



Carátula para entrega de prácticas

Facultad de Ingeniería

Laboratorio de docencia

Laboratorios de computación salas A y B

Profesor:

TISTA GARCÍA EDGAR

Asignatura:

PROGRAMACIÓN ORIENTADA A OBJETOS

Grupo:

3

No de Práctica(s):

PRACTICA #4 - CLASES Y OBJETOS

Integrante(s):

CARRILLO CERVANTES IVETTE ALEJANDRA

*No. de Equipo de
cómputo empleado:*

TRABAJO EN CASA

No. de Lista o Brigada:

08

Semestre:

2022 - 1

Fecha de entrega:

03 OCTUBRE 2021

Observaciones:

CALIFICACIÓN: _____

PRACTICA #4 – CLASES Y OBJETOS

OBJETIVO: Aplicar los conceptos básicos de la programación orientada a objetos en un lenguaje de programación.

Antes de comenzar la práctica, se activó la página de códigos 65001 desde la terminal, con el fin de que en los programas muestren los caracteres.

EJEMPLOS DE LA GUÍA

Ejemplo 1. Punto

El archivo “Punto.java” es una clase la cual tiene 2 métodos constructores, el primero es el método constructor vacío el cual sirve para crear un solo objeto sin inicializar los atributos de este, mientras que el segundo método tiene como parámetro el valor de “x” y “y” donde se situará el punto; también, tiene un método el cual se utilizará para imprimir los atributos que tiene cada punto, es decir, el valor de “x” y de “y” de cada uno de estos. El archivo “PruebaPunto.java” tiene como función crear 2 objetos tipo Punto, uno utilizando el primer constructor de la clase Punto y otro utilizando el segundo constructor de esta misma, posteriormente se imprimen los atributos de cada objeto utilizando el método imprimePunto().

```
C:\Users\aleja\OneDrive\Escritorio\FACULTAD\2022-1\P00\Prácticas\P4 P00\Carrillo Ce  
rvantes Ivette Alejandra G3 P4 V1>java PruebaPunto  
Punto[x = 5 ,y = 8]  
Punto[x = 2 ,y = 0]
```

./PruebaPunto.java

Ejemplo 2. Triángulo

El archivo “Triangulo.java” es una clase la cual tiene un método sobrecargado de 3 diferentes formas: el primero, no recibe ningún parámetro y solo retorna el valor (float) del área de un Triángulo; el segundo, recibe dos variables de tipo “float” como parámetro, la primera la base del triángulo y la segunda su altura, retorna el valor (float) del área de un triángulo según los valores dados; el tercero, recibe dos variables de tipo “int” como parámetro, la primera la base del triángulo y la segunda su altura, al igual que las anteriores, retorna el valor (float) del área de un triángulo según los valores dados. El archivo “PruebaTriangulo.java” tiene como función crear un objeto tipo Triangulo y probar cada método de la clase usando diferentes valores de la base y de la altura del triángulo según corresponda el método que se use.

```
C:\Users\aleja\OneDrive\Escritorio\FACULTAD\2022-1\P00\Prácticas\P4 P00\Carrillo Ce  
rvantes Ivette Alejandra G3 P4 V1>java PruebaTriangulo  
base:5.0  
altura:8.0  
area()-> 20.0  
area(6,2)-> 6.0  
area(5.5f,3.2f)-> 8.8
```

./PruebaTriangulo.java

ACTIVIDADES PARA EL DESARROLLO DE LA PRÁCTICA

Ejercicio 1. Computadora

Este ejercicio tiene como función diseñar un programa para la administración de computadoras para una tienda de tecnología; para su implementación, primero se creó un archivo llamado "Computadora.java", este archivo es una clase la cual tiene los siguientes 3 métodos constructores para crear un objeto tipo Computadora: el primer constructor, es un constructor vacío; el segundo constructor, cuenta con un parámetro tipo String el cual será el atributo "*nombre*" de la computadora que se va a crear; el tercer constructor, cuenta con 6 parámetros los cuales serán todos los atributos que contendrá cada computadora: nombre (String), marca (String), color (String), memoria RAM (int), precio (double) y descuento (boolean). Para diferenciar los parámetros de cada método constructor de los atributos de cada computadora se utilizó la referencia "*this*". Para llevar la cuenta de las Computadoras que existen en tienda, se creó una variable estática la cual pertenece a la clase Computadora y en cada vez que se llama a un método constructor, o bien, se crea un objeto tipo Computadora, va aumentando a uno esa variable; sin embargo, al momento de eliminar alguna computadora, va decrementando a uno dicha variable. La clase computadora, también cuenta con un método sobrecargado de 3 diferentes formas, este método se llama "descuento" y tiene la función de obtener el precio final de una computadora, dado un descuento (porcentaje más efectivo, efectivo, descuento inicial no modificable); igualmente, se creó un método para imprimir una computadora en específico.

