PATRONES SOFTWARE
LABORATORIO – P2
CONCEPTOS BÁSICOS / REFACTORIZACIÓN

Práctica 2

En esta práctica aplicaremos los conceptos de refactorización de código y aplicaremos los patrones fundamentales de Delegation, Interface y Abstract Superclass.

La **refactorización** de código se trata de la limpieza y optimización del código. Esta no se encarga de arreglar errores ni incorporar nuevas funcionalidades. Se podría decir que *altera la estructura interna del código sin cambiar su comportamiento externo*.

Enunciado:

Una universidad necesita una aplicación capaz de gestionar su personal. Existen dos tipos de personal: **(1)** personal docente e investigador (PDI) y **(2)** personal de administración y servicios (PAS).

El PDI se identifica a través de un número. Este conjunto pertenece a un departamento y a un área. Por ejemplo, el PDI "123" podría pertenecer al Dept. de Ciencias de la Computación, y estar dentro del Área de Lenguajes y Sistemas informáticos.

El PAS se identifica a través de un número. El conjunto pertenece a una sección y ocupa un cargo. Por ejemplo, el PAS "456" podría pertenecer a la sección de la Biblioteca Politécnica y ocupar el cargo de dirección.

Dentro de la gestión de la Universidad las tareas que se llevan a cabo son las siguientes:

- (1) Contratar / (2) Jubilar / (3) Modificar los datos de / (4) Mostrar los datos de PAS o PDI.
- Generar la nómina de todo el personal

La interfaz que se ofrece en primera instancia en la aplicación será de tipo texto y a través de menús.

Partiendo del código contenido en el proyecto "LabPat2", detecta los principales problemas que este presenta desde el punto de vista de la reutilización y optimización de código, y realiza una profunda refactorización del mismo. Se espera que **(como mínimo)** consigas:

- Crear una clase abstracta "Personal" que sea superclase de las clases PDI y PAS, con un atributo identificador, y que contenga un método abstracto para generar la nómina.
- Aplicar polimorfismo empleando *interfaces* de colecciones en vez de sus implementaciones. Además, parametrizar adecuadamente las mismas.
- Agrupar las interacciones por pantalla. Tener múltiples módulos que imprimen por pantalla no es mantenible. Se recomienda concentrar todos los mensajes e interacción con el usuario en la clase "MenuUniversidad".
- Realizar un correcto tratamiento de errores.
- Aplicar el patrón fundamental de *delegation e interface* para realizar el cálculo de la nómina. Se recomienda:
    - Crear una interfaz con un método para calcular la nómina.
    - Crear una clase para la nómina que implemente la interfaz y calcule la nómina dependiendo del tipo de personal. Para PDI, pagará una cantidad aleatoria entre 1000€ y 2000€. Para PAS, pagará una cantidad aleatoria entre 1000€ y 1500€.

Con el diseño que acabamos de aplicar, ¿cómo podrías incorporar una nómina especial para las pagas extra donde sumar 1000€ más a la anterior?