| Integrador Java + MySQL |
|--|
| Total de puntos 100/100 ? |
| |
| |
| Correo electrónico * |
| kevinfloresg1998@gmail.com |
| |
| 0 de 0 puntos |
| DNI * |
| Escribe tu cédula de identidad sin puntos, ni espacios |
| 40931037 |
| |
| |
| Nombre y Apellido * |
| KEVIN FLORES |
| |
| |
| Correo Electrónico * El mismo correo con el que lagracas a tu Aula Virtual (con el que estás registrado en Egg) |
| El mismo correo con el que Ingresas a tu Aula Virtual (con el que estás registrado en Egg) |
| kevinfloresg1998@gmail.com |
| |

Java Introducción

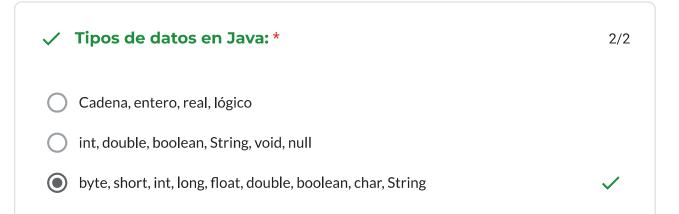
11 de 11 puntos



✓ Las variables siempre deben especificar de qué tipo son al ser *1/1 declaradas.



Falso







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

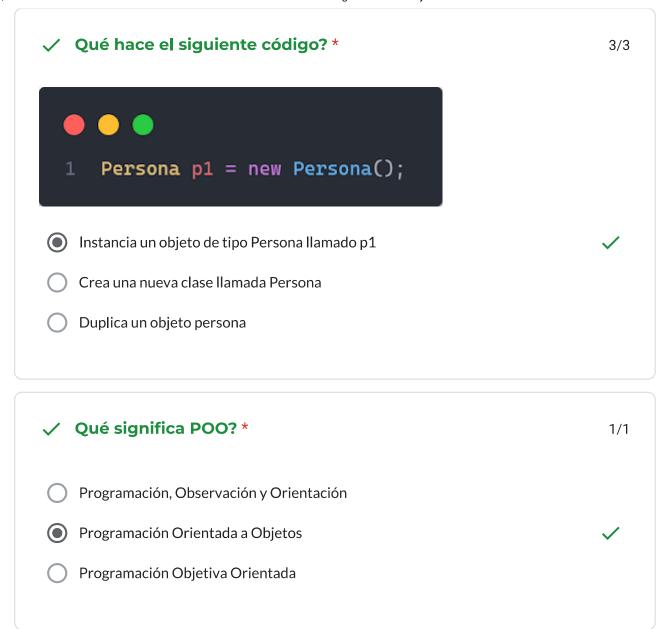
| ✓ Cual de los siguientes métodos NO es parte de la clase String? | *2/2 |
|---|----------|
| stringLength() | ✓ |
| isEmpty() | |
| .toUpperCase() | |
| oncat() | |
| | |
| | |

✓ El método main puede estar o no en un proyecto Java *2/2 funcional. Verdadero Falso

```
✓ Cuál es la salida del siguiente código? *
                                                            2/2
  1 int num1 = 6;
    int num2 = 8;
     System.out.println(num1==num2);
   8
   False
   True
```

POO 24 de 24 puntos

| ✓ ¿De que se encarga Clase servicio o control? * 2/2 |
|--|
| Se encarga de controlar que el programa no consuma memoria al instanciar objetos. |
| Se encarga de crear y modificar objetos, teniendo cada clase de entidad una clase control. |
| Se encarga de recibir los datos que ingresa el usuario por teclado |
| ✓ Cómo accedo a los atributos de un objeto? * 2/2 |
| 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 diferencia entre clases y objetos? * 2/2 |
| 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 |
| Cas clases y los objetos son los mismo |



```
✓ Cuál es la salida de este código? *
                                                                               3/3
     Persona p1 = new Persona();
     Persona p2 = new Persona();
    p1.setNombre("Albus Dumbledore");
     p2.setNombre(p1.getNombre());
    System.out.println(p1.equals(p2));
     System.out.println(p1.getNombre().equals(p2.getNombre()));
     False, False
    True, False
     True, True
    False, True
    Cuáles son los modificadores de acceso? *
                                                                               2/2
     Public, static, abstract
    Public, private, protected
     Public, notPublic
 / Qué es el encapsulamiento? *
                                                                               2/2
     Hace referencia a separar el código en "cápsulas" para mejor lectura del mismo.
    Hace referencia a los métodos constructores con y sin parámetros
     Hace referencia a crear métodos para obtener y modificar información de un
     objeto, manteniendo oculto lo que hace un objeto por dentro.
```

