



Integrador Java + MySQL

Total de puntos **83/100** ?

Correo electrónico *

jhonpaco@frba.utn.edu.ar

0 de 0 puntos

DNI *

Escribe tu cédula de identidad **sin puntos, ni espacios**

93913603

Nombre y Apellido *

Jhon Daniel Olmedo Paco

Correo Electrónico *

El mismo correo con el que Ingresas a tu Aula Virtual (con el que estás registrado en Egg)

jhonpaco@frba.utn.edu.ar

Java Introducción

11 de 11 puntos



✓ Cual de los siguientes métodos NO es parte de la clase String?

- ☐ .toUpperCase()
- ☒ .stringLength()
- ☐ .isEmpty()
- ☐ concat()

✓ Tipos de datos en Java: *

- ☐ int, double, boolean, String, void, null
- ☐ Cadena, entero, real, lógico
- ☒ byte, short, int, long, float, double, boolean, char, String

✓ Qué hace el siguiente código? *

```
1  int aux = 0;  
2  aux++;
```

- ☐ Agrega una nueva variable aux
- ☐ Multiplica por 2 a la variable aux
- ☒ Incrementa en 1 a la variable aux

✓ **Cuál es la salida del siguiente código? ***



```
1  int num1 = 6;  
2  int num2 = 8;  
3  
4  System.out.println(num1==num2);
```

- ☐ 6
- ☒ False
- ☐ 8
- ☐ True

✓ **El método main puede estar o no en un proyecto Java funcional.**

- ☒ Falso
- ☐ Verdadero

✓ **Las variables siempre deben especificar de qué tipo son al ser declaradas.**

- ☒ Verdadero
- ☐ Falso

POO

22 de 24 puntos

✓ **Cuáles son los modificadores de acceso? ***

- ☐ Public, static, abstract
- ☒ Public, private, protected
- ☐ Public, notPublic

✓ **Cómo accedo a los atributos de un objeto? ***

- ☐ Con el símbolo _ luego del nombre del objeto
- ☒ Con el símbolo . luego del nombre del objeto
- ☐ Con el símbolo ! luego del nombre del objeto

✓ **Cuál es la salida de este código? ***

```
1 Persona p1 = new Persona();
2 Persona p2 = new Persona();
3
4 p1.setNombre("Albus Dumbledore");
5 p2.setNombre(p1.getNombre());
6
7 System.out.println(p1.equals(p2));
8 System.out.println(p1.getNombre().equals(p2.getNombre()));
```

- ☒ False, True
- ☐ False, False
- ☐ True, True
- ☐ True, False

✓ **Cuál de estos servicios trabaja de la manera correcta? ***

```
1 public void conocerNombre(Persona x){
2     System.out.println("La persona se llama " + x.getNombre());
3 }
4
5 public void conocerNombre(String nombre){
6     System.out.println("La persona se llama " + nombre);
7 }
```

- ☐ El que obtiene el nombre desde un String
- ☒ El que obtiene el nombre a través de un objeto Persona

✓ **Cuál es la diferencia entre clases y objetos? ***

- ☒ Las clases son “moldes” y los objetos son modelados a partir de esos moldes, también llamados “instancias”
- ☐ Los objetos son abstracciones del mundo real, las clases son de lunes a jueves
- ☐ Las clases y los objetos son los mismo

✓ **¿De que se encarga Clase servicio o control? ***

- ☒ Se encarga de crear y modificar objetos, teniendo cada clase de entidad una clase control.
- ☐ Se encarga de controlar que el programa no consuma memoria al instanciar objeto
- ☐ Se encarga de recibir los datos que ingresa el usuario por teclado

✓ **Qué son los atributos? ***

- ☒ Variables que guardan información sobre el objeto
- ☐ Funciones que el objeto puede realizar
- ☐ Métodos de consulta o modificación de un objeto

✗ **Qué es el encapsulamiento? ***

- ☐ Hace referencia a los métodos constructores con y sin parámetros
- ☒ Hace referencia a separar el código en “cápsulas” para mejor lectura del mismo.
- ☐ Hace referencia a crear métodos para obtener y modificar información de un objeto manteniendo oculto lo que hace un objeto por dentro.

