



## EJERCICIOS DE PROGRAMACION

Ejercicios resueltos en Java, Pseint, C#, C, C++, Python, Scilab, VisualFox, Pascal

### ARREGLO DE OBJETOS

Fecha de impresión: 30-06-2024

- 246 Crear un arreglo de n estudiantes (codigo, nombre y edad), el programa debe mostrar el nombre del estudiante con mayor edad.
- 388 2. Utilizando Herencia, Encapsulaci3n y Polimorfismo. Crear un programa que tenga el control de cuentas bancarias de n usuarios. Dependiendo del tipo de cuenta "Ahorros", "Corriente", si es Ahorros por cada dep3sito tiene una bonificaci3n del 1% , por cada retiro tiene un descuento del 1.5%, si es corriente por cada dep3sito tiene un bono de 2%, no tiene descuento en los retiros. Obtener los valores de las cuentas de n personas e imprimir sus respectivos saldos con los bonos respectivos o descuentos.
- 389 5. Utilizando Herencia, Encapsulaci3n y Polimorfismo. Crear un programa que permita el control de notas de n estudiantes para n materias, cada materia tiene 3 notas, determinar si cada estudiante aprueba o no la materia, para aprobar debe tener al menor un promedio de 7, si el estudiante tiene un promedio de al menos 5, tiene derecho a un supletorio, el estudiante para aprobar con supletorio debe al menos obtener una nota de 8, el estudiante reprueba la materia si el promedio es menor que 5 o la nota del supletorio es menor que 8. Listar los alumnos con las materias y sus promedios con la aprobaci3n o no de la misma.
- 390 Realizar el programa con vectores de objetos para la gesti3n de cuatro tipos de activos: dispositivos electr3nicos, autom3viles, muebles y discos compactos. Elaborar la clase principal y las clases heredadas, de tal manera que el programa permita a3adir, modificar, borrar, buscar y ordenar los activos. Listar los activos por categor3a y buscar un determinado activo por su n3mero de serie.
- 391 1. Crear un vector de objetos para n personas, se debe ingresar datos utilizando un constructor con par3metros para inicializar el nombre, cuenta, monto. Insertar un nuevo dato indicando la posici3n en donde se desea insertar. Encontrar el n3mero de personas que tienen montos inferiores y superiores a cantidades de 4 cifras.
- 392 2. Realizar un programa que crea un vector de objetos para n empleados, se debe ingresar el nombre, la cedula, las horas trabajadas. La hora trabajada se paga a 7 USD, Si el sueldo es hasta 500 el estado es "P3simo", si el sueldo es hasta 800 el estado es "Regular", si el sueldo es hasta 1000 el estado es "Bueno", si el sueldo es superior a 1000 el estado es "Excelente".  
Imprimir el listado de los empleados incluido el estado ordenado de ascendentemente por nombre:
- 393 3. crear un vector de objetos para n estudiantes, se debe ingresar datos, utilizando un constructor con par3metros para inicializar, el nombre, la materia, el promedio. Encontrar el n3mero de estudiantes que aprueban la materia y el n3mero de estudiantes que reprueban la materia. Para aprobar el promedio debe ser al menos 7, caso contrario reprueba; adem3s el programa debe permitir insertar o eliminar un estudiante ingresando como dato la posici3n en el vector. Imprimir la lista ordenada en forma descendente de acuerdo al



## EJERCICIOS DE PROGRAMACION

### Ejercicios resueltos en Java, Pseint, C#, C, C++, Python, Scilab, VisualFox, Pascal

promedio.

