PRIMER CLASE DATOS!!!

What is an object ?

Everywhere you look in the real world, you see things or objects

plants

Animals

Cars

Houses

Now describe it !

The car is:

Black es un atributo

Has doors that open upward atributo

The car can:

Métodos:

Accelerate

Break

Turn

Play music

Es algo compuesto

Con características propias

Se define en base  a la clase

Se construyen a partir de una clase

Clase persona

Que es un paradigma?

Modelo

Patron

Ejemplar

SEGUNDA CLASE DE DATOS

Programming Paradigm

Fundamental style of computer programming

Functional paradigm modeled as mathematics and functions theory

-Functions are first-class citizens

-Scheme, lisp, F#

Imperative paradigm modeled as steps, commands and structures.

-C, C++, PHP

Structures are first-class citizens

Logic paradigm modeled as axioms, inference rules and queries

-Prolog

Object-oriented paradigm modeled as objects

-Java, C++, C#, Python

-Objects are first-class citizens

-Why has the OO paradigm being so successful ?

R/ Por el modeled del mundo real

What is OOP ?

-Design and build programs under the object-oriented paradigm

-Models software in terms similar to those that people use to describe real-world objects

Acotamientos

Acoplamiento directo: relacion entre metodos de diferentes clases

Atributos son propiedades o comportamientos

There are some important concepts to understand OOP:

* Abstraction: Vista de alto nivel, un mapa es una abstraccion de un pais

ART

No es una representacion fiel de la realidad

Capacidad del humano para hacer representaciones de la realidad

* Encapsulation:

Private and Public

Oculta detalles de implementacion de una clase particular

Meter en una caja las funcionalidades

Have the property of information hiding

Objects hide the implementation details from other objects

Microwave oven: lleva sus funciones de manera encapsulada, solo muestra interfaz para su uso

* Classes

Es un molde for example a cupcake mold

The mold define how the resulting cupcake will look like

Define la clase, pero utiliza los objetos que se crearon en base a esa clase

Como se crea una clase ?

Definir los atributos y metodos de los objectos basados en la clase

Atributo is a data element of the class

Car class has attributes such as top speed

Think of an attribute as a variable that is linked to the an object

Method are operations that make changes to the attributes of the objects

* Attribute
* Method
* Instance

Crear un objeto se refiere a un objeto creado usando una clase

Cada instancia es una entidad separada

An change on an instance does not affect other instances of the same class

Methods are executed on an specific instance

* Inheritance: El hijo solo puede tener un padre, el padre diferentes hijos, clase estudiantes es hijo de clase persona

Good way of reusing code.

“Is a” type of relationship

* Polymorphism: Un objeto se puede comportar de manera diferente

It allows to treat different objects as they were the same

A subclass can modify the methods inherited from this parent. This is called overriding

Mensajes Es el proceso mediante u objeto le envia datos a otro objeto para que ejecute un metodo

OVERLEAF

LATEX

Patron CRUD para interactuar con bases de datos

extends para heredar en Java

public class Calse extends Padre implements metodos

JFrame para dibujar Ventana en Java

Class JFrame

Servicio web

Sockets

APUNTES DE PROYECTOS

SCRABBLE

Objetivos

Lista enlazadas

Cliente servidos juego multiplayer

El servidor tiene un diccionario de palabras

Arquitectura cliente servidor

Toda comunicacion con el servidor es por socket y en formato JSON

Es un properties file

Matrices en Java

Las palabras en el clinete se maneja como listas enlazadas

El servidor lleva rastro de las palabras en la cuadricula

Biblioteca Jackson para serializar/deserializar JASON

Utilizar JavaDocs y se debe generar el HTML de la documentacion

Wiki de Github

Project management de Github

Lista de features e historias de usuario identificados de la especificacion

Distribucion de historias de usuario por criticalidad

Descomposicion de cada user story en tareas

Proyecto 1

DOTS

Juego multiplayer

Formar figuras geometricas para obtener puntos

Aplicar patrones de diseño

Hay un servidor central que consiste en una aplicacion en Java que escucha las conexiones entrants por un socket. Cada aplicacion cliente se ejecuta en una computadora y se conecta por socket al servidor. Cuando el juego va a iniciar, el servidor recibe la peticion del cliente y lo ingresa en una cola

