**INSTITUTO DE EDUCACIÓN SUPERIOR CIBERTEC**

**DIRECCIÓN ACADÉMICA**

**CARRERA PROFESIONALES**

**CURSO :** Algoritmos y Estructura de Datos

### NOTA

**PROFESOR** **:** Erick Dejo Vicente

**SEMESTRE** **:** 2023 – 02

**CICLO :** Segundo

**SECCIÓN** **:** T2IL

**GRUPO** **:** 01

**FECHA** **:** 20/11/2023

|  |
| --- |
| **ALUMNO (A) : Raúl Ichiro Rosas Chinen** |

**CASO DE LABORATORIO 3**

**Consideraciones generales:**

* La solución debe ser presentada en el archivo **1814\_AED\_Solucion\_CL3.pdf**.
* El color del texto debe ser legible, de lo contrario, la solución no será calificada.
* Lea cuidadosamente la rúbrica para cumplir con todo lo solicitado.
* Si dos o más alumnos presentan la misma solución, recibirán la nota 99 (cero anulado).
* El color del texto debe ser legible, de lo contrario, la nota será 00.

**LOGRO**

Al término del curso, el alumno implementa los conceptos de encapsulamiento, herencia y polimorfismo de la programación orientada a objetos en la construcción de aplicaciones java básicas que realizan mantenimiento de datos almacenados en arreglo de objetos y archivos planos.

**Consolidado**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Pregunta** | **Puntaje** | | **Llenar solo en caso de recalificación justificada** | |
| **Máximo** | **Obtenido** | **Sustento** | **Puntaje** |
| 1 | 10 |  |  |  |
| **2** | **10** |  |  |  |
| **Nota Recalificada** | | | |  |

Problema 1 (10 puntos)

Dadas las clases **Libro**, **ArregloLibros** y **Problema1** implemente los siguientes métodos:

En la clase **ArregloLibros**:

* Un método **buscar** que busque un libro por su código, recibido a través de un parámetro, y retorne su dirección de memoria. En caso que el código no exista, retorne **null**.

En la clase **Problema1**:

* Un método **adicionarLibros** que adicione un libro con datos tomados de la GUI evitando que el código se repita. Si el código ingresado ya existe, muestre el mensaje “El código ingresado ya existe”. Una vez adicionado el libro, actualice el listado de libros y muestre el mensaje “Libro adicionado. Cantidad actual de Libros: xxx”, donde xxx es la cantidad de libros que fueron ingresados hasta el momento.
* Un método **modificarLibro** que lea el código de un libro de la GUI y en caso que exista, disminuya su precio en 19% e incremente su stock en 31. Si el código leído no existe, muestre el mensaje “El código ingresado no existe”. Una vez modificado el libro, actualice el listado de libros y muestre el mensaje “Modificación efectuada. Nuevo precio xxx. Nuevo stock zzz”, donde xxx es el nuevo precio y zzz, el nuevo stock.
* Un método **eliminarLibro** que lea el código de un libro de la GUI y en caso que exista, lo elimine. Si el código leído no existe, muestre el mensaje “El código ingresado no existe”. Una vez eliminado el libro, actualice el listado de libros y muestre el mensaje “El libro con código xxx fue eliminado”, donde xxx es el código del libro eliminado.

No añadir ningún otro método a la clase **ArregloLibros** a parte del método **buscar**. Todas las operaciones en la clase **Problema1** deben efectuarse únicamente con los métodos proporcionados (**adicionar**, **obtener**, **tamaño** y **eliminar**) más el método **buscar**.

Problema 2 (10 puntos)

Diseñe las clases **Maquina**, **MaquinaPC** y **MaquinaLaptop**. Las clases **MaquinaPC** y **MaquinaLaptop** serán subclases de la clase **Maquina**. Para el diseño de las clases considere lo siguiente:

La clase **Maquina** debe tener:

* Atributos privados: codigo (*int*), marca (*String*), procesador (*String*).
* Un constructor que inicialice todos los atributos.
* Métodos de acceso set/get para todos los atributos.

La clase **MaquinaPC** debe tener:

* Atributos privados: codigo (*int*), marca (*String*), procesador (*String*) y cantidad de ventiladores (*int*).
* Un constructor que inicialice todos los atributos.
* Métodos de acceso set/get para todos los atributos.
* Un método que retorne la marca de la PC de la máquina de acuerdo a la siguiente tabla:

|  |  |
| --- | --- |
| **Cantidad de ventiladores** | **Marca de PC** |
| 2 | Corsair |
| 4 | CoolerMaster |
| 5 | Cougar |
| 6 | AlienWare |

La clase **MaquinaLaptop** debe tener:

* Atributos privados: codigo (*int*), marca (*String*), procesador (*String*), largo (double) y ancho (*double*).
* Un constructor que inicialice todos los atributos.
* Métodos de acceso set/get para todos los atributos.
* Un método que retorne el área de la laptop (largo \* ancho).

En clase **Problema2**, ante la pulsación del botón Procesar, cree un objeto **MaquinaPC** y un objeto **MaquinaLaptop**, ambos con datos fijos, y muestre todos sus datos usando los métodos **mostrarMaquinaPC** y **mostrarMaquinaLaptop**.

**RÚBRICA**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Problema** | **10.0 puntos** | **7.5 puntos** | **5.0 puntos** | **2.5 puntos** |
| **Problema 1**  10 Puntos | Implementa correctamente los cuatro métodos solicitados. | Implementa correctamente tres métodos solicitados. | Implementa correctamente dos métodos solicitados. | Implementa correctamente un método solicitado. |
| **Problema 2**  10 Puntos | Implementa correctamente las cuatro clases solicitadas. | Implementa correctamente tres clases solicitadas. | Implementa correctamente dos clases solicitadas. | Implementa correctamente una clase solicitada. |