Respuesta correcta

- ☒ Hace referencia a crear métodos para obtener y modificar información de un objeto manteniendo oculto lo que hace un objeto por dentro.

✓ **Qué hace el siguiente código? ***

```
1  Persona p1 = new Persona();
```

- ☒ Instancia un objeto de tipo Persona llamado p1
- ☐ Duplica un objeto persona
- ☐ Crea una nueva clase llamada Persona

✓ **Qué significa POO? ***

- ☐ Programación, Observación y Orientación
- ☒ Programación Orientada a Objetos
- ☐ Programación Objetiva Orientada

✓ **Qué es un constructor? ***

- ☐ Un método void
- ☐ Un método que nos define qué atributos tiene la clase
- ☒ Es un método que nos permite instanciar un objeto de la clase donde se encuentra

Colecciones

12 de 17 puntos

✓ **Un framework es un marco de trabajo el cual contiene un conjunto estandarizado de conceptos, prácticas y criterios para hacer frente a un tipo de problemática particular y resolver nuevos problemas de índole similar.**

- ☒ Verdadero
- ☐ Falso

✓ **Las listas son colecciones de objetos ordenados por posición donde los elementos pueden repetirse.**

- ☐ Falso
- ☒ Verdadero

✓ **Los mapas son colecciones de Llave/Valor. Las llaves son únicas pero los valores sí pueden repetirse.**

☐ Falso

☒ Verdadero

✓ **Los conjuntos o Sets pueden contener elementos repetidos, simplemente se duplica el valor**

☒ Falso

☐ Verdadero

✗ **Qué puede almacenar una colección? ***

☐ Objetos de la misma clase

☒ Objetos mezclados sin importar de qué clase son

☐ Datos primitivos, como char e int

Respuesta correcta

☒ Objetos de la misma clase



✗ **Cómo se define el tamaño de una colección? ***

- ☒ Con el método size()
- ☐ Se define al declarar la colección
- ☐ Las colecciones son dinámicas, su tamaño depende de la cantidad de objetos que le agreguemos o quitemos

Respuesta correcta

- ☒ Las colecciones son dinámicas, su tamaño depende de la cantidad de objetos que le agreguemos o quitemos

✓ **Cuál de las siguientes herramientas nos facilita recorrer una lista?**

- ☐ Un Switch con condicionales en cada caso
- ☒ Bucle ForEach
- ☐ Bucle For

✓ **El Java Collection Framework es: ***

- ☐ Una base de datos
- ☐ Un conjunto de objetos
- ☒ Un conjunto de clases e interfaces

Relaciones entre Clases

12 de 12 puntos

✓ **La composición es un tipo de relación donde un objeto complejo es conformado por objetos más pequeños.**

- ☐ Falso
- ☒ Verdadero

✓ **Las relaciones pueden ser ***

- ☒ Uno a Uno, Cero a Uno, Uno a Muchos y Cero a Muchos
- ☐ Uno a Uno y Muchos a Muchos
- ☐ Uno a Uno solamente

✓ **En una relación de agregación, un objeto depende del otro, no pueden existir individualmente.**

- ☒ Falso
- ☐ Verdadero

✓ **Las clases no actúan aisladas entre sí, al contrario, las clases están relacionadas unas con otras.**

- ☐ Falso
- ☒ Verdadero



✓ Las relaciones son siempre unidireccionales *

- ☒ Falso
- ☐ Verdadero

✓ Las relaciones entre clases realmente significan que una clase contiene una referencia a un objeto u objetos, de la otra clase en la forma de un atributo.

- ☐ Falso
- ☒ Verdadero

Herencia

15 de 15 puntos

✓ Una clase hija puede heredar de muchas clases padres. *

- ☐ Verdadero
- ☒ Falso

✓ Una clase padre puede heredar a muchas clases hijas. *

- ☒ Verdadero
- ☐ Falso



✓ **Qué hace la anotación @Override? ***

- ☐ Permite que la clase hija elimine métodos que considere innecesarios
- ☒ Permite que la clase hija haga funcionar un método de manera diferente a la clase padre
- ☐ Desbloquea nuevos métodos disponibles en Java