El servidor lleva el control complete del juego

Los clients unicamente grafican lo que el servidor les envian

Los clients a su vez envian las acciones que realicen al servidor, el cual se encarga de mantener el estado del juego complete. Toda la comunicacion entre cliente y servidor es en formato JSON

\

\

Implementar un chat

Un servidor que muestre a dos jugadores en tiempo real lo que hace el otro

Aspectos de implementacion

Sobre cliente:

No hay manejo de logica del servidor

El cliente hace pull cada n segundos para obtener el estado del juego del servidor

El cliente toma la respuesta del servidor y la dibuja en la pantalla

El usuario interactua con la malla de puntos y envia mensajes en JSON al servidor indicando la accion efectuada

No puede utilizer matrices ni ninguna clase de Java para modelar la malla del juego

La malla se implementa como una lista de listas

Sobre el servidor:

La cola de jugadores se modela como una cola implementada por los estudiantes

No puede utilizer matrices ni ninguna clase de Java para modelar la malla del juego

Biblioteca Jackson para serializar/deserializar JSON

Segundo Proyecto

inmemDB Base de Datos

Implementar una base de datos NOSQL mediante estructuras de datos lineales

Crear un esquema y borrar un esquema

08/07/2019

Inheritance

Los hijos heredan el codigo de los mayores.

-In OOP classe can also share derived class or subclass

-Se comparten los atributos

-Inheritance is a good way of reusing code

-Use it when is necessary… the right tool for the problem

-“Is a” type of r3elationship

Polymorphism

-Is tightly related to inheritance

-Allows to treat different objects as they were the same

-A subclass can modify the methods from this parent. This is called overriding

Rectangle2D

Modularity

-Modularity allows to group logically related abstractions

-Each group is a reusable module

-Agrupar clases en una carpeta

Some Java concepts

JAVA

-JVM (Java Virtual Machine) runs on top of the hardware.

-That is what allows the portability of a Java program

-JVM es una capa intermedia

Bytecode

This is native code

Java API

-Is composed of a lot of packages containing useful functionalities

JRE

JDK

IDE

Que cosas se pueden programar en Java? 09/08/2019

Aplicaciones de alta calidad, web y de red.

Es muy facil encontrar bibliotecas

Principales ambientes de Desarrollo para Java:

SDK

Netbeans

Eclipse

IntelliJ

Operadores Arimeticos

A=7

B=3

Operadores de asignacion

+= op1 = op1+=op2

-=

\*=

/=

%=

Operadores relacionales

>

<

== igualdad

!= distinto

>= Mayor o igual que

<= menor o igual que

Operadores logicos

&& if (t&&t)

|| if (t||f)

! if (!f)

Otros operadores

-Operadores unarios

C = a +-b

-Operador instanceof

Nombreobjeto instanceofNombreClase

-Operadores incrementales

++ incremento

--Decremento

-Operador de concatenacion

“+”

System.out.println(“El total es ”+ resultado + “unidades”)

Precedencia de los operadores

Tipos de datos

Byte

Short

Int

Long

Float

Double

Char

String inmutable

Boolean

Modificadores de nivel de acceso

Public: Proporciona acceso y visibilidad completqa a todos los elementos

Protected: Brinda acceso y visibilidad a los elementos dentro de un paquete, clase o subclase pero no da visibilidad a elementos externos

Private: Otorga acceso y visibilidad solo a elementos dentro de la misma clase

Sin modificador: Cuando no se especifica un modificador concede

Estructuras de control

IfElse y return

Public class IfElse

Do-while

La diferencia entre while y do-while es que la sentencia del do-while se ejecuta siempre, al menos una vez

En un while, si la condicional es falsa la primera vez, la sentencia no se ejecuta nunca.

For

switch

14/08/2019

The variable passed to the switch can be following types:

Integer

String

Arrays: Grupo de variables relacionadas que comparten el mismo tipo y nombre.

