

En la unidad 5 de nuestro curso hemos aprendido a utilizar el concepto de matriz como elemento de modelaje, pudiendo así agrupar los elementos del modelo de la solución en una estructura contenedora de dos dimensiones de tamaño fijo y estructuras contenedoras lineales de tamaño variable como elementos de modelado que permiten manejar una secuencia de objetos.

Adicionalmente he modelado (dibujar) e interpretado en un diagrama de clases las estructuras contenedoras lineales de tamaño variable. Verificar y corregir las inconsistencias que se pueden presentar entre cada uno de los elementos de diseño propuestos en el diagrama de clases y su implementación en el lenguaje de programación.

Finalmente hemos aprendido a relacionar todos los conceptos vistos en las unidades del curso. Permitiendo que desarrollemos a través de este laboratorio, un programa completo, teniendo una visión global de las etapas del proceso que se debe seguir para resolver un problema usando un computador.

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:

Events/

src/

bin/

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

Durante la presentación del examen final el semestre pasado en el auditorio Manuelita, algunos estudiantes notaron el estado defectuoso de algunas sillas.

Los profesores notificaron al terminar el examen al área encargada, pero no hubo forma de informarles exactamente cuáles de las sillas del auditorio, presentaban dicho estado.

Algunos de los estudiantes pensaron que una forma fácil de hacerlo es numerando la silletería con una fila y un número de asiento como lo hacen en los teatros de la ciudad. Así que trataron de aprovechar sus vacaciones e iniciaron definiendo algunos requerimientos funcionales.

| |
|---|
| Nombre: R1 - Crear las sillas del auditorio |
| Resumen: permite crear las sillas del auditorio, una vez se crea el objeto auditorio; por distribución física, la cantidad de sillas por fila no es igual, ya que unas filas son más largas y tienen una cantidad de sillas mayor. Tenga en cuenta que el número de la fila es un carácter, el cual inicia en la letra A para la primera fila, luego la letra B para la segunda fila y así sucesivamente. Es decir que en este método debería crearse un arreglo de caracteres con el abecedario. |
| Entradas: arreglo con la cantidad de sillas por fila. |
| Resultado: Objetos silla asignados a la matriz y en caso de no existir la silla quedará nula. |

| |
|---|
| Nombre: R2 - Reporte de Silla Defectuosa |
| Resumen: permite cambiar el estado de una silla de OPERATIVA a DEFECTUOSA, registrando la descripción del defecto. |
| Entradas: fila y número de silla, descripción defecto |
| Resultado: Un String indicando si la silla especificada se reportó defecto ó indicando si la silla no existe. |

| |
|---|
| Nombre: R3 - Calcular el % de sillas defectuosas en el auditorio. |
| Resumen: permite calcular el % de sillas defectuosas en el auditorio. Dicho cálculo se realiza con base en la cantidad de sillas disponibles en el auditorio, es decir, que no se incluyen las sillas nulas. |
| Entradas: Ninguna |
| Resultado: Porcentaje de sillas defectuosas. |

Como el tiempo de las vacaciones no les alcanzó, decidieron compartirle a sus profesores de APO1, y hemos pensado que interesante sería tener esta funcionalidad para todos los auditorios de la Universidad; para lo cual es necesario contar con la información del nombre y ubicación del auditorio al interior de la Universidad.

Y al tener la información de los auditorios, se podría llevar el registro de los eventos, que se realizan en la Universidad, para la reserva del auditorio, el cual se presta por un mínimo de dos horas y máximo por 12 horas entre las 7am y las 8pm para una misma fecha.

De cada evento podría registrarse el nombre, la fecha, la hora inicio y la hora de fin (sin minutos), el nombre del profesor responsable, el nombre de la facultad responsable del evento, la cantidad de personas que asistieron.

Tenga en cuenta que un auditorio únicamente se puede reservar para un solo evento; Por ejemplo si el auditorio Varela se encuentra reservado para el 15 de diciembre entre las 8am y 11am y una persona quisiera ingresar una nueva reserva para ese auditorio en la misma fecha entre las 10am y las 13pm.. El sistema no lo permitirá. Por el contrario, un evento puede tener reservados varios auditorios.

Al iniciar el evento, el auditorio deberá actualizarse a estado “ocupado” y el nombre del evento , todas las sillas operativas deben aparecer como disponibles, las cuales se irán ocupando aleatoriamente, teniendo en cuenta que no podrá ocupar una silla defectuosa. Cuando se cierra el evento, se actualizará la cantidad de sillas ocupadas y el auditorio vuelve a estado “disponible”, sin nombre de evento asociado.

La aplicación deberá permitir en cualquier momento ir registrando las sillas defectuosas al interior de los auditorios y cumplir con la totalidad de las necesidades expuestas.