8/28/19

Continuando los patrones de diseño.

-Singleton: Patrón que permite restringir la instanciación de una clase.

Únicamente se puede instanciar una sola clase.

public class Singleton {

private static Singleton instance = new Singleton();

private Singleton(){

-----------------

}

public void doSomething(){

---------------

}

//Usar public static Synchronized para evitar problemas con los threads (pondrá las llamadas múltiples en cola, y las ejectura 1 por 1 ).

Public static Singleton getInstance(){

If(instance == null) {

Instance = new Singleton();

}

Return instance;

}

}

Main:

Singleton.getInstance().doSomething();

Singleton var = Singleton.getInstance();

-Builder: delegar la construcción de una clase compleja a una clase complementación.

public class Phone{

private cámara;

private radio;

private touch id;

private waterproof;

public Phone(pCamara, pRadio, pTouchID, pWaterproof){

this…..

}

Public Phone(pCamara){

---------------------

}

Public Phone(Builder builder){

This.camara = builder.camara;

}

Public static class Builder{

Camara;

Radio;

TouchId;

Waterproof;  
 public Builder withCamara(pCamara){

This.camara = pCamara;

Return this;

}

Public Phone build(){

-----

Return new Phone(this);

}

}

Main:

Phone p1 = new Phone(null, null, touched, null);

Phone p2 = new Phone.Builder().withCamara(------).withRadio(------).build();

**UML**

    Unified Modeling Language

    Es un lenguaje visual. Se utilizan elementos visuales que permite interpretar lo que deseamos mediante diagramas.

* Modelo

    Una forma visual de describir un negocio y sus reglas. Existen personas que no les agrada modelar por que se les resulta tedioso o aburrido. Sin embargo, la parálisis de análisis es un punto donde únicamente se analiza y no se trabaja realmente

* Diagrama

    Forma visual de presentar los elementos descritos en el modelado. Se utilizan símbolos especiales para representar un propósito en especial

* Notación

Elementos que trabajan entre sí dentro de un diagrama

Tipos de diagrama

* Estructura

Muestra la estructura estática del sistema. Dice qué partes tiene el sistema como tal

Encontramos el diagrama de clases, de componentes, etc

* De comportamiento

Muestra el comportamiento dinámico de los elementos del sistemas

Actividad, casos de uso, etc

**Orientación a objetos en contexto de UML**

. Clase

    Plano a partir del cual los objetos son construidos. También se puede ver como una categoría para organizar a los objetos

    Las clases poseen atributos y operaciones. Los atributos son las propiedades que tiene la clase. Los datos que interactuan del objeto, ya sea que se le ingresen datos o que lean datos de este. Las operaciones son las funcionalidades que llevan a cabo, en el contexto de programación se le conocería como método.

.Objeto

    Es la instancia de una clase. Si la clase el un plano, el objeto es aquello que construimos a partir de un plano. El objeto es el que hace el trabajo como tal. Es algo que existe en el contexto del sistema.

.Herencia

    Tenemos una clase padre y otras que descienden de esta (clases hijas). Las hijas tienen todas las características de la clase padre y características propias.

    Los objetos existen en relación a otros objetos. A partir del concepto de herencia, podemos definir los conceptos de generalización (Se acerca más a la clase padre)y de especialización (Se aleja más de la clase padre).

.Asociación

    Dos clases están conectadas

    .Ligado donde la información de una clase está ligada a datos en otra clase

.Colaboración donde una clase trabaja con otra para llevar a cabo un trabajo

.Acto donde una clase actúa sobre otra. Ejemplo: el cerillo prende la vela

.Agregación

    Tenemos una clase que se le conoce como contenedora. Tiene otros objetos los cuales poseen existencia independientemente del contenedor

.Composición

    Todas las partes pertenecen a una unidad completa. Las partes no existen de manera independiente al contendor.

.Multiplicidad (Cardinalidad)

    Indican la cantidad de objetos que participan en una relación en particular. Existen tres tipos, Uno a uno, uno a muchos y muchos a muchos.

.Polimorfismo (muchas formas)

    Capacidad que tiene un objeto de actuar de diferentes maneras

    En el concepto de operaciones, se dice que la operación se puede ejecutar de diversas maneras dependiendo de el objeto con que se esté trabajando.

    En los objetos. Se crea un objeto que relaciona a otros para poder trabajar como cualquiera de los objetos está relacionando.

**Modelo 4 + 1**

* Vista lógica: nos enfocamos en las partes del sistema y las interacciones entre ellos. Enfatiza las clases y objetos
* Vista de procesos: Se visualizan los procesos o hilos. Nos muestra lo que se necesita pasar en el sistema
* Vista física: Modela el ambiente de ejecución. Mapea el software en el hardware que lo ejecutará
* Vista de Implementación: Muestra módulos  o componentes
* Vista de Casos de Uso: muestra la funcionalidad del sistema. Permite conocer los objetivos y los escenarios, da una perspectiva del sistema desde el exterior
  + Se usa elipses que representa como los objetivos que el actor desea, estos son los casos de usos
  + Se indica que actor está asociado con el caso de uso, estos se asocian con líneas o flechas