F. **Set**

E F

hai giao diện nào sau đây nằm ở đầu phân cấp trong Java Collections Framwork?

---

2 answers

A varriable declared with the **default** modifier can be accessed by \_\_\_\_\_\_

A. the class containing that variable only

B. **same package different classes**

C. different packages and different classes

D. **same package sub classes**

E. all classes

F. different package and sub classes

B D

---

your programming problem is to create a list of **unique values of part ID numbers** in a large **collection** of data respresention **orders**. Furthermore, it would be nice if the list was **in sorted order.**

You have decided to use one of the collection classes in the java.util package to construct this list. Which of the following interfaces should the ideal class implement?

A. Map

B. List

C. SortedMap

D. Set

E. SortedSet

E

---

2 answers

given:

10. abstract public class Employee {

11. **protected** abstract double getSalesAmount();

12. public double getCommision() {

13. return getSalesAmount() \* 0.15;

14. }

15. }

16. class Sales extends Employee {

17. // insert method here

18. }

Which two methods, inserted independently at line 17, correctly complete the Sales class?

A. **public double getSalesAmount() { return 1230.45; }**

B. **protected double getSalesAmount() { return 1230.45; }**

C. double getSalesAmount() { return 1230.45; }

D. private double getSalesAmount() { return 1230.45; }

A B

Phương thức abstract là protected => A và B

---

what will happen when you attempt to compile and run the following code?

class Base{

public void Base(){

System.out.println("Base");

}

}

public class In extends Base{

public static void main(String argv[]){

In i=new In();

}

}

A. **Compilation and not output at runtime**

B. Compile time error Base is a keyword

C. Output of Base

D. Runtime error Base has no valid constructor

A

Hàm Base không được gọi => chương trình vẫn chạy và không có output.

---

The java.util.Vector class provides storage for object references in the order of addition and automatically expands as needed. Which of the following classes is **closest in function to the Vector class**?

Select one:

a. java.util.LinkedList

b. **java.util.ArrayList**

c. java.util.Hashtable

d. java.util.List

B

---

Given:

10. interface Foo { int bar(); }

11. public class Sprite {

12. public int fubar( Foo foo) { return foo.bar(); }

13. public void testFoo() {

14. fubar(

15. // insert code here

16. );

17. }

18. }

Which code, inserted at line 15, allows the class Sprite to compile?

a. Foo { public int bar() { return 1; } }

b. new Foo { public int bar() { return 1; } }

c. **new Foo() { public int bar(){return 1; } }**

d. new class Foo { public int bar() { return 1; } }

C

---

(1) A value varriable contains data's value

(2) A reference variable contains the address of data.

The statement(1) is \_\_\_\_\_. and the statement(2) is\_\_\_\_\_\_\_\_\_\_\_

A. T, T

B. F, T

C. T, F

D. F, F

**A. T T**

(1) Một biến giá trị chứa giá trị dữ liệu

(2) Một biến tham chiếu chứa địa chỉ của dữ liệu.

---

What is -8 % 5

A. 3

B. -3

C. 2

D. -2

B

Trong java: a%b = (dấu của a) \* |a|%|b|

VD: -8 % 5 = -|8|%|5| = -3

Tổng quát:

a == (a / b \* b) + a % b

=> a%b = a – (a/b \*b)

VD: -8%5 = 5 – (-8/5 \* -8) = 5 – (-1\*-8) = -3

Lưu ý: % khác với mod.

VD: -12% 10 = -2 trong khi -12 mod 10 = 8

---

what will happen when you attempt to compile and run the following code?

public class **Scope**{

private int i;

public static void main(String argv[]){

Scope s = new Scope();

s.amethod();

}//End of main

public static void amethod(){

System.out.println(i);

}//end of amethod

}//End of class

A. Nothing will be printed out

B. A compile time error complaining of the scope of the varriable i

C. **A compile time error**

D. A value of 0 will be printed out

C

Method amethod là method static, do đó biến i cũng phải static.

Không thể tạo tham chiếu static cho một biến non-static.

B sai vì không liên quan đến phạm vi của biến.

---

State true or false:

If class Y extends class X, the two classes are in different package, and class X has a protected method called fun(), then any instance of Y may call the fun() method of any other instance of Y

**F**

If class **Y extends class X**, the two classes are in **different package**, and class **X has a protected method called fun()**, then **any instance of Y may call the fun() method of any other instance of Y** => SAI

---

2 answers

which of the following may appear on the right-hand side of an instanceof operator?

A. A reference

**B. An interface**

C. A varriable of primitive type

**D. A class**

E. The name of a primitive type

B D

---

What is 7 % -4?

3

---

which statement is true about this code?

1. class A

2. {

3. private static int x = 100;

4.

5. public static void main(String args[])

6. {

7. A t = new A();

8. t.x++; //101

9. A h = new A();

10. h.x++; //102

11. h = new A();

12. h.x++; //103

13. A.x++; //104

14. System.out.println("x= " + x);

15. }

16. }

A. line 13 will not compile because it is a static reference to a private varriable

B. The program compiles and the output is x = 102

C. The program compiles and the output is x = 103

D. The program compiles and the output is **x = 104**

D

---

Which of the following are correct ways of declaring an integer array named myVar

A. myVarint[];

B. **int myVar[];**

C. **int[ ] myVar;**

D. intmyVar{};

E. int [myVar];

F. myVar[] int;

B C

---

what would happen when the following code is compiled and executed?

public class Compare {

public static void main(String args[]) {

int x = 10, y;

if(x < 10)

y = 1;

if(x >= 10)

y = 2;

System.out.println("y is " + y);

}

}

A. The program compiles are prints y is 2 when executed

B. Depends on the particular implementation of the Java Virtual Machine

C. The program compiles and prints y is 1 when executed

D. The prgram throws a runtime exception

E. The program compiles and prints y is 0 when executed

F. The program **does not compile** *complaining about* ***y not being initizalized***

F

Tuy điều kiện x > 10 và x <= 10 theo logic là quét hết "các trường hợp" nhưng trình biên dịch sẽ không thể hiểu được.

Do đó biến y vẫn chưa được khởi tạo do CHỈ sử dụng điều kiện if là không đủ để trình biên dịch hiểu rằng đã quét hết "các trường hợp".

Đúng:

if(x < 10)

y = 1;

else //hoặc không cần dòng này

y = 2;

---

\_\_\_\_\_\_\_ class **reads bytes** from a file

A. **FileInputStream**

B. FileOutputStream

C. InputStream

D. FileInput

A

---

int values [] = { 1,2,3,4,5,6,7,8};

for(int i=0;i<X;++i)

System.out.println(values[i]);

Referring to the above, what (smallest) value for X will print all members of array "values"?

A. **8**

B. 9

C. None, since there is a syntax error in the array declaration

D. 7

E. 1

A

---

what does the following line of code mean?

double table[];

A. table is a varriable that refers to two numbers

B. table is a varriable to refers to a real number

C. It is not legal Java code

D. table is a varriable that **refers to an array**

D

---

which of the following is an example of a Java **bool-expression**?

A. x = 6

B. cause **==** bYes

C. 70

D. 1=2

B

---

which of the following methods of the Collections class can be used to find the largest value in a Vector?

A. Collections.maxElement()

B. We don't need any method because elements in Vector are automatically stored. Therefore, the first element contains the maximum value

C. **Collections.max()**

D. Collections.maxValue()

C

---

given the following code, what test would you need to put in place of the comment line?

//place test here

to result in an output of the string Equal

public class EqTest{

public static void main(String argv[]){

EqTest e=new EqTest();

}

EqTest(){

String s="Java";

String s2="java";

//place test here {

System.out.println("Equal");

}else

{

System.out.println("Not equal");

}

}

}

A. if(s==s2)

B. if(s.noCaseMath(s2))

C. if(s.**equalsIgnoreCase**(s2))

D. if(s.quals(s2))

C

---

a java source code will be compiled to\_\_

A. Machine code

B. **Java bytecode**

C. Operation system code

D. Assembly code

B

---

what would be the output from this code fragment?

1. int x = 0, y = 4, z = 5;

2. if (x > 2) {

3. if (y < 5) {

4. System.out.println("message one");

5. }

6. else {

7. System.out.println("message two");

8. }

9. }

10. else if (z > 5) {

11. System.out.println("message three");

12. }

13. else {

14. System.out.println("message four");

15. }

A. **message four**

B. message two

C. message one

D. message three

A

---

Whenever a method does not want to handle exception using the try block, the \_\_\_\_\_ is used?

A. **throws**

B. throwable

C. throw

D. mothrows

A

---

what is the output when the following program is run?

public class Main

{

public static void main(String args[])

{

Decrementer t = new Decrementer();

double x = 7.2;

System.out.print(x + " ");

t.decre(x);

System.out,print(x + " ");

t.decre(x);

System.out.println(x);

}

}

class Decrementer

{

public void decre(double x)

{

x = x - 2; // không ảnh hưởng đến biến x

}

}

A. 5.2 7.2 5.2

B. 7.2 5.2 3.2

C. **7.2 7.2 7.2**

D. 7.2 5.2 5.2

E. 7.2 7.2 5.2

C

TRONG hàm sẽ thực thi giảm x xuống còn 5.2 nhưng biến x thì không bị ảnh hưởng, vẫn bằng 7.2

---

2 answers

which of these statements about the value that appears in a switch statement are correct?

A. The value can be of type char

B. The value can be of type boolean

C. The value can be of type long

D. The value can be of type byte

A D

Các kiểu sử dụng được trong switch: char, byte, int, short, enum

---

Classes that are intended to be used outside the package within other programs must be declared\_\_\_\_\_\_\_

A. static

B. public

C. default

D. private

B

Các lớp được dự định sử dụng bên ngoài package trong các chương trình khác phải được khai báo public.

---

The following lists the complete contents of the file named Derived,java:

1. public class Base extends Object {

2. String objType;

3. public Base(){objType =

"I am a Base type" :

4. }

5. }

6.

7. public class Derived extends Base {

8. public Derived() { objType =

"I am a Derived type";

9. }

10. public static void main(String args[]){

11. Derived D = new Derived();

12. }

13. }

what will happen when this file is compiled?

A. The compiler will object to line 7

B. Two class files, Base.class and Derived.class, will be created

C. The compiler will object to line 1

B

Trong 1 file chỉ có duy nhất 1 class public trùng tên với file đó

---

After the following code fragment, what is the value in a?

String s;

int a;

s = "Foolish boy";

a = s.indexOf("fool");

A. random value

B. 4

C. 0

D. -1

D. **-1**

do không tìm thấy "fool" trong "Foolish boy"

---

What will happen when you attempt to compile and run the following code

class Base{

protected int i = 99;

}

public class Ab{

private int i=1;

public static void main(String argv[]){

Ab a = new Ab();

a.hallow();

}

abstract void hallow(){

System.out.println("Claines "+i);

}

}

A. Compilation and not output at runtime

B. Compilation and output of Claines 99

C. Compilation and output of Claines 1

D. **Compile time error**

D

Có phương thức abstract nhưng class Ab không khai báo abstract

---

Is the following interface a valid?

public interface TestInterface {

void aMethod(intaValue) {

System.out.println(""Hi Mom"");

}

}

a. True

b. False

A

Interface chỉ gồm các method **không có body**

---

2 answers

Which of the following statements are true?

A. There are No circumstances where an inner class may be defined as private

B. An inner class may be defined as static

C. An inner class may extend another class

D. A programmer may only provide one constructor for an anonymous class

B C

---

2 answers

Choose the valid identifiers from those listed here

A. $int

B. byte

C. BigOILongStringWithMeaninglessName

D. 1$2

E. finally

A C

Valid identifiers: chấp nhận kí tự chữ, số, dấu \_, dấu $.

Không được có space, không được có số ở đầu (D), không được trùng các từ khóa có sẵn (B, E).

---

Which of the following statements is true?

A. A final class may not contain non-final data fields

B. A final class may only contain final methods

C. **A final class cannot be extended**

D. A final class must be instantiated

C

---

2 answers

given the following

List<String> names = news ArrayList<String>();

which of the following are legal?

A. **Iterator<String> iter = names.iterator();**

B. while (String s:names)

C. **for (String s:names)**

A C

---

which of the following is legal import statement?

A. None of the other choices

B. **import java.util.Vector;**

C. static import java.util.Vector.\*;

D. import java.util.Vector static;

B

---

2 answers

which of the following methods can be legally inserted in place of the comment //Method Here?

class Base{

public void amethod(int i){ }

}

public class Scope extends Base{

public static void main(String argv[]){

}

//Method Here

}

A. **void amethod(long i) throws Exception {}**

B. public void amethod(int i) throws Exception {}

C. void amethod(int i) throws Exception {}

D. **void amethod(long i){}**

A D

---

what will happen when you attempt to compile and run this code?

class Base{

abstract public void myfunc();

public void another(){

System.out.println("Another method");

}

}

public class Abs extends Base{

public static void main(String argv[]){

Abs a = new Abs();

a.amethod();

}

public void myfunc(){

System.out.println("My func");

}

public void amethod(){

myfunc();

}

}

A. The code will compile and run, printing out the words **"My Func"**

B. The code will compile but complain at run time that the Base class has non abstract methods

C. The compiler will complain that the Base class is not declared as abstract

D. The compiler will complain that the method myfunc in the base class has no body

A

---

you want to loop through an array and stop when you come to the last element. Being a good java programmer and forgetting everything you ever knew about C/C++ you know that arrays contain information about their size. Which of the following can you use?