Para probar el método sobrecargado de diferentes formas, se instanciaron tres Computadoras y para cada una de ellas se pasaron los parámetros correspondientes a cada método para probar su función, al igual se probó el método imprimirComputadora() con cada Computadora creada.

```
System.out.println("\n * * * * *");
computadora1.imprimirComputadora();
computadora1.descuento(20, 200);
computadoras.add(computadora1);
System.out.println("\n * * * * *");
computadora2.imprimirComputadora();
computadora2.descuento(500);
computadoras.add(computadora2);
System.out.println("\n * * * * *");
computadora3.imprimirComputadora();
computadora3.descuento();
computadoras.add(computadora3);
```

Método descuento

La salida de estas líneas de código es la siguiente:

```
C:\Users\aleja\OneDrive\Escritorio\FACULTAD\2022-1\P00\Prácticas\P4 P00\Carrillo Ce
rvantes Ivette Alejandra G3 P4 V2\Computadora>java Principal

Se crearon las siguientes computadoras:

* * * * *
Computadora Computadora1
Marca: Lenovo
Color: negra
memoriaRam: 8 GB
Precio: 18499.99
¿Tiene Descuento? true

La computadora tiene un descuento del 20% más un descuento de $200
El total de su compra es: 14599.992000000002
```

```

* * * * *
Computadora Computadora2
Marca: hp
Color: rojo
memoriaRam: 12 GB
Precio: 24700.0
¿Tiene Descuento? true

La computadora tiene un descuento de: $500
El total de su compra es: 24200.0

* * * * *
Computadora Computadora3
Marca: hp
Color: gris
memoriaRam: 6 GB
Precio: 14800.99
¿Tiene Descuento? true

La computadora tiene un descuento inicial del 10%
El total de su compra es: 13320.891000000001

```

Para que sea más eficiente la administración de computadoras, se realizó un menú con las siguientes opciones, en un nuevo archivo llamado “*Principal.java*” el cual contendrá el método principal:

- Crear nueva computadora
- Modificar existente
- Ver computadoras actuales
- Eliminar Computadora
- Salir

*Se explicará cada opción con una captura del pedazo de código que le corresponde.

Para la primera opción “*Crear computadora*” se le dio a elegir al usuario si quería crear la computadora e inicializar todos sus atributos, o bien, crear la computadora y solo inicializar el atributo nombre. Para la implementación en código de esta opción, se utilizó una estructura “*switch case*” donde el usuario pueda elegir como creará cada computadora y se mando a llamar al método constructor con los parámetros correspondientes según su elección.

```

Menú de Opciones

1) Crear nueva computadora
2) Modificar existente
3) Ver computadoras actuales
4) Eliminar computadora
5) Salir

Elige una opción: 1
-----

Crear nueva computadora

1 -> Crear una computadora con todos sus atributos
2 -> Crear una computadora solo con su nombre y los demás atributos inicializarlos en cero

Elige una opción: 1

Ingresa los datos de la computadora:
Nombre: Computadora4
Marca: hp
Color: negra
Memoria RAM: 8
Precio: 16400.89
¿Tiene descuento? (true, false): false

-> Se creó la siguiente computadora:

```

```

* * * * *
Computadora Computadora4
Marca: hp
Color: negra
memoriaRam: 8 GB
Precio: 16400.89
¿Tiene Descuento? false
Elige una opción: 1
-----

Crear nueva computadora

1 -> Crear una computadora con todos sus atributos
2 -> Crear una computadora solo con su nombre y los demás atributos inicializarlos en cero
Elige una opción: 2

Ingresa los datos de la computadora:
Nombre: Computadora5

-> Se creó la siguiente computadora:

* * * * *
Computadora Computadora5
Marca: null
Color: null
memoriaRam: 0 GB
Precio: 0.0
¿Tiene Descuento? false

```

Cabe destacar que, cada vez que se iba creando una computadora, se iba agregando a un ArrayList mediante el método “.add” el cual fue creado desde un principio.

Para la segunda opción “*Modificar existente*”, se le dio a elegir al usuario que atributo quiere modificar y a que computadora en especifica. Para implementar esto, se buscó mediante un ciclo “*foreach*” (el cual recorre todo el arraylist), el atributo nombre especificado por el usuario, comparando este con cada atributo “nombre” de cada objeto, el cual se obtuvo con el método getNombre(), este se agregó se agrego a la clase “Computadora.java”. Mientras que, para modificar los atributos de cada computadora, se utilizaron métodos de acceso *setters* para cada atributo, excepto al momento de modificar todos, ahí se creó un único método set para que fuera más eficiente la modificación.

```

Menú de Opciones

1) Crear nueva computadora
2) Modificar existente
3) Ver computadoras actuales
4) Eliminar computadora
5) Salir

Elige una opción: 2
-----

Modificar computadora

-> Escribe el nombre de la computadora que quieras modificar: Computadora5

1. Nombre
2. Marca
3. Color
4. Memoria Ram
5. Precio
6. Descuento
7. TODOS

¿Qué atributo quieres modificar? 7

```

```
Modificar todos los atributos de la computadora
-> Ingresa el nombre modificado: Computadora5
-> Ingresa la marca modificada: apple
-> Ingresa el nuevo color: blanca
-> Ingresa la nueva memoria RAM: 8
-> Ingresa el nuevo precio: 30000.00
-> ¿Tiene descuento?: false

*Se modificaron todos los atributos de la computadora Computadora5
```

