



Práctica #2

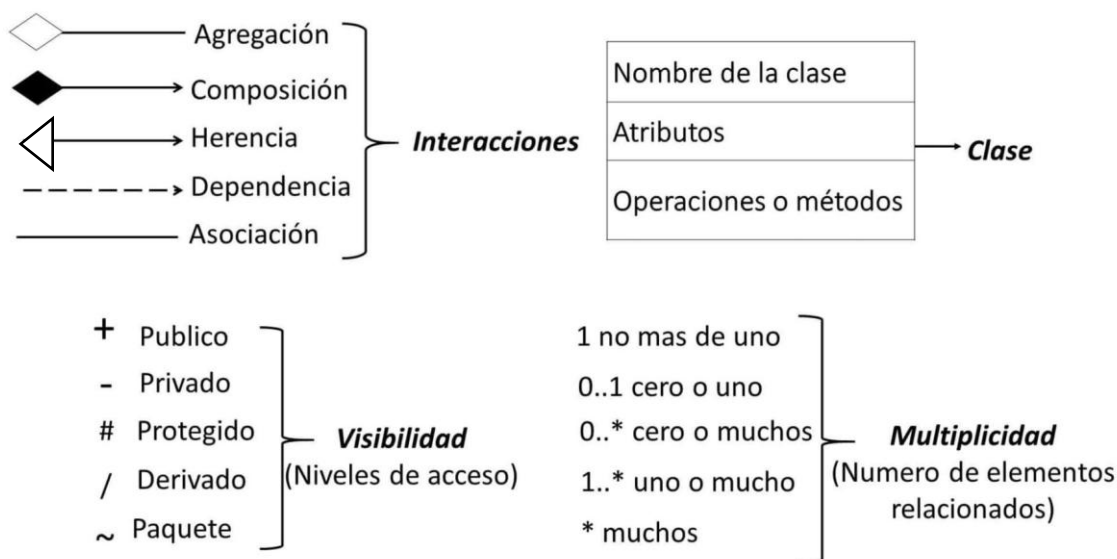
UML

Dentro de la categoría relaciones encontramos 3 tipos: relación de dependencia, relación de asociación y relación de generalización. Dentro de la categoría diagramas, vemos que UML proporciona varios tipos divididos en diagrama estructurales y de comportamiento. Los diagramas estructurales muestran los aspectos estáticos de un sistema representando el esqueleto relativamente estable, mientras que los diagramas de comportamiento muestran los aspectos dinámicos de un sistema tales como sus partes mutables.

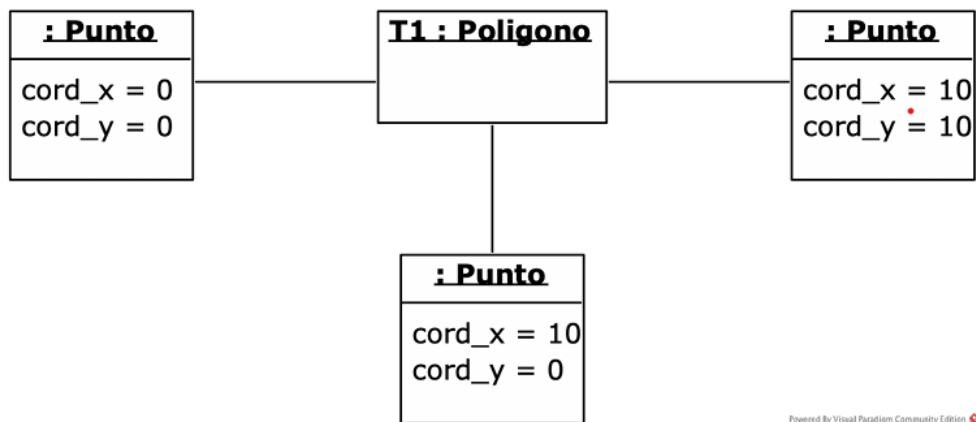
Diagramas estructurales

Entre los diagramas estructurales tenemos: diagrama de clases, diagrama de objetos, diagrama de componentes, diagrama de despliegue y diagrama de paquetes. Entre los diagramas de comportamiento tenemos: diagrama de caso de uso, diagrama de estados, diagrama de actividad, y los diagramas de interacción: diagrama de secuencia, diagrama de comunicación.