Un array de una dimension se le llama Vector y al de mas de una dimension es una Matriz

Hacer un array, tanto vector como matriz

Es una estructura estatica

Vector:

Int [] edades = new int[5];

Int [] edades = {20,19,21,26,28};

Matriz:

Operaciones con Arrays

For (int I = 0; I <2; i++)

For (int j = 0;)

Clase vector

Crea una estructura que puede crecer de forma dinamica

-Añadir elementos a una clase vector

v.addElement(“uno”)

-Determinar el tamaño

-Remover elementos

-Acceso a los elementos

Que es UML ?

Por que modelar es necesario:

* Necesito construir un buen software
* Tratar de comunicar la estructura de la forma deseada
* Para visualizar y controlar
* Para comprender el Sistema

Un modelo es una simplificación de la realidad(Abstraccion)

Hay tres tipos de UML:

* Modelo functional
* Modelo Objeto
* Modelo dinamico

Uno de los más communes de tipos de diagramas es el diagrama de clase

Diagrama de Clase

Es conocido como vista estatica

Captura los objetos de la estructura

No contiene informacion relacionada al comportamiento dinamico

Relaciones

Asociacion: Describe la conexion entre dos instancias de la clase

Depencia: Relacion entre dos elementos del modelo

Generalizacion: Relacion entre especificos y descripciones mas generales, usado por herencia y polimorfismo

Relacion: Relacion entre especificaciones e implementacion.

21/08/2019

Design Patterns

What is a pattern

* Is a **discernible regularity** in the world or in a manmade design
* One thing an expert designer know no to do is solve every problem from first principles

-Why?

-Does this make sense to you?

-Have u heard the old saying: don’t reinvent the wheel?

Is a general reusable solution to commonly occurring

Can be thought as a good practice a proven solution for a problem

Refactoring GURU

* Iterator
* Visitor

Patterns can be classified by two criteria:

Purpose and scope

* Purpose reflect what a pattern does
* Scope specifies if the pattern applies to classes or objects

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  |  |  |  |  |
|  |  | Creational | Structural | Behavioral |
| Scope | Class | Factory method | Adapter(class) | Interpreter(243)  Template method(325) |
|  | Object | Abstract Factory  Builder  Prototype  Singleton | Adapter(object)  Bridge  Composite  Decorator  Façade  Flyweight  Proxy | Chain of responsibility(223)  Command  Iterator  Mediator  Momento  Observer  State  Strategy  Visitor­­­ |

Creational patterns

* Two main focuses

Encapsulates the knowledge about which concrete classes

Hides how instances of the classes

Singleton:

Ensure that a class has one instance and provide a global point of access to it

Ensures that no other instance of a class can be created

Provides a way to access the instance

Intention: is a creational design pattern that lets you ensure that a class has only one instance

Problem:

* **Ensure that a class has just a single instance**.
* **Provide a global access point to that instance**.

Solution:

* Make the default constructor private, to prevent other objects from using the new operator with the Singleton class.
* Create a static creation method that acts as a constructor.

A screenshot of a social media post

Description automatically generated

A screenshot of a social media post

Description automatically generated

Abstract Factory:

Ofrece una interfaz para crear familias de objetos

Façade:

Define un alto nivel de interfaces que hacen subsistemas faciles de usar

Adopter:

Convierte la interfaz de una clase en otro interfaz que espera un cliente

Se aplica cunado se quiere usar una clase existente y esta inrface no match like you need

**Patrones de comportamientos**

Observador:

Define una dependencia de uno a muchos

Modelo vista controlador:

Es uno de los patrones mas communes, divide al Sistema en 3 partes

separadas:

* Vista
* Usuario
* Controlador

28/08/2019

Abstract Data Types

It is a set of objects together with a set of operations

ADTs are mathematical abstractions. Their definition does not include any way to implement the operations

When we say data types, we think of:

* Integers
* Reals
* Booleans

Each of those have operations associated whit them

Thinking on ADTs we have:

* Lists
* Queues
* Sets
* Graphs

Just like any other data type, ADTs also have operations associated

* Add
* Remove
* Contains

How can we implement an ADT in Java?

