

PARCIAL PROGRAMACION III  
20 Preguntas

NOMBRE : \_\_\_\_\_

CLASE : \_\_\_\_\_

FECHA : \_\_\_\_\_

1. Un dato de tipo booleano se caracteriza por tomar únicamente dos valores, seleccione la respuesta correcta:

- |                            |                    |                            |             |
|----------------------------|--------------------|----------------------------|-------------|
| <input type="checkbox"/> A | binary, not binary | <input type="checkbox"/> B | true, false |
| <input type="checkbox"/> C | simple, complex    | <input type="checkbox"/> D | a, b        |

2. En la programación orientada a objetos, un objeto se compone de propiedades y comportamientos donde las propiedades se representan como campos del objeto y el comportamiento se representa como método.

- |                            |       |                            |           |
|----------------------------|-------|----------------------------|-----------|
| <input type="checkbox"/> A | Falso | <input type="checkbox"/> B | Verdadero |
|----------------------------|-------|----------------------------|-----------|

3. En la programación orientada a objetos, ¿elija cuál de las dos opciones siguientes son verdaderas ?

- |                            |                                                |                            |                         |
|----------------------------|------------------------------------------------|----------------------------|-------------------------|
| <input type="checkbox"/> A | Modelado de objetos                            | <input type="checkbox"/> B | Creacion de procesos    |
| <input type="checkbox"/> C | Interacción de objetos sin un orden prescrito. | <input type="checkbox"/> D | Escribiendo Algoritmos. |

4. Puede establecer cualquier número de puntos de interrupción para su programa.

- |                            |       |                            |           |
|----------------------------|-------|----------------------------|-----------|
| <input type="checkbox"/> A | Falso | <input type="checkbox"/> B | Verdadero |
|----------------------------|-------|----------------------------|-----------|

5. Cuando el programa se ejecuta normalmente (cuando no está en modo depuración), ¿qué afirmación es cierta sobre los *Breakpoints*?

- |                            |                                                                                    |                            |                                                                                    |
|----------------------------|------------------------------------------------------------------------------------|----------------------------|------------------------------------------------------------------------------------|
| <input type="checkbox"/> A | Cualquier Breakpoint detendrá la ejecución del programa.                           | <input type="checkbox"/> B | Breakpoints detendrá la ejecución del programa en el ultimo punto de interrupción. |
| <input type="checkbox"/> C | Breakpoints detendrá la ejecución del programa en el primer punto de interrupción. | <input type="checkbox"/> D | Los Breakpoints no tendrán ningún efecto sobre la ejecución del programa.          |

6. ¿Cuál de las siguientes características son compatibles con el IDE NetBeans?

- |                            |                                                                                                                      |                            |                                                                      |
|----------------------------|----------------------------------------------------------------------------------------------------------------------|----------------------------|----------------------------------------------------------------------|
| <input type="checkbox"/> A | Una vez que NetBeans detecta un problema, no le permitirá seguir codificando hasta que el problema esté solucionado. | <input type="checkbox"/> B | NetBeans automáticamente corre código con IA.                        |
| <input type="checkbox"/> C | NetBeans ejecuta automáticamente el programa una vez que todas las llaves del código coinciden.                      | <input type="checkbox"/> D | NetBeans proporciona un atajo para formatear los espacios en blanco. |

7. ¿Cuál es la sintaxis válida para declarar e inicializar una variable String?

- |                            |                   |                            |                      |
|----------------------------|-------------------|----------------------------|----------------------|
| <input type="checkbox"/> A | String x = Java;  | <input type="checkbox"/> B | String "x" = "Java"; |
| <input type="checkbox"/> C | String x= "Java"; | <input type="checkbox"/> D | String "x" = Java;   |

8. ¿Cuáles son operadores matemáticos?

- |                            |      |                            |      |
|----------------------------|------|----------------------------|------|
| <input type="checkbox"/> A | #, @ | <input type="checkbox"/> B | /,   |
| <input type="checkbox"/> C | +, - | <input type="checkbox"/> D | ., } |

9. ¿Qué sentencia produce el resultado: 2, 4, 6, 8, 10?

