



Carátula para entrega de prácticas

Facultad de Ingeniería

Laboratorios de docencia

Laboratorio de Computación Salas A y B

Profesor(a): _____

Asignatura: _____

Grupo: _____

No de Práctica(s): _____

Integrante(s): 320095672

*No. de lista o
brigada:* _____

Semestre: _____

Fecha de entrega: _____

Observaciones: _____

CALIFICACIÓN: _____

Índice

Índice.....	1
Introducción.....	2
Ejercicio 1.....	2
Ejercicio 2.....	2
Práctica 7.....	2
Marco teórico.....	2
Desarrollo.....	3
Ejercicio 0.....	3
Ejercicio 1.....	3
Práctica 7.....	4
Resultados:.....	4
Ejecución del Ejercicio 0:.....	4
Ejecución del Ejercicio 1:.....	5
Ejecución de la Práctica 7:.....	5
Conclusiones.....	6
Referencias.....	7

Introducción

Ejercicio 1

Escribir un programa en Java para crear una clase llamada Figura con un método llamado `area()`. Se crea una clase derivada Rectángulo que sobrescribe el método `area()` para calcular el área. Y otra clase derivada Círculo que también sobrescribe el método `area()`.

Ejercicio 2

Escribir un programa en Java para crear la clase CuentaBanco con los métodos `depositar()` y `retirar()`. Se crea una clase derivada CuentaAhorro con el método sobrescrito `retirar()` que previene retiros si la cuenta no posee saldo mayor a \$100.

Práctica 7

Escribir un programa en Java para crear una jerarquía de clases para empleados de una compañía.

La clase base deberá ser Empleado, con clases derivadas Manager, Desarrollador y Programador. Cada subclase debe tener los atributos nombre, dirección, salario y nombre de trabajo. Además de implementar métodos de sobreescritura para calcular bonos, generar reportes de desempeño, y manejo de proyectos. Nota: Los métodos para generar reportes y manejo de proyectos se pueden representar con cadenas.

La realización de esta práctica nos ayuda a tener una noción sobre la herencia, el ahorro de código y la creación de jerarquías por medio de Heredar atributos y comportamientos. Dando solución a los problemas podéis modelar entidades y que estas se extiendan para dar lugar a otras entidades con mismas características.

Marco teórico

Durante la solución del problema, surge la necesidad de crear elementos que provengan de un elemento padre, es por eso que existe la herencia.

Se define a la herencia como “ forma de reutilización de software en la que se crea una nueva clase absorbiendo los miembros de una clase existente, y se mejoran con nuevas capacidades, o con modificaciones en las capacidades ya existentes “ [1]. Lo anterior indica que nosotros podremos crear entidades que provengan de otra entidad padre, pudiendo añadir más comportamientos extras de los que ya se tienen.

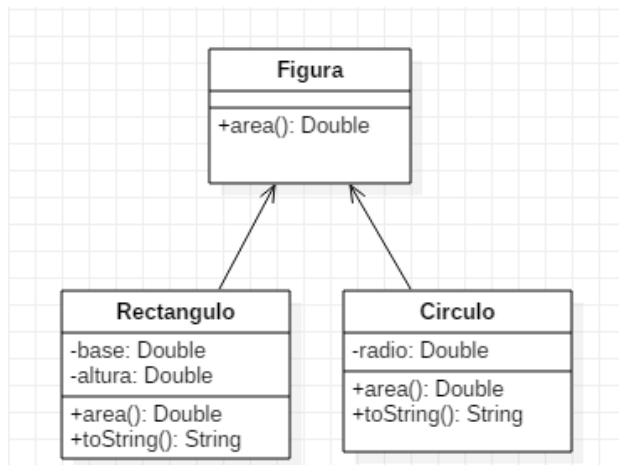
A veces necesitamos sobrescribir algunos métodos que ya están definidos para nuestro uso personal de la clase, la práctica de esta sobreescritura tiene nombre de Polimorfismo. El

polimorfismo se refiere a la capacidad de tomar varias formas o de poder comportarse de diferentes maneras según se necesite, lo usaremos para poder sobrecargar y sobrescribir.[2]

Desarrollo

Ejercicio 0.

