

INC 214 Programación 2 - Certamen 2

Profesores: **Ismael Figueroa, Eduardo Godoy**

6 de agosto de 2018

Nombre:

Rut:

Paralelo:

Puntaje:

Nota:

Instrucciones:

- 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:
Certamen_01_<NombreApellidoEstudiante_rut_paralelo>.zip. Dentro deben estar todos los archivos de código fuente requeridos.

Resultados de aprendizaje a evaluar:

1. Desarrollar algoritmos simples usando programación orientada a objetos y estructurada, en el lenguaje Java

Contenido: Este certamen evalúa los siguientes temas:

Tema	Puntajes	
	Total	Obtenido
Problema 1: Programación en lenguaje Java, sintaxis básicas e instancias.	40 pts.	
Problema 2: Sentencias de control, arreglos dinámicos y paradigma OO.	60 pts.	

1. Problema 1

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

1. **100pts.** Como Ingeniero Civil en Informática, se le ha solicitado desarrollar un sistema de registro de personas pertenecientes a la Escuela. Para eso, usted debe:

a) Pedir al usuario:

1) Los datos personales (DNI, Nombre, Apellidos, Edad, Dirección, etc.)

2) Tipo de persona (Académico, Alumno, Funcionario).

- Si es Académico, se debe registrar su grado académico (licenciado, magister o Dr.).

- Si es Alumno, se debe registrar las asignaturas que se encuentra cursando en el semestre (pueden ser más de una).

- Si es Funcionario, se debe registrar su oficio o profesión.

- La información solicitada, debe ser registrada en un archivo de texto plano, que deberá crearse, si no existe y agregar información al final si ya fue creado.

3) **35pts.** Implementar las clases requeridas utilizando el concepto de Herencia.

b) **35pts.** Implementar una clase GestionPersona que provea lo siguiente:

■ **5pts.** Un Arreglo Dinámico de tipo ArrayList como atributo.

■ **10pts.** Implementar el método crear que permita ingresar los datos requeridos para cada persona cumpliendo con el esquema de herencia agregando cada persona al arreglo.

■ **10pts.** Implementar el método guardar que recorra el Arreglo Dinámico y que tenga la responsabilidad de guardar el registro dentro del archivo identificándose claramente el tipo de persona guardada.

■ **10pts.** Implementar el método listar que lea el archivo y liste todos sus registros.

c) **30pts.** Implementar la clase PersonaMain que permita ejecutar el programa y los métodos asociados.

→ **Recomendación:** Utilice el nombre FuenteDatos para el archivo generado.

→ **Formato de entrega:** Debe subir los archivos fuente ***.java** en un comprimido **.zip** al aula virtual, con el siguiente formato:

- *ApellidoPaternoNombreC2.zip*

¿Cómo será evaluado en este trabajo?			
Ítem	Logrado	Suficiente	No Logrado
Creación de Clase Persona.java con atributos.	Aplica de forma correcta: 5pts	Aplica parcialmente con menos de 2 errores: 3pts	Aplica de forma incorrecta con 3 errores o más: 0pts
Creación de Constructor con atributos.	Aplica de forma correcta: 10pts	Aplica parcialmente con menos de errores: 5pts	Aplica de forma incorrecta con 3 errores o más: 0pts
Implementación de método esMayorDeEdad 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 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*: identificar de manera única al producto.
 - b) *nombre*: el nombre del producto.
 - c) *stock*: maneja la cantidad de productos del mismo código que actualmente se tiene en el almacen.
 - d) *precio*: precio del producto con iva incluido.

Debe crear un constructor explícito que reciba todos los atributos y los inicialice correctamente.

Todos los atributos deben ser privados. Para manipular estos valores debe implementar los metodos setX y getX, donde X es el nombre de cada atributo.

- (5pts) Crear la clase **GestionBodega** que posea el atributo listadoProductos, un arreglo dinámico que almacena objetos de tipo Producto y llamado **listaProductos**.
- La clase GestionBodega debe poseer los siguientes métodos:
 - a) (10pts) crearProducto que permita crear un producto específico y asignarle un stock inicial. Este método recibe como parámetro el precio sin IVA (19%), por lo que debe calcularse el precio actualizado a ingresar en el producto que se está creando. Además, este método debe agregar el producto creado al listado de productos de GestionBodega
 - b) (15 pts) ingresarProductosConsola que lee desde el teclado la información para la creación de productos, con datos ingresados por el usuario. La acción de crear productos debe repetirse mientras el usuario lo desee. Puede usar alguna convención de su elección para terminar la lectura. PISTA: en la implementación debe invocar a crearProducto.
 - c) (10pts) listarStock que permita visualizar el código, nombre y stock de cada producto.
 - d) (10pts) valorInventario que calcula el valor del inventario actual, mostrando la suma de precio por stock, y muestra el total con y sin IVA.
- (10pts) Crear la clase Main.java con un método main que se encargue de inicializar el programa creando la instancia de la clase GestionBodega para utilizar sus métodos asociados de la siguiente forma:
 - a) ingresarProductosConsola(), para crear 1 o más productos por parte del usuario
 - b) listarStock(), para mostrar los productos creados
 - c) valorInventario(), para mostrar los precios de los productos listados

¿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 5pts	Aplica parcialmente con menos de 2 errores 3pts	Aplica de forma incorrecta con 3 errores o más 0pts
Implementación de método crearProducto 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 ingresarProductosConsola y correcto funcionamiento.	Aplica de forma correcta 10pts	Aplica parcialmente 2 errores o menos 8pts	Aplica de forma incorrecta con 3 errores o más 0pts
Implementación de método listarStock y correcto funcionamiento.	Aplica de forma correcta 5pts	Aplica parcialmente 2 errores o menos 2pts	Aplica de forma incorrecta con 3 errores o más 0pts
Implementación de método valorInventario 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
Total de la sección	60pts	30pts	0pts

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