

JAVA

1. The standard API for accessing databases in Java is:
 - a. JPA/Hibernate
 - b. JDBC
 - c. ODBC
2. Valid argument for catch clause:
 - A. Throwable
 - B. Exception but NOT including RuntimeException
 - C. CheckedException
 - D. RuntimeException
 - E. Exception
3. Asume que este switch acepta datos de tipo entero. ¿Qué tipo de dato **no** se podría colocar en el espacio en blanco?

```
class App{  
    public static void main(String[] args) {  
        ----- dayOfWeek= 1;  
        switch (dayOfWeek) {  
            case 1:  
                System.out.println("Monday");  
            case 2:  
                System.out.println("Tuesday");  
            case 3:  
                System.out.println("Other day");  
            break;  
        }  
    }  
}
```

- A. int
 - B. long
 - C. char
 - D. None of the above
4. Elige todas las excepciones no verificadas
 - a. IllegalArgumentException
 - b. NumberFormatException
 - c. IOException
 - d. NullPointerException
 - e. Exception
 - f. ArrayIndexOutOfBoundsException

5. ¿Cuál es el resultado?

```
public class Main{  
    public static void main(String[] args) {  
        int[] ii = null;  
        for (int xx : ii) {  
            System.out.println(xx);  
        }  
    }  
}
```

```

    }
}

```

a. `java.lang.NullPointerException`

b. `null`

6. ¿Cuál es la salida?

```

public class Movie {
    public static void main(String[] args) {
        String movie = "Thriller";
        switch (movie) {
            case "Thriller":
                System.out.println("Movie Thriller");
            case "Comedy":
                System.out.println("Movie Comedy");
            case "Romance":
                System.out.println("Movie Romance");
                break;
            case "Action":
                System.out.println("Movie Action");
        }
    }
}

```

a. Movie Thriller

b. Movie Thriller Movie Comedy

c. Movie Thriller Movie Comedy Movie Romance

7. ¿Cuál es el resultado?

```

public class strings {
    public static void main(String[] args) {
        String s = "ABCD";
        s.trim();
        s.toUpperCase();
        s+= " 123";
        System.out.println(s.length());
    }
}

```

8. Usando `StringBuilder`, elige las opciones que den el mismo resultado que la concatenación original. (Elige todas las opciones aplicables).

```

public class Cadena {
    public static void main(String[] args) {
        String name = "Joe";
        int age = 31;
        String result = "My name is " + name + ", my age is " + age;
        System.out.println(result);

        StringBuilder sb = new StringBuilder();
    }
}

```

a. `sb.append("My name is " + name + ", my age is " + age);`

- b. `sb.insert("My name is " + name).append(", my age is " + age);`
- c. `sb.insert("My name is").insert(name).insert(", my age is ").insert(age);`
- d. `sb.append("My name is ").append(name).append(", my age is ").append(age);`

9. ¿Qué afirmación es correcta sobre el siguiente código?

```
public class MyStuff {
    String name;
    MyStuff(String n) {name = n;}

    public static void main(String[] args) {
        MyStuff m1 = new MyStuff("guitar");
        MyStuff m2 = new MyStuff("tv");
        System.out.println(m2.equals(m1));
    }

    public boolean equals(Object o) {
        MyStuff m = (MyStuff) o;
        if (m.name != null) {return true;}
        return false;
    }
}
```

- a. The output is true and MyStuff fulfills the Object.equals() contract
- b. The output is false and MyStuff fulfills the Object.equals() contract
- c. The output is true and MyStuff does NOT fulfill the Object.equals() contract.
- d. The output is false and MyStuff does NOT fulfill the Object.equals() contract.

10. Which two statments are true?

- A. An interface CANNOT be extended by another interface.
- B. An abstract class can be extended by a concrete class.
- C. An abstract class CANNOT be extended by an abstract class.
- D. An interface can be extended by an abstract class.
- E. An abstract class can implement an interface.
- F. An abstract class can be extended by an interface.

