

Certamen 1, Programación II  
Escuela de Ingeniería Civil Informática - Universidad de Valparaíso.  
Prof. Eduardo Godoy  
Octubre 2, 2017

---

Nombre:

Rut:

**Instrucciones:**

- El puntaje máximo del certamen es 100%, siendo el 60% el mínimo requerido para aprobar.
- El certamen es **individual**. Cualquier intento de copia, será sancionado con nota **1,0**.

1. *30pts.* De las siguientes afirmaciones, encierre en un círculo la o las alternativas correctas. Tiempo estimado para esta pregunta es 1:30 hora. Cada respuesta correcta vale *1pto.*

- |  |  |
|--|--|
| i. <i>4pts</i> El paradigma de la orientación a objeto se basa en:<br>(a) El uso de objetos y su interacción.<br>(b) El uso de clases como categorización de instancias.<br>(c) El uso de un enfoque imperativo.<br>(d) El uso de un enfoque reusable.<br>(e) Principios como herencia y abstracción.                                  | (e) Un tipo de dato de la clase.   |
| ii. <i>3pts</i> En cuanto a la programación orientada a objeto:<br>(a) Se apoya en el paradigma estructural.<br>(b) Se apoya en el paradigma orientación a objetos.<br>(c) Divide el programa en pequeñas unidades de código.<br>(d) Proporciona herramientas para modelar el mundo real.<br>(e) Es una herramienta del lenguaje JAVA. | viii. <i>3pts.</i> El principio de ocultamiento de información:<br>(a) Es una técnica que protege el estado de una entidad.<br>(b) Es indispensable en el paradigma de orientación a objeto.<br>(c) En Java, se logra utilizando los modificadores de acceso.<br>(d) Es encapsular el conocimiento de una entidad.<br>(e) Ninguna de las anteriores. |
| iii. <i>1pts.</i> Una clase es:<br>(a) Un arreglo de objetos.<br>(b) Una absorción del mundo real.<br>(c) Una herramienta de programación.<br>(d) Un puntero a memoria.<br>(e) La instancia de un objeto.  | ix. <i>2pts.</i> Un ejemplo de ocultamiento de información es:<br>(a) Un java Bean.<br>(b) La palabra reservada final.<br>(c) Una clase que posee sus atributos privados e implementa métodos públicos para acceder a ellos.<br>(d) un método privado.<br>(e) Una clase común.   |
| iv. <i>2pts.</i> Una clase es:<br>(a) Un arreglo de objetos.<br>(b) Una abstracción del mundo real.<br>(c) Un Compilador.<br>(d) Un punto a memoria.<br>(e) La instancia de un objeto.   | x. <i>2pts.</i> Respecto a la herencia, las clases:<br>(a) Heredan sólo los métodos privados.<br>(b) Heredan el comportamiento completo de la clase padre.<br>(c) Heredan sólo el comportamiento que se desea utilizar.<br>(d) En Java, se implementan con la palabra implements.<br>(e) En Java, se implementan con la palabra extends.             |
| v. <i>3pts.</i> Respecto a una clase:<br>(a) Puede o no tener atributos.<br>(b) Puede o no tener métodos.<br>(c) Puede o no tener constructor.<br>(d) Puede o no tener un nombre.<br>(e) Puede o no tener un tipo.   | xi. <i>3pts.</i> El polimorfismo:<br>(a) El mismo nombre implementa distintas funcionalidades.<br>(b) Una funcionalidad implementada con distintos nombres.<br>(c) Es una característica de JAVA.<br>(d) Es una característica de POO.<br>(e) Un ejemplo es el símbolo %.  |
| vi. <i>2pts.</i> Un objeto es:<br>(a) Un tipo de dato de la clase.<br>(b) La instancia de una clase.<br>(c) Una abstracción del mundo real.<br>(d) Un sub-conjunto de atributos y métodos de la clase.<br>(e) Siempre estático.  | xii. <i>2pts.</i> El método <i>main</i> :<br>(a) Puede no ser void.<br>(b) Debe ser static.<br>(c) Puede no llevar argumentos de entrada.<br>(d) Debe retornar un valor.<br>(e) Debe incluirse en un programa.   |
| vii. <i>1pts.</i> Los objetos se comunican a través de:<br>(a) Mensajes.<br>(b) La instancia de una clase.<br>(c) Datos.<br>(d) Atributos.   | xiii. <i>2pts.</i> Respecto al manejo de excepciones:<br>(a) finally nunca se ejecuta.<br>(b) Debe ser static.<br>(c) Se compone de las palabras reservadas try, catch y finally.<br>(d) La zona del código encerrada dentro de una try/catch se llama zona segura.<br>(e) catch nunca se ejecuta.   |

2. *70pts*. Cree un programa que permita realizar una correcta gestion de bodega de un antiguo almacen, para esto se requiere implementar lo siguiente.