- 394 4. Realice un algoritmo para crear un vector de objetos para  $n$  personas, se debe ingresar datos utilizando un constructor con parámetros para inicializar el nombre, la fecha de nacimiento, el sexo. De acuerdo a la fecha de nacimiento y la del sistema calcular la edad (años) determinar si la persona es mayor o menor de edad, encontrar el número de personas mayores de edad y el número de personas menores de edad. Imprimir el listado de personas ordenado de acuerdo a la edad.
- 395 5. Realice un algoritmo para crear un vector de objetos para  $n$  productos, se debe ingresar datos utilizando un constructor con parámetros para inicializar el código, nombre, precio de costo y precio de venta. El precio de venta se calcula de acuerdo al precio de costo, si el precio de costo esta entre 0 y 10, el precio de venta es más el 25%, si el precio de costo esta 10 y 30, el precio de venta es más el 20%, si el precio de costo esta 30 y 50, el precio de venta es más el 15%, si el precio de costo esta 50 y 80, el precio de venta es más el 10%, si el precio de costo mayor que 80, el precio de venta es más el 8%. Encontrar los precios de venta, ordenar de forma ascendente y descendente por precio de venta o por nombre, buscar un determinado producto por el nombre y mostrar la información del producto. Imprimir la lista de precios de venta ordenado por nombre.
- 396 Realizar el programa con vectores de objetos que permita el control de préstamos para  $N$  libros, utilizando constructores, los datos de cada libro son: código, nombre, autor, editorial, materia, año de edición, número de copias. El programa debe determinar si un libro está disponible para ser prestado o no dependiendo del número de copias y sus préstamos. Crear un formulario con todas las opciones de mantenimiento, es decir, insertar, modificar, buscar, borrar. El programa además debe ordenar los datos por cualquiera de sus campos. Determinar el número de libros prestados y no prestados.
- 397 Crear un vector de objetos para  $n$  empleados, se debe ingresar desde teclado utilizando un constructor con parámetros para inicializar el nombre, la edad, el sueldo. Si la edad es hasta 35, tiene una bonificación del 5%, si la edad es hasta 45 tiene una bonificación de 10%, si la edad es superior a 45 tiene una bonificación del 15%. Crear un menú con las siguientes opciones:
- a) Ingresar datos: Ingresa los datos en el vector
  - b) Buscar: Encontrar los datos del empleado que tenga el mejor sueldo incluido la bonificación, utilice el método de búsqueda secuencial
  - c) Ordenar: Utilizando el método selección, ordenar en forma descendente por la edad de los empleados.
- 400 Estructura de datos, Arbol binario, muestra la organización del árbol, altura y profundidad.
- 401 Ingresar números no repetidos a un árbol binario, realizar los recorridos en orden, preorden, postorden. mostrar el árbol



## EJERCICIOS DE PROGRAMACION

### Ejercicios resueltos en Java, Pseint, C#, C, C++, Python, Scilab, VisualFox, Pascal

- 522 Utilizando POO, crear un programa, con el nombre de proyecto : Estudiante y una clase en el mismo paquete llamada Operaci3n. Que realice lo siguiente :  
Pedir por consola un n3mero de estudiantes, luego solicitar los atributos: nombre, apellido, c3dula, edad, nota 1,nota2 de cada estudiante, a continuaci3n crear los siguiente m3todos: promedio de las notas de cada estudiante, estudiantes aprobados, estudiantes reprobados,(tomando en cuenta que un estudiante aprueba con la nota de 7 o mas, promediando nota1 y nota2), promedio de las edades de los estudiantes, la nota m3s alta de los estudiantes y el estudiante) , para el siguiente ejercicio haga uso de arreglos y llame a los m3todos desde la clase principal e imprima por consola lo requerido.
- 581 Realiza correctamente el siguiente programa: Crear la clase Elevador con los siguientes campos: Cantidad de personas y Peso del viaje. Elaborar un programa que lea los datos de los viajes del d3a: Cantidad de personas y Peso del viaje. Y al finalizar imprima Cantidad de Viajes, Personas Transportadas, Peso Transportado, Promedio de personas por Viaje y Promedio de peso por Viaje.
- 815 Cargue dos arreglos, uno con los nombres de los candidatos y el otro con la cantidad de votos obtenidos en las elecciones de un pa3s. El programa debe permitir trabajar con n candidatos. (con Programaci3n Orientada a Objetos)  
imprimir los nombres de los tres candidatos con mayor cantidad de votos.
- 838 La alcald3a de Bogot3 ha recibido en los 3ltimos tiempos muchas denuncias por abusos en el cobro de los servicios p3blicos, como principio de soluci3n a este tema, el distrito ha decidido realizar el desarrollo de una aplicaci3n de escritorio por consola que permita de al usuario detectar si le est3n haciendo un cobro no apropiado a su consumo de cada servicio p3blico.  
Usted ha sido contratado por el distrito para desarrollar la aplicaci3n que permite a los usuarios a saber cu3nto deben pagar por Servicio P3blico: AGUA, LUZ o GAS NATURAL, bajo las siguientes condiciones:  
â€¢Si el estrato socioecon3mico donde se encuentra el predio es 1 o 2: hay un descuento del50% sobre el total de la factura.  
â€¢Si el estrato socioecon3mico donde se encuentra el predio es 3 o 4: hay un descuento del10% sobre el total de la factura.  
â€¢Si el estrato socioecon3mico donde se encuentra el predio es 5 o 6: hay un sobrecostodel 25% sobre el total de la factura.  
â€¢Para el GAS NATURAL, el valor del metro c3bico es \$5234.  
â€¢Para el AGUA, el valor del metro c3bico es \$9256.  
â€¢Para el servicio de ENERG3A (LUZ), el valor de cada KW (kilowatt) es de \$6200.  
El valor por factura de cada uno de los servicios se calcula teniendo en cuenta la siguiente f3rmula: CONSUMO x VALOR UNIDAD  
El usuario debe poder escoger qu3 servicio p3blico quiere pagar interactuando con un MEN3 (Ayuda: Haga uso de Switch/Case) y considerando todas las condiciones presentadas previamente (Ayuda: Haga uso de if/else if/else).  
Posdata: debido a los da3os en infraestructura ocasionados por hinchas de equipos de futbol, todas las facturas tienen un impuesto del 1% sobre el valor de la factura.