11. ¿Cuál es el resultado?

```
public class DoWhile {
    public static void main(String[] args) {
        int ii = 2;
        do {
            System.out.println(ii);
        } while (--ii);
    }
}
```

- a. 2 1 0
- b. 1 0
- c. null
- d. An infinite loop
- e. Compilation fails

12. ¿Cuál es el resultado?

```
public class Bucle {  
    public static void main(String[] args) {  
        int[] at = { 1, 2 };  
        for (int x : at) {  
            System.out.println(x + ", ");  
            if (x < at.length) {  
                break;  
            }  
        }  
    }  
}
```

- a. 1, 2,
- b. No compila
- c. 1,
- d. 1, 2

13.

```
Int a = 10; int b = 37; int z = 0; int w = 0;  
If (a==b) {z=3; } else if (a>b) {z=6;}  
w = 10 * z;
```

- a. 30
- b. 0**
- c. 60

14. ¿Cuál es el resultado?

```
class Control {  
    public static void main(String[] args) {  
        int i = 42;  
        String s = (i < 40) ? "Greater than" : false;  
        System.out.println(s);  
    }  
}
```

- a. Error de compilación**
- b. Greater then
- c. Greater then false
- d. false

15. ¿El siguiente código compila?

```
public class estructura {  
    public static void main(String[] args) {  
        int num = 1;  
        switch (num) {  
            case 1:  
            case 2:  
                System.out.println("Caso 2");  
            case 3:  
                System.out.println("Caso 3");  
        }  
    }  
}
```

```

        break;
    }
}

```

16. ¿Cuál de los siguientes códigos podría estar presente en un método que no retorna ningún valor? (Elige dos opciones)

- a. return null;
- b. return void;
- c. return;
- d. Omitir el uso de la palabra clave return

17. What two keywords may precede the word 'class' in a class declaration?

- a. Volatil
- b. Public
- c. Local
- d. Synchronized
- e. Static

18. Which three are bad practices?

- a. Checking for an IOException and ensuring that the program can recover if one occurs.
- b. Checking for ArrayIndexOutOfBoundsException and ensuring that the program can recover if one occurs.
- c. Checking for FileNotFoundException to inform a user that a filename entered is not valid.
- d. Checking for Error and, if necessary, restarting the program to ensure that users are unaware problems.
- e. Checking for ArrayIndexOutOfBoundsException when iterating through an array to determine when all elements have been visited.

19. _____ occurs when a subclass redefines a method from its superclass, while _____ happens when multiple methods in the same class share the same name but differ in their parameters.

- a. Overloading / Overriding
- b. Overriding / Overloading
- c. Inheriting / Overriding
- d. Overloading / Inheriting

20. What changes will make this code compile?

```

class X{
    X(){ }
    private void one(){ }
}
public class Y extends X{
    Y(){ }
    private void two(){ one(); }
    public static void main(String[] args){
        new Y().two();
    }
}

```

}

- A. Adding the public modifier to the declaration of class X.
- B. Adding the protected modifier to the class X() constructor.
- C. Changing the private modifier on the declaration of the one() method to protected.
- D. Removing the Y() constructor.
- E. Removing the private modifier from the two() method

21. ¿Cuáles de las siguientes opciones son instanciaciones e inicializaciones válidas de un arreglo multidimensional? Elige dos

- A. `int[][] array2D = {{0, 1, 2, 4}{5, 6}};`
- B. `int[][] array2D = new int[][2];`
`array2D[0][0] = 1;`
`array2D[0][1] = 2;`
`array2D[1][0] = 3;`
`array2D[1][1] = 4;`
`int[] array3D = new int[2][2][2];`
- C. `int[][][] array3D = {{{0, 1}, {2, 3}, {4, 5}}};`
`int[] array = {0, 1};`
- D. `array3D[0][0] = array;`
`array3D[0][1] = array;`
`array3D[1][0] = array;`
`array3D[1][1] = array;`

22. ¿Cuál es la salida?

```
public class Main {  
    public static void main(String[] args) {  
        int x = 2;  
        if(x==2) System.out.println("A");  
        else System.out.println("B");  
        else System.out.println("C");  
    }  
}
```

23. Concatenación de strings e ints

```
public static void main(String[] args) {  
    System.out.println("Result:" + 3 + 5);  
    System.out.println("Result:" + (3 + 5));  
}
```