A. myarray.**length**;

B. myarray.length();

C. myarray.size();

D. myarray.size

A

---

2 answers

which of the following statements are true with package?

A. In Java, a package is a combination of class, interfaces and sun-packages

B. A package in Java can be created by including a package statement as the first statement in a Java program

C. The classes in a package must not be saved under a folder that bears the same name as the package

A C

A. Trong Java, một package là sự kết hợp của lớp, giao diện và sun-package

B. Một package trong Java có thể được tạo bằng cách bao gồm một câu lệnh package làm câu lệnh đầu tiên trong chương trình Java => SAI

C. Các lớp trong một package không được lưu trong một thư mục có cùng tên với package

---

3 answer

given:

11. **public** interface Status {

12. /\* insert code here \*/ int MY\_VALUE = 10;

12. }

which three are valid on line 12?

A. native

B. abstract

C. **static**

D. **final**

E. **public**

F. protected

C D E

---

consider the following application:

1. class Q6 {

2. public static void main(String args[]) {

3. Holder h = new Holder();

4. h.held = 100;

5. h.bump(h);

6. System.out.println(h.held);

7. }

8. }

9.

10. class Holder {

11. public int held;

12. public void bump(Holder theHolder) {

13. theHoler.held++;

14. }

15. }

16.}

what value is printed out at line 6?

A. 100

B. 1

C. 101

D. 0

C

---

If your method overrides one of its supperclass's methods, you can invoke the overridden method through the keyword parent

F

Nếu phương thức của bạn ghi đè một trong các phương thức của supperclass, bạn có thể gọi phương thức được ghi đè thông qua từ khóa cha.

---

select a correct statement:

A. void f(double d, int ... x, int y) {}

B. void f(int ... x, int ... y) {}

C. void f(int ... x, int y) {}

D. void f(int x, int ... y) {}

D

void f(int x, int ... y) {}

---

which of the following is INCORRECT?

A. char c = \u1234;

B. char c = 0x1234;

C. String x = "ABC" + 2;

D. char c = '\u1234';

A

char c = \u1234 => SAI

(\u: chuyển đổi unicode)

---

Suppose the current directory dose not contain a sub-directory named "dir"

What happens when you try to compile and run the following application?

10. import java.io.\*;

11. public class Main {

12. public static void main(String argv[]) {

13. try {

14. File d = new File("dir");

15. File f = new File(d,"fi.txt"); // sai

16. if(!f.exists()) {

17. f.createNewFile();

18. }

19. }catch (IOException e) {

20. e.printStackTrace ();

21. }

22. }

23. }

A. Line 17 is never excuted;

B. Line 15 creates a directory named "dir" and a file "fi.txt" within it

C. Line 17 creates a directory named "dir" and a file "fi.txt" within it

D. **Line 15 throws an exception at runtime**

E. Line 17 throws an exception at runtime

D

---

If you need a Set implementation that provides **value-ordered iteration**, which class should you use?

A. TreeSet

B. HashSet

C. LinkHashSet

A

**TreeSet**

Nếu bạn cần triển khai Set cung cấp phép lặp theo thứ tự giá trị, bạn nên sử dụng lớp nào?

---

2 answers

consider the following class definition:

1. public class Test extends Base {

2. public Test(int i) {

3. }

4. public Test(int j, int k) {

5. super(j, k);

6. }

7. }

which of the following forms of constructor must exist explicitly in the definition of the Base class? Assume Test and Base are in the same package

A. Base(int j) { }

B. **Base() { }**

C. **Base(int j, int k) { }**

D. Base(int j, int k, int l) { }

B C

---

when a negative long is cast to a byte, what is the possible value of the result?

A. The value may be negative, zero or positive

B. Positive

C. Negative

D. Zero

A

**The value may be negative, zero or positive**

---

which of the following signatures is valid for the main() method entry point of an application

A. public static void main()

B. public void main(String [] arg)

C. public static int main(String [] arg)

D. public static void main(String[] args)

D

**public static void main(String[] args)**

---

select the most correct statement

A. An interface may be declared as public, protected or private

B. An interface may be declared as public, protected, private or abstract

C. **An interface may be declared as public or abstract**

D. An interface may be declared as public or protected

C

---

which of the following statements is correct?

A. only primitives are converted automatically: to change the type of an object reference, you have to do a cast

B. Arithmetic promotion of object references requires explicit casting

C. **Both primitives and object references can be both converted and cast**

D. Only object references are converted automatically: to change the type of a primitive, you have to do a cast

C

---

Which of the following is true?

A. Both >> and >>> operators carry the sign bit when shifting right

B. The >>> operator carries the sign bit when shifting right. The >> zero-fills bits that have been shifted out

C. Both >> and >>> operators zero-fills bits that have been shifted out

D. The **>>** operator carries the sign bit when shifting right. The **>>** zero-fills bits that have been shifted out

D

---

how can you force garbage collection of an object?

A. Set all references to the object to new values (null, for example)

B. Call System.gc(). passing in a reference to the object to be garbage-collected

C. Call Runtime.gc()

D. **Garbage collection cannot be forced**

E. Call System.gc()

D

---

given the following code, what is the expected outcome?

public class Test {

public static void main(String [] a) {

int [] b = [1,2,3,4,5,6,7,8,9,0];

System.out.println("a[2]=" + a[2]);

}

}

A. **the code does not compile**

B. "a[2]=2" is printed out to the console

C. The code compiles but does not output anything

D. "a[2]=3" is printed out to the console

A

---

a generic class can have only one type parameter

F

một lớp chung chỉ có thể có một loại tham số => SAI

---

what will the following code print out?

public class Oct{

public static void main(String argv[]){

Oct o = new Oct();

o.amethod();

}

public void amethod(){

int oi=012;

System.out.println(oi);

}

}

A. 10.0

B. 12

C. 012

D. 10

B. **12**

---

you can prevent a class from being sub-classed by using the\_\_\_\_\_\_\_\_ keyword in the class's declaratio

A. private

B. static

C. final

D. protected

C

bạn có thể ngăn một lớp bị phân thành lớp con bằng cách sử dụng từ khóa\_\_\_\_\_\_\_\_ trong phần khai báo của lớp => FINAL

---

Class SomeException:

1. public class SomeException {

2. }

Class A:

1. public class A {

2. public void doSomething() { }

3. }

Class B:

1. public class B extends A {

2. public void doSomething() **throws SomeException { }** // sai

3. }

Which is true about the two classes? (Choose one.)

a. Compilation of both classes will fail.

b. Compilation of both classes will succeed.

c. Compilation of class A will fail. Compilation of class B will succeed.

d. Compilation of class B will fail. Compilation of class A will succeed.

D

---

what happens when you attempt to compile and run these two files the same directory?

//File P1.java

package MyPackage;

class P1{

void afancymethod(){

System.out.println("What a fancy method");

}

}

//File P2.java

public class P2 extends P1{

public static void main(String argv[]){

P2 p2 = new P2();

p2.afancymethod();

}

}

A. **P1 compiles cleanly but P2 has an error at compile time**

B. Both compile and P2 outputs "What a fancy method" when run

C. Both compile but P2 has an error at run time

D. Neither will compile

A

---

2 answers

Which of the following are true?

A. **Primitives are passed by value**

B. Primitives are passed by reference

C. **References are passed by reference**

D. References are passed by value

A D

---

Suppose class X contains the following method:

void doSomething(int a, float b) { ... } Which of the following methods may appear in class Y, which extends X? (Choose one.)

A. private void doSomething(int a, float b) throws java.io.IOException { ...}

B. public void doSomething(int a, float b) throws java.io.IOException { ...}

C. private void doSomething(int a, float b) { ... }

D. public void doSomething(int a, float b) { ... }

D

public void doSomething(int a, float b) { ... }

---

Statement 1: A protected member of a class can be accessed from any class in the same package and from a subclass that is outside the package. Statement 2: A member of a class that is declared private can not be accessed only within the class but nowhere outside the class. Choose the correct answer?

Select one:

a. Both Statement 1 and Statement 2 are FALSE

b. Both Statement 1 and Statement 2 are TRUE

c. Statement 1 is FALSE, Statement 2 is TRUE

d. Statement 1 is TRUE, Statement 2 is FALSE

B**. Both Statement 1 and Statement 2 are TRUE**

Mệnh đề 1: Một thành viên được bảo vệ của một lớp có thể được truy cập từ bất kỳ lớp nào trong cùng một gói và từ một lớp con nằm ngoài gói.

Mệnh đề 2: Một thành viên của một lớp được khai báo là private không thể chỉ được truy cập trong lớp mà không thể truy cập được bên ngoài lớp.

---

2 anwsers

given the following statments:

int [] a = {9,41,49,13,32};

int sum = 0;

which of the following are legal ways to calculate the sum of the elements of the array a?

A. for(int k : a) sum += a[k]; //a[k], sai

B. for(int i=0; i<a.length; i++) sum += a[i];

C. for(int k : a) sum +=k;

D.for(int i=0; i<=a.length; i++) sum += a[i]; //<=, sai

E. sum += a[int i : a]; //sai

B C

---

what is the output when you try to compile and run the following program?

public class Main{

public static void main(String argv[]){

String s = "Hi there";

int pos = s.indexOf(" ");

String r = s.substring(0,pos);

String s2 = new String(new char[] {'H','i'});

if(r.equals(s2))

System.out.println("EQUAL ");

else

System.out.println("NOT EQUAL ");

System.out.println();

}

}

A. There is a compile error in the program

B. EQUAL

C. NOT EQUAL

D. EQUAL NOT EQUAL

B. **EQUAL**.

---

what will happen if you try to compile and run the following code:

public class MyClass {

public static void main(String arguments[]) {

amethod(arguments);

}

*public void amethod*(String[] arguments) {

System.out.println(arguments);

System.out.println(arguments[1]);

}

}

A. error array must include parameter

B. error method main not correct

C. **error Can't make static reference to void amethod**

D. amethod must be declared with String

C

Phương thức amethod phải khai báo static

---

consider the following line of code:

int[] x = new int[25];

After execution, which statement is true?

A. x[25] is 0

B. x[0] is null

C. x[24] is undefined

D**. x[24] is 0**

D

---

when mutiple methods exist within the same class with different method signatures, this is known as what?

A. There is nothing one can say

B. Message passing

C. Overriding methods

D. A headache

E. Method overloading

E

khi nhiều phương thức tồn tại trong cùng một lớp nhưng signature khác nhau

---

Which of the following statements is false?

A. Values stored in TreeSet are automatically sorted

B. **Vector does not allow duplicate elements**

C. ArrayList can duplicate elements

D. TreeSet does not allow duplicate elements

B

---

2 answers

given the following code fragment:

switch(x) {

case 100:

System.out.println("One hundred";break;

case 200:

System.out.println("two hundred";break;

case 300:

System.out.println("three hundred";break;

}

choose all of the declarations of x that will not cause a compiler error (KHÔNG gây ra lỗi cho trình biên dịch)

A. byte x = 100 //byte max 256 => không chấp nhận case 300

B. long x = 400 //kiểu long không chấp nhận

C. short x = 200

D. int x = 300

C D

---

which of the following is legal loop construction

A. int j = 0;

for(int k = 0; j+k != 10; i++;k++) {

System.out.println("j=" + j + ". k=" +k);

}

B. int j=0;

do {

System.out.println("j=" + j++);

if(j==3) continue loop;

} while (j<10);

C. int i = 3;

while (i) {

System.out.println("i is " + i);

}

D. while (int i<7) {

i++;

System.out.println("i is " + i);

}

A

---

what does the following code fragment print out at lint 9?

1.FileOutputStream fos = new FileOutputStream("xx");

2.for (byte b=10; b<50; b++)

3. fos.write(b);

4.fos.close();

5.RandomAccessFile raf = new RandomAccessFile("xx","r");

6.raf.seek(10);

7.int i = raf.read();

8.raf.close()

9.System.out.println("i =" + i);

A. The output is i = 20

B. The output is i = 10

C. The output is i = 30

D. There is no output because the code throws an exception at line 1

E. There is no output because the code throws an exception at line 5

a

---

which of the following declarations is INCORRECT?

A. String s = "Hi!";

B. int[] ar = new int(3);

C. double d = 1.3E + 21;

D. double d = 1.3D;

b

---

Suppose salaries is an array containing **floats**. Which of the following are valid loop control statements for processing each element of salaries

A. for (int i::salaries)

B. for (float f::salaries)

C. **for (float f:salaries)**

D. for (int i:salaries)

C

---

2 answers

which of the following operators can perform promotion on their operands?

A. +

B. - -

C. ++

D. !

E. -

A E

+ -

---

3 answers

you want the program to print 3 to the output. What of the following values for x will do this

switch(x){

case(1);

System.out.println("1");

case(2);

case(3);

System.out.println("3");

}

A. any value

B. 3

C. 2

D. 1

E. 0

F. 4

B C D

3 2 1

---

which of the following lines will compile without warning or error

A. byte b = 257; //vượt quá 256

B. boolean b = null; //null chỉ được chấp nhận bởi reference types

C**. int i = 10; char c = '\0';**

D. char c = "a"; //char phải dùng ' '

E. float = 1.3; //1.3 kiểu double, đúng: 1.3f

C

---

which of the following may legally appear as the new type **(between the parentheses)** int a cast operation?

A. Arrays of classes

B. Arrays of interfaces

C. Classes

D. Interfaces

E. **All of the others**

---

12. Which of the following may legally appear as the new type (between the parentheses) in a cast operation?

A. Abstract classes

B. Final classes

C. Primitives

D. All of the above

**All of the above**

---

E

điều nào sau đây có thể xuất hiện một cách hợp pháp dưới dạng kiểu mới (giữa các dấu ngoặc đơn) trong một phép toán ép kiểu?