## EJERCICIOS DE PROGRAMACION

### Ejercicios resueltos en Java, Pseint, C#, C, C++, Python, Scilab, VisualFox, Pascal

Desarrolle la solución utilizando el lenguaje de programación JAVA y el IDE Netbeans. Diseñe un diagrama de clases que siga el estilo arquitectural MVC, en donde queden representados: paquetes, clases, atributos, métodos y relaciones

- 974 Crea una clase Banco con atributos nombre del banco, ciudad, cantidad de cuentas bancarias y la lista de objetos de cuentas bancarias de ahorro de sus respectivos clientes. Cada cuenta de ahorro se creará mediante la clase CuentaAhorro (bancaria) con atributos para el número de cuenta (un entero largo), el CI del cliente (cadena de texto) y el saldo actual. Define en la clase CuentaAhorro los siguientes métodos:
- Constructor por defecto y constructor con parámetros de CI, saldo e intereses
  - Métodos de acceso y modificadores para cada atributo de la clase CuentaAhorro.
  - Acreditar(double): permitir depositar una cantidad de dinero en la cuenta.
  - Debitar(double): permitir sacar una cantidad de dinero de la cuenta (si hay saldo suficiente).
  - Método Impresión que nos permita mostrar todos los datos de la cuenta.
- Define en la clase Banco los siguientes métodos:
- Constructor por defecto vacío y constructor con parámetros de nombre del banco, ciudad y cantidad de cuentas bancarias de ahorro.
  - Métodos de acceso y modificadores para los atributos de nombre del banco, ciudad y cantidad de cuentas bancarias de ahorro.
  - AdicionarCuentaAhorro(CuentaAhorro obj): que permita adicionar un objeto de cuenta de ahorro a la lista de objetos de cuenta bancaria de ahorros.
  - AcreditarCuentaAhorro(int NoCuenta, int monto): permitir depositar una cantidad de dinero en la cuenta conocido su número de cuenta.
  - DebitarCuentaAhorro(int NoCuenta, int monto): permitir sacar una cantidad de dinero de la cuenta (si hay saldo suficiente), conocido su número de cuenta.
  - DatosCuentaAhorro(int NoCuenta): permita mostrar todos los datos de la cuenta conocido su número de cuenta.
1. Crear un objeto de la clase Banco mediante el constructor con parámetros de nombre del banco, ciudad y cantidad de cuentas bancarias de ahorro.
  2. Implementar un menú de opciones:
    1. Adicionar una cuenta bancaria de ahorro conocido el número de cuenta el, CI del cliente y el saldo actual.
    2. Acreditar dinero a la cuenta bancaria conocido su número de cuenta y el monto a depositar.
    3. Debitar dinero de la cuenta bancaria conocido su número de cuenta y el monto a sacar.
    4. Mostrar los datos de la cuenta bancaria conocido su número de cuenta.
- 1048 Calculadora básica, Ejercicio Enciclopedia de Visual C# (Javier Cevallos)
- 1194 Una clase Prueba como se aprecia en la figura, donde se evidencie por medio de la creación de objetos todas las sobrecargas posibles de las funciones constructoras de la clase, las variables i, j y k son de tipo entero. La Función miembro Mostrar(), muestra el valor de i, la Función miembro Mostrar2() llama a la Función miembro Mostrar() y muestra j, finalmente la Función miembro Mostrar3() llama a la Función miembro Mostrar2() y muestra k. Se deben tener mensajes en cada uno de los constructores donde se evidencie cual de los mismos es el que



## EJERCICIOS DE PROGRAMACION

### Ejercicios resueltos en Java, Pseint, C#, C, C++, Python, Scilab, VisualFox, Pascal

estÃ¡ inicializando los objetos respectivos.