Classes can be used to implement ADTs because they allow to hide the implementation details

Doing operations on the ADTs is just calling methods of the class

ADT is not an OO concept

In the OO context, and ADT can be the same as an interface or a class with some characteristics

Which operations should be inside an ADT ??

That is a design decision of the programmer

When implementing an ADT be careful of not exposing implementations details

ARRAYS

* They are objects to a very limited extend
* There is no keyword with which all arrays are declared
* The may be considered instances of an understood array class

How to declare:

Int[ ] a;

Int a[ ];

Int[ ] a = new int[10];

Int[ ] b ={5, 4, 2, 1};

How to access elements

Mediante indices

Some limitations

No se puede cambiar el tamaño, habria que hacer uno nuevo y meter los elementos del anterior

Changing the size of the array is expensive:

* Create a new array
* Copying all data from the array with the old size to the array with the new size

The data in the array are next to each other sequentially in memory

MATRIX

A matrix can be thought as a two dimensional array or an array of arrays

Int [ ] [ ] nums = new int [5] [4]

X = nums[2][1]

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | 0 | 1 | 2 | 3 |
| 0 |  |  |  |  |
| 1 |  |  |  |  |
| 2 |  |  |  |  |
| 3 |  |  |  |  |
| 4 |  |  |  |  |

LINKED LISTS

Linear data structure where each element (node) is a separate object a not necessarily adjacent in memory

A dynamic data structures

* The numbers of nodes is no fixed
* A list can grow and shrink on demand
* Any application which has to deal with an unknow number of objects will need to use

A node contains s data field that is a reference to another node. This is how the nodes are linked

So each node holds is composed of data and a reference to another node in the list

To increase performance, specially when inserting at the end, u can add a reference to the last node(double-ended list)

DOUBLE-LINKED LISTS

Is a variation of the simple list

The advantage is that ease backward navigation of the list

Each node has two references to other nodes instead of one

The downside of double-linked list is dealing with more links

30/08/2019

Linked List

-There is more than one specific way to implement a linked list

Double linked lists

Puedo pasar del primero nodo hasta el ultimo

The downside of double-linked list is dealing with more links while inserting/deleting nodes.

Nodes are little bigger because they have an extra link

Insert at the beginning

Circular Lists

Podemos ir desde el inicio hasta el final y nos podemos devolver

Every node has a successor and the “last” element is succeeded by the “first” element

Deleting

location.next = location.next.next:

Inserting

El siguiente del anterior va a ser el nuevo nodo

When to use a circular lists

-Applications that need to access both ends of the list

STACKS

ultimo en entrar, primero en salir

A stack allows access to only one data item:

Processors use stacks intensively

Compilers use stacks to interpret the programming language expressions

06/09/2019

LinkedList

Array list

What is algorithm analysis?

Is a method for estimating the source

Why analyze an algorithm

-Classify problems and algorithms by difficulty

-Predict performance, compare algorithms, tune parameters

-Better understand and improve implementations and algorithms

Factors affecting Run Time

Compiler

Time(CPU)

Space(Disk Memory)

Network Bandwidth

Peripheral Devices

Any resource in the computer

Good measures, Bad measures:

Empirical

* Benchmark

Simulational

* Simulated Data
* Test cases

Analytical

* Mathematical Model (time & space)

Computational Complexity

Classifies computational problems according their difficulty

Time complexity

Space Complexity

Time complexity

* Classifies the amount of time taken by an algorithm
* Focus on dominant operations, the ones which perform more ”primitive” operations

CPU Performance

* Cycles per instruction
* Instructions per cycle
* Instruction per second
* Elementary instruction vs Composed instructions

Constant Time Instructions

-Basic Arithmetic Operators

-Bitwise operators

-Logical operators

-Jumps

Best, worst and average

-Best case scenario: resource usage at least

-Worst case Scenario: resource usage at most

-Average case scenario: resource usage on average

Big-O notation

Work can be calculated as a function of the size of the input to the algorithm

It is possible to express an approximation of this function using a mathematical notation called order of magnitude or Big-O

antgonto

SORT ALGORITHMS