Elementos y símbolos en los diagramas de clases UML

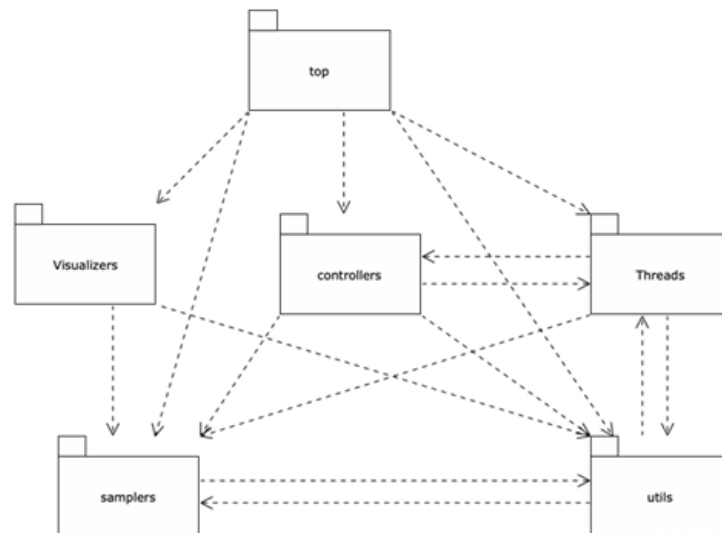


- Suponga el siguiente diagrama de objetos:
 1. Haga el diagrama de clases correspondiente
 2. Suponga dos triángulos con un lado común. Hacer un diagrama de objetos donde: “un punto pertenece a uno o más polígonos”
 3. Hacer un diagrama de objetos de un polígono con cuatro puntos: (0,0), (1,0), (0,1) y (1,1)



Powered By Visual Paradigm Community Edition

- Supongamos el siguiente diagrama de paquetes:



1. ¿Qué significa que el paquete “top” tenga una dependencia con el paquete “controllers”?
2. ¿Con cuántos paquetes tiene dependencias el paquete “Threads”? ¿Cuántas dependencias son de entrada y cuántas de salida? ¿Si se hace un cambio en alguna de las clases del paquete “Threads” qué podría suceder?
3. Hablando en términos de modularidad, ¿este diagrama muestra un código con bajo o alto acoplamiento entre los paquetes? ¿Esto es bueno o malo?

Diagramas de Comportamiento

A diferencia de los diagramas estructurales, los diagramas de comportamiento muestran como se comporta un sistema de información de forma dinámica. Es decir, describe los cambios que sufre un sistema a través del tiempo cuando está en ejecución. Entre los diagramas de comportamiento tenemos: los diagramas de interacción (secuencia y comunicación), los diagramas de actividad y estado, y los diagramas de casos de uso.

Ejercicios

- a) Muestre en un diagrama de secuencias los detalles de las siguientes instrucciones escritas en Java:

1. `(miLista.getItem()).hacerAlgo()`
2. `miColeccion.getEstudiante(i).getNombre()`
3. `lista.add(textField.getText())`

- b) Dado el siguiente código en Java, cree el diagrama de secuencia que representa el pase de mensaje entre las dos clases:

```
public class PruebaE1{
    public void f(){ g(); }
    public void g(){ h(); }
    public void h(){ System.out.println("Algo"); }
}

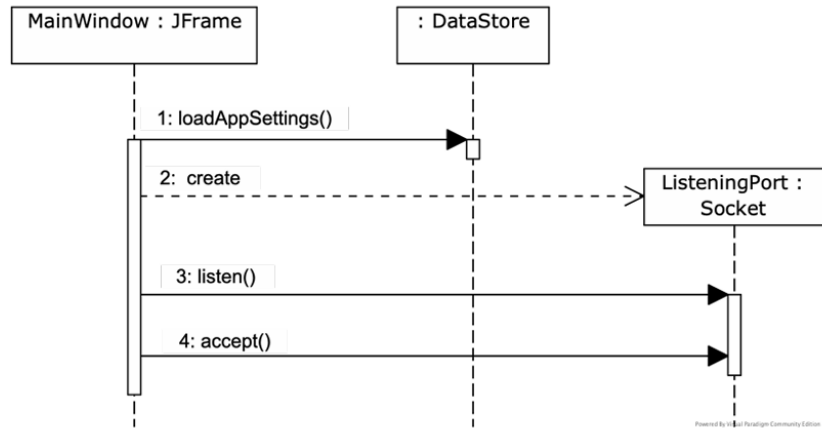
public class ClienteE1 {
    public static void main(String[] args) {
        Prueba p = new PruebaE1();
        p.f();
    }
}
```

- c) Dado el siguiente código, cree el diagrama de secuencia que representa el pase de mensaje entre las dos clases:

```
public class PruebaE2{
    private static PruebaE2 p;
    public static void f(){ p = new PruebaE2(); p.g(); }
    public void g() { this.h(); }
    public void h() { System.out.println("Prueba algo"); }
}

public class ClienteE2 {
    public static void main(String[] args) {
        PruebaE2.f();
    }
}
```

- d) Dado el siguiente diagrama de secuencia, cree el diagrama de comunicación correspondiente:



- e) Haga el diagrama de comunicación correspondiente al ejercicio “b”
 f) Dado el siguiente código, cree el diagrama de comunicación correspondiente:

```

Compra c = new Compra();
c.setCosto(x)
c.setFecha(f.getFecha());
l.add(c);
  