---

which line contains only legal statements?

A. String x = "Hello"; int y = 9; y += x; //khai báo 2 lần y

B. String x = "Hello"; int y = 9; x = x + y; //hợp lệ

C. String x = "Hello"; int y = 9; y = y + x; //tương tự A

D. String x = "Hello"; int y = 9; if(x == y) {} //không thể so sánh String và int

B

---

what is the output of the following code:

class Main {

public static void main(String[] args) {

short s = 259;

byte b = (byte) s;

System.out.println(b);

}

}

A. -3

B. 3

C. 259

D. -259

E. 256

B. 3

byte (8 bit) => từ -127 đến 128 (256 giá trị) => 259 – 256 = 3

---

what is the output of the following code

class Main {

static int k = 10;

static {k += 5:}

public static void main(String args[]){

}

static {k += 5;}

}

A. 15

B. 5

C. 10

D. 20

E. a compile-time error

D. **20**

---

2 answers

given the following class definition, which of the following methods could be legally place after the comment //here?

public class Rid{

public void amethod(int i, String s){}

//here

}

A. **public void amethod(String s, int i){}**

B. public void amethod(int i, String mystring){}

C. **public void Amethod(int i, String s){}**

D. public int amethod(int i, String s){}

A C

(overloading)

---

Assuming any exception handing has been set up, which of the following will create an instance of the RandomAccessFile class?

A. RandomAccessFile raf=new RandomAccess("myfile.txt");

B. RandomAccessFile raf=new RandomAccess(new File("myfile.txt"));

C. RandomAccessFile raf=new RandomAccess**("myfile.txt","rw");**

D. RandomAccessFile raf=new RandomAccess(new DataInputStream());

C

RandomAccessFile là một điểm bất thường trong kiến ​​trúc Java I / O. Nó đi xuống trực tiếp từ Đối tượng và không phải là một phần của kiến ​​trúc Luồng.

---

given the following class:

public class Xyz implements java.io.Serializable {

public int iAmPublic;

private int iAmPrivate;

static int iAmStatic;

transient int iAmTransient;

}

Assuming the class does perform custom serialization, which fields is/are **written** when an **instance of Xyz is serialized**?

A. **iAmPublic, iAmPrivate**

B. iAmPublic

C. iAmStatic, iAmTransient

D. iAmTransient

A

---

what will happen when you attempt to compile and run the following code

int Output=10;

boolean b1 = false;

if(b1==true)&&((Output+=10)==20)){ //tại thời điểm Output+=10 thì Output vẫn bằng 10

System.out.println("We are equal "+Output);

}else

{

System.out.println("Not equal! "+Output);

}

A. Compilation and output of "Not equal! 20"

B. Compilation and output of "We are equal 10"

C. Compilation and output of **"Not equal! 10"**

D. Compile error, attempting to peform binary comparison on logical data type

C

---

QN=125  
(122) Suppose class A has a method called doSomething(), with default access.  
Suppose class B extends A and overrides doSomething(). Which access modes may **not apply** to **B's version** of doSomething()? (Choose one)  
a. public  
b. private  
c. protected  
d. Default

B

(private < default < protected < public)

QN=126  
(117) Suppose class Supe, in package packagea, has a method called  
doSomething(). Suppose class Subby, in package packageb, overrides doSomething(). What access modes may **Subby's** version of the method have? (Choose two.)  
a. public  
b. protected  
c. Default  
d. private

A, B

---

QN=103  
(4924) public class Test{  
public static void main(String[] args){

Object ob1= new Object();

Object ob2= ob1; //object ob2 bằng ob1 => == và equals đều thỏa mãn  
if(ob1.equals(ob2)) System.out.println("ob1 equals ob2");

if(ob1==ob2) System.out.println("ob1==ob2");

System.out.println("Have a nice day!");  
}  
  
}  
  
What is the output?  
a. ob1 equals ob2  
ob1==ob2  
Have a nice day!  
b. ob1 equals ob2  
Have a nice day!  
c. ob1==ob2  
Have a nice day!  
d. None of the above

A

---

QN=102  
(77) public class Test{  
public static void main(String[] args){

Object ob1= new Object();

Object ob2= new Object();

//ob2 là một object mới khác hoàn toàn ob1 => không xảy ra 2 mệnh đề if ở dưới  
if(ob1.equals(ob2)) System.out.println("ob1 equals ob2");

if(ob1==ob2) System.out.println("ob1==ob2");

System.out.println("Have a nice day!");  
}  
  
}  
  
What is the output?  
a. ob1 equals ob2  
Have a nice day!  
b. ob1==ob2  
Have a nice day!  
c. Have a nice day!  
d. No output

C

The following code will point1.

if(new Boolean("true") == new Boolean("true"))

2. System.out.println("True");

3. else

4. System.out.println("False");

A. Compilation error

B. No compilation error , but runtime exception

C. Prints "True"

D. Prints "False"

E. Depends on the particular implementation of the Java Virtual Machine

D

Giữa 2 đối tượng mới new, không thể so sánh ==.

49.class Employee{

private String dob;

public Employee(String dob){

this.dob = dob;

}

public int getAge(){

String[] t = dob.split("/");

int m = Interger.parseInt(t[1]);

int y = Interger.parseInt(t[2]);

if(m<6)

return (2021 -y);

else

return (2021 -y + 1);

}

}

public class Main{

public static void main(String[] args){

System.out.println(new Employee("11/06/2000").getAge());

}

}

22

---

6. What will be the output of following code?abstract class A{public A()

{System.out.println("Father");}

abstract public void out();}

class B extend A{public B()

{System.out.println("Son");}

@Overidepublic void out()

{System.out.println("Father loves son!");}

}

public class Main1

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

{A.b= new B();

b.out();

}

}

Father

Son

Father loves son!

---

11.

When an int value is add to a double value , what is the type of the result?

A. double

B. int

C. float

D. You can't add a short to a float

A

---

10.

Which of the following signatures is valid for the main() method entry point of an application?

A. public static void main()

B. public static void main(String [] args)

C. public void main (String [] arg)

D. public static int main(String [] arg)

B

---

8. You want to loop through an array and stop when you come to the last element. Being a good java programmer and forgetting everything you ever knew about C/C++ you khow that arrays contain information about their size . Which of the following can you use?

A. myarray.length();

B. myarray.length;

C. myarray.size;

D. myarray.size;

B

---

7. which of the following statements is true ?

A. Constructors cannot have a visibility modifier

B. Constructor can be marked public and protected, but not private

C. Constructors can only have a primitive return type

D. Constructors are not inherited

D

---

5. What is difference between class and object ?(Select the best answer)

A. A class describes object behavior and an object defines class properties

B. A class is a template for objects and an object is an instance of a class

C. Class and object are essentially the same , they only differ in name

D. An object is a logical entity while class is a physical entity

A

---

4. In order to use the TreeSet class, the class that describes elements must implement the .... interface

A. java.lang.Comparable

B. java.lang.Comparing

C. java.util.Comparable

D. java.util.treeset.Compare

A

---

3. If all three top-level elements occur in a source file , they must appear in which order?

A. Imports, package declarations, classes/ interfaces/enums

B. Class/ interfaces/enums, imports, package declarations

C. Package declaration must come first; order for import and class/ intefaces/enum definitions is not significant

D. Package declaration, import, class/interface/enum definitions

E. Import must come first , order for package declaration and class/interface/ enum definitions is not significant

D

---

14. Which of the following may overide a method which signature is void ABC(int i)?

A. private void ABC(int k)

B. public void ABC(int k)

C. int ABC(int k)

D. private int ABC(int k)

B

---

13. What will happen when you attempt to compile and run the following code: int Output = 10; boolean b1 = false;if((b1 == true )&& (Output + 10) == 20){System.out.println ("We are equal "+ Output);}else{System.out.println ("Not equal" + Output);}

A. Compile error, attempting to perform binary comparision on logical data type

B. Compilation and output of "We are equal 10"

C. Compilation and output of "Not equal 20"

D. Compilation and output of "Not equal 10"

D

---

15.Given the following Java code://===========================================

class **Ladder**{

private final void step(){

System.out.println("Ladder");

}

}

public class Lift extends Ladder{

public final void step(){

System.out.println("Lift");

}

public static void main(String [] args){

new Lift().step();

}

}//=========================================

What is the result ?

A. Lift

B. Ladder

C. Ladder

Lift

D. Lift

A

---

17. Given the following Java code:

public class ElectricCooker implements Appliance {

public void cook() {

}

}

//2

abstract class ElectricRiceCooker1 extends ElectricCooker {

}

//4

abstract class ElectricRiceCooker2 extends ElectricCooker {

public void cook(double time) {

}

//5

}

//6

class ElectricRiceCooker3 extends ElectricCooker {

//7

public void ricecook() {

}

///8

}

interface Appliance {

public void cook();

}

what is the result?

nothing

---

18. Suppose A and B are correctly defined class , X and Y are correctly defined interfaces and all statment in branket pair() are also correct. Which of the following is the corrrect declaration of the class Test?

A. class Test extend A, B implements X,Y {}

B. class Test extend A,B implements X{ }

C. class Test extends A implements X,Y{}

D. class Test extend A, implements X,Y{}

D

---

21. Suppose a class has public visibility in this class we define a protected method. Which of the following statements is correct?

A. This method is only accessible from inside the class itself and from inside all subclass

B. In a class, you cannot declare methods with lower visibility than the visibility of the class in which it is defined

C. From within protected method you do not have access to public methods

D. This method is accessible from within the class itself and from when all classes are defined in the same package as the class itself

D

---

28. Java array always start of index 0

---

29. Which of the following statements is true ?

A class may extend only one other class but may implement many interfaces

---

31.

Which of the following statements regarding "the final " modifier are true? (select 3)

A. A variable defined as "final" is a constant

B. A class declared as being "final" can be subclassed

C. A method declared "final " cannot be overridden in the subclass

D. A class declared as being "final " cannot be subclassed

E. A method declared "final" can be overridden in the subclass

A B C

---

40. What must a class do to implement an interface? (Select correct answer)

A. It must provide all of the methods in the interface and identify the interface in its implements clause

B. It must provide all of the methods in an interface

C. It must identify the interface in its implements clause

D. It must provide all of the methods in the interface and identify the interface in its implements clause and does not extend any other class

D

---

35. public class Test{

public static void main(String[] args){

Object ob1 = new Object();

Object ob2 = ob1;if(ob1.equal (ob2)

System.out.println("ob1 equal ob2");

if( ob1 == ob2)

System.out.println("ob1 = ob2");

System.out.println("Have a nice day !");}}

A. ob1 equals ob2

ob1 = ob2

Have a nice day!

---

You required to print out all the elements of a list.

Which of the following is NOT valid?

List<String > names = new ArrayList<();

names.add("ToLan");

names.add("Vu Ha");

names .add("Cao Anh");

C. names.parralleStream().forEach(System.out.println);

---

38. Given the following code , which of the result that follow would you expect?

1. package TestInterfaces;

2. interface Demo{

3. protected void Input();

4. public void Output();

5. public void Draw();

6. }

A. The code will not compile because of line 3

B. The code will not compile because of line 4

C. The code will not compile because of line 5

A

---

47. The following code will point1.

if(new Boolean("true") == new Boolean("true"))

2. System.out.println("True");

3. else

4. System.out.println("False");

A. Compilation error

B. No compilation error , but runtime exception

C. Prints "True"

D. Prints "False"

E. Depends on the particular implementation of the Java Virtual Machine

D

---

48. Given the following class definitionpublic class Upton{

public static void main(String [] args){}

public void amethod(int i){}

/// Here}

Which of the following would be illegal to place after the comment //Here?

public int amethod(int z){}

---

Is the following code legal?

try {

} finally {

}

Có, nó hợp pháp - và rất hữu ích.

Một câu lệnh try không nhất thiết phải có khối catch nếu nó có khối finally.

Nếu mã trong câu lệnh try có nhiều điểm thoát và không có mệnh đề catch liên quan, thì mã trong khối finally được thực thi bất kể khối try được thoát như thế nào. Vì vậy, nó là hợp lý để cung cấp một khối finally bất cứ khi nào có mã luôn phải được thực thi. Điều này bao gồm mã khôi phục tài nguyên, chẳng hạn như mã để đóng các luồng I / O.

