

U3-Ejercicio N° 1

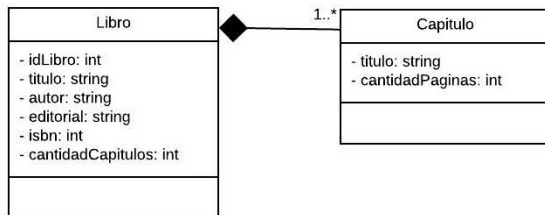
Grade: 100/100



Due: Monday, June 8, 2020 at 11:59 pm

Composición

Usted es el programador de una empresa de software. El analista le ha entregado la siguiente parte del diagrama de clases del nuevo software que están desarrollando.



A. Implemente las clases del diagrama anterior.

B. Defina una clase *ManejaLibros* que permita manejar los n libros que posee la biblioteca usuaria del software.

C. Implemente un programa principal que permita:

- a. Cargar los datos de los libros en una instancia de la clase *ManejaLibros*. Para esto debe considerar que la biblioteca ha provisto un archivo *libros.csv* con los datos de los libros. Este archivo presenta la siguiente estructura lógica: en una línea están los datos del libro y a continuación, una línea por cada capítulo con sus respectivos datos. Esto se repite para cada libro. A continuación se da un ejemplo.

```

10001, Python 3 Object oriented Programming, Dusty Phillips, Packt Publishing, 1849511268, 5
Object-oriented Design, 26
Objects in Python, 31
When objects are Alike, 36
Expecting the Unexpected, 28
When to use Object-oriented Programming, 34
10002, Learning Object-Oriented Programming, Gastón C. Hillar, Packt Publishing, 9781785
Objects Everywhere, 25
Classes and Instances, 43
Organization of Object-Oriented Code, 38
  
```

- b. A través de un **menú de opciones** cumplir con las siguientes funcionalidades:

1. Ingresar el identificador de un libro y mostrar título del libro, nombre de cada uno de sus capítulos y cantidad total de páginas de un libro.
2. Dada una palabra, mostrar título y autor de los libros que contienen la palabra dada en su título o en el título de alguno de sus capítulos.

Submissions



Assignment submitted

1 item · On time

Re-submit Assignment

Comments

There are no comments

Write a comment

Post

U3-Ejercicio N° 2

Grade: 100/100



Due: Monday, June 8, 2020 at 11:59 pm

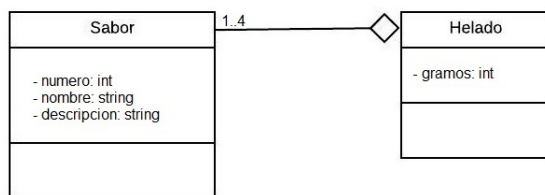
Agregación

La heladería “El conito” requiere una aplicación para el sector de ventas que registre los helados vendidos de modo que facilite el análisis de estos datos.

Descripción

1. La heladería presenta distintos sabores de helado.
2. Cada sabor tiene asociado un número que comienza en 1 y es correlativo.
3. De cada sabor se registra el número, el nombre y una breve descripción.
4. Los helados se distinguen por el peso expresado en gramos.
5. Los tipos de helados que se venden son de 100gr, 150 gr, 250 gr, 500 gr y 1000gr.
6. De un helado vendido se registra el peso del helado y el/los sabor/es pedidos.

Diagrama de clases



La aplicación que usted debe implementar requiere:

- A. Implementar las clases del diagrama UML dado.
- B. Definir una clase *ManejaSabores* que permita manejar los n sabores que la heladería presenta a la venta.
- C. Definir una clase *ManejaHelados* que registre y gestione a través de una lista los helados vendidos.
- D. Implementar un programa principal que permita:
 - a. Cargar los datos de los sabores en una instancia de la clase *ManejaSabores*. Estos datos se encuentran en el archivo *sabores.csv*.
 - b. A través de un menú de opciones realice las siguientes funcionalidades:
 1. Registrar un helado vendido (instancia de la clase *helado*).
 2. Mostrar el nombre de los 5 sabores de helado más pedidos.
 3. Ingresar un número de sabor y estimar el total de gramos vendidos. Para un helado se estima la cantidad de gramos de cada sabor dividiendo los gramos del helado en la cantidad de sabores. Por ejemplo, si se vendió un helado de 1000 gr de chocolate, frutilla, limón y americana. Se estima que en este helado se vendió de cada sabor $1000 / 4 = 250\text{gr}$.
 4. Ingresar por teclado un tipo de helado y mostrar los sabores vendidos en ese tamaño considerando todos los helados vendidos.