- 1212 Clase Estudiante y clase Docente heredadas pÃºblicamente de la clase Persona como se aprecia en la Figura 1. Se deben generar las funciones constructoras suficientes y las funciones miembro para que se pueda visualizar la salida de la Figura 2, tenga presente que la funciÃ³n Mostrar() se debe anular en las Subclases y utilizar las funciones Mostrar() de la Superclase por medio del Operador de Ãmbito de ResoluciÃ³n cuando sea necesario, adicionalmente la funciÃ³n NotaFinal() retorna un valor flotante y los constructores de Estudiante inicializan las notas en 0 (cero).
- 1213 Una central de distribuciÃ³n de quÃ©micos distribuye Alcohol hacia diferentes almacenes sucursales. Disponen de un muelle de carga a donde van llegando tanques de alcohol de entre 3000 y 9000 litros, con pesos variables en funciÃ³n de las circunstancias de la producciÃ³n. La empresa dispone de una flota de camiones con capacidades de carga de entre 18000 y 28000 litros.  
Se pretende establecer un protocolo consistente en cargar 20 camiones diarios. Cada camiÃ³n se quiere cargar como mÃ¡ximo a su lÃmite de capacidad debiendo partir si con la siguiente saca en la lÃnea de producciÃ³n se fuera a exceder su capacidad.  
La empresa quiere desarrollar una soluciÃ³n bÃ¡sica de programaciÃ³n que le pida al operario encargado de carga la capacidad del camiÃ³n y el peso de las sacas, indicÃ¡ndole si debe cargar los tanques de alcohol o despachar el camiÃ³n para comenzar a cargar otro.
- 1214 Una lista enlazada es un TDA que permite almacenar datos de forma organizada, al igual que los arreglos o vectores, pero a diferencia de estos, su estructura es dinÃ¡mica, por lo que no tenemos que saber el nÃºmero de elementos que puede contener.  
Crear un programa que mediante listas enlazadas, permita realizar la inserciÃ³n ordenada, bÃ¡squeda, listado y eliminaciÃ³n de estudiantes de una aula.  
Para efecto del ejercicio, se realizarÃ¡ todo el programa en un solo archivo (como se muestra en la imagen anterior), declarando las clases Estudiante y MetodosAula en el archivo principal del proyecto Aula
- â Debe primeramente realizarse la declaraciÃ³n de la clase Estudiante (Nodo), que serÃ¡ la que represente a cada uno de los elementos de la lista enlazada. Esta clase deberÃ¡ estar formada por dos campos de informaciÃ³n Id y nombre (IdentificaciÃ³n y nombre) y un puntero sig hacia el siguiente Estudiante (Nodo), asÃ­ como de su correspondiente constructor y mÃ©todos Set y Get.
  - â Debe crearse clase MetodosAula, con un atributo head de tipo Estudiante, que serÃ¡ el encargado de apuntar al primer elemento de la lista, asÃ­ como de su correspondiente constructor y un objeto de tipo Scanner que nos servirÃ¡ para leer informaciÃ³n desde teclado.
  - â Se debe crear el mÃ©todo principal main, el mismo que debe contener:
    - â Un objeto de tipo MetodosAula, que se utilizarÃ¡ para comunicarse con los mÃ©todos a crearse.
    - â Un proceso repetitivo que permita presentar en pantalla las diferentes operaciones a realizar y recibir desde teclado la opciÃ³n escogida.
    - â Un proceso selectivo switch, que permita ejecutar el mÃ©todo de la opciÃ³n escogida por el usuario.





## EJERCICIOS DE PROGRAMACION

### Ejercicios resueltos en Java, Pseint, C#, C, C++, Python, Scilab, VisualFox, Pascal

• En la clase MetodosAula deberá crearse los siguientes métodos:

• ListaVacia. Que devolverá una respuesta de tipo boolean

• Insertar. Encargado de insertar un nuevo elemento a la lista enlazada. Este elemento debe ser insertado de forma ordenada de acuerdo al Id del nuevo estudiante.

• Recorrer. Para la presentación de los elementos en la lista.

• Buscar. Presentar el nombre del elemento buscado o un mensaje de error en el caso de no existir.

• Eliminar. Quitar de la lista el elemento solicitado.

1252 Para la realización del ejercicio se deberá considerar.

Para efecto del ejercicio, se realizará todo el programa con la misma modalidad del ejercicio del primer bimestre (como se muestra en la imagen anterior), declarando las clases Nodo, MetodosABB y la clase principal del proyecto main

• Debe primeramente realizarse la declaración de la clase Nodo, que será la que represente a cada uno de los elementos que forman el árbol. Esta clase deberá estar formada por los campos de datos Id y nombre y dos variables de tipo puntero izq y der, que estarán encargados de realizar el enlace desde los nodos hacia los sub árboles izquierdo y derecho, así como de su correspondiente constructor.

• Debe crearse clase MetodosABB, con un atributo raíz de tipo Nodo, que será el encargado de apuntar al primer elemento del árbol, así como de su correspondiente constructor.

• Se debe crear el método principal main, en el mismo que debemos definir:

Un objeto de tipo MetodosABB, que se utilizará para comunicarse con los demás métodos a crearse. Un proceso repetitivo que permita recibir desde teclado los datos a insertar en el árbol binario de búsqueda (id y nombre) y enviar la información para la creación de cada uno de los nodos.

• Un método Insertar Nodo, que se encargará de realizar las siguientes tareas: (operación de INSERCIÓN)

Crear un nuevo nodo con la información leída en el método principal. Buscar el lugar en donde le corresponde ubicarse al nuevo nodo

• Enlazar el nuevo nodo en el lugar adecuado. • Un método eliminarNodo, que se encargará de realizar las siguientes tareas: (operación de ELIMINACIÓN)

. Mediante un procedimiento de búsqueda localizar el nodo a eliminar

Realizar el redireccionamiento de enlaces para eliminar el nodo. Por medio de un método Reemplazar, actualizar la información del nodo a reemplazar con la información del nodo ubicado según el algoritmo descrito.

