Práctica 1:

Asignatura: Programación Web, 3º Grado de Ingeniería Informática
Escuela Politécnica Superior de Córdoba - Universidad de Córdoba
2023 - 2024

Trabajo realizado por:
-Enrique de los Reyes Montilla
i12remoe@uco.es
-Manuel García Obrero
i82gaobm@uco.es
-Francisco José Mellado Ortiz
i12meorf@uco.es
-Lucía Téllez López
i12telol@uco.es

Para la realización de este informe lo vamos a dividir en 4 grandes bloques. El primero es, las clases, Persona, Monitor, Asistente, Actividad, Campamento e Inscripción. El segundo, son todas las clases Gestor. Tercero, el main y los ficheros. El cuarto documentación:

-En el primer bloque hemos decidido las siguientes implementaciones:

- En las clases Persona con Monitor y Asistentes se ha considerado que, dado que tienen características muy similares, se creó una clase abstracta con dichas características y estas heredan de ellas.
- En la clase Inscripciones(factoría) y las de registros(producto) se ha utilizado un patrón de diseño Abstract Factory
- Por otro lado, en Campamentos tuvimos que hablar con la profesora ya que tenían puesto en constructor parametrizado que hagamos una lista de Asistentes. La cuál, decidimos no ponerla ya que sería más óptimo que esta la consigamos a partir de los gestores.
- Hemos añadido solo a las variables internas al final un "_" debido a que como vimos el año pasado en la asignatura de Programación Orientado a Objetos, es una buena práctica para diferenciar una variable privada de una declarada en la función. Además, de así no tener muchos nombres diferentes y siempre usar el mismo con la diferenciación del "_".

-En el segundo bloque hemos decidido las siguientes implementaciones:

- En todos los gestores hemos implementado el patrón de diseño Singleton.
- También, hemos decidido que estos gestores puedan modificar las variables ID de las clases del primer bloque. Esto es debido, a que ahora no puede tener mucho sentido que se cambie, solamente sea porque nos hayamos equivocado escribiendo. Pero para un futuro, pensamos crear un modo usuario y un modo administrador y el administrador si pueda cambiarlo. En cambio el usuario no.
- Se ha creado varios "Exception" para poder finalizar el programa si se hace introduce algo que no se debiera, para poder asegurar la integridad de los datos, haciendo que sale un error y se finalice el programa sin ningún guardado de los datos actuales, ya que estarían erróneos.

-En el tercer bloque hemos decidido las siguientes implementaciones:

- En el main hemos decidido crear un bucle con un menú con todas nuestras funcionalidades que hemos hecho. En este menú irás moviéndote a partir de enviar pon pantallas un número. Este número corresponderá alguna función que inmediatamente se ejecutará y mostrará por pantalla el resultado, volviendo del nuevo al bucle principal.
- Además, cada vez que abre el programa, cargaremos en listas todo lo que tenemos en los ficheros y antes de cerrad lo volveremos a volcar. Lo que significa que estos solo se usarán al principio y al final del programa para que nuestra "base de datos" sea más integra.

- También para poder empezar a funcionar, hay que revisar si están bien la ruta de los ficheros, que se encuentra al principio del main, con la vuestra. Ya que si no estos ficheros no funciona y daría error por no encontrarlos, debido a que hemos puesto la ruta nuestra personal. Igualmente, como advertencia, hemos enviado los ficheros vacíos para que vosotros podáis hacer las pruebas desde cero.

-Por último el cuarto bloque, como documentación hemos estado usando Javadoc para resolver nuestras dudas y sacar información útil para el trabajo. Un ejemplo puede ser, de los paquetes y clases de java: https://docs.oracle.com/javase/8/docs/api/