



**Universidad Nacional de San Luis**  
**Área de Programación y Metodologías de Desarrollo de Software**  
**Ingeniería en Informática – Ingeniería en Computación**  
**Programación II**  
**Práctico N° 3**  
**Lenguaje Java: Herencia y Polimorfismo**

**Ejercicio 1:** define los siguientes conceptos: a) Herencia, b) Subclase y c) Superclase y d) Polimorfismo.

**Ejercicio 2:** explique la diferencia entre this y super.

**Ejercicio 3:** explica cada uno de los modificadores de acceso a nivel de clases y miembros que provee Java. ¿Cuándo utilizaría el modificador de acceso protected?

**Ejercicio 4:** investiga cual será la consecuencia de declarar un método como final. ¿Qué pasa si es la clase la que se declara como final?

**Ejercicio 5:** define la clase Persona, la misma posee las siguientes variables de instancia: nombre, tipo de documento, número de documento, estado civil y fecha de nacimiento. Define los métodos observadores y modificadores para cada una de las variables de instancia. Defina un método que muestre por pantalla los datos de una persona.

**Ejercicio 6:** define la clase Estudiante a partir de la clase Persona (ejercicio anterior). La clase estudiante debe permitir almacenar: nombre, apellido, tipo documento, número documento, estado civil, carrera y año que cursa. Además, implementa los métodos necesarios para modificar la carrera y año que cursa el estudiante y los métodos para mostrar todos los datos del estudiante.

**Ejercicio 7:** define la clase EstudianteADistancia la cual, además de los datos de un Estudiante como el definido en el ejercicio anterior, tiene además la ciudad en la que reside el estudiante, usuario de aula virtual y contraseña para la misma. Realiza los métodos que consideres necesarios para hacer uso de esta clase y para mostrar todos los datos del estudiante a distancia.

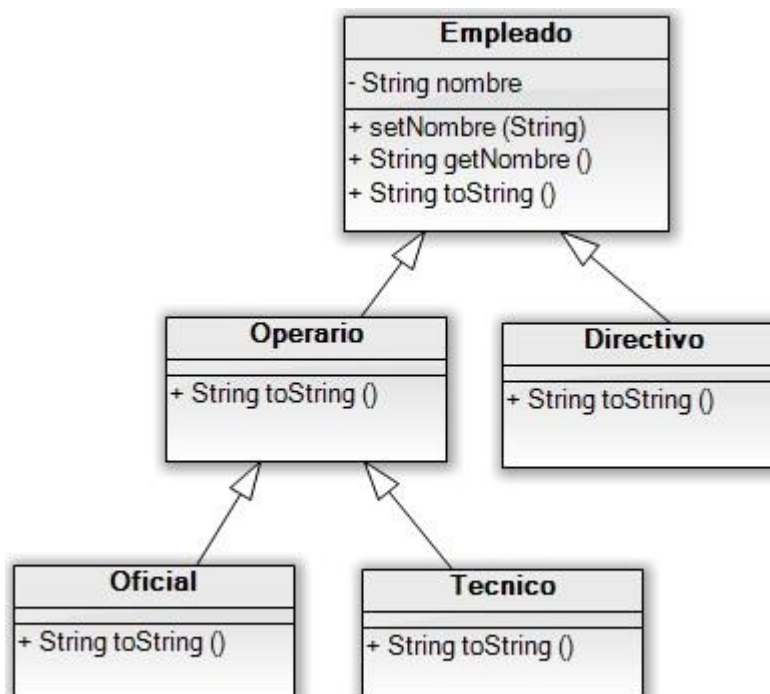
**Ejercicio 8:** desarrolla un programa que permita al usuario ingresar una cantidad n de estudiantes a distancia y completar todos sus campos. Una vez finalizada la carga, debe imprimir todos los estudiantes a distancia del año que el usuario indique.

*Nota:* tenga en cuenta el ejercicio anterior.

**Ejercicio 9:** construye una clase final Math2 que incluya funciones que devuelvan, respectivamente, el máximo, el mínimo, la sumatoria, la media aritmética y el promedio entero (se ajusta por el techo del valor) de un arreglo de números reales dado como parámetro.

*Nota:* una clase declarada como final no podrá ser “extendida”, es decir, no se puede heredar de dicha clase.

**Ejercicio 10:** Codifique la siguiente jerarquía de clases (clases, atributos y métodos) representada por el siguiente diagrama UML:



Nota: si desconoce la notación UML, tenga en cuenta que la flecha apunta al padre, es decir que por ejemplo la clase Operario es hija de la clase Empleado.

**Ejercicio 11:** Acceso a atributos y métodos. En base a los ejercicios 5, 6 y 7; determine en cada caso si es correcto o no y justifique.

1. Desde la clase Persona se puede acceder a los métodos definidos en los ejercicios previos.
2. Desde la clase Persona solo se puede acceder a los atributos públicos de la misma.
3. Desde la clase Estudiante se puede acceder a todos los atributos de la superclase Persona
4. La clase Estudiante hereda sólo los métodos y atributos públicos de Persona.
5. La clase EstudianteADistancia hereda todos los atributos y métodos de la clase Persona.
6. Un objeto de la clase Estudiante puede acceder a los métodos de la clase Persona.
7. Un objeto de la clase Persona puede acceder a los métodos de la clase Estudiante.

**Ejercicio 12:** una empresa de informática necesita llevar un registro de todos sus empleados que se encuentran en la oficina central, para esto define un programa en Java que incluye lo siguiente:

1. Clase Empleado con los siguientes atributos: nombre, dni (de tipo String), edad, casado (de tipo boolean) y salario. Además, se deben proveer los siguientes métodos:
  1. Constructor con y sin parámetros de entrada.
  2. Método que permita mostrar la clasificación según la edad de acuerdo al siguiente algoritmo: si edad es  $\leq 21$ , Principiante; si edad es  $\geq 22$  y  $\leq 35$ , Intermedio; si edad es  $> 35$ , entonces se debería clasificar al empleado como Senior.



3. Imprimir los datos del empleado por pantalla.
4. Un método que permita aumentar el salario en un porcentaje.
2. Clase Programador (especialización de Empleado). Clase que hereda de Empleado todos los atributos y métodos. Además tiene los atributos `lineasDeCodigoPorHora`, `lenguajeDominante` (de tipo String) y los constructores con y sin parámetros.

**Ejercicio 13:** defina e implemente lo siguiente:

1. Clase Nota. Una nota contiene un identificador numérico y una línea de texto. Defina constructor, getters, setters y toString.
2. Clase NotaAlarma. Una nota que además contiene la hora en la que sonará la alarma. Defina constructor, getters, setters y toString.
3. Clase BlocNotas que modela un bloc de notas en el que se pueden introducir notas, listar todas las notas, eliminar una nota mediante su posición en el bloc de notas o saber cuántas notas contiene el bloc de notas.
4. Clase Prueba que cree un bloc de Notas de ejemplo y pruebe las operaciones que soporta.

**Ejercicio 14:** implementa el ejercicio 14 del práctico 2 usando herencia y polimorfismo.