. Finalmente el método Listar: (operación de RECORRIDO)

• Se encargará de presentar en forma ordenada según el Id, la siguiente información: Id y nombre de cada nodo.

1280 Representar el siguiente árbol mediante código. Mostrar por pantalla sus elementos.

1289 El periódico TodoFut desea contratarlo para programar una clase que permita calcular el costo de venta de postales para el álbum del Mundial Qatar-2022. Se calculan según la cantidad de postales que se desee adquirir, así como al grupo que pertenece.



## EJERCICIOS DE PROGRAMACION

### Ejercicios resueltos en Java, Pseint, C#, C, C++, Python, Scilab, VisualFox, Pascal

Esta primera versi3n del sistema debe funcionar, como si fuese un tipo de calculadora, donde se le solicite al usuario los datos requeridos para la compra de postales, que debe de realizar autom3ticamente el costo. Los valores resultantes deben ser expuestos mediante mensajes que muestren la informaci3n almacenada.

Validar que no se repita las postales compradas.

Es requerido que defina la clase Postal, con los valores que considere necesario incluir, definir y desarrollar los m3todos "establecer" y "obtener" para cada atributo definido, as3 como un m3todo para el c3lculo del costo total.

Tambi3n tiene que crear otra clase llamada CalculadoraPostal el cual ser3 la interface gr3fica para solicitar los datos al usuario. Esta clase debe instanciar la clase Postal, para lo que corresponda.

Se debe solicitar al usuario, c3dula de identidad, nombre completo y el tipo de secci3n del 3lbum donde quiere comprar las postales.

1292 En un equipo de baloncesto se tienen varios jugadores. De cada jugador se tiene su nombre y la cantidad de puntos que anot3, por cada uno de los 5 partidos que jug3. Se desea obtener un reporte tipo tabla de todos los jugadores del equipo:

Nombre , Total de puntos, Nivel de anotaci3n. Ejm. PEPITO, 67 , Bueno .....

Total xxx jugadores xxxxx,

Nombre del mejor anotador xxxxxxxxxxxxxxxxxxxx,

Nombre del peor anotador xxxxxxxxxxxxxxxxxxxx,

• El total de puntos es la sumatoria de los puntos que anot3 en todos los juegos,

• El nivel de anotaci3n es un comentario que indica:

o Deficiente, si el total de puntos es menor de 40

o Bueno, si el total de puntos est3 entre 40 y 90

o Excelente, si el total de puntos es mayor de 90

• El total de jugadores es el n3mero de jugadores que se dieron de alta y tambi3n se desea conocer el Total de puntos anotados por todos los jugadores

• Nombre del mejor jugador, es el nombre del jugador que tiene el total de puntos mayor

• Nombre del peor jugador, es el nombre del jugador que tiene el total de puntos menor

Se pide:

• Utilizar un men3 de opciones necesarias (utilizar funciones)

• Utilizar atributos privados, constructores sobrecargados, m3todos set y get

• Utilizar una lista din3mica (vector) para almacenar los datos de los jugadores

1298 En la imagen, se muestra un diagrama UML. Con base al diagrama, definir e implementar la clase que corresponda.

Un docente utiliza una lista de estudiantes y calificaciones, entre las operaciones que el docente realiza se tienen:

1. Ingresar los nombres de los estudiantes y sus notas

Nota: Al ingreso se establece el estado (Aprobado nota mayor a 70 puntos)

Las notas deben estar en el rango de 0 a 100.



## EJERCICIOS DE PROGRAMACION

### Ejercicios resueltos en Java, Pseint, C#, C, C++, Python, Scilab, VisualFox, Pascal

2. Listar las calificaciones (ordenados por nombre),
3. Consulta las notas de un estudiante,
4. Corregir nota de un estudiante

diseñar el programa para realizar las tareas mencionadas aplicando cualquier método de ordenar y búsqueda

1300 La empresa Codor\_Feliz, lo cual su negocio consiste en una granja que se dedica a la crianza y reproducción de codornices. Las codornices se venden con frecuencia, es por eso que la cantidad de codornices en la granja varía a final de la semana.

La empresa invierte en comida fresca cada semana, es por eso que ellos compran el alimento todos los lunes. Codor\_Feliz separa las codornices en 3 tipos según edad: Polluelos, Jóvenes y adultos. Es por esta razón que se debe de comprar alimentos con diferente contenido de proteína:

Codorniz Tipo de Alimento Precio de alimento por saco Cantidad de Codornices que se alimentan por saco

Polluelo Crecimiento (6% Proteína) \$ 5840 1000

Joven Desarrollo (8% Proteína) \$ 8045 800

Adulto Mantenimiento (13% Proteína) \$ 13200 500

Por lo tanto, se le solicita un programa para calcular la cantidad de sacos por comprar según la cantidad de codornices en la granja. Si hay un número impar de codornices, el sistema debe agregar un saco entero de alimento adicional ya que no se puede conseguir medio saco de producto.