- A. Result: 8 Result: 8
- B. Result: 35 Result: 8
- C. Result: 8 Result: 35
- D. Result: 35 Result: 35

APX

1. Cuáles son los dos tipos de step que existen Batch APX
 - a. Creator, Processor
 - b. Processor, Task
 - c. Chunk, Reader
 - d. Chunk, Task
2. De la lista siguiente, indique cuales son las invocaciones restringidas en la Arquitectura Batch:
 - a. Invocación WebServices, servicios REST
 - b. Acceso a recursos externos vía HTTP
 - c. Uso de Pre y Post Acciones
 - d. Escritura en Registro de Operaciones
 - e. Escritura en el Diario Electrónico
3. ¿Qué nivel de profundidad es permitido al realizar invocaciones de librería a librería?
 - a. Profundidades a 3 niveles: Transacción > Librería > Librería > Librería
 - b. Profundidades a 5 niveles: Transacción > Librería > Librería > Librería > Librería
 - c. Profundidades a 7 niveles: Transacción > Librería > Librería > Librería > Librería > Librería
 - d. Profundidades a 9 niveles: Transacción > Librería > Librería > Librería > Librería > Librería > Librería > Librería
4. Se debe evitar en una transacción o biblioteca, dentro de su lógica, invocar más de _____ librerías a la vez
 - a. 7
 - b. 15
 - c. 5
 - d. 9
5. Está permitido que la aplicación modifique o informe una variable de entorno de la JVM
 - a. Verdadero
 - b. Falso
6. El acceso a los datos debe realizarse utilizando las utilidades que Arquitectura proporciona desde el asistente de biblioteca. En el caso de bases de datos relacionales, ¿Qué utilidad se debe usar?
 - a. APIConector
 - b. Elastic
 - c. No conozco la respuesta
 - d. Datio
 - e. JDBC

7. Este patrón suele representarse en una librería que contiene un solo método “execute” con parámetros que actúan como filtros y que engloba diversas funcionalidades dispares en función de sus parámetros de entrada
 - a. Contenedor Mágico
 - b. CRUD
 - c. Paginacion
 - d. Blob
8. La información utilizada en la lógica de negocio de la transacción debe incluirse en _____.
 - a. Los archivos locales.
 - b. Las variables globales.
 - c. Los parámetros de entrada.
 - d. Las variables locales.
9. No es una característica de blob

Tips para el aprobar el examen

1. Repasar y estudiar todo el material de las guías de Java realizadas con Miguel durante la academia.
2. Si no entiendes algún ejercicio o código de las guías, te recomiendo usar [Cursor](#), una plataforma de desarrollo asistida por IA que se integra con editores de código. Esta herramienta te permite hacer preguntas específicas sobre el código y también ofrece sugerencias para mejorarlo.
3. En caso de tener complicaciones con respecto algún tema de Java en particular, recomiendo leer el libro de [OCA JAVA SE 8](#), ya que sintetiza bien la información y todo está muy bien explicado, con ejemplos claros y fáciles de entender.
4. Repasar y estudiar las guías de APX que se encuentran en la carpeta de drive compartida por Catalina.
5. El examen consta de 40 preguntas que deben responderse en 120 minutos, por lo que es fundamental gestionar el tiempo de manera eficiente. Aunque lo ideal es dedicar un máximo de 3 minutos por pregunta, cada una presenta desafíos diferentes. Algunas requieren un análisis más profundo, mientras que otras se resuelven rápidamente. Mi estrategia consistió en revisar y analizar cuidadosamente cada pregunta, evaluando todas las opciones de respuesta, incluso aquellas que parecían algo obvias, y descartando las menos probables, ya que tanto las preguntas como las respuestas pueden ser engañosas.

En general, tomé el tiempo necesario para analizar, asegurándome de no superar los 3 minutos por pregunta, aunque algunas resultaron tan sencillas que las respondí de un minuto a dos. Es crucial no apresurarse y tomar el tiempo necesario para reflexionar, pero también evitar quedar atrapado (a) en una sola pregunta.