```

- g) Dado el siguiente código, cree el diagrama de comunicación que representa el pase de mensaje entre las dos clases:

```

public class Registro {
    private CatalogoProductos catalogo;
    private Venta venta;

    public Registro(CatalogoProductos cp) {...}

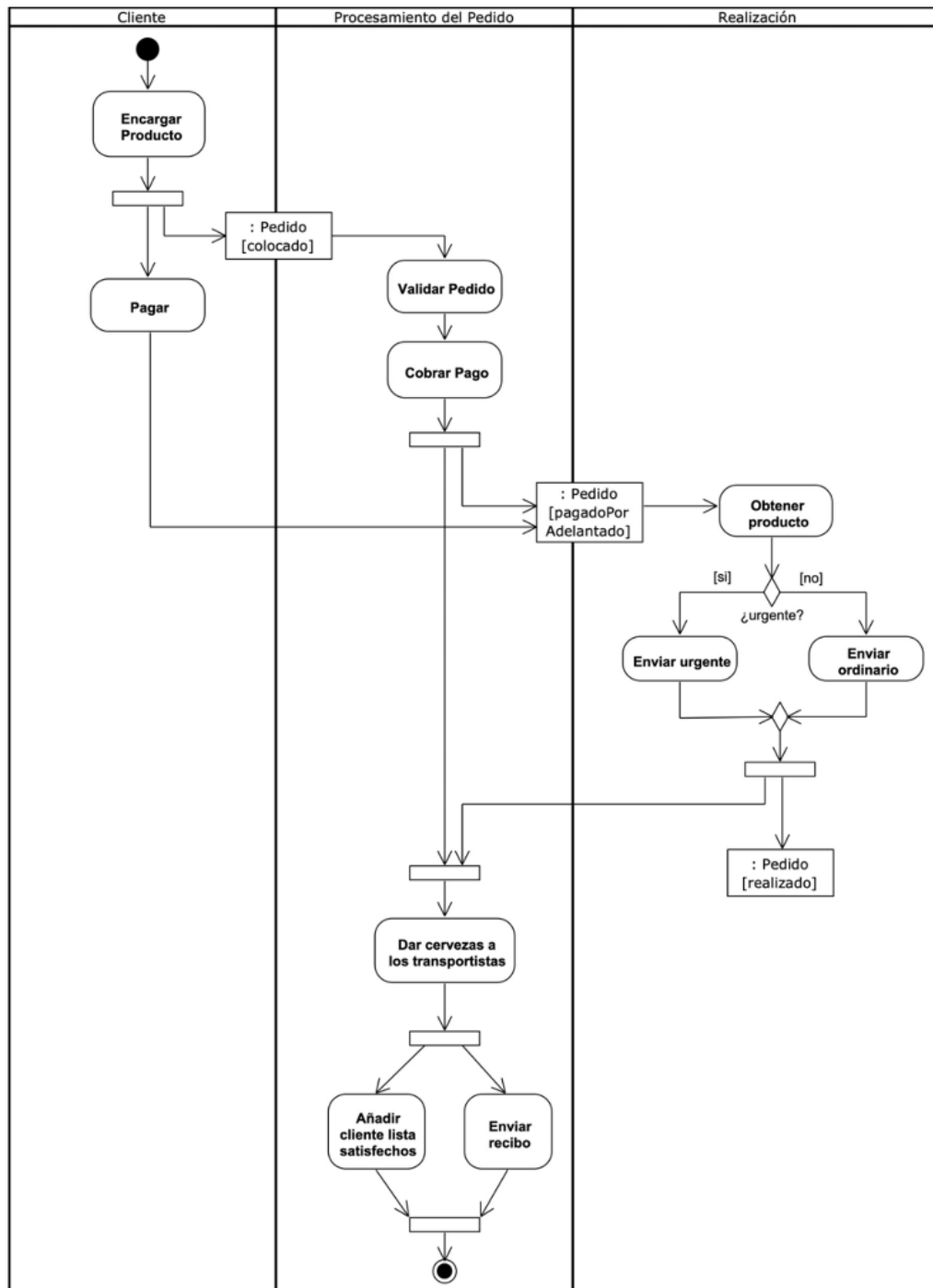
    public void finalizarVenta() {...}
    public void crearNuevaVenta() {...}

    public void introducirArticulo(ArticuloID id, int cant) {
        EspecificacionProducto espec = catalogo.getEspecificacion(id);
        venta.crearLineaVenta(espec, cant);
    }
}

public class Venta {
    private Date fechaHora = new Date();
    private List lineasVenta = new ArrayList();
    ...
    public void crearLineaVenta(espec, cant){
        lineasVenta.añadir(new LineaVenta(espec, cant));
    }
}
  
```

- h) Un vehículo puede tener tres estados: detenido, avanzando y retrocediendo. El estado avanzando puede a su vez tener tres estados: en primera, en segunda o en tercera. Según esta descripción haga el diagrama de estado correspondiente.
 i) Convierta el diagrama de estado del ejercicio anterior (h) en un diagrama de actividad

- j) Vea el siguiente diagrama de actividades y describa qué indica el mismo. Identifique en el diagrama cuántos objetos hay, los distintos estados de esos objetos, cuántas actividades hay, donde hay bifurcaciones, uniones de bifurcación, nodos de decisiones, para qué se usan las particiones o carriles. Según el diagrama, ¿en qué momento los transportistas van a recibir cervezas? ¿qué se hace primero, añadir el cliente a la lista de satisfechos o enviar el recibo?



- k) Convierta el diagrama de actividad del ejercicio anterior (j) en un diagrama de estado