Es importante indicar que el proveedor de alimento le hace un descuento por cada huevo que la empresa le da a cambio. El proveedor le dará un máximo de 20% de descuento sobre el total de la compra si la empresa le da una suma mayor o igual a 300 huevos. El descuento es relativo, por ejemplo: Si la empresa le da 150 huevos, tendrá un descuento de 10%.

Considere lo siguiente:

• El programa debe solicitar la cantidad de codornices en la granja según las 3 clasificaciones por edad.

• Tiene que crear al menos una clase "Codorniz" para el cálculo de los sacos por comprar. Esta clase tiene que tener 3 métodos para calcular el precio de los 3 diferentes tipos de codorniz.

• Declare los atributos de la cantidad de codornices como privados y cree los métodos setter y getter para el acceso y asignación de valores.

• Debe incluir el manejo de excepciones: el programa solo puede recibir números enteros.

• Debe existir una clase llamada "Descuento" .

• En la clase descuento debe de haber una función para calcular el descuento en relación a la cantidad de huevos de codorniz.

• Imprimir el total por pagar con y sin descuento.

1304 EJERCICIO HERENCIA PERSONA.

Elabora una clase llamada Persona que siga las siguientes condiciones:

• Sus atributos son: nombre, edad, INE, sexo (H hombre, M mujer), peso y altura. No queremos que se accedan directamente a ellos. Piensa que modificador de acceso es el más





## EJERCICIOS DE PROGRAMACION

### Ejercicios resueltos en Java, Pseint, C#, C, C++, Python, Scilab, VisualFox, Pascal

adecuado, también en su tipo.

• Por defecto, todos los atributos menos el IFE serán valores por defecto según su tipo (0 números, cadena vacía para String, etc.). Sexo será hombre por defecto.

• Se implantarán constructores:

• Un constructor por defecto.

• Un constructor con el nombre, edad y sexo, el resto por defecto.

• Un constructor con todos los atributos como parámetro.

• Los métodos que se implementarán son:

• `calcularIMC()`: calcula si la persona está en su peso ideal (peso en kg/(altura<sup>2</sup> en m)), si esta fórmula devuelve un valor menor que 20, la función devuelve un -1, si devuelve un número entre 20 y 25 (incluidos), significa que está por debajo de su peso ideal la función devuelve un 0 y si devuelve un valor mayor que 25 significa que tiene sobrepeso, la función devuelve un 1.

• `esMayorDeEdad()`: indica si es mayor de edad, devuelve un booleano.

• `comprobarSexo(char sexo)`: comprueba que el sexo introducido es correcto. Si no es correcto, será H.

• `ToString()`: devuelve toda la información del objeto.

• `generaINE()`: genera un número aleatorio de 8 cifras, genera a partir de este su número su letra correspondiente. Este método será invocado cuando se construya el objeto.

• Métodos set de cada parámetro, excepto de INE. Elabora una clase Empleado que herede de la clase Persona

• Sus atributos: Cargo, Departamento, horasTrabajadas y pagoPorHora

• Elabora constructores • Los métodos que se implementarán son: o `Calcular Sueldo` o `ToString` con todos los datos Elabora una clase Estudiante que herede de la clase Persona

• Sus atributos: Colegio, semestre

• Elabora constructores

• Los métodos que se implementarán son: o `ToString` Ahora, crea una clase principal que haga lo siguiente:

• Pide por teclado el nombre, la edad, sexo, peso y altura.

Crea 3 objetos de la clase Persona, el primer objeto obtendrá las anteriores variables pedidas por teclado, el segundo objeto obtendrá todos los datos en asignados desde el constructor con parámetros y el último por defecto, para este último utiliza los métodos set para darle a los atributos un valor.

• Crea 1 objeto de la clase Trabajador solicita los datos necesarios desde teclado

• Crea 1 objeto de la clase Estudiantes, asigna los valores, utiliza el constructor que reciba los parámetros a excepción de aquellos que deban ser asignados.

• Para cada objeto, deberá comprobar si está en su peso ideal, tiene sobrepeso o por debajo de su peso ideal con un mensaje. Para el caso de estudiantes se ha decidido que todos aquellos que presenten sobrepeso deberán iniciar un programa de entrenamiento y aquellos que estén por debajo del peso ideal deberán asistir con el nutricionista y seguir una dieta balanceada. Para los trabajadores se ha decidido que todos aquellos que estén fuera del peso ideal deberán asistir a pláticas con el objeto de atender esta situación de salud. Deberán indicar en el mensaje que debe hacer cada persona de acuerdo con lo mencionado.

• Indicar para cada objeto si es mayor de edad.

• Por último, mostrar la información de cada objeto.