How many bits does a float contain?  
A. 1  
B. 8  
C. 16  
D. 32  
E. 64

D. 32

|  |  |  |
| --- | --- | --- |
| **Data Type** | **Default Value** | **Default size** |
| boolean | false | 1 bit |
| char | '\u0000' | 2 byte |
| byte | 0 | 1 byte |
| short | 0 | 2 byte |
| int | 0 | 4 byte |
| long | 0L | 8 byte |
| float | 0.0f | 4 byte |
| double | 0.0d | 8 byte |

## **Toán tử số học trong Java (Arithmetic operations)**

|  |  |  |
| --- | --- | --- |
| **Toán tử** | **Miêu tả** | **Ví dụ** |
| + | Phép cộng | A + B sẽ cho kết quả 30 |
| - | Phép trừ: trừ toán hạng trái cho toán hạng phải | A - B sẽ cho kết quả -10 |
| \* | Phép nhân | A \* B sẽ cho kết quả 200 |
| / | Phép chia: chia toán hạng trái cho toán hạng phải | B / A sẽ cho kết quả 2 |
| % | Phép chia lấy phần dư: Lấy phần dư của phép chia toán hạng trái cho toán hạng phải | B % A sẽ cho kết quả 0 |
| ++ | Phép lượng gia: lượng gia giá trị toán hạng thêm 1 | B++ sẽ cho kết quả 21 |
| -- | Phép lượng giảm: lượng giảm giá trị toán hạng đi 1 | B-- sẽ cho kết quả 19 |

## **Toán tử quan hệ trong Java**

|  |  |  |
| --- | --- | --- |
| **Toán tử** | **Miêu tả** | **Ví dụ** |
| == | Kiểm tra nếu giá trị của hai toán hạng có cân bằng hay không, nếu có thì điều kiện là true. | (A == B) là không true. |
| != | Kiểm tra nếu giá trị hai toán hạng là cân bằng hay không, nếu không cân bằng, thì điều kiện là true | (A != B) là true. |
| > | Kiểm tra nếu toán hạng trái có lớn hơn toán hạng phải hay không, nếu có thì điều kiện là true | (A > B) là không true. |
| < | Kiểm tra nếu toán hạng phải có lớn hơn toán hạng trái hay không, nếu có thì điều kiện là true | (A < B) là true. |
| >= | Kiểm tra nếu toán hạng trái có lớn hơn hoặc bằng toán hạng phải hay không, nếu có thì điều kiện là true | (A >= B) là không true. |
| <= | Kiểm tra nếu toán hạng phải có lớn hơn hoặc bằng toán hạng trái hay không, nếu có thì điều kiện là true | (A <= B) là true. |

**Toán tử dịch:**

|  |  |  |
| --- | --- | --- |
| << | Toán tử dịch trái. Giá trị toán hạng trái được dịch chuyển sang trái bởi số các bit được xác định bởi toán hạng bên phải. | A << 2 sẽ cho kết quả 240, hay là 1111 0000 |
| >> | Toán tử dịch phải. Giá trị toán hạng trái được dịch chuyển sang phải bởi số các bit được xác định bởi toán hạng bên phải | A >> 2 sẽ cho kết quả 15, hay là 1111 |
| >>> | Toán tử dịch phải và điền 0 vào chỗ trống | A >>>2 sẽ cho kết quả 15, hay là 0000 1111 |

7. Consider the following application:

1. class Q7 {

2. public static void main(String args[]) {

3. double d = 12.3;

4. Decrementer dec = new Decrementer();

5. dec.decrement(d);

6. System.out.println(d);

7. }

8. } 9.

10. class Decrementer {

11. public void decrement(double decMe) {

12. decMe = decMe - 1.0;

13. }

14. }

What value is printed out at line 6?

A. 0.0

B. 1.0

C. 12.3

D. 11.3

**12.3**

---

18. Which of the following are legal?

A. char c = 0x1234;

B. char c = \u1234;

C. char c = '\u1234';

**char c = '\u1234';**

---

19. Consider the following code:

1. StringBuffer sbuf = new StringBuffer();

2. sbuf = null;

3. System.gc();

Choose all true statements:

A. After line 2 executes, the StringBuffer object is garbage collected.

B. After line 3 executes, the StringBuffer object is garbage collected.

C. After line 2 executes, the StringBuffer object is eligible for garbage collection.

D. After line 3 executes, the StringBuffer object is eligible for garbage collection.

After **line 2** executes, the StringBuffer object is **eligible for** garbage collection.

---

3. Which of the following expressions results in a positive value in x?

A. int x = -1; x = x >>> 5;

B. int x = -1; x = x >>> 32;

C. byte x = -1; x = x >>> 5;

D. int x = -1; x = x >> 5;

**int x = -1; x = x >>> 5;**

---

11. Suppose ob1 and ob2 are references to instances of java.lang.Object. If (ob1 == ob2) is false, can ob1.equals(ob2) ever be true?

A. Yes

B. No

**No**

---

15. Which of the following operations might **throw an ArithmeticException**?

A. >>

B. >>>

C. <<

D. None of these

**None of these**

---

16. Which of the following operations might **throw an ArithmeticException**?

A. +

B. -

C. \*

D. /

E. None of these

**/**

---

20. What is -50 >> 1?

A. A negative number with very large magnitude.

B. A positive number with very large magnitude.

C. -100

D. -25

E. 100

F. 25

**-25**

---

3. What is the minimal modification that will make this code compile correctly?

1. final class Aaa

2. {

3. int xxx;

4. void yyy() { xxx = 1; }

5. } 6.

7.

8. class Bbb extends Aaa

9. {

10. final Aaa finalref = new Aaa(); 11.

12. final void yyy()

13. {

14. System.out.println("In method yyy()");

15. finalref.xxx = 12345;

16. }

17. }

A. On line 1, remove the final modifier.

B. On line 10, remove the final modifier.

C. Remove line 15.

D. On lines 1 and 10, remove the final modifier.

E. The code will compile as is. No modification is needed.

**On line 1, remove the final modifier.**

---

4. Which of the following statements is true?

A. Transient methods may not be overridden.

B. Transient methods must be overridden.

C. Transient classes may not be serialized.

D. Transient variables must be static.

E. Transient variables are not serialized.

**Transient variables** are **not serialized.**

---

8. Which modifier or modifiers should be used to denote a variable that should not be written out as part of its **class's persistent state**? (Choose the shortest possible answer.)

A. private

B. protected

C. private protected

D. transient

E. volatile

**transient**

---

9. This question concerns the following class definition:

1. package abcde; 2.

3. public class Bird {

4. protected static int referenceCount = 0;

5. public Bird() { referenceCount++; }

6. protected void fly() { /\* Flap wings, etc. \*/ }

7. static int getRefCount() { return referenceCount; }

8. }

Which statement is true about class Bird and the following class Parrot?

1. package abcde; 2.

3. class Parrot extends abcde.Bird {

4. public void fly() {

5. /\* Parrot-specific flight code. \*/

6. }

7. public int getRefCount() {

8. return referenceCount;

9. }

10. }

A. Compilation of Parrot.java fails at line 4 because method fly() is protected in the superclass, and classes Bird and Parrot are in the same package.

B. Compilation of Parrot.java fails at line 4 because method fly() is protected in the superclass and public in the subclass, and methods may not be overridden to be more public.

C. Compilation of Parrot.java fails at line 7 because method getRefCount() is static in the superclass, and static methods may not be overridden to be nonstatic.

D. Compilation of Parrot.java succeeds, but a runtime exception is thrown if method fly()

is ever called on an instance of class Parrot.

E. Compilation of Parrot.java succeeds, but a runtime exception is thrown if method

getRefCount() is ever called on an instance of class Parrot.

Compilation of Parrot.java **fails at line 7** because method getRefCount() is static in the superclass, and static methods may not be overridden to be nonstatic.

---

10. This question concerns the following class definition:

1. package abcde; 2.

3. public class Bird {

4. protected static int referenceCount = 0;

5. public Bird() { referenceCount++; }

6. protected void fly() { /\* Flap wings, etc. \*/ }

7. static int getRefCount() { return referenceCount; }

8. }

Which statement is true about class Bird and the following class Nightingale?

1. package singers; 2.

3. class Nightingale extends abcde.Bird {

4. Nightingale() { referenceCount++; } 5.

6. public static void main(String args[]) {

7. System.out.print("Before: " + referenceCount);

8. Nightingale florence = new Nightingale();

9. System.out.println(" After: " + referenceCount);

10. florence.fly();

11. }

12. }

A. The program will compile and execute. The output will be Before: 0 After: 2.

B. The program will compile and execute. The output will be Before: 0 After: 1.

C. Compilation of Nightingale will fail at line 4 because static members cannot be overridden.

D. Compilation of Nightingale will fail at line 10 because method fly() is protected in the superclass.

E. Compilation of Nightingale will succeed, but an exception will be thrown at line 10, because method fly() is protected in the superclass.

The program will **compile and execute**. The output will be **Before: 0 After: 2.**

---

12. Which of the following statements are true?

A. An abstract class may be instantiated.

B. An abstract class must contain at least one abstract method.

C. An abstract class must contain at least one abstract data field.

D. An abstract class must be overridden.

E. An abstract class must declare that it implements an interface.

F. None of the above.

**None of the above.**

A. Một lớp trừu tượng có thể được khởi tạo.

B. Một lớp trừu tượng phải chứa ít nhất một phương thức trừu tượng.

C. Một lớp trừu tượng phải chứa ít nhất một trường dữ liệu trừu tượng.

D. Một lớp trừu tượng phải được ghi đè.

E. Một lớp trừu tượng phải khai báo rằng nó thực hiện một giao diện.

---

13. Suppose interface Inty defines five methods. Suppose class Classy declares that it implements Inty but does not provide implementations for any of the five interface methods. Which is/are true?

A. The class will not compile.

B. The class will compile if it is declared public.

C. The class will compile if it is declared abstract.

D. The class may not be instantiated.

The class will compile if it is **declared abstract.**

The class **may not be instantiated.**

Lớp sẽ biên dịch nếu nó được khai báo là trừu tượng.

Lớp có thể không được khởi tạo.

---

14. Which of the following may be declared final? (Choose all that apply.)

A. Classes

B. Data

C. Methods

**Classes**

**Data**

**Methods**

---

15. Which of the following may follow the static keyword? (Choose all that apply.)

A. Class definitions

B. Data

C. Methods

D. Code blocks enclosed in curly brackets

**Data**

**Methods**

**Code blocks enclosed in curly brackets**

---

16. Suppose class A has a method called doSomething(), with default access. Suppose class B extends A and overrides doSomething(). Which access modes **may apply** to B's version of doSomething()? (Choose all that apply.)

A. public

B. private

C. protected

D. Default

**public**

**protected**

**Default**

(chú ý phân biệt với **may NOT apply**)

---

19. Which of the following statements are true?

A. A final class must be instantiated.

B. A final class may only contain final methods.

C. A final class may not contain non-final data fields.

D. A final class may not be extended.

E. None of the above.

A final class **may not be extended.**

---

6. Which of the following statements is true? (Choose one.)

A. Object references can be converted in assignments but not in method calls.

B. Object references can be converted in method calls but not in assignments.

C. Object references can be converted in both method calls and assignments, but the rules governing these conversions are very different.

D. Object references can be converted in both method calls and assignments, and the rules governing these conversions are identical.

E. Object references can never be converted.

Object references can be converted in **both method calls and assignments**, and the rules governing these conversions are identical.

Các tham chiếu đối tượng có thể được chuyển đổi trong cả lệnh gọi và phép gán phương thức, đồng thời các quy tắc điều chỉnh các chuyển đổi này là giống hệt nhau.

---

7. Consider the following code. Which line will not compile?

1. Object ob = new Object();

2. String[] stringarr = new String[50];

3. Float floater = new Float(3.14f);

4. ob = stringarr;

5. ob = stringarr[5];

6. floater = ob;

7. ob = floater;

A. Line 4

B. Line 5

C. Line 6

D. Line 7

Line 6

---

9. Consider the following code:

1. Cat sunflower;

2. Washer wawa;

3. SwampThing pogo; 4.

5. sunflower = new Cat();

6. wawa = sunflower;

7. pogo = (SwampThing)wawa;

Which of the following statements is true? (Choose one.)

A. Line 6 will not compile; an explicit cast is required to convert a Cat to a Washer.

B. Line 7 will not compile, because you cannot cast an interface to a class.

C. The code will compile and run, but the cast in line 7 is not required and can be eliminated.

D. The code will compile but will throw an exception at line 7, because runtime conversion from an interface to a class is not permitted.

E. The code will compile but will throw an exception at line 7, because the runtime class of

wawa cannot be converted to type SwampThing.

The code will **compile but will throw an exception at line 7**, because the runtime class of wawa cannot be converted to type SwampThing.

Mã sẽ biên dịch nhưng sẽ ném ra một exception ở dòng 7, vì lớp thời gian chạy của wawa không thể được chuyển đổi thành kiểu SwampThing.

---

10. Consider the following code:

1. **Raccoon rocky**;

2. SwampThing pogo;

3. Washer w;

4.

5. rocky = new Raccoon();

6. w = rocky;

7. pogo = w; //

Which of the following statements is true? (Choose one.)

A. Line 6 will not compile; an explicit cast is required to convert a Raccoon to a Washer.

B. Line 7 will not compile; an explicit cast is required to convert a Washer to a SwampThing.

C. The code will compile and run.

D. The code will compile but will throw an exception at line 7, because runtime conversion from an interface to a class is not permitted.

E. The code will compile but will throw an exception at line 7, because the runtime class of w

cannot be converted to type SwampThing.

**Line 7 will not compile**; an explicit cast is required to convert a Washer to a SwampThing.

Dòng 7 sẽ không biên dịch; cần có cast rõ ràng để chuyển đổi Washer thành SwampThing.

---

14. Suppose the type of xarr is an array of XXX, and the type of yarr is an array of YYY. When is the assignment xarr = yarr; legal?

A. Sometimes

B. Always

C. Never

**Sometimes**

---

15. When is x & y an int? (Choose one).

A. Always

B. Sometimes

C. When neither x nor y is a float, a long, or a double

**Sometimes**

---

16. What are the legal types for whatsMyType?

short s = 10; whatsMyType = !s;

A. short

B. int

C. There are no possible legal types.

**There are no possible legal types.**

---

20. What is the difference between the rules for method-call conversion and the rules for assignment conversion?

A. There is no difference; the rules are the same.

B. Method-call conversion supports narrowing, assignment conversion does not.

C. Assignment conversion supports narrowing, method-call conversion does not.

D. Method-call conversion supports narrowing if the method declares that it throws

ClassCastException.