Posted Wed May 6, 2020 at 10:50 am

Comments

There are no comments

Submissions



Assignment submitted

1 item · On time

[Re-submit Assignment](#)

U3-Ejercicio N° 3

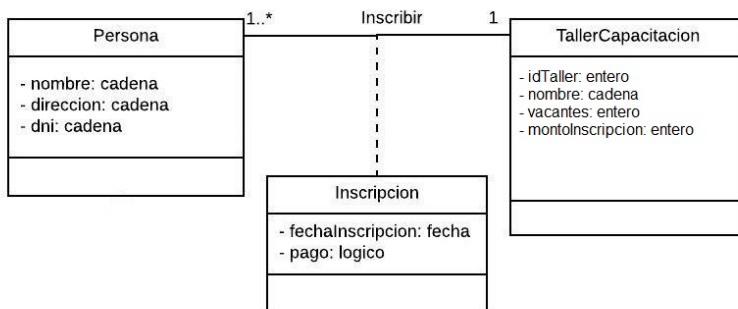
Grade: 100/100 

Due: Monday, June 8, 2020 at 11:59 pm

Clase asociación

Con su grupo de trabajo deben desarrollar una aplicación para manejar las inscripciones de las personas a los talleres de capacitación que se dictan en una escuela complementaria. Al momento de la inscripción la persona puede o no abonar el costo del taller.

a- Defina las clases involucradas en el siguiente diagrama.



a- Defina una clase con un atributo arreglo para almacenar instancias de la clase *TallerCapitacion*. El archivo "Talleres.csv" contiene en la primera línea la cantidad de talleres, y a continuación los datos de cada uno de ellos.

b- Defina una clase colección (manejador), tipo lista para almacenar las instancias de la clase *Persona*.

c- Defina una clase colección (manejador), tipo arreglo para almacenar instancias de la clase *Inscripcion*.

e- Implementar un programa que permita:

1. **Cargar los datos de los talleres** en la clase *TallerCapitacion* a partir de los datos registrados en el archivo.
2. **Inscribir una persona en un taller:** Se registra la inscripción (con el atributo pago en False) y la cantidad de vacantes del taller debe ser actualizada.
3. **Consultar inscripción:** Ingresar el DNI de una persona, si está inscripta mostrar el nombre del taller en el que se inscribió y el monto que adeuda.
4. **Consultar inscriptos:** Ingresar el identificador de un taller y listar los datos de los alumnos que se inscribieron en él.
5. **Registrar pago:** Ingresar el DNI de una persona y registrar el pago (dar al atributo pago el valor True).
6. **Guardar inscripciones:** Generar un nuevo archivo que contenga los siguientes datos de las inscripciones: DNI de la persona, idTaller, fechaInscripcion y pago.

Submissions



Assignment submitted

1 item · On time

Re-submit Assignment

U3-Ejercicio N° 4

Grade: 100/100 

Due: Monday, June 8, 2020 at 11:59 pm

Herencia y métodos de ligadura dinámica

Descripción del sistema

1. En una empresa constructora existen tres tipos de empleados: de planta, contratados y externos.
2. De cada empleado se registra: DNI, nombre, dirección y teléfono.
3. Si el empleado es de planta se registra: sueldo básico y antigüedad.
4. Si el empleado es contratado se registra: fecha de inicio y de finalización de contrato, cantidad de horas trabajadas y valor por hora (es el mismo para todos los empleados contratados).
5. Los empleados externos realizan tareas especiales estas tareas pueden ser carpintería, electricidad, plomería. Para este tipo de empleados se registra: tarea, fecha de inicio, fecha de finalización, monto viático, el costo de la obra y monto del seguro de vida.
6. El sueldo de un empleado contratado se calcula con la siguiente fórmula:

$$\text{Sueldo} = \text{cantidad de horas trabajadas} * \text{valor de la hora}$$

7. El sueldo de un empleado de planta se calcula con la siguiente fórmula:

$$\text{Sueldo} = \text{sueldo básico} + 1\% \text{ del sueldo básico} * \text{antigüedad}$$

8. El sueldo de un empleado externo se calcula:

$$\text{Sueldo} = \text{costo de la obra} - \text{viático} - \text{monto del seguro de vida}$$

Para el desarrollo del sistema requerido usted debe:

- a- Definir la jerarquía de clases correspondiente a la descripción dada.
- b- Definir una clase colección basada en un arreglo cuyo tamaño se debe ingresar por teclado, para almacenar los empleados de la empresa.
- c- Almacenar en memoria principal los empleados de la empresa, obteniendo los datos de los archivos "planta.csv", "contratados.csv", "externos.csv" que contienen los datos de cada uno de los tipos de empleados.
- d- Implementar un programa que a partir de la información almacenada en memoria permita:
 1. **Registrar horas:** Ingresar el DNI de un empleado y la cantidad de horas trabajadas hoy e incrementar la cantidad de las horas trabajadas del empleado.
 2. **Total de tarea:** Dada una tarea mostrar el monto a pagar para ella. Solo se consideran las tareas que no han finalizado.
 3. **Ayuda:** La empresa dará una ayuda solidaria a los empleados cuyo sueldo sea inferior a \$25000; listar nombre, dirección y DNI de los empleados que les corresponde la ayuda.
 4. **Calcular sueldo:** Mostrar nombre, teléfono y sueldo a cobrar de todos los empleados.

Posted Wed May 6, 2020 at 10:56 am

Comments

There are no comments

Submissions



Assignment submitted

1 item · On time

U3-Ejercicio N° 5

Grade: 100/100 

Due: Monday, June 8, 2020 at 11:59 pm

Ejercicio N° 5 - Interfaces

Defina una interface con los siguientes métodos:

a- **insertarElemento**: para insertar un objeto en una posición determinada en una colección, teniendo en cuenta el manejo de excepciones cuando la posición donde se vaya a insertar no sea válida.

b- **agregarElemento**: para agregar un elemento al final de una colección.

c- **mostrarElemento**: dada una posición de la colección, mostrar los datos del elemento almacenado en dicha posición si esa posición es válida, en caso de que no sea válida lanzar una excepción que controle el error.

Posted Mon May 18, 2020 at 8:09 pm

Comments

There are no comments

Submissions



Assignment submitted

1 item · On time

U3-Ejercicio N° 6

Grade: 70/100



Due: Monday, June 8, 2020 at 11:59 pm

Descripción del sistema

La concesionaria de automóviles "TuAuto.com" se dedica a la venta de vehículos nuevos y usados. Para ello cuenta con un archivo, "vehiculos.json", con la estructura necesaria para almacenar vehículos nuevos y vehículos usados.

De cada vehículo desea registrar el modelo (ej. Palio, Focus, etc.), cantidad de puertas, color y el precio base de venta. Los vehículos nuevos son de una misma marca, en cambio los usados pueden ser de cualquier marca, por ello este también es un dato que debe registrar.

Además, para los usados se registrará la patente, el año y el kilometraje.

De un vehículo nuevo también registra la versión (base o full).

El importe de venta de cada vehículo se calcula en función del precio base y de sus características. Para ello se deben considerar las siguientes reglas de negocio:

- Importe de venta del vehículo usado es el precio base, menos el 1% por cada año de antigüedad (año actual – año del vehículo), menos el 2% si el kilometraje supera los 100.000. Ambos porcentajes se calculan sobre el precio base.
- Importe de venta del vehículo nuevo es el precio base, más 10% por gastos de patentamiento, mas 2% si es full. Ambos porcentajes se calculan sobre el precio base.

El analista del concesionario, le solicita a usted que desarrolle una aplicación, con las siguientes restricciones:

La colección de vehículos **debe** implementarse usando una **Lista definida por el programador**, donde los nodos deberán almacenar vehículos.

La Lista deberá proveer los métodos y atributos necesarios para que sea un iterable, e implementar la interface del ejercicio anterior.

Para cumplir con lo solicitado por el analista, usted deberá:

- a) Definir la jerarquía de clases correspondiente a la problemática planteada.
- b) Leer y procesar el archivo "vehiculos.json" para almacenar en la Lista los vehículos de la concesionaria.
- c) Implemente un programa que a través de un menú de opciones permita:
 1. Insertar un vehículo en la colección en una posición determinada.
 2. Agregar un vehículo a la colección.
 3. Dada una posición de la Lista: Mostrar por pantalla qué tipo de objeto se encuentra almacenado en dicha posición.
 4. Dada la patente de un vehículo usado, modificar el precio base, y luego mostrar el precio de venta.
 5. Mostrar todos los datos, incluido el importe de venta, del vehículo más económico.
 6. Mostrar para todos los vehículos que la concesionaria tiene a la venta, modelo, cantidad de puertas e importe de venta.
 7. Almacenar los objetos de la colección Lista en el archivo "vehiculos.json".

Posted Mon May 18, 2020 at 11:29 pm

Comments

There are no comments

Write a comment

Post

Submissions



Assignment submitted

1 item · On time

Re-submit Assignment

U3-Ejercicio N° 7

Grade: 100/100 

Due: Monday, June 8, 2020 at 11:59 pm

Herencia Múltiple, polimorfismo

En el ámbito universitario los agentes que ahí trabajan se pueden clasificar de la siguiente manera: Docentes, investigadores, docente investigador y personal de apoyo.