## EJERCICIOS DE PROGRAMACION

### Ejercicios resueltos en Java, Pseint, C#, C, C++, Python, Scilab, VisualFox, Pascal

- 1314 Una compañía paga semanalmente a sus empleados, quienes se dividen en cuatro tipos: empleados asalariados que reciben un salario semanal fijo, sin importar el número de horas trabajadas; empleados por horas, que perciben un sueldo por hora y pago por tiempo extra (es decir, 1.5 veces la tarifa de su salario por horas), por todas las horas trabajadas que excedan a 40 horas; empleados por comisión, que perciben un porcentaje de sus ventas y empleados asalariados por comisión, que obtienen un salario base más un porcentaje de sus ventas. Para este periodo de pago, la compañía ha decidido recompensar a los empleados asalariados por comisión, agregando un 10% a sus salarios base. La compañía desea implementar una aplicación que realice sus cálculos de nómina en forma polimórfica. (VER IMAGEN).  
La clase Empleado tendrá los métodos calcularIngreso() e imprimir() como métodos virtuales puros (abstractos). El método calcularIngreso será abstracto dado que cada cálculo dependerá de la clase de empleado. El método imprimir deberá desplegar los datos de cada empleado.  
En tu clase Principal que contiene el método main() crea un vector de la clase Empleado. Permite el alta de empleados mediante un menú que te permita elegir el tipo de empleado.
- 1317 Ejercicio. Registro de clientes de un banco  
Sigue los pasos que se indican a continuación para realizar el ejercicio propuesto:  
a) Define una clase base "Persona" con los atributos generales de una persona. Identifica por lo menos 3 atributos para la clase.  
b) Paso 2. Establece 2 clases derivadas que hereden de tu clase Persona. Estas clases son Cliente y Empleado. Identifica por lo menos 3 atributos específicos para cada clase.  
c) Para cada una de las clases hijas define un método registro, que reciba como parámetros los atributos de la clase y los asigne a los miembros privados.  
d) Realiza el modelado de tus clases.  
e) Realiza un programa en C++ donde implementes las clases utilizando herencias.  
f) Utiliza encapsulamiento en alguna de las propiedades.
- 1319 Crear una superclase llamada Electrodomesticos con las siguientes características:  
Sus atributos son id (int), precio base (double), color (string), consumo energético (char, letras entre A y F) y peso (int).  
Por defecto, el color será blanco, el consumo energético será F, el precioBase es de \$1000 y el peso de 5 kg.  
Los colores disponibles son blanco, negro, rojo, azul y gris  
Los constructores que se implementarán serán: constructor por defecto; constructor con el precio y peso y el resto por defecto; constructor con todos los atributos.  
Los métodos que implementará serán:  
- `set` y `get` de todos los atributos (variables miembro). Valida datos, en el consumo energético (char letra): comprueba que la letra es correcta, sino es correcta usará la letra por defecto. En el caso del color comprueba que es correcto, sino lo es usa el color por defecto.  
- `calcularPrecio()`, se podrá sobrescribir en las subclasses, según el consumo energético y su tamaño aumentará su precio. Deberá regresar el valor del precio. Esta es la lista de precios: (ver imagen).  
Implementa una función miembro `imprimir()` para mostrar los datos, deberá ser una



## EJERCICIOS DE PROGRAMACION

### Ejercicios resueltos en Java, Pseint, C#, C, C++, Python, Scilab, VisualFox, Pascal

función virtual pura

Crea una subclase llamada Lavadora con las siguientes características:

Su atributo es carga (int). Por defecto, la carga es de 5 kg.

Los constructores que se implementarán serán: constructor por defecto; constructor con el precio y peso, el resto por defecto; constructor con la carga y el resto de atributos heredados

Los métodos que se implementará serán:

• Constructores

• Método set y get, imprimir

• calcularPrecio(), si tiene una carga mayor de 30 kg, aumentará el precio \$500, sino es así no se incrementará el precio.

Crea una subclase llamada Televisor con las siguientes características:

Sus atributos son resolución (int - en pulgadas-) y Smart (booleano). Por defecto, la resolución

es de 24 pulgadas y smart será falso.

Los constructores que se implementarán serán: constructor por defecto; constructor con el precio y peso. El resto por defecto; constructor con la resolución, el smart y el resto de atributos heredados

Los métodos que se implementará serán:

• Constructores

• Método set y get, imprimir

• calcularPrecio(), si tiene una resolución mayor de 40 pulgadas, se incrementará el precio

30% y si es Smart, se aumentará \$500.

En la Clase Principal realizar lo siguiente dentro del método main().

Deberás utilizar una lista de objetos de la clase Electrodomésticos

Utiliza un menú de opciones para realizar lo siguiente:

1. Alta
2. Consulta
3. Modificación
4. Baja
5. Salir

El alta de objetos podrá ser de la clase Lavadora y de la clase Televisor; inicializa todos los campos con valores apropiados.