- (*10pts*) Crear La clase **Producto** que tenga los siguientes atributos:
  - (a) código: de tipo entero, cuya responsabilidad es identificar al producto.
  - (b) nombre: de tipo string, que contiene al nombre del producto.
  - (c) stock: de tipo entero, dato encargado de manejar la cantidad de productos del mismo nombre que actualmente se tiene en el almacen.
  - (d) precio: precio del producto con iva incluido.

Además debe contener los métodos get y set asociados a cada atributo.

- (*5pts*) Crear la Clase **GestionBodega** que posea como atributo: un arreglo de tipo dinámico de tipo **Producto** y llamado **listaProductos** que permita almacenar los productos que creará el usuario.

La clase **GestionBodega** debe poseer los siguientes métodos

- (a) (*15pts*) Codificar el método **crearProducto** que permita crear un producto específico y asignarle un stock inicial. Se debe considerar que al asignarle el precio al producto, el sistema debe agregarle de forma automática el iva (19%) sobre el precio ingresado. luego de esto se proceder a guardar el resultado en el atributo precio de la clase producto. Para finalizar agregandolo al arreglo listaProductos. La acción de crear productos debe repetirse mientras el usuario lo desee.
- (b) (*10pts*) Codificar el método **listarStock** que permita visualizar el código, nombre y stock de cada producto.
- (c) (*20pts*) Generar el método **venderProductos**. La responsabilidad de este método es mostrar un lista de productos con su código, nombre y precio cuyo stock asociado sea mayor a 0. Luego el usuario irá seleccionando en base a al código cada producto que le vende a un cliente, el sistema debe ser capaz de ir descontando 1 al inventario de cada producto y a la vez debe ir sumando los precios de cada producto seleccionado acumulandolo en una variable llamada precioVenta). La venta del productos se repetirá hasta que el usuario lo desee. Una vez finalizada la venta de productos, el sistema debe desplegar el precio total de la venta realizada (precioVenta).
- (*5pts*) Implementar la clase **GestionBodegaImpl** que posea el método **main** encargada de inicializar el sistema creando una instancia de la clase **GestionBodega** y utilizar sus métodos asociados.
- (*5pts*) Resuelve el problema utilizando el paradigma de Orientación a Objetos.

- Enviar respuesta a eduardo.gl@gmail.com con asunto: Certame 1 - Lenguajes de Programación desde correop institucional UV.
- Las clases deben estar comprimidas en zip y el nombre del archivo resultante debe ser bajo el siguiente formato: nombre\_apellido\_rut.zip
- Tiempo estimado de resolución de pregunta dos es 24 horas.
- El proyecto debe estar entregado al día siguiente, martes 3 de Octubre antes de las 21:01 hrs, si el tiempo de entrega es excedido la entrega quedará inválida, obtenido 0 pto en ese ítem.

¿Cómo será evaluado en la pregunta 2?			
Tópico	Logrado	Medianamente logrado	No logrado
Construir Clase Producto	10pts Crea la clase atómica Producto con sus atributos/métodos.	5pts Crea la clase atómica Producto con algunos atributo o algunos métodos get y set requeridos en el problema.	0pts No crea la clase atómica Producto.
Crea clase GestionBodega y su atributo	5pts Define e implementa correctamente la clase GestionBodega con su atributos Atributo: listaProducto.	2pts Define clase pero no atributo requerido para el problema.	0pts No define ni los atributos ni métodos.
En GestionBodega contruir método crearProducto	15pts Crea de crearProducto de forma correcta.	7pts Define el método crearProducto en otra clase o no cumple con la totalidad de lo requerido.	0pts No define el método crearProducto o no cumple con lo requerido.
En GestionBodega contruir método listarStock	10pts Crea de listarStock de forma correcta.	5pts Define el método listarStock en otra clase o no cumple con la totalidad de lo requerido.	0pts No define el método listarStock o no cumple con lo requerido.
En GestionBodega contruir método venderProductos	20pts Crea de venderProductos de forma correcta.	10pts Define el método venderProductos en otra clase o no cumple con la totalidad de lo requerido.	0pts No define el método venderProductos o no cumple con lo requerido.
Construir clase GestionBodegaImpl y su método asociado	5pts Define e implementa correctamente la clase GestionBodega con su atributos Atributo: listaProducto.	2pts Define clase pero su método no cumple con lo requerido en el problema.	0pts No define ni los atributos ni métodos.
Paradigma Orientación a Objetos	5pts Resuelve el problema utilizando el POO.	2pts Utiliza parte del POO para resolver el problema.	0pts No utiliza el POO para dar solución al problema.
Total máximo puntaje pregunta 2	70pts	31pts	0pts