UNIVERSIDAD ICESI

Departamento de TIC
Algoritmos y Programación I
Tarea Integradora 3 - Unidad 3 y 4

En la unidad 3 de nuestro curso hemos trabajado sobre la aplicación de mecanismos de reutilización como son la herencia y el polimorfismo. Estas técnicas enriquecen el comportamiento de las clases modeladas como parte de una solución en el diagrama de clases.

En la unidad 4 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.

Esta tarea integradora presenta una actividad en la cual se requiere aplicar todos los conocimientos adquiridos hasta el momento. Por tanto, esta tarea es un instrumento para verificar el cumplimiento de los objetivos que han sido planteados para la unidad 3 y 4 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 (Lista de requerimientos funcionales en el formato visto en la clase de ingeniería de software 1, descárguelo aquí).
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. El modelo debe ser elaborado digitalmente, pero NO generado automáticamente.
3. Trazabilidad del Análisis al Diseño. Una tabla a tres columnas en la que se relaciona cada requerimiento con el método o métodos que permiten satisfacer dicho requerimiento.
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 en la 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.
7. Elaboración de un video con la ejecución del programa.
8. Subir a moodle los puntos anteriores el plazo máximo es lunes 6 de diciembre.

Recuerde que puede encontrar la Rúbrica de la tarea integradora en el siguiente link..

Nota:

- Usted debe entregar 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:

dataCenter/
src/
bin/
doc/

UNIVERSIDAD ICESI

Departamento de TIC
Algoritmos y Programación I
Tarea Integradora 3 - Unidad 3 y 4

- 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/. El directorio doc tendrá toda la documentación de análisis y diseño
- Su código debería compilar de acuerdo con lo explicado en la diapositiva 15 de esta presentación: http://tinyurl.com/y3bd9bg2

A continuación, encontrará el enunciado de la tarea integradora 3 con él fin de complementar dicha solución.

Enunciado

El nuevo rector de la universidad Icesi tiene entre sus planes de modernización de la universidad la construcción de un DataCenter que sirva para soportar los procesos de investigación y para generar ingresos a la universidad por medio del alquiler de los cuartos de servidores. Entonces, se construirá en el campus un nuevo edificio de un solo piso que estará dividido en 8 corredores, en cada corredor habrá 50 minicuartos. Cada minicuarto será asignado a un proyecto de investigación, o alquilado a una empresa identificada con un nit y un nombre, tendrá un indicador para saber si el minicuarto está ubicado en un ventanal del edificio y se registrará el valor del alquiler, el cual consta de dos partes: un valor base igual para todos los minicuartos y un valor que depende de la ubicación de la siguiente manera:

- Minicuarto ubicado en Ventana: tiene un descuento del 10%
- Minicuarto ubicado en el séptimo corredor: tiene un descuento del 15%
- Minicuarto ubicado entre el segundo y sexto corredor, tiene un recargo del 25%

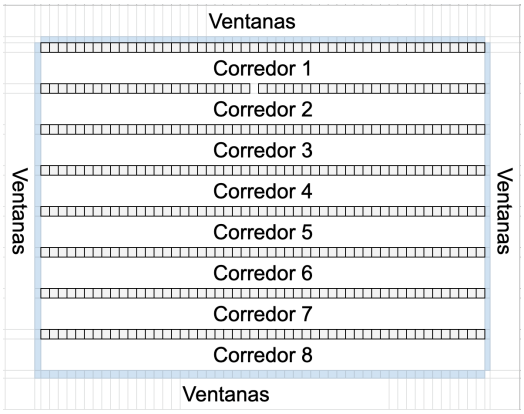

Fig.1 plan del edificio del Data Center

Cada minicuarto se ha identificado con un número único dentro del edificio y en su interior se ubica un armazón metálico especial llamado RACK y en cada RACK se ubican varios servidores. La empresa inquilina decide la cantidad de servidores alojados, pero si se utilizan menos de 4 se le cobra a la empresa un 15% adicional al valor del alquiler, como especie de multa por subutilizar el procesamiento.

De cada servidor se guardará la siguiente información:

- Cantidad de memoria cache (en GB)
- Número de procesadores

- Marca del procesador (INTEL, AMD)
- Cantidad de memoria RAM (en GB)
- Cantidad de discos
- Capacidad de los discos (en teras)

Se desea simular el sistema de apagado de los minicuartos en forma de letras:

- ➢ Letra L: apaga los primeros minicuartos de todos los corredores, junto con los minicuartos del primer corredor.
- ➢ Letra Z: apaga los minicuartos del primer y último corredor, junto con los minicuartos de la diagonal inversa.
- ➢ Letra H: apaga los minicuartos ubicados en los corredores impares
- ➢ Letra O: apaga los minicuartos ubicados en las ventanas
- ➢ Letra M: Pregunta al usuario una columna *N* y apaga todos los minicuartos en la columna N
- ➢ Letra P: apaga los minicuartos de un corredor (dado el corredor. Recuerde que para el usuario los corredores inician en uno)

El nuevo rector considera que con lo aprendido en su curso de Algoritmos y Programación 1 cada estudiante puede hacer una propuesta de una aplicación que permita administrar el nuevo datacenter y programar de forma automática el prendido y apagado de los minicuartos.
Las funcionalidades requeridas para la aplicación son:

1. Generar un listado con lo minicuartos disponibles, mostrando el corredor, la ubicación (ventana o no ventana), la columna y el valor del alquiler.
2. Alquilar un minicuarto, aquí será importante dejar el registro de la fecha de alquiler, registrar los servidores y calcular el valor del alquiler mensual que una empresa debe pagar. Si el minicuarto se alquila a un proyecto de investigación, la empresa es la universidad Icesi y se debe guardar además el número de registro del proyecto.
3. Cancelar el alquiler para un minicuarto o para todos los minicuartos de una empresa, lo que haría que el minicuarto se encuentre en estado disponible. Recuerde que debe eliminar los servidores asociados al RACK. Antes de proceder a la cancelación deberá desplegar la capacidad de procesamiento (indicando total de capacidad en disco, total de memoria RAM).
4. Mostrar un mapa del datacenter en la que se puedan ver cuáles minicuartos están prendidos y cuáles están apagados. Solo los minicuartos que están alquilados pueden estar prendidos (los otros están apagados por consumo de energía)
5. Simular el prendido de todos los minicuartos (sin importar el estado) para probar los protocolos de apagado.
6. Simular el apagado de los minicuartos de acuerdo a la letra ingresada, después de apagar los cuartos se debe mostrar el mapa del datacenter en donde puedan observarse los cambios.