La opción de consulta deberá desplegar los datos de cada objeto.

La opción de modificación podrá modificar el precio base y deberás buscar el objeto por su id

La opción de borrar eliminará el objeto de acuerdo al id seleccionado.

1328 Utilizando Programación Orientada a Objetos, crear un arreglo de que almacene n fechas, crear la clase Fecha con 2 constructores, uno con parámetros día,mes,año y otro sin parámetros; en este caso asignar la fecha del sistema. Ingresar una fecha dd, mm, aaaa y realizar la búsqueda secuencial y búsqueda binaria.

1368 TAREA 1: El departamento de recursos humanos requiere el desarrollo de una calculadora de salarios genérica donde se pueda registrar los salarios brutos (salario sin deducciones) de los empleados y determine el salario neto (salario menos deducciones).



## EJERCICIOS DE PROGRAMACION

### Ejercicios resueltos en Java, Pseint, C#, C, C++, Python, Scilab, VisualFox, Pascal

La calculadora debe ser capaz de calcular las deducciones de ley, las cuales actualmente son:  
Cuota del Seguro Social: corresponde a un rebajo del 5,5% sobre el salario bruto.

Cuota PensiÃ³n: corresponde un rebajo del 4,17% sobre el salario bruto

Ahorro obligatorio Banco Popular: corresponde a un rebajo del 1% sobre el salario bruto.

Impuesto de renta: rebajo que se calcula con base en los siguientes rangos:

Sobre el exceso de \$941.000 y hasta \$1.381.000, 10%.

Sobre el exceso de \$1.381.000 y hasta \$2.423.000, 15%.

Sobre el exceso de \$2.423.000 y hasta \$4.845.000, 20%.

Sobre el exceso de \$4.845.000, 25%.

De esta forma, se le estÃ¡ solicitando crear una clase cSalario donde se puede ingresar el salario bruto como parÃ¡metro al constructor. AdemÃ¡s debe ser capaz de calcular las deducciones indicadas anteriormente mediante mÃ©todos respectivos.

Finalmente se debe incluir en esta clase un mÃ©todo salarioNeto que calcule el monto del salario neto.

Para validar la correcta construcciÃ³n de la clase cSalario, se debe crear una clase principal (que contiene el mÃ©todo main) de defina un arreglo de cSalario que agregue salarios mientras el usuario indique que tiene mas salarios que ingresar.

Al final debe recorrer el arreglo, mostrando sus valores en pantalla (salario bruto, deducciones y salario neto) y calcular el monto promedio de salarios ingresados, el salario mayor, el salario menor y la sumatoria de todos los salarios y sus deducciones (conocer cuanto es el total de cada rubro).

1369 TAREA 2. Con base en la Tarea 1, el Departamento de Recursos Humanos requiere ampliar el uso de la calculadora de salarios, por lo que requiere aplicar las siguientes mejoras:

â€¢ Generar una clase nueva, cSalario\_Deducciones\_Voluntarias, la cual es una clase que hereda de la clase cSalario, que incluye el manejo de deducciones voluntarias al salario.

â€¢ Las nuevas deducciones, al ser opcionales, tienen la posibilidad de que el usuario indique en cada Ã­tem cuales deben ser utilizadas y cuales no.

â€¢ Las nuevas deducciones son:

- o AsociaciÃ³n Solidarista, rebajo del 3% del monto bruto del salario
- o Cooperativa, rebajo del 1% del monto bruto del Salario
- o Ahorro NavideÃ±o, rebajo de un monto porcentual indicado por el usuario.
- o Otros rebajos, rebajo de un monto fijo indicado por el usuario.

â€¢ Esta tarea debe aplicar los mismos requerimientos de usabilidad descritos en la tarea 1, por lo que usted debe validar cuales mÃ©todos de la clase padre deben ser modificados en la clase a desarrollar en esta tarea.

AdemÃ¡s se requiere que la tarea se ejecute sobre una interfaz grÃ¡fica (GUI), que permita al usuario visualizar todas las opciones del salario y poder seleccionar cuales de las deducciones opcionales requiere incluir en cada Ã­tem.

Para validar la correcta construcciÃ³n de la clase cSalario\_Deducciones\_Voluntarias, se debe crear una clase principal (que contiene el mÃ©todo main) donde se genere la instancia



## EJERCICIOS DE PROGRAMACION

**Ejercicios resueltos en Java, Pseint, C#, C, C++, Python, Scilab, VisualFox, Pascal**

de la clase asÃ- como los componentes de la interface grÃ;fica de la pantalla a desarrollar.

Al final de la ejecuciÃ³n, igual que en la tarea 1, se debe mostrar un resumen de los cÃ;culos efectuados, mostrando el promedio de salarios ingresados, el salario mayor, el salario menor y la sumatoria de todos los salarios y sus deducciones (conocer cuanto es el total de cada rubro)