- |                            |                                                                 |                            |                                                                       |
|----------------------------|-----------------------------------------------------------------|----------------------------|-----------------------------------------------------------------------|
| <input type="checkbox"/> A | for (int i = 0; i < 8; i += 2) {<br>System.out.print(i + " ");  | <input type="checkbox"/> B | for (int i = 1; i < 10; i += 2) {<br>System.out.print(i + " ");<br>}  |
| <input type="checkbox"/> C | for (int i = 0; i < 10; i += 2) {<br>System.out.print(i + " "); | <input type="checkbox"/> D | for (int i = 2; i <= 10; i += 2) {<br>System.out.print(i + " ");<br>} |

10. ¿Cuáles es la asignación válida de a en Java?

- |                            |                |                            |               |
|----------------------------|----------------|----------------------------|---------------|
| <input type="checkbox"/> A | int a; a = 10; | <input type="checkbox"/> B | int a = 10"}  |
| <input type="checkbox"/> C | int a = 10     | <input type="checkbox"/> D | int a = "10"; |

11. ¿Cuál es la salida? `int[] arr = new int[2]; for(int i=0; i < arr.length; i++){  
System.out.print("bye "); }`

- |                            |             |                            |     |
|----------------------------|-------------|----------------------------|-----|
| <input type="checkbox"/> A | bye bye bye | <input type="checkbox"/> B | by  |
| <input type="checkbox"/> C | bye bye     | <input type="checkbox"/> D | bye |

12.

```
.What is the output? public static void main
(String args[]) {
    int x = 100;
    int y = x;
    y++;
    System.out.println("Value of x is " + x);
    System.out.println("Value of y is " + y);
}
```

Identifique los nombres de dos variables utilizadas en el código dado.

A

Valor en x es 100Valor en y es 111

B

Valor en x es 100Valor en y es 101

C

Valor en x es 1000Valor en y es 101

D

Valor en x es 0Valor en y es 101

13. Cual es la salida del siguiente código:

```
public class IndiceIMC {
    public static void main(String args[]) { int masa = 91; // Masa en kilogramos
    double estatura = 1.77; // Estatura en metros
    double IMC = masa/Math.pow(estatura, 2); /* Calcular el índice de masa corporal */
    System.out.println("La persona tiene una masa = " + masa + " kilogramos y estatura = " +
    estatura + " metros");
    if (IMC < 16) {
        System.out.println("La persona tiene delgadez severa.");
    } else if (IMC < 17) {
        System.out.println("La persona tiene delgadez moderada.");
    } else if (IMC < 18.5) {
        System.out.println("La persona tiene delgadez leve.");
    } else if (IMC < 25) {
        System.out.println("La persona tiene peso normal.");
    } else if (IMC < 30) {
        System.out.println("La persona tiene sobrepeso.");
    } else if (IMC < 35) {
        System.out.println("La persona tiene obesidad leve.");
    } else if (IMC < 40) {
        System.out.println("La persona tiene obesidad media.");
    } else {
        System.out.println("La persona tiene obesidad mórbida.");
    }
    }
}
```

A

La persona tiene obesidad leve

B

La persona tiene obesidad mórbida.

C

La persona tiene sobrepeso.

D

La persona tiene obesidad mórbida.

14. Cuantas clases tienen una relacion de herencia que se declaran dentro del siguiente codigo?

```

public abstract class Figura {
    private String tipo;
    public Figura(String tipo) {
        this.tipo = tipo;
    }
    // getters & setters
    public abstract double area();
}

public class Triangulo extends Figura {
    private int base, altura;
    public Triangulo(String tipo, int base, int altura) {
        super(tipo); this.setBase(base); this.setAltura(altura);
    }
    // getters & setters @Override
    public double area() {
        return (double) this.base * this.altura / 2;
    }
}

public class Cuadrado extends Figura {
    private int lado;
    public Cuadrado(String tipo, int lado) {
        super(tipo); this.setLado(lado);
    }
    // getters & setters @Override
    public double area() {
        return (double) this.lado * this.lado;
    }
}

public void polimorfismo(Figura una){
    System.out.println("Tipo: " + una.getTipo()); System.out.println("Area: " + una.area());
}