Para la tercera opción “*Ver computadoras actuales*”, igualmente se recorrió el ArrayList el cual contiene todas las computadoras creadas, y en cada iteración se fue llamando al método “*imprimirComputadora*”.

```
Menú de Opciones
1) Crear nueva computadora
2) Modificar existente
3) Ver computadoras actuales
4) Eliminar computadora
5) Salir

Elige una opción: 3
-----

Computadoras actuales

-> El número de computadoras actuales son: 5

* * * * *
Computadora Computadora1
Marca: Lenovo
Color: negra
memoriaRam: 8 GB
Precio: 18499.99
¿Tiene Descuento? true

* * * * *
Computadora Computadora2
Marca: hp
Color: rojo
memoriaRam: 12 GB
Precio: 24700.0
¿Tiene Descuento? true

* * * * *
Computadora Computadora3
Marca: hp
Color: gris
memoriaRam: 6 GB
Precio: 14800.99
¿Tiene Descuento? true

* * * * *
Computadora Computadora4
Marca: hp
Color: negra
memoriaRam: 8 GB
Precio: 16400.89
¿Tiene Descuento? false

* * * * *
Computadora Computadora5
Marca: apple
Color: blanca
memoriaRam: 8 GB
Precio: 30000.0
¿Tiene Descuento? false

* * * * *
```

Finalmente, para la cuarta opción “*Eliminar computadora*”, se le dio a elegir al usuario que computadora eliminar y al igual que en las opciones anteriores, se recorrió el ArrayList el cual contiene todas las computadoras creadas y se fue comparando el nombre de cada computadora mediante el método `getNombre()`, con el nombre ingresado por el usuario y si estos coincidían, se eliminaba la computadora mediante el método “*remove()*”. Para comprobar la función de esta opción, se eliminaron las computadoras 1,3 y 5 antes creadas.

```
Menú de Opciones
1) Crear nueva computadora
2) Modificar existente
3) Ver computadoras actuales
4) Eliminar computadora
5) Salir
Elige una opción: 4
-----
Eliminar computadora
-> Escribe el nombre de la computadora que quieras eliminar: Computadora1
Eliminaste la computadora con nombre: Computadora1

Menú de Opciones
1) Crear nueva computadora
2) Modificar existente
3) Ver computadoras actuales
4) Eliminar computadora
5) Salir
Elige una opción: 4
-----
Eliminar computadora
-> Escribe el nombre de la computadora que quieras eliminar: Computadora3
Eliminaste la computadora con nombre: Computadora3

Menú de Opciones
1) Crear nueva computadora
2) Modificar existente
3) Ver computadoras actuales
4) Eliminar computadora
5) Salir
Elige una opción: 4
-----
Eliminar computadora
-> Escribe el nombre de la computadora que quieras eliminar: Computadora5
Eliminaste la computadora con nombre: Computadora5
```

Para comprobar que las Computadoras fueron eliminadas, se imprimieron todas las computadoras existentes:

```
Menú de Opciones
1) Crear nueva computadora
2) Modificar existente
3) Ver computadoras actuales
4) Eliminar computadora
5) Salir
Elige una opción: 4
```

```

-----
Eliminar computadora
-> Escribe el nombre de la computadora que quieras eliminar: Computadora5
Eliminaste la computadora con nombre: Computadora5
-----

Menú de Opciones
1) Crear nueva computadora
2) Modificar existente
3) Ver computadoras actuales
4) Eliminar computadora
5) Salir
Elige una opción: 3
-----

Computadoras actuales
-> El número de computadoras actuales son: 2

* * * * *
Computadora Computadora2
Marca: hp
Color: rojo
memoriaRam: 12 GB
Precio: 24700.0
¿Tiene Descuento? true

* * * * *
Computadora Computadora4
Marca: hp
Color: negra
memoriaRam: 8 GB
Precio: 16400.89
¿Tiene Descuento? false

* * * * *

Menú de Opciones
1) Crear nueva computadora
2) Modificar existente
3) Ver computadoras actuales
4) Eliminar computadora
5) Salir
Elige una opción: 5
-----

Seleccionaste salir

Byeeee (:

```

./Computadora./Principal.java

Conclusiones

Se cumplieron con los objetivos de esta práctica, ya que se aplicaron los conceptos básicos de la programación orientada a objetos, tales como son instanciar objetos y las clases, así como se vio más a detalle la implementación de referencias this, métodos de acceso y miembros de clase vistos en clase.

Se cumplieron con todas las actividades de la práctica; sin embargo, me costó un poco de trabajo encontrar la forma de obtener una computadora mediante un atributo, después recordé el método get y la tarea de la biblioteca Strings donde se buscaron los métodos más importantes de esta, utilicé un método descrito en dicha tarea “compareTo” para comparar dos cadenas String.

Considero que esta práctica tuvo los elementos necesarios para analizar los conceptos básicos de la Programación Orientada a Objetos. (: