**INSTITUTO DE EDUCACION SUPERIOR CIBERTEC**

**DIRECCIÓN ACADÉMICA**

**CARRERA PROFESIONALES**

**CURSO :** 4683 – Algoritmo y Estructura de Datos

### NOTA

**PROFESOR :** Henry Christopher Ilizarbe Monteagudo

**SEMESTRE :** 2024 - I

**CICLO :** Segundo

**SECCIÓN :** T2PL

**DURACIÓN :** 2 días

**FECHA INICIO :** Sábado, 6 de Julio desde las 12:00 horas

**FECHA FIN :** Lunes, 8 de Junio a las 23:59 horas

|  |
| --- |
| **ALUMNO (A) :** |

**EVALUACIÓN PARCIAL – EF**

**Consideraciones generales:**

* Considerar el orden, la limpieza y la claridad de las respuestas.
* Está permitido el uso o consulta de cuadernos, separatas, libros o cualquier material de la asignatura durante el desarrollo de la evaluación.
* Inicie el proyecto con el nombre T2PL\_EF\_XXXX (donde XXXX serán sus apellidos y nombres)
* Subir el proyecto al Blackboard

**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** | | | |  |

**Pregunta 01**

Implemente la clase Factura en el paquete clase con los atributos privados: ruc (String), empresa (String), unidades (int) y precio unitario (double); un constructor que inicialice los atributos privados, los métodos de acceso público set/get y el método importeFacturado (unidades \* precio unitario).

Implemente la clase ArregloFacturas en el paquete arreglo con el atributo privado fac (ArrayList de tipo Factrura) e implemente:

Métodos básicos:

* Un constructor que cree el ArrayList y adicione las DirMem de ocho objetos Factura.
* Método adicionar que recibe la DirMem de una nueva factura y la adiciona al ArrayList.
* Método tamaño que retorne la cantidad de objetos Factura registrados hasta ese momento.
* Método obtener que reciba una posición y retorne la DirMem del objeto Factura respectivo.

Métodos complementarios:

* Retorne suma de todos los importes facturados.
* Retorne importe promedio facturado.
* Retorne menor importe facturado.
* Retorne mayor importe facturado.
* Retorne nombre de la primera empresa cuyo importe facturado sea menor al importe promedio facturado.

En la clase principal:

* Considere la declaración global ArregloFacturas af = new ArregloFacturas();
* Implemente la pulsación del botón Adicionar que agregue una nueva factura.
* Implemente la pulsación del botón Reportar que muestre los retornos de los métodos complementarios.

**Pregunta 02**

Implemente en el paquete padre, la clase Licencia con los atributos protegidos de tipo cadena: clase, categoría, fechaEmision (dd/mm/aa) y fechaRenovacion (dd/mm/aa), un constructor que inicialice los atributos, el método datosDelBrevete() que retorne en una cadena los dados de la licencia más un código generado que se obtiene de concatenar clase y fechaEmision.

Implemente en el paquete hijos, la clase Particular que herede (usando extends) la clase padre Licencia y agregue el atributo privado dni de tipo entero. A través del constructor reciba los cinco aributos y derive (usando super) los cuatro primeros a la clase padre Licencia. Implemente el método público datosLicenciaParticular() que retorne en una cadena los datos completos de la licencia incluyendo un segundo código generado a través del método público segundoCodigo() que se obtiene de concatenar categoría y dni.

Implemente en el paquete hijos, la clase Profesional que herede (usando extends) la clase padre Licencia y agregue el atributo privado ruc de tipo cadena. A través del constructor reciba los cinco aributos y derive (usando super) los cuatro primeros a la clase padre Licencia. Implemente el método público datosLicenciaProfesional() que retorne en una cadena los datos completos de la licencia incluyendo un segundo código generado a través del método público segundoCodigo() que se obtiene de concatenar categoría y ruc.

A la pulsación del botón Procesar en la clase principal cree un objeto de tipo Licencia, Particular y Profesional respectivamente. Finalmente, visualice a través de tres métodos listado la información completa de cada objeto.

Cambie la sintaxis de los métodos que retornan los datos completos de cada clase por el de datosCompletos() y haga uso de super en las clases hijos para diferenciarlos.

Utilice un solo método listado que reciba como parámetro la referencia de la clase padre Licencia, haga uso del operador instanceof para distinguir el tipo de objeto recibido y visualice la información completa de los objetos.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Rúbrica** | | | | | |
| **Puntos** | **10 puntos** | **7 puntos** | **5 puntos** | **3 puntos** | **0 puntos** |
| **10 Puntos** | Implementa lo solicitado de forma completa y correcta. | Implementa lo solicitado de forma completa pero con errores. | Implementa lo solicitado de forma parcial. | Implementa lo solicitado de forma parcial y con errores. | No implementa ninguna funcionalidad solicitada. |
| **10 Puntos** | Implementa lo solicitado de forma completa y correcta. | Implementa lo solicitado de forma completa pero con errores. | Implementa lo solicitado de forma parcial. | Implementa lo solicitado de forma parcial y con errores. | No implementa ninguna funcionalidad solicitada. |