```

Relación de herencia Relación de herencia

- |                            |                                    |                            |                      |
|----------------------------|------------------------------------|----------------------------|----------------------|
| <input type="checkbox"/> A | Figura, Triangulo y cuadrado       | <input type="checkbox"/> B | Tipo y Area          |
| <input type="checkbox"/> C | Tipo, Figura, Triangulo y cuadrado | <input type="checkbox"/> D | Triangulo y cuadrado |

15.

```

1. package ordenamiento;
2.
3. public class Ordena {
4.     public static void insercion(Integer[] data) {
5.         for (int i = 1; i < data.length; i++) {
6.             Integer aux = data[i];
7.             int j = i - 1;
8.             while (j >= 0 && data[j] > aux) {
9.                 data[j + 1] = data[j];
10.                j--;
11.            }
12.            data[j + 1] = aux;
13.        }
14.    }
15. }

```

¿Segun el codigo presentado identifique el metodo de *ordenación* del arreglo?

A

ordenación burbuja

B

ordenación quick sort

C

ordenación por inserción

D

ordenación merge sort

16.

Es una operación que consiste en disponer un conjunto de datos en algún determinado orden con respecto a uno de los campos de los elementos del conjunto.

A

Algoritmo

B

Concatenación

C

Ordenación

D

inserción

17.

```

int [] numeros=new int[20];
System.out.print("El resultado es");
for (int i=0;i<numeros.length;i+=2){
    numeros[i]=i;
    System.out.println(numeros[i]);
}

```

¿Cuál es el resultado del siguiente código?

A

Los números pares

B

Los números pares incluido el 0 y el 20

C

Los números pares hasta el 18

D

Ninguna de las anteriores

18.

¿Qué es un arreglo?

A

Conjunto de variables del mismo tipo cuyo acceso se realiza por índice

B

Son colecciones de variables de un mismo tipo que se referencian utilizando un nombre común. Un arreglo consta de posiciones de memoria continua.

C

ninguna de las anteriores

D

Datos o valores ordenados para acceso por índice

19.

¿Cuál es la salida del siguiente codigo?

```

char grade = 'A'; switch (grade) { case 'A': System.out.println("En hora buena!"); case 'B': System.out.println("Si estudiaste"); case 'C': System.out.println("Aprobarás"); case 'D': System.out.println("tú primer parcial"); notcase 'F': System.out.println("Failed"); }

```

A

En hora buena! Si estudiaste Aprobarás tú primer parcial Failed

B

En hora buena! Si estudiaste Aprobarás tú primer parcial

C

Si no estudiaste tú primer parcial Failed

D

Failed. Good Work Average Barely Passing Failed

20.

```
// Fragment 1
String inputString =
JOptionPane.showInputDialog("??");
int input = Integer.parseInt(inputString);
input++;

// Fragment 2
int input = Integer.parseInt
(JOptionPane.showInputDialog("??")) + 1;
```

Estos dos fragmentos de código realizan la misma tarea.

☐ A

Verdadero

☐ B

Falso

21.

Consiste en buscar el valor mayor a medida que hace una pasada y, después de completar la pasada, lo pone en la ubicación correcta.

☐ A

ordenamiento por inserción

☐ B

ordenamiento por selección

☐ C

ordenamiento por burbuja

☐ D

ordenamiento por intercambio

27. Given the import statement:  
import java.awt.font.TextLayout;  
which is the package name?

☐ A java.awt.font

☐ B java.awt

☐ C awt.font

☐ D java

28. The classes of the Java class library are organized into packages.

☐ A True

☐ B False

29. The JFrame and JOptionPane classes are in the javax.swing package. Which two will import those classes?

☐ A import javax.swing.J\*;

☐ B import javax.swing.\*;

☐ C import javax.swing;

☐ D import javax.swing.JOptionPane;javax.swing.JFrame;

30. Which statement is true about packages?

☐ A A package contains a group of related classes.