Para este ejercicio se crearon las siguientes clases:

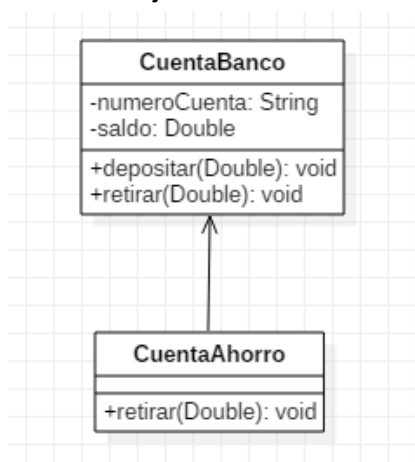


Tenemos a la clase padre, **Figura**, donde hereda a 2 clases: **Rectangulo** y **Circulo**.

Figura solo tiene el método `area`, **Rectangulo** crea sus atributos y sobrescribe los métodos `area` y `toString`, **Circulo** igual crea su atributo y de igual forma sobrescribe el método `area` y `toString`.

Ejercicio 1.

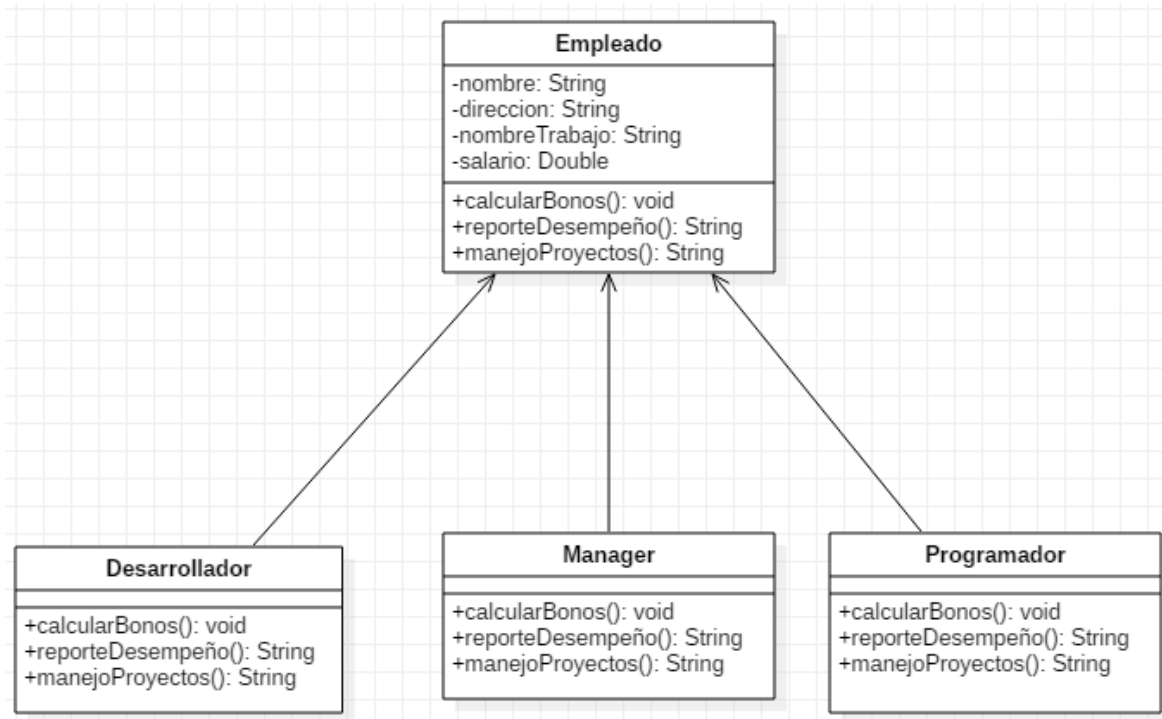
Para este ejercicio se crearon las siguientes clases:



Se tiene la clase padre **CuentaBanco** con sus atributos y sus métodos, también se tiene la clase heredada **CuentaAhorro** que solamente sobrescribe el método `retira`.

Práctica 7.

En esta práctica se crearon las siguientes clases:



Donde se tiene a la clase base Empleado con sus respectivos atributos y métodos. Aparecen también las clases heredadas Desarrollador, Manager y Programador, donde solamente sobrescriben los métodos indicados. Se tiene implícito los métodos getters y setters.

Resultados:

Ejecución del Ejercicio 0:

```
[luillilol@tostadora E0]$ java mx.unam.fi.poo.g1.p7.Ejercicio0
El area del rectangulo es: 30.0

El area del circulo es: 78.53981633974483
[luillilol@tostadora E0]$
```

Se crean los objetos Circulo y Rectangulo y se calcula el área para mostrarla. Ambos métodos son heredados de la clase Figura aunque esta clase no se muestra.

Ejecución del Ejercicio 1:

```
[luillilol@tostadora E1]$ java mx.unam.fi.poo.g1.p7.Ejercicio1
Se crea un objeto CuentaBanco (C/b No. CB1234) con un saldo inicial de $500
Se depositan $1000 a la cuenta CB1234
Saldo nuevo: 1500.0
Se retiran $600 de la cuenta CB1234
Saldo nuevo: 900.0

    Creando un objeto CuentaAhorro (C/a No. CA1000) con un saldo inicial de $ 450
Se retiran $300 de la cuenta CA1000
Saldo nuevo: 150.0

    Creando un objeto CuentaAhorro (C/a No. CA1001) con un saldo inicial de $300
Se intenta retiro por $250...
Se requiere un saldo de al menos 100
Saldo actual: 300.0
[luillilol@tostadora E1]$ |
```

Se crean objetos de clase CuentaBanco y también de CuentaAhorro, se van mostrando sus datos y también se van haciendo movimientos de “dinero” en las cuentas mediante los métodos que se crearon.

Ejecución de la Práctica 7:

```
[luillilol@tostadora P7]$ java -jar Practica7.jar
Mánager:
Se otorga un bono al manager:

Nombre: Luis Falcón
Puesto: Manager
Dirección: Nezahualcóyotl
Salario: 104000.0

- El desempeño del Mánager fue bueno, mejor que el promedio
- Supervisando al equipo de desarrolladores y programadores encargados de crear la nueva aplicación móvil.

Desarrollador:
Se otorga un bono al desarrollador:

Nombre: Fernando Samuel
Puesto: Desarrollador
Dirección: Av. Ermita
Salario: 60000.0

- El desempeño del Desarrollador fue normal, los resultados no salieron de la expectativa y no hubo ningún problema.
- Trabajando actualmente en la planificación de un software completo para el registro de alumnos a un sistema nacional de colegios

Programador:
Se otorga un bono al programador:
```

```
Programador:
Se otorga un bono al programador:

Nombre: Andreco Juarez
Puesto: Programador
Dirección: Tláhuac
Salario: 33000.0

- El desempeño del Programador es muy malo, no es eficaz y es muy lento en los proyectos que la
empresa necesita
- Trabajando en la corrección del desastre de código que otro compañero programador hizo
[luillilol@tostadora P7]$ |
```

Se crean 3 objetos, 1 de Manager, 1 de Programador y 1 de Desarrollador.

Se muestran sus datos y también se ocupan los métodos creados, como el de calcular bonos, reporte desempeño y manejo de proyectos.

Conclusiones

Gracias a los conceptos vistos en la sección del Marco Teórico se pudieron crear ciertas entidades que pudieron heredar a otras para su uso. La entidad padre cumplió la función de crear un “modelo” para crear a las demás entidades con sus atributos y comportamientos ya establecidos, con la libertad de modificarlos. Es importante destacar que estos conocimientos son de gran importancia al momento de resolver problemas, ya que nos otorgan una organización mejor y también facilita su solución. Mediante se va avanzando de prácticas, los conocimientos vistos se van acumulando y cada vez se tiene un mejor dominio de este, facilitando el estudio de los conceptos siguientes.

.

Referencias.

[1] P. J. Deitel and H. M. Deitel, *Cómo programar JAVA*. 7th. México: PEARSON EDUCATION, 2008, p. 379.

[2] A. Parra (2023, Mar 23) . Polimorfismo en Java [Online]. Available:
<https://www.programandojava.com/blog/polimorfismo-en-java/>