✓ **La Sub Clase hereda atributos y métodos. ***

- ☒ Verdadero
- ☐ Falso

✓ **La Sub Clase hereda el constructor ***

- ☐ Verdadero
- ☒ Falso

✓ **Cuándo usamos la palabra reservada super? ***

- ☐ Cuando queremos acceder a los métodos, atributos y constructores de la clase inferior
- ☒ Cuando queremos acceder a los métodos, atributos y constructores de la clase superior
- ☐ Cuando queremos referirnos a una clase que está muy bien hecha



✓ **Qué es la Super Clase? ***

- ☐ Una clase con muchos métodos
- ☒ Como se llama a la clase superior(Padre o Madre)
- ☐ Una clase con superpoderes

✓ **En el siguiente código ¿cuál es la clase padre? ***

```
1 public class Persona extends SerVivo{  
2  
3     String nombre;  
4  
5 }
```

- ☒ SerVivo
- ☐ public class
- ☐ Persona
- ☐ Java.util

✓ **Para qué sirve el modificador Protected? ***

- ☐ Para proteger el código para que no sea borrado
- ☐ Para que suene a que sabemos más que los demás
- ☒ Para que los atributos sean accesibles sólo por las clases que heredan sin usar getters ni setters



✓ **Cuál es el resultado del siguiente código? siguiendo el ejemplo que venimos viendo**

```
1 Persona p1 = new Persona();  
2 String nombre = null;  
3  
4 try {  
5     if (!nombre.equals(null)) {  
6         p1.setNombre(nombre);  
7     }  
8 } catch (Exception e) {  
9     System.out.println("El nombre no puede ser nulo");  
10 }
```

- ☒ Se imprime por consola "El nombre no puede ser nulo"
- ☐ El nombre de p1 queda vacío
- ☐ El nombre de p1 es null

✓ **Un objeto Exception contiene información sobre un error ocurrido.**

- ☒ Verdadero
- ☐ Falso

✓ **Una excepción es un evento que ocurre durante la ejecución de un programa que interrumpe el flujo normal de las instrucciones del programa.**

- ☒ Verdadero
- ☐ Falso



✗ Sólo se puede usar 1 bloque catch por bloque try. *

☒ Verdadero

☐ Falso

Respuesta correcta

☒ Falso

✓ Para aclarar que un método puede lanzar un error se coloca la palabra "extends".

☐ Verdadero

☒ Falso

✗ Si una excepción se produce en el bloque catch, puede atraparse en el bloque try y manejarla.

☐ Falso

☒ Verdadero

Respuesta correcta

☒ Falso

✓ Cuando se lanza una excepción no hay nada más que hacer, termina el programa.

☐ Verdadero

☒ Falso

MySQL

2 de 8 puntos

Resultados del ejercicio de los candados

✗ Candado D *☐ Posición 1☒ Posición 2☐ Posición 3☐ Posición 4

Respuesta correcta

☒ Posición 4**✗ Candado A ***☐ Clave: 16832☒ Clave: 15935☐ Clave: 14043☐ Clave: 13539

Respuesta correcta

☒ Clave: 14043

✗ **Candado B ***

- ☐ Posición 1
- ☐ Posición 2
- ☐ Posición 3
- ☒ Posición 4

Respuesta correcta

- ☒ Posición 3

✗ **Candado C ***

- ☐ Clave: 631
- ☒ Clave: 963
- ☐ Clave: 145
- ☐ Clave: 601

Respuesta correcta

- ☒ Clave: 631



✗ **Candado C ***

- ☐ Posición 1
- ☐ Posición 2
- ☒ Posición 3
- ☐ Posición 4

Respuesta correcta

- ☒ Posición 1

✗ **Candado B ***

- ☐ Clave: 4380
- ☐ Clave: 4830
- ☐ Clave: 3480
- ☒ Clave: 3830

Respuesta correcta

- ☒ Clave: 3480

✓ **Candado D ***

- ☒ Clave: 191
- ☐ Clave: 1063
- ☐ Clave: 153
- ☐ Clave: 101

✓ **Candado A ***

- ☐ Posición 1
- ☒ Posición 2
- ☐ Posición 3
- ☐ Posición 4

El formulario se creó en Egg Cooperation.

Google Formularios