☐ B Packages of the Java class library do not contain related classes.

☐ C A package makes it difficult to locate the related classes.

☐ D A package doesn't contain a group of related classes.

31. Which of the following wild card character is used to import all the classes in a particular package?

☐ A ;

☐ B \*

☐ C !

☐ D ~

32. Import statements are placed above the class definition.

☐ A True

☐ B False

33. Which package is implicitly imported?

☐ A java.lang

☐ B java.io

☐ C java.math

☐ D java.awt

34. Which of the following are the arguments in the following method?

```
Employee emp = new Employee();  
emp.calculateSalary(100000, 3.2, 15);
```

- |                            |                                       |                            |                                   |
|----------------------------|---------------------------------------|----------------------------|-----------------------------------|
| <input type="checkbox"/> A | emp.calculateSalary(100000, 3.2, 15); | <input type="checkbox"/> B | calculateSalary(100000, 3.2, 15); |
| <input type="checkbox"/> C | 100000, 3.2, 15                       | <input type="checkbox"/> D | emp                               |

35. public void simpleInterest(double principal, int roofYears, double interestRate){  
System.out.println("The interest rate is "+interestRate);  
} How many arguments does the following method accept?

- |                            |   |                            |   |
|----------------------------|---|----------------------------|---|
| <input type="checkbox"/> A | 0 | <input type="checkbox"/> B | 1 |
| <input type="checkbox"/> C | 3 | <input type="checkbox"/> D | 2 |

36. Object instantiation is done using what keyword?

- |                            |        |                            |          |
|----------------------------|--------|----------------------------|----------|
| <input type="checkbox"/> A | System | <input type="checkbox"/> B | new      |
| <input type="checkbox"/> C | void   | <input type="checkbox"/> D | instance |

37. Which of the following statements are true?  
(Choose all correct answers)

- |                            |                                                                  |                            |                                                            |
|----------------------------|------------------------------------------------------------------|----------------------------|------------------------------------------------------------|
| <input type="checkbox"/> A | Methods can never be written with more than four parameters.     | <input type="checkbox"/> B | Methods can be written with any number of parameters.      |
| <input type="checkbox"/> C | Methods cannot be written with parameters.                       | <input type="checkbox"/> D | Parameter values can be used within the method code block. |
| <input type="checkbox"/> E | Parameter values can never be used within the method code block. |                            |                                                            |

38. void type methods don't return any values

- |                            |      |                            |       |
|----------------------------|------|----------------------------|-------|
| <input type="checkbox"/> A | True | <input type="checkbox"/> B | False |
|----------------------------|------|----------------------------|-------|

39. You're designing banking software and need to store 10000 customer accounts with information on the accountholder's name, balance, and interest rate. The best approach is store 30000 separate variables in the main method.

- |                            |       |                            |      |
|----------------------------|-------|----------------------------|------|
| <input type="checkbox"/> A | False | <input type="checkbox"/> B | True |
|----------------------------|-------|----------------------------|------|



40. In Java, methods usually hold the properties of an object.

- ☐ A False ☐ B True

41. Which of the following scenarios would be ideal for writing a method?

- ☐ A To group similar data types together ☐ B When you don't find similar lines of code to describe an object's behavior.
- ☐ C For every five to six lines of code. ☐ D When you don't want to repeat similar lines of code to describe an object's behavior.

42. Which two are valid import statements of the Scanner class?

- ☐ A `import java.util.Scanner;` ☐ B `import java.util.*;`
- ☐ C `import java.util;` ☐ D `import java.*;`

43. The import statement consists of two parts.

```
import package.className;
```

One is the package name and the other is the classname.