There is no difference; the rules are the same.

---

2. Consider the following code:

1. outer: for (int i = 0; i < 2; i++) {

2. for (int j = 0; j < 3; j++) {

3. if (i == j) {

4. continue outer;

5. }

6. System.out.println("i = " + i + " j = " + j);

7. }

8. }

Which lines would be part of the output? (Choose all that apply.)

A. i = 0 j = 0

B. i = 0 j = 1

C. i = 0 j = 2

D. i = 1 j = 0

E. i = 1 j = 1

**i = 1 j = 0**

---

3. Which of the following are legal loop constructions? (Choose all that apply.)

A. while (int i<7) { i++;

System.out.println("i is " + i);

}

B. int i = 3; while (i) {

System.out.println("i is " + i);

}

C. int j = 0;

for (int k=0, j+k != 10; j++,k++) { System.out.println("j=" + j + ", k=" + k);

}

D. int j=0; do {

System.out.println("j=" + j++); if (j==3)

continue loop;

} while (j<10);

C. int j = 0;

for (int k=0, j+k != 10; j++,k++) { System.out.println("j=" + j + ", k=" + k);

}

---

5. Which statement is true about the following code fragment?

1. int j = 2;

2. switch (j) {

3. case 2:

4. System.out.println("value is two");

5. case 2 + 1:

6. System.out.println("value is three");

7. break;

8. default:

9. System.out.println("value is " + j);

10. break;

11. }

A. The code is illegal because of the expression at line 5.

B. The acceptable types for the variable j, as the argument to the switch() construct, could be any of byte, short, int, or long.

C. The output would be the text value is two.

D. The output would be the text value is two followed by the text value is three.

E. The output would be the text value is two, followed by the text value is three, fol- lowed by the text value is 2.

**The output would be the text value is two followed by the text value is three.**

---

17. When is it appropriate to pass a cause to an exception's constructor?

A. Always

B. When the exception is being thrown in response to catching of a different exception type

C. When the exception is being thrown from a public method

D. When the exception is being thrown from a private method

When the exception is being thrown **in response to catching of a different exception type**

---

19. When does an exception's stack trace (dấu vết) get recorded (được ghi lại) in the exception object?

A. When the exception is constructed

B. When the exception is thrown

C. When the exception is caught

D. When the exception's printStackTrace() method is called

**When the exception is constructed**

---

20. When is it appropriate to write code that constructs and throws an error?

(phù hợp để constructs và throws 1 lỗi)

A. When a public method's preconditions are violated

B. When a public method's postconditions are violated

C. When a nonpublic method's preconditions are violated

D. When a nonpublic method's postconditions are violated

E. Never

**Never**

---

5. Consider the following classes, declared in separate source files:

1. public class Base {

2. public void method(int i) {

3. System.out.print("Value is " + i);

4. }

5. }

1. public class Sub extends Base {

2. public void method(int j) {

3. System.out.print("This value is " + j);

4. }

5. public void method(String s) {

6. System.out.print("I was passed " + s);

7. }

8. public static void main(String args[]) {

9. Base b1 = new Base();

10. Base b2 = new Sub();

11. b1.method(5);

12. b2.method(6);

13. }

14. }

What output results when the main method of the class Sub is run?

A. Value is 5Value is 6

B. This value is 5This value is 6

C. Value is 5This value is 6

D. This value is 5Value is 6

E. I was passed 5I was passed 6

**Value is 5This value is 6**

---

9. Which of the following statements are true? (Choose all that apply.)

A. Given that Inner is a nonstatic class declared inside a public class Outer and that appro- priate constructor forms are defined, an instance of Inner can be constructed like this: new Outer().new Inner()

B. If an anonymous inner class inside the class Outer is defined to implement the interface

ActionListener, it can be constructed like this: new Outer().new ActionListener()

C. Given that Inner is a nonstatic class declared inside a public class Outer and that appro- priate constructor forms are defined, an instance of Inner can be constructed in a static method like this: new Inner()

D. An anonymous class instance that implements the interface MyInterface can be constructed and returned from a method like this:

1. return new MyInterface(int x) {

2. int x;

3. public MyInterface(int x) {

4. this.x = x;5. }6. };

*Given that Inner is a nonstatic class declared inside a public class Outer and that appro- priate constructor forms are defined, an instance of Inner can be constructed like this:* **new Outer().new Inner()**

---

11. Which of the following may **override** a method whose signature is void xyz(float f)?

A. void xyz(float f)

B. public void xyz(float f)

C. private void xyz(float f)

D. public int xyz(float f)

E. private int xyz(float f)

**void xyz(float f)**

**public void xyz(float f)**

override => giống signature

---

14. Suppose x and y are of type TrafficLightState, which is an enum. What is the best way to test whether x and y refer to the same constant?

A. if (x == y)

B. if (x.equals(y))

C. if (x.toString().equals(y.toString()))

D. if (x.hashCode() == y.hashCode())

**if (x == y)**

---

15. Which of the following restrictions apply to anonymous inner classes?

A. They must be defined inside a code block.

B. They may only read and write final variables of the enclosing class.

C. They may only call final methods of the enclosing class.

D. They may not call the enclosing class' synchronized methods.

**They must be defined inside a code block.**

---

18. Which methods return an enum constant's name?

A. getName()

B. name()

C. toString()

D. nameString()

E. getNameString()

**name()**

**toString()**

---

1. Which one statement is true concerning the following code?

1. class Greebo extends java.util.Vector

2. implements Runnable {

3. public void run(String message) {

4. System.out.println("in run() method: " +

5. message);

6. }

7. } 8.

9. class GreeboTest {

10. public static void main(String args[]) {

12. Greebo g = new Greebo();

13. Thread t = new Thread(g);

14. t.start();

15. }

16. }

A. There will be a compiler error, because class Greebo does not correctly implement the

Runnable interface.

B. There will be a compiler error at line 13, because you cannot pass a parameter to the constructor of a Thread.

C. The code will compile correctly but will crash with an exception at line 13.

D. The code will compile correctly but will crash with an exception at line 14.

E. The code will compile correctly and will execute without throwing any exceptions.

There will be a **compiler error**, because class **Greebo does not correctly** implement the Runnable interface.

---

2. Which one statement is always true about the following application?

1. class HiPri extends Thread {

2. HiPri() {

3. setPriority(10);

4. } 5.

6. public void run() {

7. System.out.println(

8. "Another thread starting up.");

9. while (true) { }

10. } 11.

12. public static void main(String args[]) {

13. HiPri hp1 = new HiPri();

14. HiPri hp2 = new HiPri();

15. HiPri hp3 = new HiPri();

16. hp1.start();

17. hp2.start();

18. hp3.start();

19. }

20. }

A. When the application is run, thread hp1 will execute; threads hp2 and hp3 will never get the CPU.

B. When the application is run, thread hp1 will execute to completion, thread hp2 will execute to completion, then thread hp3 will execute to completion.

C. When the application is run, all three threads (hp1, hp2, and hp3) will execute concurrently, taking time-sliced turns in the CPU.

D. None of the above scenarios can be guaranteed to happen in all cases.

When the application is run, all three **threads (hp1, hp2, and hp3)** will execute concurrently, taking time-sliced turns in the CPU.

---

6. If you attempt to compile and execute the following application, will it ever print out the message In xxx?

1. class TestThread3 extends Thread {

2. public void run() {

3. System.out.println("Running");

4. System.out.println("Done");

5. } 6.

7. private void xxx() {

8. System.out.println("In xxx");

9. }

10.

11. public static void main(String args[]) {

12. TestThread3 ttt = new TestThread3();

13. ttt.xxx();

14. ttt.start();

12. }

13. }

A. Yes

B. No

**Yes**

---

7. A Java monitor must either extend Thread or implement Runnable.

A. True

B. False

**False**

Một trình giám sát Java phải mở rộng Thread hoặc triển khai Runnable.

---

8. Which of the following methods in the Thread class are deprecated?

A. suspend() and resume()

B. wait() and notify()

C. start() and stop()

D. sleep() and yield()

**suspend() and resume()**

---

9. Which of the following statements about threads is true?

A. Every thread starts executing with a priority of 5.

B. Threads inherit their priority from their parent thread.

C. Threads are guaranteed to run with the priority that you set using the setPriority()

method.

D. Thread priority is an integer ranging from 1 to 100.

**Threads inherit their priority from their parent thread.**

Các luồng kế thừa mức độ ưu tiên của chúng từ luồng parent của chúng.

---

10. Which of the following statements about the wait() and notify() methods is true?

A. The wait() and notify() methods can be called outside synchronized code.

B. The programmer can specify which thread should be notified in a notify() method call.

C. The thread that calls wait() goes into the monitor's pool of waiting threads.

D. The thread that calls notify() gives up the lock.

**The thread that calls wait() goes into the monitor's pool of waiting threads.**

---

13. How many locks does an object have?

A. One

B. One for each method

C. One for each synchronized method

D. One for each non-static synchronized method

**One**

---

14. Is it possible to write code that can execute only if the current thread owns multiple locks?

A. Yes.

B. No.

Yes.

Có thể viết mã chỉ có thể thực thi nếu luồng hiện tại sở hữu nhiều khóa không?

---

17. How do you prevent shared data from being corrupted in a multithreaded environment?

A. Mark all variables as synchronized.

B. Mark all variables as volatile.

C. Use only static variables.

D. Access the variables only via synchronized methods.

**Access the variables only via synchronized methods.**

---

18. How can you ensure that multithreaded code does not deadlock?

A. Synchronize access to all shared variables.

B. Make sure all threads yield from time to time.

C. Vary the priorities of your threads.

D. A, B, and C do not ensure that multithreaded code does not deadlock.

**A, B, and C do not ensure** that multithreaded code does not deadlock.

---

3. Suppose you want to write a class that offers static methods to compute hyperbolic trigonometric functions. You decide to subclass java.lang.Math and provide the new functionality as a set of static methods. Which one statement is true about this strategy?

A. The strategy works.

B. The strategy works, provided the new methods are public.

C. The strategy works, provided the new methods are not private.

D. The strategy fails because you cannot subclass java.lang.Math.

E. The strategy fails because you cannot add static methods to a subclass.

The strategy **fails** because you **cannot subclass java.lang.Math.**

---

4. Which one statement is true about the following code fragment?

1. import java.lang.Math;

2. Math myMath = new Math(); //sai từ đây

3. System.out.println("cosine of 0.123 = " +

4. myMath.cos(0.123));

A. Compilation fails at line 2.

B. Compilation fails at line 3 or 4.

C. Compilation succeeds, although the import on line 1 is not necessary. During execution, an exception is thrown at line 3 or 4.

D. Compilation succeeds. The import on line 1 is necessary. During execution, an exception is thrown at line 3 or 4.

E. Compilation succeeds and no exception is thrown during execution.

**Compilation fails at line 2.**

---

5. Which one statement is true about the following code fragment?

1. String s = "abcde";

2. StringBuffer s1 = new StringBuffer("abcde");

3. if (s.equals(s1))

4. s1 = null;

5. if (s1.equals(s))

6. s = null;

A. Compilation fails at line 1 because the String constructor must be called explicitly.

B. Compilation fails at line 3 because s and s1 have different types.

C. Compilation succeeds. During execution, an exception is thrown at line 3.

D. Compilation succeeds. During execution, an exception is thrown at line 5.

E. Compilation succeeds. No exception is thrown during execution.

**Compilation succeeds. No exception is thrown during execution.**

---

6. In the following code fragment, after execution of line 1, sbuf references an instance of the

StringBuffer class. After execution of line 2, sbuf still references the same instance.

1. StringBuffer sbuf = new StringBuffer("abcde");

2. sbuf.insert**(3, "xyz");**

A. True

B. False

**True**

---

7. In the following code fragment, after execution of line 1, sbuf references an instance of the

StringBuffer class. After execution of line 2, sbuf still references the same instance.

1. StringBuffer sbuf = new StringBuffer("abcde");

2. sbuf.append**("xyz");**

A. True

B. False

**True**

---

8. In the following code fragment, line 4 is executed.

1. String s1 = "xyz";

2. String s2 = "xyz";

3. if (s1 == s2)

4. System.out.println("Line 4");

A. True

B. False

**True**

---

9. In the following code fragment, line 4 is executed.

1. String s1 = "xyz";

2. String s2 = new String(s1);

3. if (s1 == s2)

4. System.out.println("Line 4");

A. True

B. False

**False**

Line 4 không execute vì dùng new để khởi tạo s2 => không ==

---

11. Which of the following are legal? (Choose all that apply.)

A. List<String> theList = new Vector<String>;

B. List<String> theList = new Vector<String>();

C. Vector <String> theVec = new Vector<String>;

D. Vector <String> theVec = new Vector<String>();

List<String> theList = new Vector<String>();

Vector <String> theVec = new Vector<String>();

---

17. Which line of code tells a scanner called sc to use a single digit as a delimiter?

A. sc.useDelimiter("d");

B. sc.useDelimiter("\d");

C. sc.useDelimiter("\\d");

D. sc.useDelimiter("d+");

E. sc.useDelimiter("\d+");

F. sc.useDelimiter("\\d+");

**sc.useDelimiter("\\d");**

---

20. Which of the following statements are true?

A. StringBuilder is generally faster than StringBuffer.

B. StringBuffer is generally faster than StringBuilder.

C. StringBuilder is threadsafe; StringBuffer is not.

D. StringBuffer is threadsafe; StringBuilder is not.

**StringBuilder is generally faster than StringBuffer.**

**StringBuffer is threadsafe; StringBuilder is not.**

---

2. Which of the statements below are true? (Choose all that apply.)

A. When you construct an instance of File, if you do not use the file-naming semantics of the local machine, the constructor will throw an IOException.

