Universidad Nacional de Colombia
Programacion Orientada a Objetos
Cristhian David Amaya 25481092

Taller 2

1. Responda las siguientes preguntas:

a. ¿Cuál es la importancia de la abstracción en el proceso de modelado?

La abstraccion es un proceso mediante el cual los objetos creados a partir de una clase realizan su funcion de forma transparente al usuario aislados del contexto del programa, de esta forma el usuario se ocupa unicamente de suministrar la informacion necesaria al objeto para que realice sus funciones sin preocuparse del modo en que este lo hace, ademas de proveer cierta seguridad en cuanto a derechos de autor por ejemplo generando bibiotecas de vinculos dimamicos o estaticos que permiten utilizar el codigo construido sin que el usuario tanga que ver mas que el archivo de cabeceras donde se definen las funciones y clases de las que puede disponer y los parametros necesarios para su funcionamiento, se puede dotar a las las clases construidas de portabilidad en terminos de que podrian ser independientes del lenguaje en que fueron construidas pudiendose vincular a programas construidos en otros lenguajes de programacion.

Su importancia en el proceso de modelado radica en una sintesis de procesos, es decir que varios procesos se pueden entender como uno solo reduciendo el tiempo necesario para concebir o modelar un programa siendo este al mismo tiempo mas comprencible al verlo desde una perspectiva mas global, un ejemplo practico de esto podria ser el diseño de un programa usando una api grafica, para el cual se requieren crear varias ventanas con las que el usuario interactuara para darle ordenes, en nuestro proceso de modelado prodriamos tener que dibujar un TabControl, para ello se tendria que usar una librería grafica como por ejemplo OpenGL mediante el cual se dibujaria un cuadrado que delimitaria el control, un segundo cuadrado al interior del primero para el area de la pestaña, un tercer cuadrado dentro del primero donde estaran otros cuadrados para seleccionar la pestaña a la cual se quiere acceder, ademas de crear los metodos necesarios para su correcto funcionamiento, pero usando una api grafica todo lo anterior se prodria resumir en el llamdo a una clase que creara el control y otro llamado a un metodo para mostrarlo, de lo anterior se puede evidenciar que muy dificilmente modelando el programa mediante el primer metodo se entenderia que se esta mostrando al usurio un TabControl, mientras que mediante el segundo con la simple llamada (HwndTab=Create(“TabControl”,...);show(HwndTab);) se deduce que se esta creando y mostrando al usuario un TabControl

b. ¿Cuál es el código cliente (client code) y donde podemos encontrarlo?

El codigo cliente es quel que hace uso de una clase para construir objetos y de esta forma realizar alguna funcion, por lo que se puede en contrar en cualquier programa que se construlla de forma orientada a objetos, ya sea en una funcion que realice una simple llamada para imprimir en pantalla o para crear otros objetos.

c. ¿Cuáles son las similitudes y diferencias entre las variables de tipo primitivo (Primitive-type) y las variables de tipo referencia (reference-type)?

Las similitudes entre las varibles de tipo primitivo y de refencia radican en que se pueden entender como un puntero que hace referencia a una ubicación de memoria, en el caso de las variables primitivas de un tamaño igual al tipo de la variable, mientras que a diferencia de C y C++ las diferencias en java radican en que en las variables primitivas ese espacio de memoria al que apunta la variable almacena directamente el valor que se le asigno a la variable, mientras que para las variables de refencia teniendo en cuenta que estas pueden estar compuestas por varios tipos de variables primitivas, ese espacio de memoria almacena

las direcciones de memoria donde se encuentran las variables que componen al objeto, es decir como similitudes se tienen que los dos tipos hacen referencia a una ubicación de memoria, y como diferencias que las primeras almacenan directamente en ese espacio de memoria el valor asignado mientras que las segundas almacenan direcciones de las variables que la componen.

d. Explique cuándo y por qué una clase debe proveer los métodos get/set para una variable de instancia.

Teniendo en cuenta que una variable de intancia es unica para cada objeto, es decir que su modificacion unicamente afecta al objeto al cual pertenece, no es necesario proporcionar los metodos get y set para una variable pero es considerado como una buena practica de programacion, ya que el objeto que use esa variable puede requerirla en determinado formato o con ciertas caracteristicas para manejarla, por lo que permitiendo al usuario su modificacion directa podria este no proporcionarla con los requerimientos generando posibles fallas en el programa, de modo que restringiendo la variable al usuario se obliga a este a setearla mediante la llamada a un metodo que tranformaria la entrada al formato necesario y despues de esto seteria la variable, siendo el metodo esa capa de filtrado que aseguraria que el objeto disponga de la variable con las caracteristacas adecuadas evitando fallas en el funcionamiento ademas de liberar al usuario de la generalmente engorrosa tarea de darle algun formato a sus entradas, como ejemplo de lo anterior podriamos tener una interfaz software para determinado dispositivo en donde se requeriria que el usuario definiera por ejemplo un sensor del cual capturar informacion, siendo esta una direccion del puerto de algun microcontrolador, el usuario mirando el manual de operación del dispositivo podria determinar que se trata del puerto usb, dando como entrada al programa “usb”, lo que causaria que el programa no encontrara esta direccion ya que requeriria una direccion fisica en formato hexadecimal, pero mediante la llamada a un metodo esta entrada coincidiria con algun valor suministrando al programa una direccion valida de la cual podria obtener la informacion necesaria.

e. El Garbage Collector elimina los objetos que ya no se utilizan proporcionando una gestión de memoria automática. ¿Cómo la JVM realiza el manejo de recursos? ¿Cuándo y cómo funciona el Garbage Collector?

Teniendo en cuenta que en java existen 3 tipos de memoria Datos, Heap y Stack , la gestion de recursos de Java se realiza de forma automatica al inicio de la ejecucion del programa sumistrando cierta cantidad de memoria que puede ser modificada si es necesario, a medida que se crean variables en tiempo de ejecucion estas son almacenadas en la zona Heap de memoria que tiene un tamaño ya definido por lo que cuando se eliminan las refencias a esta memoria desde la zona zona stack que se crea en tiempo de compilacion y almacena las variables locales, la memoria heap se va llenando de informacion no necesaria, y es aquí donde se traduce la gestion de meria dinamica mediante el Garbage Collector de java que actua sobre la zona de memoria Heap que almacena las variables creadas de forma dinamica siendo un proceso de baja prioridad lo que quiere decir que se ejecuta cuando el procesador no tiene procesos de mayor prioridad para ejecutar y libera la memoria de las variables que no tienen referencia desde la zona stack de memoria.