

ICI 202 Programación 2 - Certamen 01

Prof. Ismael Figueroa, Eduardo Godoy

4 de mayo de 2018

Nombre:

Rut:

Paralelo:

Instrucciones:

Puntaje:

Nota:

- El puntaje máximo es 100 puntos.
- Tiempo máximo: 120 minutos.
- El trabajo es **individual**. Cualquier intento de copia, será sancionado según dicta el reglamento de la carrera.
- Se debe subir al aula virtual un archivo comprimido con el siguiente formato
Certamen₀₁ < NombreApellidoEstudiante_rut_paralelo > .zip, dentro deben estar los códigos fuentes requeridos.

Resultados de aprendizaje a evaluar:

1. Características de Orientación a objetos.
2. Modelamiento y su implementación.

Contenido: Este certamen evalúa los siguientes temas:

Tema	Puntajes	
	Total	Obtenido
Problema 2: Programación en lenguaje java, sintaxis básicas e instancias.	40 pts.	
Problema 3: Sentencias de control, arreglos dinámicos y paradigma OO.	60 pts.	

1. Problema 1

1. **40pts.** Implemente un programa en lenguaje java que permita instanciar objeto del tipo persona e identificar si cumple o no con la mayoría de edad, para este objetivo se debe implementar lo siguiente:

1. Una clase **Persona** que posea:

- a) (10pts) Características: *nombre(s)*, *apellido*, *edad*. Considerar el tipo de dato idóneo para cada uno de los atributos y definir el valor por omisión acorde a cada tipo.
- b) (15pts) Definir su comportamiento asociado: **esMayorDeEdad()** para determinar lo requerido. Determinando si la persona es o no mayor de edad mostrando mensaje por pantalla mediante un mensaje en pantalla.

c) (15pts) Crear Clase Main.java que posea el método main(String [] args) encargada de:

- 1) (5pts) Inicializar el programa creando la instancia de la clase MedicionPersona. Además se deben generar la instancia de los siguientes objetos.
- 2) (5pts) Primer Objeto:
 - nombre = Pedro
 - apellido=Figuerola
 - edad = 16;
- 3) (5pts) Segundo Objeto:
 - nombre = Juan.
 - apellido=Godoy.
 - edad = 20;

¿Cómo será evaluado en este trabajo?			
Ítem	Logrado	Suficiente	No Logrado
Creación de Clase Persona.java con atributos.	Aplica de forma correcta 10pts	Aplica parcialmente con menos de 2 errores 5pts	Aplica de forma incorrecta con 3 errores o más 0pts
Implementación de método (comportamientos) esMayorDeEdad asociados y correcto funcionamiento.	Aplica de forma correcta 15pts	Aplica parcialmente 2 errores o menos 8pts	Aplica de forma incorrecta con 3 errores o más 0pts
Implementar clase instanciadora Main con método main y crea instancias de objetos y llamadas a métodos requeridos .	Aplica de forma correcta 15pts	Aplica parcialmente con menos de 2 errores 8pts	Aplica de forma incorrecta con 3 errores o más 0pts
Total de la sección	40pts	21pts	0pts

Nota: En caso de que el ítem no esté presente, tiene ponderación cero.

2. Problema 2

1. **60pts.** Cree un programa que permita realizar una correcta gestion de bodega de un antiguo almacen, para esto se requiere 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.

- (10pts) 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.
- (20pts) La clase GestionBodega debe poseer los siguientes métodos:
 - a) (10pts) 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 procederá 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.
- (10pts) Crear Clase Main.java que posea el método main(String [] args) encargada de inicializar el programa creando la instancia de la clase GestionBodega y utilizar sus métodos asociados de la siguiente forma:
 - a) crearProducto() ->se crea primer producto.
 - b) crearProducto() ->se crea segundo producto.
 - c) listarStock() ->se listan ambos productos.
- (10pts) Resuelve el problema utilizando el paradigma de Orientación a Objetos.

¿Cómo será evaluado en este trabajo?			
Ítem	Logrado	Suficiente	No Logrado
Creación de Clase Producto.java con atributos.	Aplica de forma correcta 10pts	Aplica parcialmente con menos de 2 errores 5pts	Aplica de forma incorrecta con 3 errores o más 0pts
Creación de Clase GestionBodega.java con arreglo dinámico implementado.	Aplica de forma correcta 10pts	Aplica parcialmente con menos de 2 errores 5pts	Aplica de forma incorrecta con 3 errores o más 0pts
Implementación de método (comportamientos) crearProducto asociados y correcto funcionamiento.	Aplica de forma correcta 10pts	Aplica parcialmente 2 errores o menos 5pts	Aplica de forma incorrecta con 3 errores o más 0pts
Implementación de método (comportamientos) listarStock asociados y correcto funcionamiento.	Aplica de forma correcta 10pts	Aplica parcialmente 2 errores o menos 5pts	Aplica de forma incorrecta con 3 errores o más 0pts
Implementar clase instanciadora Main con método main y crea instancias de objetos y llamadas a métodos requeridos .	Aplica de forma correcta 10pts	Aplica parcialmente con menos de 2 errores 5pts	Aplica de forma incorrecta con 3 errores o más 0pts
Cumple con el paradigma orientación a objetos y compilación correcta	Aplica de forma correcta 10pts	Aplica parcialmente con deficiencias en el paradigma 5 pto	Aplica de forma incorrecta con x errores o más 0pts
Total de la sección	60pts	30pts	0pts

Nota: En caso de que el ítem no esté presente, tiene ponderación cero.