- ☐ A False ☐ B True

44. Which is a valid way of calling the testMethod in the TestClass? Assume a testInstance has been created.

```
public void testMethod(int x, double y){  
    System.out.println(x/y);  
}
```

- ☐ A `testInstance.testMethod(3.5);` ☐ B `testInstance.testMethod(10, 3.5, 0);`
- ☐ C `testInstance.testMethod(3.5, 10);` ☐ D `testInstance.testMethod(10);`
- ☐ E `testInstance.testMethod(10, 3.5);`

45. Once an object is instantiated, how might its fields and methods be accessed in Java?
- ☐ A Using the double-colon(::) operator      ☐ B Using the dot(.) operator
- ☐ C Using the colon(:) operator      ☐ D Using the comma(,) operator
46. An argument is a value that's passed during a method call
- ☐ A False      ☐ B True
47. Methods allow all instance of a class to share same behaviors.
- ☐ A False      ☐ B True
48. You need to generate random integer values between 0 and 80 (inclusive). Which statement should you use?
- ☐ A `nextInt(0-79);`      ☐ B `nextInt(81);`
- ☐ C `nextInt(80);`      ☐ D `nextInt();`
49. Which values are returned by the method `nextBoolean();`
- ☐ A Either a true or false.      ☐ B Nothing is returned.
- ☐ C Returns the next value.      ☐ D An integer value.
50. Using the Random class requires an import statement.
- ☐ A True      ☐ B False
51. Which class is used to generate random numbers?
- ☐ A Integer      ☐ B Number
- ☐ C Random      ☐ D Double

52. Which two are the features of the Math class?

- ☐ A The Math methods can be invoked without creating an instance of a Math object.
- ☐ B You don't have to worry about the data type returned from a Math method.
- ☐ C Common math functions like square root are taken care of in the language.
- ☐ D Math methods can be invoked with Strings as arguments.

53. What is the package name which contains Math class?

- ☐ A java.net
- ☐ B java.lang
- ☐ C java.awt
- ☐ D java.io

54. What is the approximate value of PI?

- ☐ A 2.718
- ☐ B 0
- ☐ C 3.141
- ☐ D The value varies.

55. Which is NOT true?

- ☐ A Static methods must be of return void.
- ☐ B Static methods can be invoked through the class name.
- ☐ C A class can have multiple static methods.
- ☐ D Static methods can be invoked through an instance of a class

56. 

```
public static void main(String args[]) {  
    String greeting = "Java World!";  
    String w = greeting.replace("a", "A");  
    System.out.println(w);  
}
```

 What is the output?

- ☐ A Java World!
- ☐ B JAvA World!
- ☐ C JAvA World!
- ☐ D JavA World!

57. String objects are immutable.

- ☐ A False
- ☐ B True

58. 

```
public static void main(String args[]) {  
    String greeting = "Java World!";  
    String w = greeting.substring(7, 11);  
    System.out.println(w);  
}
```

 What is the output?
- ☐ A orld! ☐ B rld!
- ☐ C rld ☐ D ld!
59. 

```
public static void main(String args[]) {  
    String alphaNumeric = "Java World!" + 8;  
    System.out.println(alphaNumeric);  
}
```

 What is the output?
- ☐ A Java World! 8 ☐ B Compilation error.
- ☐ C Java World! + 8 ☐ D Java World!8
60. 

```
public static void main(String args[]) {  
    String firstString = "Java";  
    firstString = firstString.concat("World");  
    System.out.println(firstString);  
}
```

 What is the output of the following code?
- ☐ A Java World ☐ B Java
- ☐ C JavaWorld ☐ D World
61. A String is a sequence characters.
- ☐ A False ☐ B True
62. The String class must be imported using java.lang.String;
- ☐ A False ☐ B True
63. The indexOf() method returns the index value of a character in the string.
- ☐ A False ☐ B True
64. The String concat() method concatenates only String data types.
- ☐ A True ☐ B False
65. The replaceFirst() method replaces only the first occurrence of matching character pattern in a string.
- ☐ A False ☐ B True

66. Which method returns the length of a String?

- |                            |             |                            |               |
|----------------------------|-------------|----------------------------|---------------|
| <input type="checkbox"/> A | compareTo() | <input type="checkbox"/> B | charAt()      |
| <input type="checkbox"/> C | length()    | <input type="checkbox"/> D | findLength () |

67. The Math class methods can be called without creating an instance of a Math object.

- |                            |      |                            |       |
|----------------------------|------|----------------------------|-------|
| <input type="checkbox"/> A | True | <input type="checkbox"/> B | False |
|----------------------------|------|----------------------------|-------|