B. When you construct an instance of File, if the corresponding file does not exist on the local file system, one will be created.

C. When an instance of File is garbage collected, the corresponding file on the local file system is deleted.

D. None of the above.

None of the above.

A. Khi bạn xây dựng một thể hiện của Tệp, nếu bạn không sử dụng ngữ nghĩa đặt tên tệp của máy cục bộ, thì hàm tạo sẽ ném ra một IOException. B. Khi bạn xây dựng một thể hiện của Tệp, nếu tệp tương ứng không tồn tại trên hệ thống tệp cục bộ, một thể hiện sẽ được tạo.

C. Khi một thể hiện của Tệp được thu thập rác, tệp tương ứng trên hệ thống tệp cục bộ sẽ bị xóa.

---

9. You execute the following code in an empty directory. What is the result?

1. File f1 = new File("dirname");

2. File f2 = new File(f1, "filename");

A. A new directory called dirname is created in the current working directory.

B. A new directory called dirname is created in the current working directory. A new file called

filename is created in directory dirname.

C. A new directory called dirname and a new file called filename are created, both in the current working directory.

D. A new file called filename is created in the current working directory.

E. No directory is created, and no file is created.

**No directory is created, and no file is created.**

---

10. What is the result of attempting to compile and execute the following code fragment? Assume that the code fragment is part of an application that has write permission in the current working directory. Also assume that before execution, the current working directory does not contain a file called datafile.

1. try {

2. RandomAccessFile raf = new

3. RandomAccessFile("datafile" ,"rw");

4. BufferedOutputStream bos = new

5. BufferedOutputStream(raf);

6. DataOutputStream dos = new

7. DataOutputStream(bos);

8. dos.writeDouble(Math.PI);

9. dos.close();

10. bos.close();

11. raf.close();

12. }

13. catch (IOException e) { }

A. The code fails to compile.

B. The code compiles but throws an exception at line 4.

C. The code compiles and executes but has no effect on the local file system.

D. The code compiles and executes; afterward, the current working directory contains a file called datafile.

The code **fails to compile.**

---

13. Suppose class A extends Object; class B extends A; and class C extends B. Of these, only class C implements java.io.**Serializable**. Which of the following must be true in order to avoid an exception during deserialization of an instance of C?

A. A must have a no-args constructor.

B. B must have a no-args constructor.

C. C must have a no-args constructor.

D. There are no restrictions regarding no-args constructors.

**B** *must have a no-args constructor.*

---

14. Suppose class A extends Object; Class B extends A; and class C extends B. Of these, only class C implements java.io.**Externalizable**. Which of the following must be true in order to avoid an exception during deserialization of an instance of C?

A. A must have a no-args constructor.

B. B must have a no-args constructor.

C. C must have a no-args constructor.

D. There are no restrictions regarding no-args constructors.

**C** *must have a no-args constructor.*

---

18. What happens when you try to compile and run the following application?

1. import java.io.\*; 2.

3. public class Xxx {

4. public static void main(String[] args) {

5. try {

6. File f = new File("xxx.ser");

7. FileOutputStream fos = new FileOutputStream(f);

8. ObjectOutputStream oos = new ObjectOutputStream(fos);

9. oos.writeObject(new Object());

10. oos.close();

11. fos.close();

12. }

13. catch (Exception x) { }

14. }

15. }

A. Compiler error at line 9.

B. An exception is thrown at line 9.

C. An exception is thrown at line 10.

D. No compiler error and no exception.

**An exception is thrown at line 9.**

---

5. Given a class with a public variable theTint of type Color, which of the following methods are consistent with the JavaBeans naming standards?

A. public Color getColor()

B. public Color getTint()

C. public Color getTheTint()

D. public Color gettheTint()

E. public Color get\_theTint()

public Color **getTheTint()**

(hàm get trình biên dịch tạo)

---

6. Which of the following statements are true regarding the following method?

void callMe(String... names) { }

A. It doesn't compile.

B. Within the method, names is an array containing Strings.

C. Within the method, names is a list containing Strings.

D. The method may be called only from within the enclosing class.

**Within the method, names is an array containing Strings.**

---

8. Given the following class:

class A extends java.util.Vector { private A(int x) { super(x); }

}

Which statements are true?

A. The compiler creates a default constructor with public access.

B. The compiler creates a default constructor with protected access.

C. The compiler creates a default constructor with default access.

D. The compiler creates a default constructor with private access.

E. The compiler does not create a default constructor.

**The compiler does not create a default constructor.**

---

9. Which of the following types are legal arguments of a switch statement?

A. enums

B. bytes

C. longs

D. floats

E. strings

**enums**

**bytes**

---

10. Given the following:

int[] ages = { 9, 41, 49 }; int sum = 0;

Which of the following are legal ways to add the elements of the array?

A. for (int i=0; i<ages.length; i++) sum += ages[i];

B. for (int i=0; i<=ages.length; i++) sum += ages[i];

C. for (int i:ages) sum += i;

D. sum += ages[int i:ages];

**for (int i=0; i<ages.length; i++) sum += ages[i];**

**for (int i:ages) sum += i;**

---

11. Which lines check that x is equal to four? Assume assertions are enabled at compile time and runtime.

A. assert x == 4;

B. assert x != 4;

C. assert x == 4 : "x is not 4";

D. assert x != 4 : "x is not 4";

**assert x == 4;**

**assert x == 4 : "x is not 4";**

---

14. ObjectStreamException extends IOException. NotSerializableException extends ObjectStreamException. AWTException does not extend any of these. All are checked exceptions. The callMe() method throws NotSerializableException.What does the following code print out? Choose all lines that are printed.

try { callMe();

System.out.println("I threw");

}

catch (ObjectStreamException x) { System.out.println("Object stream");

}

catch (IOException x) { System.out.println("IO");

}

catch (Exception x) { System.out.println("Exception");

}

finally { System.out.println("Finally");

}

A. I threw

B. Object Stream

C. IO

D. Exception

E. Finally

**Object Stream**

**Finally**

---

15. While testing some code that you are developing, you notice that an **ArrayIndexOutOf- BoundsException** is thrown. What is the appropriate reaction?

A. Enclose the offending code in a try block, with a catch block for

ArrayIndexOutOfBoundsException that does nothing.

B. Enclose the offending code in a try block, with a catch block for

ArrayIndexOutOfBoundsException that prints out a descriptive message.

C. Declare that the method that contains the offending code throws

ArrayIndexOutOfBoundsException.

D. None of the above.

**None of the above.**

---

Suppose you want to use a DateFormat to format an instance of Date. What factors influence the string returned by DateFormat's format() method?

A. The operating system

B. The style, which is one of SHORT, MEDIUM, or LONG

C. The style, which is one of SHORT, MEDIUM, LONG, or FULL

D. The locale

The style, which is one of **SHORT, MEDIUM, LONG, or FULL**

The locale

---

25. How do you generate a string representing the value of a float f in a format appropriate for a locale loc?

A. NumberFormat nf = NumberFormat.getInstance(loc);

String s = nf.format(f);

B. NumberFormat nf =

new NumberFormat(loc); String s = nf.format(f);

C. NumberFormat nf = NumberFormat.getInstance();

String s = nf.format(f, loc);

D. NumberFormat nf =

new NumberFormat(loc); String s = nf.format(f, loc);

**NumberFormat nf = NumberFormat.getInstance(loc);**

**String s = nf.format(f);**

---

26. Given the following code:

1. String scanMe = "aeiou9876543210AEIOU";

2. Scanner scanner = new Scanner(scanMe);

3. String delim = ?????; // WHAT GOES HERE?

4. scanner.useDelimiter(delim);

5. while (scanner.hasNext())

6. System.out.println(scanner.next());

What code at line 3 produces the following output?

aeiou AEIOU

A. String delim = "d+";

B. String delim = "\d+";

C. String delim = "\\d+";

D. String delim = "d\*";

E. String delim = "\d\*";

F. String delim = "\\d\*";

String delim = **"\\d+";**

---

27. Which line prints double d in a **left-justified** field that is 20 characters wide, with 15 characters to the right of the decimal point?

A. System.out.format("%20.5f", d);

B. System.out.format("%20.15f", d);

C. System.out.format("%-20.5f", d);

D. System.out.format("%-20.15f", d);

System.out.format**("%-20.15f", d);**

---

29. What will be the outcome when the following application is executed?

public class ThreadTest { public void newThread() {

Thread t = new Thread() { public void run() { System.out.println("Going to sleep");

try { sleep(5000);

} catch (InterruptedException e) {}

System.out.println("Waking up");

}

};

t.start(); try {

t.join();

} catch (InterruptedException e) {} System.out.println("All done");

}

public static void main(String [] args) { new ThreadTest().newThread();

}

}

A. The code prints "Going to sleep," then "Waking up," and then "All done."

B. The code prints "All done," then "Going to sleep," and then "Waking up."

C. The code prints "All done" only.

D. The code prints "Going to sleep" and then "Waking up."

E. The code does not compile.

The code prints "**Going to sleep," then "Waking up," and then "All done."**

---

What happens when you try to compile the following code and run the Zebra application?

class Animal { float weight; Animal(float weight) {

this.weight = weight;

}

}

class Zebra extends Animal {

public static void main(String[] args) { Animal a = new Animal(222.2f);

Zebra z = new Zebra();

}

}

A. Class Animal generates a compiler error.

B. Class Zebra generates a compiler error.

C. The code compiles without error. The application throws an exception when the Animal

constructor is called.

D. The code compiles without error. The application throws an exception when the Zebra

constructor is called.

E. The code compiles and runs without error.

**Class Zebra** generates a compiler error.

---

Given the following code:

1. class Xyz {

2. float f;

3. Xyz() {

4. ??? // What goes here?

5. }

6. Xyz(float f) {

7. this.f = f;

8. }

9. }

What code at line 4 results in a class that compiles?

A. super();

B. this(1.23f);

C. this(1.23f); super();

D. super(1.23f); this(1.23f);

**super();**

**this(1.23f);**

---

What relationship does the extends keyword represent?

A. "is a"

B. "has a"

C. Polymorphism

D. Multivariance

E. Overloading

**"is a"**

Mối quan hệ mà từ khóa extends biểu thị là mối quan hệ "is a"

---

When should objects stored in a Set implement the java.util.Comparable interface?

A. Always

B. When the Set is generic

C. When the Set is a HashSet

D. When the Set is a TreeSet

E. Never

**When the Set is a TreeSet**

---

Given the following class:

class Xyzzy { int a, b;

public boolean equals(Object x) { Xyzzy that = (Xyzzy)x;

return this.a == that.a;

}

Which methods below honor the hash code contract?

A. public int hashCode() { return a; }

B. public int hashCode() { return b; }

C. public int hashCode() { return a+b;

}

D. public int hashCode() { return a\*b;

}

E. public int hashCode() { return (int)Math.random();

}

A E

**public int hashCode() { return a; }**

**public int hashCode() { return (int)Math.random();**

**}**

---

Give the following declarations: Vector plainVec; Vector<String> fancyVec;

If you want a vector in which you know you will only store strings, what are the advantages of using fancyVec rather than plainVec?

A. Attempting to add anything other than a string to fancyVec results in a compiler error.

B. Attempting to add anything other than a string to fancyVec causes a runtime exception to be thrown.

C. Attempting to add anything other than a string to fancyVec causes a checked exception to be thrown.

D. Adding a string to fancyVec takes less time than adding one to plainVec.

E. The methods of fancyVec are synchronized.

**Attempting to add anything other than a string to fancyVec results in a compiler error.**

---

The declaration of the java.util.Collection interface is

interface Collection <E>

The addAll() method of that interface takes a single argument, which is a reference to a collection whose elements are compatible with E. What is the declaration of the addAll() method?

A. public boolean addAll(Collection c)

B. public boolean

addAll(Collection c extends E)

C. public boolean

addAll(Collection ? extends E)

D. public boolean

addAll(Collection<? extends E> c)

**D. public boolean**

**addAll(Collection<? extends E> c)**

---

Given the following class:

package ocean; public class Fish {

protected int size;

protected void swim() { }

}

Which of the following may appear in a subclass of Fish named Tuna that is not in the ocean

package?

A. void swim() { };

B. public void swim() { };

C. size = 12;

D. (new Tuna()).size = 12;

B C

**public void swim() { }**

**size = 12;**

---

47. Given the following classes:

public class Wrapper { public int x;

}

public class Tester {

private static void bump(int n, Wrapper w) { n++;

w.x++;

}

public static void main(String[] args) { int n = 10;

Wrapper w = new Wrapper();

w.x = 10; bump(n, w);

// Now what are n and w.x?

}

}

When the application runs, what are the values of n and w.x after the call to bump() in the

main() method?

A. n is 10, w.x is 10

B. n is 11, w.x is 10

C. n is 10, w.x is 11

D. n is 11, w.x is 11

**n is 10, w.x is 11**

---

When does the string created on line 2 become eligible for garbage collection?

1. String s = "aaa";

2. String t = new String(s);

3. t += "zzz";

4. t = t.substring(0);

5. t = null;

A. After line 3

B. After line 4

C. After line 5

D. The string created on line 2 does not become eligible for garbage collection in this code.

**After line 3**

---

What is -15 % -10?

