



Universidad Nacional de San Luis
Área de Programación y Metodologías de Desarrollo de Software
Ingeniería en Informática – Ingeniería en Computación
Programación II
Práctico N° 4
Lenguaje Java: Clases Abstractas e Interfaces

Ejercicio 1: define los siguientes conceptos: a) Interfaz, b) Clase Abstracta, c) Ligadura Dinámica, d) Ligadura Estática. ¿Qué es “herencia múltiple”?

Ejercicio 2: ¿Cuáles son las diferencias y similitudes entre clases abstractas e interfaces? ¿Por qué se relaciona el concepto de interfaces con el de herencia múltiple?

Ejercicio 3: define la clase *Persona* (Abstracta) y las subclases *LicEnComputacion* e *IngInformatico*. Además de todos los atributos y métodos que consideres necesarios, se debe definir el método *titulo()* el cual retorna un String con el título del profesional y en caso de que no posea devuelve “Sr.”. Luego, construye un programa principal que: i) permita al usuario ingresar *n* licenciados o ingenieros en un mismo arreglo, y ii) muestre los nombres de las personas que están en el arreglo anteponiendo su título.

Ejercicio 4: define la clase abstracta *Figura* y las subclases *Triangulo* y *Circulo*. Para realizar una buena definición ten en cuenta: a) ¿Qué atributos son comunes? y b) ¿Qué métodos deberían ser abstractos? Las subclases deben ser capaces de mostrar todos los atributos de la misma. ¿Cómo sería la reestructuración si existieran las clases *Equilátero*, *Isósceles* y *Escaleno*?

Ejercicio 5: define la clase abstracta *Barco* y las subclases *Crucero* y *Carguero*. Define todos los atributos y métodos que consideres necesarios. Luego construye un programa principal que: i) permita al usuario ingresar *n* cruceros o cargueros en un mismo arreglo, y ii) muestre el contenido del arreglo.

Ejercicio 6: desarrolla la clase *Puerto* que permite mantener el *nombre del puerto*, *dirección*, *código postal* y un arreglo que mantenga los barcos atracados. Además de los métodos que consideres necesarios, implementa los siguientes métodos para esta clase:

- a) Un método que muestre por pantalla los cruceros en el puerto.
- b) Un método que muestre por pantalla los cargueros en el puerto.
- c) *cargarNBarcos(...)* que permite agregar barcos al puerto a partir de un arreglo de barcos pasado como parámetro.
- d) *cantidadCruceros()* que muestra por pantalla la cantidad de cruceros en el puerto.
- e) Desarrolla un programa principal que permita crear dos puertos mediante un menú de opciones. El menú debe posibilitar ingresar *n* barcos al puerto determinado por el usuario, agregar los barcos de un puerto a otro y mostrar los barcos de un puerto según lo determine el usuario.

Ejercicio 7: usando lo desarrollado en el ejercicio anterior, implementa el programa principal y el menú de opciones para que permita al usuario definir *n* puertos y administrarlos.

Ejercicio 8: realiza las siguientes actividades:

1. Define una clase abstracta *Cuenta* con los siguientes atributos: *numeroCuenta*, *saldo* y *cliente*

(atributo de la clase Persona). Cuenta tiene los siguientes métodos:

1. constructor parametrizado que recibe un cliente y un número de cuenta;
 2. abstract retirar(double) que permitirá sacar una cantidad de la cuenta (si hay saldo).
 3. abstract actualizarSaldo(): actualizará el saldo de la cuenta, pero cada cuenta lo hace de una forma diferente teniendo en cuenta el interés.
2. Define las subclases de Cuenta que se describen a continuación:
1. CuentaCorriente: cuenta normal con un interés fijo del 1.5%.
 2. CuentaAhorro: esta cuenta tiene como atributos el interés variable a lo largo del año y un saldo mínimo necesario. Al retirar dinero hay que tener en cuenta que no se sobrepase el saldo mínimo.

Define los métodos que creas necesarios en cada clase y desarrolla un programa que cree dos cuentas de distintos tipos y pruebe sus características.

Ejercicio 9: crea una jerarquía de clases donde la clase abstracta *Ave* sea la clase padre y *Canario* y *Paloma* las subclases; en conjunto define todos los atributos y métodos necesarios. Por otra parte considera la jerarquía *Persona* y las subclases *LicEnComputacion* e *IngInformatico*, implementadas en el ejercicio 3. Define la interfaz *Cantante* que posee un único método *canta()*. ¿Cuáles de las clases en las jerarquías anteriores deberían implementar esta interfaz?

Ejercicio 10: construye una interfaz *Comparador* que incluya los siguientes métodos:

- a) *esMayor(Object b)* devuelve verdadero si el objeto que envía el mensaje es mayor que *b*.
- b) *esMenor(Object b)* devuelve verdadero si el objeto que envía el mensaje es menor que *b*.
- c) *esIgual(Object b)* devuelve verdadero si el objeto que envía el mensaje es igual que *b*.

Posteriormente que las clases *Persona*, *Barco* y *Materia* implementen dicha interfaz considerando el criterio de comparación que más prefieras.

Ejercicio 11: define los siguientes elementos:

- a) interface *Puerta* con los métodos *abrir* y *cerrar*.
- b) interface *PuertaBloqueable* derivada de *Puerta*, con los métodos *bloquea* y *desbloquea*.
- c) interface *Alarma* con los métodos *alarmaActivada?*, *activarAlarma* y *desactivarAlarma*.
- d) clase *ComponentedeCoche* con los atributos *descripción*, *peso* y *coste*, y un método que muestre dichos atributos.
- e) clase *PuertaCoche*, con el atributo boolean *estaBloqueada*, y que extienda la clase *ComponentedeCoche* e implemente las interfaces *Alarma* y *Puertabloqueable*.

Escriba un programa que pruebe la clase *PuertaCoche* y todas las operaciones que admite.

Ejercicio 12: Realice las siguientes actividades:

1. Defina la clase *Jugador* la cual almacena el nombre y el deporte que practica un jugador.
2. Defina la clase *JugadorDeBasquet* la cual permite registrar la cantidad de triples que ha encestado un jugador.
3. Defina la clase *JugadorDeRugby* la cual permite registrar la cantidad de tackles que le han realizado a un jugador.
4. Defina un programa principal que:
 - a) Declare un arreglo de cinco jugadores.



- b) Incorpore al arreglo jugadores, jugadores de básquet y jugadores de rugby. El arreglo debe contener al menos un jugador de cada tipo.
- c) Imprima por pantalla los jugadores que no son de básquet ni de rugby.
- d) Imprima el arreglo.

Nota: En este ejercicio ud. debe definir todos los métodos necesarios para la correcta definición de las clases. Además, si ud. considera necesario definir métodos auxiliares para elaborar la solución puede llevar adelante dicha tarea.

Ejercicio 14: teniendo en mente la implementación del ejercicio 13 del práctico 3, agregue a la misma el uso de clase Abstracta. ¿Significativamente en qué cambia?