<https://github.com/vmware/singleton.git>

Información general del proyecto:

Este proyecto es un archivo de java que utiliza el framework de Spring.boot para conectarse a tanto la base de datos de una página web, como para manipular los datos de empleados. Este proyecto usa el patrón estructural de mvc para ordenar su código y de la mano del framework aplican el patrón en tecnologías web.

Información y estructura del fragmento de donde se usa MVC:

El proyecto tech-jobs-mvc-master es un proyecto hecho en java que utiliza el framework de springframework.boot, para acceder de manera veloz a la información de trabajadores en una base de datos. La aplicación empieza en el TechJobsApplication, donde se llama la función SpringApplication.run() donde se empieza a correr la aplicación en el framework y se especifica que es el inicio de la aplicación con el comando @SpringBootAplication. Luego, en un paquete que se llama org.launchcode.controllers se especifica que los archivos son controladores con el comando de Spring.boot @Controller. Este tipo de archivo se conecta a páginas web con vista. En el archivo @RequestMapping, sirve para traducir las funciones definidas en el archivo a un formato utilizable en HTTP. En general, este controlador se encarga de manejar solicitudes y mandarlos al modelo y retornar la vista “index”, decide que se le muestra al usuario. Por otro lado, el controlador ListController se encarga de manejar las solicitudes relacionadas con filtrar trabajos en el archivo según los inputs que el usuario haya dado. Para eso al igual que el controlador pasado, utiliza los comandos @Controller y @RequestMapping entre otros y almacena la información relevante en el HashMap de nombre columnChoices. Finalmente, el controlador SearchController aunque en implementación está incompleto, se supone que debe manejar y procesar los datos que el usuario mete en la vista. Para ese fin, se usa el comando de spring.boot @RequestParam.

En el archivo JobData.java es el archivo que funciona como modelo del proyecto. En este archivo hay métodos que manejan las siguientes funciones: carga de datos *(loadData()*) , métodos para traer datos *(findAll() , findbycolummnandvalue() , findByValue()* ). En resumen estos métodos permiten buscar y filtrar de diferentes maneras. Cabe recalcar, que en este proyecto, la viststa está generada por la misma página web, que está implementada fuera de estre proyecto en particular. No obstante, si se puede decir que el proyecto implementa el patrón de diseño MVC.



Infromación del patron MVC:

El patrón MVC es un patrón estructural utlilizado en el desarrollo de software para dividir el código en tres componentes principales. El Modelo se dedica a implementar la lógica del proyecto y almacenar los datos que se utilizarán en el proyecto. La Vista, es la parte del código que determina cómo un usuario va a interactuar con la aplicación. Existen vistas por consola o GUI que pueden incluir páginas web, o ventanas diseñadas para la aplicación. El controlador es un intermediario entre la vista y el modelo, esta parte se encarga de “digerir” la información de entrada en un formato que sea amigable para el modelo.

Preguntas del PDF:

¿Por qué tiene sentido haber utilizado el patrón en este proyecto?¿Que ventajas tiene?

El patrón MVC es muy útil para que el desarrollo de código sea mucho más fácil de implementar y que para un tercero leer. En el desarrollo de páginas web, es muy importante, pues permite a un desarrollar partir el problema en piezas, la parte con la que un usuario interactúa , la parte en la que la aplicación cumple sus funciones, y la que comunica ambas. Esta metodología funciona para reducir el acoplamiento y que el principio de single responsibility se cumpla con mayor facilidad.

¿Qué desventajas tiene haber utilizado el patrón?

En una aplicación pequeña utilizar el patrón MVC puede ser contraproducente, pues una implementación relativamente simple puede hacerse más compleja al introducir más paquetes y archivos de los que son necesarios. En este tipo de casos es posible cometer errores en la sincronización y comunicación de datos, que puede ser una complicación innecesaria si el proyecto es pequeño.

¿De que otra forma se le ocurre haber solucionado el problema?

Pues, realmente el proyecto se podría hacer todo en un solo archivo, entonces la implementación sería igual, pero en una única clase. Este patrón se utiliza para que desarrollar el código sea más fácil y que leerlo después sea más sencillo. No obstante, la problemática se puede ignorar completamente sin que el proyecto sufra en funcionalidad, el problema es que este tipo de código es muy difícil de mantener y no es recomendable en el más mínimo, incluso si es funcional.