A. 0

B. 5

C. 10

D. -5

E. -1

**-5**

---

method is used to wait for a client to initiate communications.

a. wait()

b. accept()

c. listen()

**accept()**

---

1. public class A {

2. public String doit(int x, int y) {

3. return "a";

4. }

5.

6. public String doit(int... vals) {

7. return "b";

8. }

9. }

Given:

25. A a=new A();

26. System.out.println(a.doit(4, 5));

What is the result? (Choose one.)

a. Line 26 prints "a" to System.out.

b. Line 26 prints "b" to System.out.

c. An exception is thrown at line 26 at runtime.

d. Compilation of class A will fail due to an error in line 6.

**Line 26 prints "a" to System.out.**

---

1. public class A {

2. public void method1() {

3. B b=new B();

4. b.method2();

5. // more code here

6. }

7.}

1. public class B {

2. public void method2() {

3. C c=new C();

4. c.method3();

5. // more code here

6. }

7.}

1. public class C {

2. public void method3() {

3. // more code here

4. }

5.}

25. try {

26. A a=new A();

27. a.method1();

28. }catch (Exception e) {

29. System.out.print("an error occurred");

30. }

Which two are true if a NullPointerException is thrown on line 3 of class

C? (Choose two.)

a. The application will crash.

b. The code on line 29 will be executed.

c. The code on line 5 of class A will execute.

d. The code on line 5 of class B will execute.

e. The exception will be propagated back to line 27.

**The code on line 29 will be executed.**

**The exception will be propagated back to line 27.**

---

10. public class ClassA {

11. public void methodA() {

12. ClassB classB = new ClassB();

13. classB.getValue();

14. }

15.} And:

20. class ClassB {

21. public ClassC classC;

22.

23. public String getValue() {

24. return classC.getValue();

25. }

26.} And:

30. class ClassC {

31. public String value;

32.

33. public String getValue() {

34. value = "ClassB";

35. return value;

36. }

37.}

ClassA a = new ClassA();

a.methodA();

What is the result? (Choose one.)

a. Compilation fails.

b. ClassC is displayed.

c. The code runs with no output.

d. An exception is thrown at runtime.

**An exception is thrown at runtime.**

---

11. public class Bootchy {

12. int bootch;

13. String snootch;

14.

15. public Bootchy() {

16. this("snootchy");

17. System.out.print("first ");

18. }

19.

20. public Bootchy(String snootch) {

21. this(420, "snootchy");

22. System.out.print("second ");

23. }

24.

25. public Bootchy(int bootch, String snootch) {

26. this.bootch = bootch;

27. this.snootch = snootch;

28. System.out.print("third ");

29. }

30.

31. public static void main(String[] args) {

32. Bootchy b = new Bootchy();

33. System.out.print(b.snootch +" " + b.bootch);

34. }

35. }

What is the result? (Choose one.)

a. snootchy 420 third second first

b. snootchy 420 first second third

c. first second third snootchy 420

d. third second first snootchy 420

e. third first second snootchy 420

f. first second first third snootchy 420

**third second first snootchy 420**

---

A monitor called mon has 10 threads in its waiting pool; all these waiting

threads have the same priority. One of the threads is thr1. How can you notify thr1 so that it alone moves from the Waiting state to the Ready state? (Choose one.)

a. Execute notify(thr1); from within synchronized code of mon.

b. Execute mon.notify(thr1); from synchronized code of any object.

c. Execute thr1.notify(); from synchronized code of any object.

d. Execute thr1.notify(); from any code (synchronized or not) of any object.

e. You cannot specify which thread will get notified.

**You cannot specify which thread will get notified.**

---

A signed data type has an equal number of non-zero positive and negative

values available.

a. True

b. False

**False**

Kiểu dữ liệu signed có sẵn một số lượng các giá trị âm và dương khác 0 bằng nhau.

---

A thread wants to make a second thread ineligible for execution. To do

this, the first thread can call the yield() method on the second thread.

a. True

b. False

**False**

Một luồng muốn tạo một luồng thứ hai không đủ điều kiện để thực thi. Để thực hiện việc này, luồng đầu tiên có thể gọi phương thức yime () trên luồng thứ hai.

---

QN=21

(195) A thread's run() method includes the following lines:

1. try {

2. sleep(100);

3. } catch (InterruptedException e) { }

Assuming the thread is not interrupted, which one of the following statements is correct?

a. The code will not compile, because exceptions cannot be caught in a

thread's run() method.

b. At line 2, the thread will stop running. Execution will resume in, at most,

100 milliseconds.

c. At line 2, the thread will stop running. It will resume running in exactly

100 milliseconds.

d. At line 2, the thread will stop running. It will resume running some time

after 100 milliseconds have elapsed.

***At line 2****, the thread will stop running. It will resume running some time*

*after* ***100 milliseconds* have elapsed.**

---

Consider the following application:

1. class Q6 {

2. public static void main(String args[]) {

3. Holder h = new Holder();

4. h.held = 100;

5. h.bump(h);

6. System.out.println(h.held);

7. }

8. }

9.

10. class Holder {

11. public int held;

12. public void bump(Holder theHolder) {

13. theHolder.held++;

14 }

15. }

15. }

What value is printed out at line 6?

a. 0

b. 1

c. 100

d. 101

**101**

---

Give:

11. public static Iterator reverse(List list) {

12. Collections.reverse(list);

13. return list.iterator();

14. }

15. public static void main(String[] args) {

16. List list = new ArrayList();

17. list.add(" 1"); list.add("2"); list.add("3");

18. for (Object obj: reverse(list))

19. System.out.print(obj + ",");

20. }

'What is the result? (Choose one.)

a. 3, 2, 1,

b. 1, 2, 3,

c. Compilation fails.

d. The code runs with no output.

e. An exception is thrown at runtime.

**Compilation fails.**

---

Given arrays a1 and a2, which call returns true if a1 and a2 have the same length, and a1[i].equals(a2[i]) for every legal index i? (Choose one.)

a. java.util.Arrays.equals(a1, a2);

b. java.util.Arrays.compare(a1, a2);

c. java.util.List.compare(a1, a2);

**java.util.Arrays.equals(a1, a2);**

---

Given the following code, and making no other changes, which

combination of access modifiers (public, protected, or private) can legally be placed before aMethod() on line 3 and be placed before aMethod() on line 8? (Choose one.)

1. class SuperDuper

2. {

3. void aMethod() { }

4. }

5.

6. class Sub extends SuperDuper

7. {

8. void aMethod() { }

9. }

a. line 3: public; line 8: private

b. line 3: protected; line 8: private

c. line 3: default; line 8: private

d. line 3: private; line 8: protected

e. line 3: public; line 8: protected

**line 3: private; line 8: protected**

---

Given:

1. public interface A {

2. String DEFAULT\_GREETING = "Hello World";

3. public void method1();

4. }

A programmer wants to create an interface called B that has A as its parent. Which interface declaration is correct? (Choose one.)

a. public interface B extends A { }

b. public interface B implements A {}

c. public interface B instanceOf A {}

d. public interface B inheritsFrom A { }

**public interface B extends A { }**

---

Given:

10. class Nav{

11. public enum Direction { NORTH, SOUTH, EAST, WEST }

12. }

13. public class Sprite{

14. // insert code here

15. }

Which code, inserted at line 14, allows the Sprite class to compile? (Choose one.)

a. Direction d = NORTH;

b. Nav.Direction d = NORTH;

c. Direction d = Direction.NORTH;

d. Nav.Direction d = Nav.Direction.NORTH;

**Nav.Direction d = Nav.Direction.NORTH;**

---

Given:

10. interface Foo { int bar(); }

11. public class Sprite {

12. public int fubar( Foo foo) { return foo.bar(); }

13. public void testFoo() {

14. fubar(

15. // insert code here

16. );

17. }

18. }

Which code, inserted at line 15, allows the class Sprite to compile? (Choose one.)

a. Foo { public int bar() { return 1; } }

b. new Foo { public int bar() { return 1; } }

c. new Foo() { public int bar(){return 1; } }

d. new class Foo { public int bar() { return 1; } }

**new Foo() { public int bar(){return 1; } }**

---

Given:

11. public static void main(String[] args) {

12. try {

13. args=null;

14. args[0] = "test";

15. System.out.println(args[0]);

16. }catch (Exception ex) {

17. System.out.println("Exception");

18. }catch (NullPointerException npe) {

19. System.out.println("NullPointerException");

20. }

21. }

What is the result? (Choose one.)

a. Test

b. Exception

c. Compilation fails.

d. NullPointerException

**Compilation fails.**

---

11. public static void parse(String str) {

12. try {

13. float f= Float.parseFloat(str);

14. } catch (NumberFormatException nfe) {

15. f = 0;

16. } finally {

17. System.out.println(f);

18. }

19. }

20. public static void main(String[] args) {

21. parse("invalid");

22. }

What is the result? (Choose one.)

a. 0.0

b. Compilation fails.

c. A ParseException is thrown by the parse method at runtime.

d. A NumberFormatException is thrown by the parse method at runtime.

**Compilation fails.**

---

Given:

11. String test = "This is a test";

12. String[] tokens = test.split("\s");

13. System.out.println(tokens.length);

What is the result? (Choose one.)

a. 0

b. 1

c. 4

d. Compilation fails.

e. An exception is thrown at runtime.

**Compilation fails.**

---

Given:

12. public class AssertStuff {

14. public static void main(String [] args) {

15. int x= 5;

16. int y= 7;

18. assert (x > y): "stuff";

19. System.out.println("passed");

20. }

21. }

And these command line invocations:

java AssertStuff java -ea AssertStuff

What is the result? (Choose one.)

a. Passed

Stuff

b. Stuff

Passed

c. passed

An AssertionError is thrown with the word "stuff" added to the stack trace.

d. passed

An AssertionError is thrown without the word "stuff" added to the stack trace.

e. passed

An AssertionException is thrown with the word "stuff" added to the stack trace.

f. passed

An AssertionException is thrown without the word "stuff" added to the stack trace.

**passed**

**An AssertionError is thrown with the word "stuff" added to the stack trace.**

---

QN=68 (1516)

Given:

12. public class Test {

13. public enum Dogs {collie, harrier};

14. public static void main(String [] args) {

15. Dogs myDog = Dogs.collie;

16. switch (myDog) {

17. case collie:

18. System.out.print("collie ");

19. case harrier:

20. System.out.print("harrier ");

21. }

22. }

23. }

What is the result? (Choose one.)

a. collie

b. harrier

c. Compilation fails.

d. collie harrier

e. An exception is thrown at runtime.

**collie harrier**

---

Given:

23. Object [] myObjects = {

24. new Integer(12),

25. new String("foo"),

26. new Integer(5),

27. new Boolean(true)

28. };

29. java.util.Array.sort(myObjects);

30. for( int i=0; i<myObjects.length; i++) {

31. System.out.print(myObjects[i].toString());

32. System.out.print(" ");

33. }

What is the result? (Choose one.)

a. Compilation fails due to an error in line 23.

b. Compilation fails due to an error in line 29.

c. A ClassCastException occurs in line 29.

d. A ClassCastException occurs in line 31.

e. The value of all four objects prints in natural order.

**Compilation fails due to an error in line 29.**

---

Given:

55. int []x= {1, 2,3,4, 5};

56. int y[] =x;

57. System.out.println(y[2]);

Which is true? (Choose one.)

a. Line 57 will print the value 2.

b. Line 57 will print the value 3.

c. Compilation will fail because of an error in line 55.

d. Compilation will fail because of an error in line 56.

**Line 57 will print the value 3.**

---

Given:

10. class Line {

11. public static class Point { }

12. }

13.

14. class Triangle {

15. // insert code here

16. }

Which code, inserted at line 15, creates an instance of the Point class defined in Line? (Choose one.)

a. Point p = new Point();

b. Line.Point p = new Line.Point();

c. The Point class cannot be instatiated at line 15.

d. Line l = new Line() ; Point p = new l.Point();

**Line.Point p = new Line.Point();**

---

Given:

class A {

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

public static void main(String[] args) {

try { ((A)new B()).process(); }

catch (Exception e) { System.out.print("Exception "); }

}

}

class B extends A {

public void process() throws RuntimeException {

super.process();

if (true) throw new RuntimeException(); System.out.print("B");

}

}

What is the result? (Choose one.)

a. Exception

b. A Exception

c. A Exception B

d. A B Exception

e. Compilation fails because of an error in line: public void process() throws

RuntimeException

f. Compilation fails because of an error in line: try { ((A)new

B()).process(); }

**A Exception**

---

How can you ensure that multithreaded code does not deadlock? (Choose

one.)

a. Synchronize access to all shared variables.

b. Make sure all threads yield from time to time.

c. Vary the priorities of your threads.

d. There is no single technique that can guarantee non-deadlocking code.

**There is no single technique that can guarantee non-deadlocking code**

---

In the following code fragment, after execution of line 1, sbuf references

an instance of the StringBuffer class. After execution of line 2, sbuf still references the same instance.

1. StringBuffer sbuf = new StringBuffer("FPT");

2. sbuf.append("-University");

a. True

b. False

**True**

---

In the following code fragment, after execution of line 1, sbuf references

an instance of the StringBuffer class. After execution of line 2, sbuf still references the same instance.

1. StringBuffer sbuf = new StringBuffer("FPT");

2. sbuf.insert(3, "-University");

a. True

b. False

**True**

---

QN=97

(76) Is it possible to define a class called Thing so that the following method

can return true under certain circumstances?

boolean weird(Thing s) { Integer x = new Integer(5); return s.equals(x);

}

a. Yes

b. No

**Yes**

---

QN=98

(205) Is it possible to write code that can execute only if the current thread owns

multiple locks?

a. Yes

b. No

**Yes**

---

QN=100

(305) MVC is short call of

a. Model-View-Controller

b. Multiple-View-Controller

c. Metal-View-Controller

**Model-View-Controller**

---

QN=101

(129) public class Test{

public static void main(String[] args){

byte b = 2; byte b1 = 3; b = b \* b1;

System.out.println("b="+b);

}

}

What is the output?

a. b=6

b. No output because of compile error at line: b = b \* b1;

c. No output because of compile error at line: System.out.println("b="+b);

d. No output because of compile error at line: byte b = 2;

e. No output because of compile error at line: byte b = 3;

No output because of compile error at line: **b = b \* b1;**

---

Select correct statement(s) about remote class.(choose one)

a. It must extend java.rmi.server.UnicastRemoteObject.

b. It must implement the remote interface.

c. It is the class whose methods provide services to clients.

d. All the others choices

All the others choices

---

Select correct statements about remote interface. (choose 1)

a. A remote interface is an interface that describes the remotely accessible

methods of a remote object.

b. All remote interfaces must extend java.rmi.Remote.

c. All methods in a remote interface must throw java.rmi.RemoteException

d. The type of a remote reference is a remote interface

e. All the others choices

All the others choices

---

Select INCORRECT statement about deserialize. (choose 1)

a. Any JVM that tries to deserialize an object must have access to that

object's class definition.

b. We use readObject() method of ObjectOutputStream class to deserialize.

c. The readObject method deserializes the next object in the stream and

traverses its references to other objects recursively to deserialize all objects that are reachable from it.

**We use readObject() method of ObjectOutputStream class to deserialize.**

---

Suppose a source file contains a large number of import statements and one

class definition. How do the imports affect the time required to load the class? (Choose one.)

a. Class loading takes no additional time.

b. Class loading takes slightly more time.

c. Class loading takes significantly more time.

**Class loading takes no additional time.**

---

Suppose the type of xarr is an array of XXX, and the type of yarr is an

array of YYY. When is the assignment xarr = yarr; legal? (Choose one.)

a. Sometimes

b. Always

c. Never

d. None of the others choices

**Sometimes**

---

Suppose you are writing a class that will provide custom deserialization.

The class implements java.io.Serializable (not java.io.Externalizable). What access mode should the readObject() method have? (Choose one.)

a. public

b. protected

c. default

d. private

**private**

---

Suppose you are writing a class that will provide custom serialization. The

class implements java.io.Serializable (not java.io.Externalizable). What access mode should the writeObject() method have? (Choose one.)

a. public

b. protected

c. default

d. private

**private**

---

Suppose you want to create a custom thread class by extending

java.lang.Thread in order to provide some special functionality. Which of the following must you do? (Choose one.)

a. Declare that your class implements java.lang.Runnable.

b. Override run().

c. Override start().

d. Make sure that all access to all data is via synchronized methods.

**Override run().**

---

The class is the primary class that has the driver information.

a. DriverManager

b. Driver

c. ODBCDriver

d. None of the others

**DriverManager**

---

The element method alters the contents of a Queue.

a. True

b. False

**False**

Phương thức element thay đổi nội dung của Hàng đợi.

---

There are two classes in Java to enable communication using datagrams

namely.

a. DataPacket and DataSocket

b. DatagramPacket and DatagramSocket

c. DatagramPack and DatagramSock

**DataPacket and DataSocket**

---

URL referring to databases use the form:

a. protocol:subprotocol:datasoursename

b. protocol:datasoursename

c. jdbc:odbc:datasoursename

d. jdbc:datasoursename

**protocol:subprotocol:datasoursename**

---

What does the following code fragment print out at line 9? (Choose one.)

1. FileOutputStream fos = new FileOutputStream("xx");

2. for (byte b=10; b<50; b++)

3. fos.write(b);

4. fos.close();

5. RandomAccessFile raf = new RandomAccessFile("xx", "r");

6. raf.seek(10);

7. int i = raf.read();

8. raf.close()

9. System.out.println("i = " + i);

a. The output is i = 30.

b. The output is i = 20.

c. The output is i = 10.

d. There is no output because the code throws an exception at line 1.

e. There is no output because the code throws an exception at line 5.

**The output is i = 20.**

---

What does the following code print?

public class A

{

static int x;

public static void main(String[] args) { A that1 = new A();

A that2 = new A();

that1.x = 5; that2.x = 1000; x = -1;

System.out.println(x);

}

}

a. 0

b. 5

c. 1000

d. -1

**-1**

---

QN=153

(59) What happens when you try to compile and run the following code?

public class Q {

static String s;

public static void main(String[] args) { System.out.println(">>" + s + "<<");

}

}

a. The code does not compile

b. The code compiles, and prints out >><<

c. The code compiles, and prints out >>null<<

**The code compiles, and prints out >>null<<**

---

What happens when you try to compile and run this application? (Choose one.)

1. import java.util.\*;

2.

3. public class Apple {

4. public static void main(String[] a) {

5. Set<Apple> set = new TreeSet<Apple>();

6. set.add(new Apple());

7. set.add(new Apple());

8. set.add(new Apple());

9. }

10. }

a. Compiler error.

b. An exception is thrown at line 6.

c. An exception is thrown at line 7.

d. An exception is thrown at line 8.

e. No exception is thrown.

**An exception is thrown at line 7.**

---

What is **-50 >> 2**

a. A negative number with very large magnitude.

b. A positive number with very large magnitude.

c. -13

d. -25

e. 13

f. 25

**-13**

---

What is the range of values that can be assigned to a variable of type **byte**?

(Choose one.)

a. Depends on the underlying hardware

b. 0 through 2^8 − 1

c. 0 through 2^16 − 1

d. −2^7 through 2^7 − 1

e. −2^15 through 2^15 − 1

**−2^7 through 2^7 − 1**

---

What is the range of values that can be assigned to a variable of type **short**?

(Choose one.)

a. Depends on the underlying hardware

b. 0 through 2^16 − 1

c. 0 through 2^32 − 1

d. −2^15 through 2^15 − 1

e. −2^31 through 2^31 − 1

**−2^15 through 2^15 − 1**

---

When a **byte is added to a char**, what is the type of the result?

a. byte

b. char

c. int

d. short

e. You can't add a byte to a char.

**int**

---

When a short is added to a float, what is the type of the result?

a. short

b. int

c. float

d. You can't add a short to a float.

**float**

---

When comparing java.io.BufferedWriter to java.io.FileWriter, which

capability exists as a method in only one of the two? (Choose one.)

a. closing the stream

b. flushing the stream

c. writing to the stream

d. marking a location in the stream

e. writing a line separator to the stream

**writing a line separator to the stream**

---

Which of the following are legal **loop definitions**? (Choose one.)

a. while (int a = 0) { /\* whatever \*/ }

b. while (int a == 0) { /\* whatever \*/ }

c. do { /\* whatever \*/ } while (int a = 0)

d. do { /\* whatever \*/ } while (int a == 0)

e. for (int a==0; a<100; a++) { /\* whatever \*/ }

f. None of the above.

**None of the above.**

---

Which of the following are legal? (Choose two.)

a. double d = 1.2d;

b. double d = 1.2D;

c. double d = 1.2d5;

d. double d = 1.2D5;

**double d = 1.2d;**

**double d = 1.2D;**

---

Which of the following are legal? (Choose two.)

a. int a = abcd;

b. int b = ABCD;

c. int c = 0xabcd;

d. int d = 0XABCD;

e. int f = 0ABCD;

**int c = 0xabcd;**

**int d = 0XABCD;**

---

Which of the following are methods of the **java.util.SortedSet** interface?

(Choose one.)

a. first

b. last

c. headSet

d. tailSet

e. subSet

f. All the above

All the above

---

Which of the following are true? (Choose one.)

a. System.out has a println() method.

b. System.out has a format() method.

c. System.err has a println() method.

d. System.err has a format () method.

e. All the above

All the above

---

Which of the following are true? (Choose two.)

a. An enum definition should declare that it extends java.lang.Enum.

b. An enum may be subclassed.

c. An enum may contain public method definitions.

d. An enum may contain private data.

An enum may contain **public method definitions.**

An enum may contain **private data.**

---

Which of the following are valid arguments to the **DataInputStream** constructor? (Choose one.)

a. File

b. FileReader

c. FileInputStream

d. RandomAccessFile

**FileInputStream**

---

Which of the following are valid mode strings for the RandomAccessFile

constructor? (Choose one.)

a. "r"

b. "rw"

c. "rws"

d. "rwd"

e. All the above

All the above

---

Which of the following calls may be made from a non-static synchronized

method? (Choose one.)

a. A call to the same method of the current object.

b. A call to the same method of a different instance of the current class.

c. A call to a different synchronized method of the current object.

d. A call to a static synchronized method of the current class.

e. All the above

All the above

---

Which of the following classes implement java.util.List? (Choose two.)

a. java.util.ArrayList

b. java.util.HashMap

c. java.util.TreeSet

d. java.util.Stack

**java.util.ArrayList**

**java.util.Stack**

---

Which of the following classes implements a **FIFO Queue**? (Choose one.)

a. HashSet

b. LinkedList

c. PriorityQueue

d. CopyOnWriteArraySet

**LinkedList**

---

Which of the following interfaces does **not allow duplicate** objects?

(Choose one.)

a. Queue

b. Set

c. List

**Set**

---

Which of the following is not appropriate situations for **assertions**?

(Câu lệnh Assertion trong Java giúp phát hiện lỗi bằng cách kiểm tra đoạn mã mà người lập trình cho là đúng.)

(Choose one)

a. Preconditions of a public method

b. Postconditions of a public method

c. Preconditions of a private method

d. Postconditions of a private method

Preconditions of a public method

Điều kiện tiên quyết của một public method

---

Which of the following is NOT a valid comment:

a. /\*\*\* comment \*\*/

b. /\*\* comment \*\*/

c. /\* comment

d. // comment

**/\* comment**

---

Which of the following is the most appropriate way to handle invalid

arguments in a public method?

a. Throw java.lang.InvalidArgumentException.

b. Throw java.lang.IllegalArgumentException.

c. Check for argument validity in an assert statement, which throws

AssertionError when the arguments are invalid.

d. Use non-assert code to check for argument validity. If invalid arguments

are detected, explicitly throw AssertionError.

**Throw java.lang.IllegalArgumentException.**

**(illegal)**

---

Which of the following is true? (Choose one.)

a. Readers have methods that can read and return floats and doubles.

b. Readers have methods that can read and return floats.

c. Readers have methods that can read and return doubles.

d. Readers have methods that can read and return ints.

e. None of the above

None of the above

Readers and Writers only deal with character I/O

---

Which of the following is(are) true? (Choose one.)

a. An enum definition may contain the main() method of an application.

b. You can call an enum's toString() method.

c. You can call an enum's wait() method.

d. You can call an enum's notify() method.

e. All the above

**All the above**

A. Định nghĩa enum có thể chứa phương thức main () của một ứng dụng.

B. Bạn có thể gọi một phương thức toString () của enum.

C. Bạn có thể gọi một phương thức enum's wait ().

D. Bạn có thể gọi một phương thức enum's Inform ().

---

Which of the statements below are true? (Choose one.)

a. To change the current working directory, call the setWorkingDirectory()

method of the File class.

b. To change the current working directory, call the cd() method of the File

class.

c. To change the current working directory, call the

changeWorkingDirectory() method of the File class.

d. None of the above

**None of the above**

---

Which one statement is true about the following code fragment?

1. String s = "FPT";

2. StringBuffer s1 = new StringBuffer("FPT");

3. if (s.equals(s1))

4. s1 = null;

5. if (s1.equals(s))

6. s = null;

a. Compilation fails at line 1 because the String constructor must be called

explicitly.

b. Compilation fails at line 3 because s and s1 have different types.

c. Compilation succeeds. During execution, an exception is thrown at line 3.

d. Compilation succeeds. During execution, an exception is thrown at line 5.

e. Compilation succeeds. No exception is thrown during execution.

**Compilation succeeds. No exception is thrown during execution.**

---

Which statement is true about this application? (Choose one.)

1. class StaticStuff

2 {

3. static int x = 10;

4.

5. static { x += 5; }

6.

7. public static void main(String args[])

8. {

9. System.out.println("x = " + x);

10. }

11.

12. static {x /= 5; }

13. }

a. Lines 5 and 12 will not compile because the method names and return

types are missing.

b. Line 12 will not compile because you can only have one static initializer.

c. The code compiles and execution produces the output x = 10.

d. The code compiles and execution produces the output x = 15.

e. The code compiles and execution produces the output x = 3.

**The code compiles and execution produces the output x = 3.**

---

Which statement is true about this code? (Choose one.)

1. class HasStatic

2. {

3. private static int x = 100;

4.

5. public static void main(String args[])

6. {

7. HasStatic hs1 = new HasStatic();

8. hs1.x++;

9. HasStatic hs2 = new HasStatic();

10. hs2.x++;

11. hs1 = new HasStatic();

12. hs1.x++;

13. HasStatic.x++;

14. System.out.println("x = " + x);

15. }

16. }

a. Line 8 will not compile because it is a static reference to a

variable.

b. Line 13 will not compile because it is a static reference to a private

variable.

c. The program compiles and the output is x = 102.

d. The program compiles and the output is x = 103.

e. The program compiles and the output is x = 104.

The program compiles and the output is **x = 104.**

---