

Programación Avanzada

LABORATORIO 0

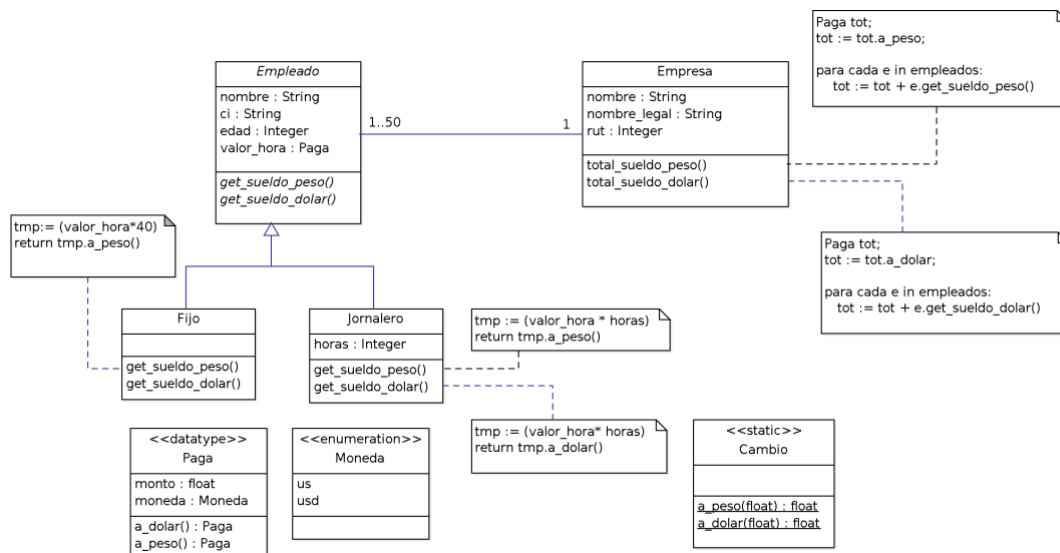
Consideraciones Generales:

- La entrega deberá realizarse hasta el **martes 4 de abril a las 12:00 hs.**

Con este laboratorio se espera que el estudiante adquiera competencias en la implementación de operaciones básicas, uso básico del lenguaje y reafirmar conceptos presentados en el curso.

Problema

Se desea implementar un sistema para el registro de sueldos de los empleados de una empresa. Para esto considerar el siguiente diagrama:



Ejercicio 1

Implementar en C++:

- 1) El enumerado *Moneda*, representado con los valores: "us" y "usd" para peso y dólar respectivamente.
- 2) La clase estática *Cambio*, con las siguientes operaciones:
 - a) `a_pesos(float importe)` : que dado un importe lo convierte a pesos utilizando una cotización fija¹.
 - b) `a_dolar(float importe)` : que dado un importe lo convierte a dólares utilizando una cotización fija.
- 3) El datatype *Paga*, que representará la paga realizada a los empleados.
 - a) El constructor por defecto deberá crear una *Paga* con monto 0 y moneda *us*.
 - b) Deberá contar con dos operaciones que utilizarán la clase estática *Cambio*:
 - i) `a_dolar()` : Retorna una *Paga* donde si la moneda es "us" realiza la conversión utilizando *Cambio*.
 - ii) `a_peso()` : Retorna una *Paga* donde si la moneda es "usd" realiza la conversión utilizando *Cambio*.
 - c) Agregar la sobrecarga de las siguientes operaciones:
 - i) Asignación (operador =)
 - ii) Multiplicación (operador *)
 - (1) Si realizamos: `Paga(120, Moneda::us) * 30` el resultado debe ser `Paga(3600, Moneda::us)`.
 - iii) Suma (operador +)
 - (1) Si realizamos: `Paga(3200, Moneda::us) + Paga(1200, Moneda::us)` el resultado debe ser `Paga(4400, Moneda::us)`.
 - (2) Si realizamos: `Paga(3200, Moneda::us) + Paga(500, Moneda::usd)` o en el orden inverso de moneda se debe lanzar una excepción correspondiente.
 - d) Agregar la sobrecarga del operador de inserción de flujo (ej. <<) en un objeto de tipo `std::ostream`. El mismo debe imprimir en el flujo de salida el datatype *Paga* con el siguiente formato: *Monto Moneda*
 Por ejemplo la sentencia `cout << Paga(15400, Moneda::usd) << "\n"` debe imprimir "15.400 usd" y un salto de línea en la salida estándar (observar la inclusión del carácter separador ".").
 - e) Realizar un procedimiento `main()` que permita, mediante un menú, probar todas las funcionalidades implementadas en las partes anteriores.

¹ Considerar una cotización constante de 40.15 pesos por dólar.

Ejercicio 2

Para poder llevar a cabo el registro de sueldos, se pide implementar en C++:

- 1) Las clases *Empleado* y *Fijo* y *Jornalero* como derivadas de la clase abstracta *Empleado*. Esta última tendrá como *pseudoatributo* un puntero a la clase *Empresa*.
- 2) La clase *Empresa*. Como se muestra en el diagrama, contiene un *pseudoatributo* que será un arreglo de *Empleado*. Considerarlo de tamaño fijo².
- 3) Realizar un procedimiento `main()` que permita, mediante un menú, dar de alta una empresa, sus respectivos empleados así como calcular el *total por concepto de sueldo* tanto en *dólares* como en *pesos*.

Nota:

En todos los casos utilice el manejo de excepciones para indicar la excepción que corresponda. Por ejemplo, cuando los parámetros indicados no sean válidos (ej. el constructor de *Paga* recibe un monto negativo), se lanzará la excepción `std::invalid_argument`. Utilice el siguiente enlace como referencia:

<http://www.cplusplus.com/doc/tutorial/exceptions>.

Por más información sobre C++: <http://www.cplusplus.com/doc/tutorial/>.

² Definir la constante `MAX_EMPLEADO` con el valor 50.