La información del personal universitario se encuentra almacenada en un archivo denominado "personal.json".

El Centro de cómputos de la UNSJ, lo contrata a usted como programador, para que desarrolle una aplicación que permita procesar el archivo .json donde guarda la información del personal que trabaja en la misma. Para ello le provee la siguiente narrativa, que ha sido elaborada por el analista funcional:

Narrativa

De todo el personal se conoce la siguiente información: cuil, apellido, nombre, sueldo básico y antigüedad.

Si es un docente se registra además carrera en la que dicta clases, cargo y cátedra.

Si es personal de apoyo se registra además categoría.

Si es un investigador se registra además área de investigación y tipo de investigación

Si es un docente investigador se registra además carrera en la que dicta clases, cargo, cátedra, área de investigación y tipo de investigación, categoría en el programa de incentivos de investigación, importe extra por docencia e investigación.

Para cumplir con las tareas solicitadas por el analista, usted deberá:

- a) Definir la jerarquía de clases correspondiente a la narrativa dada.
- b) Almacenar en una colección tipo Lista definida por el programador, los agentes de la Universidad, obteniendo los datos del archivo "personal.json", la misma deberá implementar la interfaz definida en el ejercicio 5. Los nodos de la lista, serán referencias a objetos que representen objetos de la clase base de la jerarquía.
- c) Implementar un programa principal con un menú de opciones que permita testear las siguientes acciones:
 1. 1. Insertar a agentes a la colección.
 2. Agregar agentes a la colección.
 3. Dada una posición de la lista: Mostrar por pantalla que tipo de agente se encuentra almacenado en dicha posición.
 4. Ingresar por teclado el nombre de una carrera y generar un listado ordenado por nombre con todos los datos de los agentes que se desempeñan como docentes investigadores.
 5. Dada un área de investigación, contar la cantidad de agentes que son docente investigador, y la cantidad de investigadores que trabajen en ese área.
 6. Recorrer la colección y generar un listado que muestre nombre y apellido, tipo de Agente y sueldo de todos los agentes, ordenado por apellido.
 7. Dada una categoría de investigación (I, II, III, IV o V), leída desde teclado, listar apellido, nombre e importe extra por docencia e investigación, de todos los docentes investigadores que poseen esa categoría, al final del listado deberá mostrar el total de dinero que la Secretaría de Investigación debe solicitar al Ministerio en concepto de importe extra que cobran los docentes investigadores de la categoría solicitada.
 8. Almacenar los datos de todos los agentes en el archivo "personal.json"

Submissions



Assignment submitted

1 item · On time

Re-submit Assignment

U3-Ejercicio N° 8

Grade: 50/100



Due: Monday, June 8, 2020 at 11:59 pm

Usar la narrativa del Ejercicio N° 4

Se ha agregado nuevos requerimientos al sistema, y usted como programador, debe darles solución:

El sistema prevé que los empleados serán administrados a través de una colección. Existen dos interfaces distintas para operar con empleados, la interfaz del **Tesorero**, que solamente puede acceder a los gastos que la empresa tiene en concepto de sueldos, para ello, dado un número de documento, podrá consultar el gasto de sueldos para el empleado al que pertenece dicho número de documento.

La interfaz del **Gerente**, permite modificar el sueldo básico de un empleado de planta, el valor que se paga por hora de un empleado contratado o el valor que se paga por viático de un empleado externo; para ello debe proveerse el número de documento del empleado, y el valor que corresponda según lo que se quiera modificar.

Usted deberá crear las interfaces mencionadas con las funcionalidades solicitadas.

class ITesorero (Interface)

```
def gastosSueldoPorEmpleado ( dni):  
    pass
```

class IGerente (Interface)

```
def modificarBasicoEPlanta(dni, nuevoBasico):  
    pass  
  
def modificarViaticoEExterno(dni, nuevoViatico):  
    pass  
  
def modificarValorEPorHora(dni, nuevoValorHora):  
    pass
```

En el programa principal agregue una opción para el Tesorero y un menú de opciones para Gerente. Para ello deberá autenticar al usuario que quiere acceder, si es tesorero, accederá a las funcionalidades de tesorero y si es gerente, accederá a las funciones de gerente. La autenticación del usuario se hace pidiendo nombre de usuario y contraseña.

Usuario/contraseña para Tesorero: uTesorero/ag@74ck

Usuario/contraseña para Gerente: uGerente/ufC77#1

Posted Mon May 18, 2020 at 11:42 pm

Comments

There are no comments

Submissions



Assignment submitted

1 item · On time