

En la unidad 3 de nuestro curso hemos trabajado en el análisis de un problema construyendo un modelo con los elementos que intervienen en el problema y especificando los servicios que el programa debe ofrecer, bajo el paradigma de programación orientado a objetos.

Aprendimos a construir las clases que implementan el modelo de la solución del problema, identificando de manera informal los métodos de una clase y clasificarlos en métodos constructores, de consulta y de modificación. Utilizamos una arquitectura para un programa que permita repartir de manera adecuada las responsabilidades entre la interfaz de usuario y el modelo de la solución, y cómo relacionar dichos componentes.

Finalmente hemos aprendido a relacionar todos los conceptos vistos en las tres primeras unidades del curso.

El presente laboratorio les presenta una actividad en la cual se requiere aplicar todos los conocimientos adquiridos en estas unidades y verificar de esta manera el cumplimiento de los objetivos que han sido planteados para la unidad 3 descritos en el programa del curso.

Para llevar a cabo este ejercicio es necesario realizar las actividades listadas a continuación:

Actividades

Lleve a cabo las siguientes actividades de cada una de las etapas de desarrollo de software:

1. Análisis del problema (Definición de uno o más diagramas de flujo que representen la solución planteada. Excepto para métodos constructores y para métodos get y set)
2. Diseño de la solución. Elabore un diagrama de clases que modele la solución del problema de acuerdo con las buenas prácticas y los patrones de diseño revisados hasta el momento en el curso. Su diagrama debe incluir el paquete modelo y el de interfaz de usuario.
3. Realice un diagrama de objetos que satisfaga su diagrama de clases.
4. Implementación en Java. Incluya en la implementación, los comentarios descriptivos sobre los atributos y métodos de cada clase. Recuerde que todos los artefactos generados de fase de diseño e implementación deben ser en inglés.
5. Documentación en JavaDoc (Debe entregarse el JavaDoc generado y ubicarlo en la carpeta docs).
6. Usar GitHub como repositorio de código fuente y documentación utilizando la estructura de carpetas aprendida en clase. Recuerde que se debe evidenciar su avance a través de los días en el laboratorio.
7. Subir a moodle los puntos anteriores el plazo máximo es 29 de septiembre de 2019

Recuerde que puede encontrar la Rúbrica laboratorio en el siguiente enlace.

Nota: Usted debe entregar un archivo en formato pdf con toda la documentación (análisis) y la URL de su repositorio GitHub donde se deben encontrar los archivos de codificación en sus respectivos paquetes.

Tenga en cuenta que su repositorio GitHub debe presentar una estructura base como por ejemplo:

Angels/
src/

Dentro de los directorios src/ y bin/ estarán presentes estos directorios(representando cada uno de sus paquetes):

ui/
model/

El directorio src (source code) contiene sus clases .java dentro del directorio ui/ y model/. Por otro lado el directorio bin (binary files) contiene los archivos .class en el directorio ui/ y model/.

Su código debería compilar de acuerdo con lo explicado en la diapositiva 13 de esta presentación:
<http://tinyurl.com/y3bd9bg2>

A continuación, encontrará un enunciado que narra de forma detallada la situación problemática que se espera usted solucione.

Enunciado

A una amiga de su profesora de Algoritmos le gustan mucho los ángeles y le ha comentado que sería interesante contar con una pequeña aplicación que le recuerde las fechas importantes, tipo, color representativo, entre otros.

Su profesora, al tener la necesidad de plantearles un problema a sus estudiantes de APO1, comenzó a preguntarle a su amiga y esto fue lo que le contó:

Los ángeles están categorizados en varios tipos: arcángeles, querubines, serafines.

Todos los ángeles de un mismo tipo se agrupan en una legión, por ejemplo: los ángeles arcángeles están agrupados en una legión llamada "Máxima Superior". Los ángeles querubines están agrupados en una legión llamada "Liberi" y por último los serafines se agrupan en la legión "Seraphim".

Nota: Para este ejercicio solo nos interesa la legión de los arcángeles.

Cada arcángel tiene un nombre (el cual termina en la sílaba "el"), una foto, una oración, una fecha de celebración (día y mes) y un poder. Entre algunos de los poderes reconocidos están: la protección, la salud, la abundancia, la justicia, la lealtad, pero el usuario puede crear más poderes. Cada arcángel se le asigna una vela especial, la cual tiene un color, un tamaño, una esencia y un grado de iluminancia.

La aplicación debe permitir:

1. Ingresar los arcángeles (solo existe un arcángel por nombre y solo existe un arcángel por poder)
2. Contar los arcángeles ingresados
3. Desplegar la información del arcángel dado su nombre
4. Desplegar la información del arcángel dado su poder
5. Desplegar las celebraciones a realizar dado un mes (debe mostrar el nombre del arcángel, el día de la celebración, el color y la esencia de su vela)
6. Desplegar todas las celebraciones: Revisa cada uno de los ángeles creados y va concatenando la respuesta con la fecha de celebración, así: nombre del ángel: fecha de celebración, nombre del ángel: fecha de celebración. Ejemplo: Miguel: 29 de septiembre, Rafael: 10 de junio.
7. Revisar próximo viernes es un punto sorpresa