✓ Cuál de estos servicios trabaja de la manera correcta? * 3/3 public void conocerNombre(Persona x){ System.out.println("La persona se llama " + x.getNombre()); public void conocerNombre(String nombre){ System.out.println("La persona se llama " + nombre); El que obtiene el nombre desde un String El que obtiene el nombre a través de un objeto Persona / Qué es un constructor? * 2/2 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 🗸 Un método void Qué son los atributos? * 2/2 Variables que guardan información sobre el objeto Métodos de consulta o modificación de un objeto Funciones que el objeto puede realizar **Colecciones**

17 de 17 puntos

| ~ | Cuál de las siguientes herramientas nos facilita recorrer una lista? | * 2/2 |
|----------|--|--------------|
| 0 | Bucle For | |
| • | Bucle ForEach | ✓ |
| 0 | Un Switch con condicionales en cada caso | |
| ~ | Las listas son colecciones de objetos ordenados por posición donde los elementos pueden repetirse. | *2/2 |
| • | Veradadero | ✓ |
| 0 | Falso | |
| / | 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. | *2/2 |
| • | Verdadero | ✓ |
| 0 | Falso | |
| ~ | Los mapas son colecciones de Llave/Valor. Las llaves son únicas pero los valores sí pueden repetirse. | *2/2 |
| • | Verdadero | ✓ |
| 0 | Falso | |
| | | |

| ~ | Los conjuntos o Sets pueden contener elementos repetidos, simplemente se duplica el valor | *2/2 |
|----------|---|----------|
| 0 | Verdadero | |
| • | Falso | ~ |
| / | Cómo se define el tamaño de una colección? * | 3/3 |
| 0 | Con el método size() | |
| 0 | Se define al declarar la colección | |
| • | Las colecciones son dinámicas, su tamaño depende de la cantidad de objetos que le agreguemos o quitemos | ✓ |
| ✓ | Qué puede almacenar una colección? * | 2/2 |
| 0 | Datos primitivos, como char e int | |
| 0 | Objetos mezclados sin importar de qué clase son | |
| • | Objetos de la misma clase | ~ |
| / | El Java Collection Framework es: * | 2/2 |
| 0 | Una base de datos | |
| 0 | Un conjunto de objetos | |
| • | Un conjunto de clases e interfac | ✓ |
| | | |

Relaciones entre Clases

!

12 de 12 puntos

| Las clases no actúan aisladas entre sí, al contrario, las clases están relacionadas unas con otras. | *2/2 |
|---|----------|
| Verdadero | ✓ |
| ○ Falso | |
| En una relación de agregación, un objeto depende del otro, no pueden existir individualmente. | *2/2 |
| Verdadero | |
| Falso | ✓ |
| Las relaciones entre clases realmente significan que una clas contiene una referencia a un objeto u objetos, de la otra clase en la forma de un atributo. | |
| ○ Falso | |
| Verdadero | ✓ |
| ✓ Las relaciones son siempre unidireccionales * | 2/2 |
| Falso | ✓ |
| Verdadero | |
| | |

| La composición es un tipo de relación donde un objeto complejo es conformado por objetos más pequeños. | * 2/2 |
|---|--------------|
| Falso | |
| Verdadero | ~ |
| ✓ Las relaciones pueden ser * | 2/2 |
| Uno a Uno y Muchos a Muchos | |
| Uno a Uno solamente | |
| Uno a Uno, Cero a Uno, Uno a Muchos y Cero a Muchos | ✓ |
| Herencia 15 de 1 | 5 puntos |
| ✓ Qué es la Super Clase? * | 2/2 |
| Una clase con superpoderes | |
| Una clase con muchos métodos | |
| Como se llama a la clase superior(Padre o Madre) | ✓ |
| ✓ Una clase hija puede heredar de muchas clases padres. * | 1/1 |
| Verdadero | |
| Falso | ✓ |

| ✓ Para qué sirve el modificador Protected? * | 2/2 |
|--|----------|
| Para que los atributos sean accesibles sólo por las clases que heredan sin usar getters ni setters | ~ |
| Para que suene a que sabemos más que los demás | |
| Para proteger el código para que no sea borrado | |
| ✓ La Sub Clase hereda atributos y métodos. * | 1/1 |
| Verdadero | ✓ |
| ○ Falso | |
| ✓ La Sub Clase hereda el constructor * | 1/1 |
| Falso | ✓ |
| Verdadero | |
| ✓ Una clase padre puede heredar a muchas clases hijas. * | 2/2 |
| ○ Falso | |
| Verdadero | ✓ |

| ✓ | Qué hace la anotación @Override? * | 1/1 |
|----------|---|-----|
| • | Permite que la clase hija haga funcionar un método de manera diferente a la clase padre | / |
| 0 | Permite que la clase hija elimine métodos que considere innecesarios | |
| 0 | Desbloquea nuevos métodos disponibles en Java | |
| | | |

```
En el siguiente código ¿cuál es la clase padre? *
                                                                   3/3
  public class Persona extends SerVivo{
      String nombre;
  }
public class
Persona
SerVivo
Java.util
```

```
✓ Cuándo usamos la palabra reservada super? *
                                                                                 2/2
     Cuando queremos acceder a los métodos, atributos y constructores de la clase
     inferior
     Cuando queremos referirnos a una clase que está muy bien hecha
     Cuando queremos acceder a los métodos, atributos y constructores de la clase
     superior
```

Manejo de Excepciones

13 de 13 puntos

| Una excepción es un evento que ocurre durante la ejecución de un programa que interrumpe el flujo normal de las instrucciones del programa. | *1/1 |
|---|----------|
| ○ Falso | |
| Verdadero | ✓ |
| | |



✓ Cuál es el resultado del siguiente código? siguiendo el *****4/4 ejemplo que venimos viendo

```
Persona p1 = new Persona();
  String nombre = null;
     if (!nombre.equals(null)) {
          pl.setNombre(nombre);
8 } catch (Exception e) {
       System.out.println("El nombre no puede ser nulo");
```

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

| / | Cuando se lanza una excepción no hay nada más que h termina el programa. | acer, *1/1 |
|----------|---|-------------------|
| • |) Falso | ~ |
| C | Verdadero | |
| ~ | Un objeto Exception contiene información sobre un errocurrido. | or *2/2 |
| • | Verdadero | ✓ |
| C |) Falso | |
| ~ | Para aclarar que un método puede lanzar un error se co palabra "extends". | oloca la *1/1 |
| • |) Falso | ~ |
| C | Verdadero | |
| ~ | Si una excepción se produce en el bloque catch, puede atraparse en el bloque try y manejarla. | *2/2 |
| • |) Falso | ~ |
| C | Verdadero | |
| My: | SQL | 8 de 8 puntos |
| Resul | tados del ejercicio de los candados | |

| ✓ Candado B * | 1/1 |
|----------------|----------|
| O Clave: 3830 | |
| Clave: 4380 | |
| O Clave: 3480 | ✓ |
| Clave: 4830 | |
| ✓ Candado A * | 1/1 |
| Clave: 16832 | |
| Clave: 15935 | |
| O Clave: 14043 | ✓ |
| O Clave: 13539 | |
| ✓ Candado D * | 1/1 |
| O Posición 1 | |
| O Posición 2 | |
| O Posición 3 | |
| Posición 4 | ~ |

| ✓ Candado C * | 1/1 |
|---------------|----------|
| O Clave: 631 | ✓ |
| O Clave: 963 | |
| O Clave: 145 | |
| O Clave: 601 | |
| ✓ Candado C * | 1/1 |
| Posición 1 | ✓ |
| O Posición 2 | |
| O Posición 3 | |
| O Posición 4 | |
| | |
| ✓ Candado D * | 1/1 |
| O Clave: 191 | ✓ |
| O Clave: 1063 | |
| O Clave: 153 | |
| Clave: 101 | |
| | |

| ✓ Candado B * | 1/1 |
|---------------------------|----------|
| O Posición 1 | |
| O Posición 2 | |
| Posición 3 | ~ |
| O Posición 4 | |
| | |
| | |
| ✓ Candado A * | 1/1 |
| ✓ Candado A * Posición 1 | 1/1 |
| | 1/1 |
| O Posición 1 | 1/1 |
| Posición 1 Posición 2 | 1/1 |

El formulario se creó en Egg Cooperation.

Google Formularios