



Universidad Nacional Autónoma de México

Facultad de Ingeniería



Laboratorio de docencia
Laboratorios de computación
Laboratorios de computación A y
B

Práctica #5 Fundamentos y Sintaxis del lenguaje

Profesor: Tista García Edgar

Asignatura: Programación Orientada a Objetos

Grupo: 3

No de Práctica(s): 5

Integrante(s): Félix Flores Paul Jaime

Semestre: 2019-2

Fecha de entrega: 13-03-2019

Observaciones:

CALIFICACIÓN: _____

OBJETIVO.

Aplicar el concepto de abstracción para el diseño de clases que integran una solución, utilizando el encapsulamiento para proteger la información y ocultar la implementación

DESARROLLO.

Ejercicio 1) Manual de laboratorio.

En esta primera practica del manual de laboratorio,hicimos una clase publica llamada Circulo a la igual íbamos a llamar mediante una instancia por medio de una instancia en la clase principal PruebaFiguras.

Si una variable de instancia se declara como *public*, cualquier método que tenga una referencia a un objeto que contenga esta variable de instancia podrá leer o escribir en ella.

```
Microsoft Windows [Versión 10.0.17134.590]
(c) 2018 Microsoft Corporation. Todos los derechos reservados.

C:\Users\andro>cd Desktop

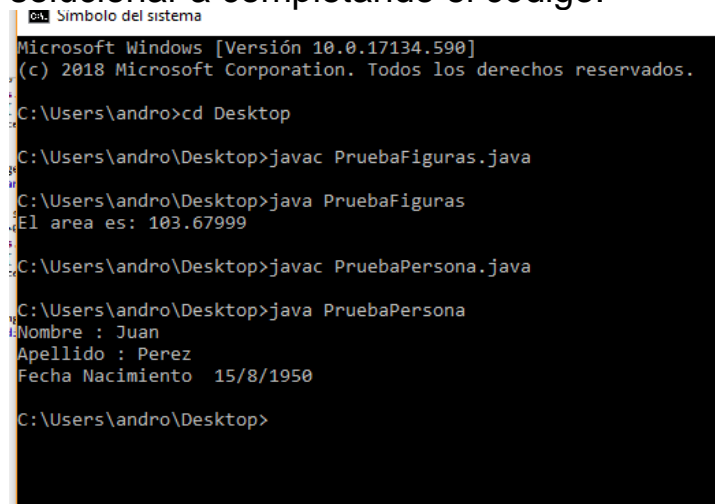
C:\Users\andro\Desktop>javac PruebaFiguras.java

C:\Users\andro\Desktop>java PruebaFiguras
El area es: 103.67999

C:\Users\andro\Desktop>_
```

Ejercicio 2) Manual de laboratorio

En esta parte creamos dos clases, la clase fecha y la clase persona, que van hacer llamadas mediante métodos dentro de la clase principal en esta parte usamos getters y setters para obtener y recibir información. En esta arte tuve problemas, ya que el código de la página estaba cortado y mal editado, me tardé mucho en descifrar porque marcaba muchos errores, pero lo pude solucionar a completando el código.



```
CS: Símbolo del sistema
Microsoft Windows [Versión 10.0.17134.590]
(c) 2018 Microsoft Corporation. Todos los derechos reservados.

C:\Users\andro>cd Desktop

C:\Users\andro\Desktop>javac PruebaFiguras.java

C:\Users\andro\Desktop>java PruebaFiguras
El area es: 103.67999

C:\Users\andro\Desktop>javac PruebaPersona.java

C:\Users\andro\Desktop>java PruebaPersona
Nombre : Juan
Apellido : Perez
Fecha Nacimiento  15/8/1950

C:\Users\andro\Desktop>
```

Ejercicio 1:

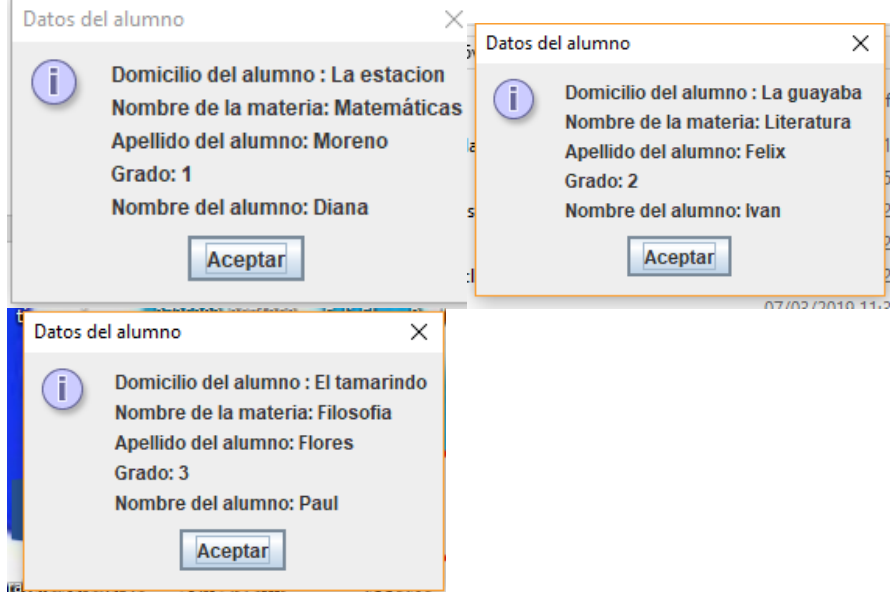
En este primer ejercicio tuve problemas para hacer dos paquetes, ya que no entendía cómo hacerlo, así que decidí solucionarlo metiendo todo dentro de un paquete. Mi programa funciona de esta manera, metí como super-clase a empleado, ya que de aquí iba a tomar los datos de todos los empleados y así usar extends para heredar a las otras clases, después herede todo a director, poniéndolo como privado el suelo extra y después pase todo a gerente el cual iba a llevar de la mano a operativos. En mi menú principal hice que el usuario metiera los datos del director, quería meter los demás datos, pero no sé por qué razón no me dejaba, así que decidí hacerlos estáticos, así de este modo use los constructores en cada clase para obtener la información de las clases heredadas.

The image displays five overlapping Java dialog boxes, each titled with a role and containing a list of attributes and an 'Aceptar' button. Each dialog box includes an information icon (i) in a blue circle.

- Datos del Directivo**
 - Salario Adicional: 1234456.0
 - Salario base: 500000.0
 - Nombre: Rafael
 - Apellido: Ramo
 - Puesto: Principal
 - Aceptar
- Datos del Gerente**
 - Area: dos
 - Subordinados: 2
 - Salario Base: 1982.99
 - Nombre: Denisse
 - Apellido: Miranda
 - Puesto: Administradora
 - Aceptar
- Datos del Gerente** (second instance)
 - Area: dos
 - Subordinados: 2
 - Salario Base: 1253.99
 - Nombre: Diana
 - Apellido: Garcia
 - Puesto: Administradora
 - Aceptar
- Datos del operativo** (first instance)
 - Turno: Matutino
 - Jornada laboral: 8
 - Area: tres
 - Subordinados: 1
 - Salario Base: 100.99
 - Nombre: Ivan
 - Apellido: Moreno
 - Puesto: Cargador
 - Aceptar
- Datos del operativo** (second instance)
 - Turno: Vespertino
 - Jornada laboral: 24
 - Area: tres
 - Subordinados: 1
 - Salario Base: 500.99
 - Nombre: Guillermo
 - Apellido: Flores
 - Puesto: Supervisor
 - Aceptar
- Datos del operativo** (third instance)
 - Turno: Nocturno
 - Jornada laboral: 12
 - Area: tres
 - Subordinados: 1
 - Salario Base: 500.99
 - Nombre: Alejandra
 - Apellido: Martinez
 - Puesto: Guardia de seguridad
 - Aceptar

Ejercicio 2)

En el ejercicio 2, era parecido al anterior pero más sencillo, decidí hacerlos estáticos, porque igual tenía el problema de mandarlos a llamar uno por uno, también use la herencia, definí los atributos de cada clase como se pueden ver en las capturas de pantalla y mandarlos a llamar dentro de la clase principal.



Después use los getters y setters para modificar la información de las materias y el domicilio

```
run:
Obteniendo materia adicional del alumno 2...Espanol
Obteniendo el domicilio adicional del alumno 2...Montes
Obteniendo materia adicional del alumno 3...Computacion
Obteniendo el domicilio adicional del alumno 3...Pedregal
```

Conclusiones

Después de realizar la presente práctica, se puede concluir que se cumplieron los objetivos, debido a que se logró identificar el uso de las diferentes clases que nos ofrece POO, así como el uso de clases envolventes vistas en clase. Se logró comprenderlos al implementarlos en Java, así como un poco del conocimiento de los getters y setters.

Con los ejercicios vistos en clases y al aplicarlos al lenguaje de programación orientado a objetos, los conceptos quedaron más claros.

En cuanto la dificultad de los ejercicios, tuve complicación en la implementación de trabajar en dos paquetes al mismo tiempo, al final no me salió como yo hubiese querido.

Solucione mis problemas viendo tutoriales de youtube, aunque no estaban tan completos, me quedo un poco más claro el uso de clases, así como el encapsulamiento, como el uso de los getters y setters.

Los ejercicios de la práctica fueron muy buenos para comprender de una mejor manera el funcionamiento del encapsulamiento, me gusto ver la parte de heredación ya que se me hizo muy interesante como es que pasaban sus atributos dependiendo del tipo de clase, sin duda creo que puedo mejorar mi programa con nuevos conocimientos, los ejercicios fueron buenos para poner en práctica los conocimientos adquiridos en la materia de Programación Orientada a Objetos.