68. You need to generate random integer values in the range 2 through 10. This code fragment will produce the desired result.

```
Random r = new Random();  
r.nextInt(9) + 2;
```

- |                            |       |                            |      |
|----------------------------|-------|----------------------------|------|
| <input type="checkbox"/> A | False | <input type="checkbox"/> B | True |
|----------------------------|-------|----------------------------|------|

69. A break statement causes control to transfer to the end of the switch statement.

- |                            |       |                            |      |
|----------------------------|-------|----------------------------|------|
| <input type="checkbox"/> A | False | <input type="checkbox"/> B | True |
|----------------------------|-------|----------------------------|------|

70. Which two of the following data types can be used in a switch statement?

- |                            |        |                            |         |
|----------------------------|--------|----------------------------|---------|
| <input type="checkbox"/> A | int    | <input type="checkbox"/> B | boolean |
| <input type="checkbox"/> C | String | <input type="checkbox"/> D | float   |

71. What is the output?

```
public static void main(String args[]) {  
    char ch ='c';  
    switch(ch) {  
        case 'a':  
        case 'e':  
        case 'i':  
        case 'o':  
        case 'u':  
            System.out.println("Vowels");  
            break;  
        default:  
            System.out.println("Consonants");  
    }  
}
```

- |                            |                   |                            |            |
|----------------------------|-------------------|----------------------------|------------|
| <input type="checkbox"/> A | Vowels            | <input type="checkbox"/> B | Consonants |
| <input type="checkbox"/> C | Compilation error | <input type="checkbox"/> D | Vowels     |

72.

```
char grade = 'A';
switch (grade) {
    case 'A':
        System.out.println("Congratulations!");    case 'B':
        System.out.println("Good work!");
    case 'C':
        System.out.println("Average");
    case 'D':
        System.out.println("Barely passing");
    case 'F':
        System.out.println("Failed");
}
```

What is the output?

☐ A

Failed

☐ B

A

☐ CCongratulations! Good Work Average  
Barely Passing Failed☐ D

Congratulations

73.

A String comparison with == compares the Strings' locations in memory and not the content of the String.

☐ A

False

☐ B

True

74.

How should Strings be compared?

☐ A

The equals() method

☐ B

==

☐ C

=

☐ D

~=

75.

```
public static void main(String[] args) {
    String name = "Java";
    String language = "Programming";
    String fullName = name + language;
    boolean test = fullName.equals(name + language);
    System.out.println(test);
}
```

What is the output?

☐ A

False

☐ B

JavaProgramming

☐ C

Java Programming

☐ D

True





1) Un dato de tipo booleano se cancela por tomar únicamente dos valores, seleccione la respuesta correcta:

→ true, false

B) Rta por que los booleanos reciben verdaderos o falso

2) En la programación orientada, un objeto se compone de propiedades y comportamientos donde las propiedades se representan como campo del objeto y el comportamiento se representa como método.

→ verdadero

B) porque los métodos se representan una acción y las propiedades representan una cantidad que representa en la programación orientada a objetos

3) En la programación orientada a objetos, un objeto se compone de propiedades y comportamientos donde las propiedades se representan como campos del objeto y el comportamiento se representa como método.

→ modelo de objetos

A) Porque es un paradigma de programación, esto es, un modelo o un estilo de programación que proporciona unas guías acerca de como trabajar en él y está basado en los conceptos de clases y objetos.

4) Para establecer cualquier número de puntos de interrupción para su programa.

→ Verdadero

B) Porque uno tiene n puntos que tenga dependiendo de las necesidades

5) Cuando el programa se ejecuta normalmente (cuando no está en modo depuración), ¿qué afirmación es cierta sobre los breakpoints?

→ Los breakpoints no tendrán ningún efecto sobre la ejecución del programa.

D) porque no está en modo de depuración

6) ¿Cuál de las siguientes características son compatibles con el IDE NetBeans?

→ Una vez que NetBeans detecta un problema, no le permitirá seguir codificando hasta que el problema esté solucionado

A) porque no te deja compilar sino lo solucionas, sin lo solucionas te deja continuar

7) ¿Cuál es la sintaxis válida para declarar e inicializar una variable String?

C) String x = "JAVA"; Porque la variable se define sin comillas ya que es su nombre, como es un String solo toma cadenas de texto, y los textos van entre comillas en caso de querer declararlos.



8) ¿cuáles son operadores matemáticos?

DÍA MES AÑO

C) +, - Porque esta se usa para operadores y para hacer valores que aumenten gradualmente, y las otras son para definir variables

9) ¿qué sentencia produce el resultado: 2, 4, 6, 8, 10?

D) `for (int i=2; i<=10; i+=2) { System.out.print(i + " "); }` porque este for inicia en 2, y muestra i por cada pasada, así que muestra el número 2, crece en aumentos de 2 hasta llegar a 10, al llegar a 10, muestra el número y termina.

10) ¿cuáles es la asignación válida de a en Java?

C) `int a=10` Porque como en el punto 7, es claro con este caso a) es solo el nombre por lo que, no va en comillas, como es tipo int solo toma números pequeños, por lo que se pone el número sin comillas ni nada a la hora de declarar

11) ¿cuál es la salida? `int [] arr = new int[2]; for (int i=0; i<arr.length; i++) { System.out.print("bye"); }`

C) bye bye Porque es un array con tamaño 2, el for repite hasta que sea menor al tamaño del arreglo o sea el número 2, lo imprime 2 veces porque i inicia desde 0

12) Código → What is the output? `public static void main`

```
{String args []} {  
    int x=100;  
    int y=x;  
    y++;  
    System.out.println("value of x is "+x);  
    System.out.println("value of y is "+y);  
}
```

Identifique los nombre de dos variables utilizadas en el código dado.

B) Valor en X es 100 Valor en Y es 101 Porque X es igual a 100 y toma el valor de X por lo que es también igual a 100 luego ponen `y++` lo que significa que a y se le suma 1, por lo que queda `X=100` y `Y=101`

13) ¿cuál es la salida del siguiente código?

Código → `public class IndiceIMC {  
 public static void main (String args []) {  
 int masa=91; //masa en Kilogramos  
 double estatura=1.77 //Estatura en metros  
 double IMC=masa/Math.pow(estatura, 2); //Calcular el índice de la masa corporal  
 System.out.println("la persona tiene una masa=" + masa + " kilogramos y estatura=" +  
 estatura + " metros");  
 if (IMC<16) {`

nessan



DIA MES AÑO

```
System.out.println("La persona tiene delgadez severa.");
} else if (IMC < 17) {
    System.out.println("La persona tiene delgadez moderada.");
} else if (IMC < 18.5) {
    System.out.println("La persona tiene delgadez leve.");
} else if (IMC < 25) {
    System.out.println("La persona tiene peso normal.");
} else if (IMC < 30) {
    System.out.println("La persona tiene Sobrepeso.");
} else if (IMC < 35) {
    System.out.println("La persona tiene obesidad leve.");
} else if (IMC < 40) {
    System.out.println("La persona tiene obesidad media.");
} else {
    System.out.println("La persona tiene obesidad mórbida.");
}
}
}
```

c) La persona tiene Sobrepeso. Porque la persona tiene Sobrepeso, nos dan muestra a 11 estatura 1,77 y sacas el IMC que nos da 29,09 según el código Si el IMC es menor a 30 el output será "Persona tiene sobrepeso".

14) Cuantas clases tiene una relación de herencia que se declaran dentro del siguiente código?

```
public abstract class Figura {
    private String tipo;
    public Figura(String tipo) {
        this.tipo = tipo;
    }
    // getters & setters
    public abstract double area();
}
public class Triangulo extends Figura {
    private int base, altura;
    public Triangulo(String tipo, int base, int altura) {
        super(tipo); this.setAltura(altura);
    }
    // getters & setters @override
    public double area() {
        return (double) this.base * this.altura / 2;
    }
}
public class Cuadrado extends Figura {
    private int lado;
    public Cuadrado(String tipo, int lado) {
        super(tipo); this.setLado(lado);
    }
}
```



3  
//getters y setters @override

```
public double area() {  
    return (double) this.lado * this.lado;  
}
```

```
public void Polimorfismo (Figura una) {
```

```
    System.out.println("Tipo: " + una.getTipo()); System.out.println("Area: " + una.area());
```

```
}  
Relación de herencia Relación de herencia.
```

A) Figura, triángulo y cuadrado porque estas dos (2) clases tienen relación de herencia ya que a la hora de declararlas tienen un `extends` Figuras lo que significa que va a heredar algún atributo de la clase figura por lo que la figura cuadrado y triángulo tiene relación de herencia, figuras a heredar a los más dos.

15) 1 `Package` ordenamiento

2

3 `public class` ordena {

4 `public void` static inserción (Integer [] data) {

5 `for` (int i = 0; i < data.length; i++) {

6 `Integer` aux = data[i];

7 `int` j = i - 1;

8 `while` (j >= 0 && data[j] > aux) {

9 data[j+1] = data[j];

10 j++;

11 }

12 data[j+1] = aux;

13 }

14 }

15 }

¿Según el código presentado identifique el método de ordenación de arreglo?

C) ordenación por inserción porque ya que toma el segundo valor del arreglo también hace un intercambio en un `while` y en un `for`.

16) Es una operación que consiste en disponer un conjunto de datos en algún determinado orden con respecto a uno de los campos de los elementos del conjunto.

C) ordenación porque forman un union y los cambia con arcos, ordenando dependiendo de lo que usará al usarlos lo asignará una posición para que quede ordenado (en este caso de menor a mayor).

A) `int [] numeros = new int [20];`

`System.out.println("El resultado es");`

`for` (int i = 0; i < numeros.length; i++) {

numeros[i] = i;



DIA MES AÑO `System.out.println(numeros[i]);`  
`}`

¿Cuál es el resultado del código siguiente?

C) Los números pares hasta el 18 por que el for toma un valor 100 como hicimos en 2, as array se le asigna el valor de 1, por lo que su valor inicial es 0, y por lo que aumenta 2 en 2, cada valor que se le ira asignando al array, y se ira mostrando, hasta llegar a 18, ya que si sube a 20 se termina el ciclo

18) ¿Que es un arreglo?

B) Son colecciones de variables de un mismo tipo que se referencian utilizando un nombre común. Un arreglo consta de posiciones de memoria continua. Porque ya que un arreglo son muchos guardan en posiciones de memorias que son llamadas por un indice, lo mismo que dice el punto b)

19) ¿Cuál es la salida del siguiente código?

```
char grade = 'A';
switch (grade) {
    case 'A': System.out.println("En hora buena!");
    case 'B': System.out.println("Si estudiaste");
    case 'C': System.out.println("Aprobaras");
    case 'D': System.out.println("tu primer parcial");
    default: System.out.println("Failed");
}
```

B) En hora buena! Si estudiaste Aprobaras tu primer parcial. Porque al no haber un break, al cumplirse un caso (en este ejemplo A), se va el mostrar ese caso y todo lo que haya es el switch ya que no hay un break que lo detenga, el not case no se muestra ya que esta más denotado, si fuera o un default dasla cosas distintas

20) // Fragment 1

```
String InputString =
OptionPane.showInputDialog("??");
int input = Integer.parseInt(InputString);
input++;
```

// Fragment 2

```
int input = Integer.parseInt(
OptionPane.showInputDialog("??")) + 1;
```

Estos fragmentos de código realizan la misma tarea.

A) Verdadero por que en general ambos hacen lo mismo, convertir un string a un int, parece que cambia mucho pero el resultado final es el mismo ya que al int en ambos casos se le incrementa en 1.

21) Consiste en buscar el valor mayor a medida que hace una pasada y, después de completar la pasada, lo pone en la ubicación correcta.

B) Ordenamiento por selección porque la selección se busca el valor mínimo o máximo, se verifica en todo el arreglo no se ubica, hasta organizando todo.