

POO

1. Introducción

1.1 Introducción a la programación



Paradigmas de Programación

- Paradigma.
 - RAE. Teoría o conjunto de teorías cuyo núcleo central se acepta sin cuestionar y que suministra la base y modelo para resolver problemas y avanzar en el conocimiento
- División principal: Imperativo VS Declarativo
 - Imperativo: como se debe procesar
 - Estructurado
 - Basada en estructuras de control
 - Orientado a Objetos
 - Orientado a Eventos
 - La ejecución viene determinada por los sucesos que ocurren en el sistema (UI...)
 - Orientado a Aspectos
 - Requisitos transversales (monitorización, errores, seguridad...)
 - Declarativo: que se debe procesar
 - Funcional
 - Reactivo
 - Lógico
- **Java.** Lenguaje de programación que implementa varios paradigmas.
- **Framework.** Es un entorno de trabajo para la organización y realización de software.

Paradigmas de Programación

Imperativo

```
public class Imperative {
    public static void main(String[] args) {
        System.out.println("POO");

        final int MIDDAY = 12;
        final String NAME = "Jesús";
        if (LocalDateTime.now().getHour() < MIDDAY) {
            System.out.println("Buenos días " + NAME + "!");
        } else {
            System.out.println("Buenas tardes " + NAME + "!");
        }

        final int[] list = {0, 1, 2, 3};
        int accumulator = 0;
        for (int i = 0; i < list.length; i++) {
            accumulator += list[i];
        }
        System.out.println("Suma (for i): " + accumulator);
        int max = -1; // solo para listas con valores naturales
        for (int item : list) {
            if (item > max) {
                max = item;
            }
        }
        System.out.println("Máximo (for each): " + max);
    }
}
```

Programación Orientada a Objetos

Declarativo

```
public class Declarative {
    public static void main(String[] args) {

        final int[] list = {0, 1, 2, 3};
        int sum = Arrays.stream(list)
            .sum(); //declarative
        System.out.println("Suma: " + sum);

        int max = Arrays.stream(list)
            .max() //declarative
            .orElse(-1); //declarative
        System.out.println("Máximo: " + max);
    }
}
```

Paradigmas de Programación

Programación Orientado a Objetos. Conocimiento humano

- Abstracción
 - Acción y efecto de separar por medio de una operación intelectual un rasgo o una cualidad de algo para analizarlos aisladamente o considerarlos en su pura esencia o noción (*RAE*)
- Encapsulación
 - Es el proceso por el que se ocultan los detalles del soporte de las características esenciales de una abstracción.
- Modularidad
 - La modularidad es el proceso de descomposición de un sistema en un conjunto de piezas poco acoplados y cohesivos.
 - Bajo acoplamiento & alta cohesión
- Jerarquía
 - Es una organización de un conjunto de elementos en grados o niveles

Paradigmas de Programación

Programación Orientado a Objetos. Elementos

- Clases y objetos.
 - Una clase es la descripción de una abstracción que sirve de molde para crear objetos.
 - Describe los atributos y los métodos sobre esos atributos.
 - Los objetos son las instancias de las clases, que permiten ejecutar los métodos y pueden alterar el valor de los atributos.
- Atributos y estado.
 - Los atributos son cada uno de los posibles datos de una clase.
 - El estado representa el valor de los atributos de un objeto en un momento dado.
- Métodos y mensajes.
 - Los métodos son las funciones u operaciones que pueden ejecutar los objetos sobre sus atributos. Pueden disponer de un grupo de parámetros.
 - Los mensajes son las invocaciones de los métodos.

Paradigmas de Programación

Programación Orientado a Objetos. Elementos

- Herencia y polimorfismo.
 - La Herencia es el mecanismo por el que se pueden crear nuevas clases a partir de otras existentes. Debiera responder a “es un”, ejemplo: Circulo hereda de Figura, por lo tanto, un Circulo es una Figura.
 - Las clases ascendientes se llaman superclases (o clases padres).
 - Las clases descendientes se llaman subclases (o clases hijas).
 - Se heredan los atributos, pudiendo añadir nuevos atributos. No se pueden eliminar atributos de la superclase.
 - Se heredan los métodos pudiendo añadir nuevos métodos. Se pueden sustituir los métodos heredados mediante, pero siempre que se respete el Principio de Sustitución de Liskov.
 - El polimorfismo es la capacidad de mandar un mensaje a un objeto de la clase padre, pero realmente lo recibe una clase hija que desconocemos su tipo.

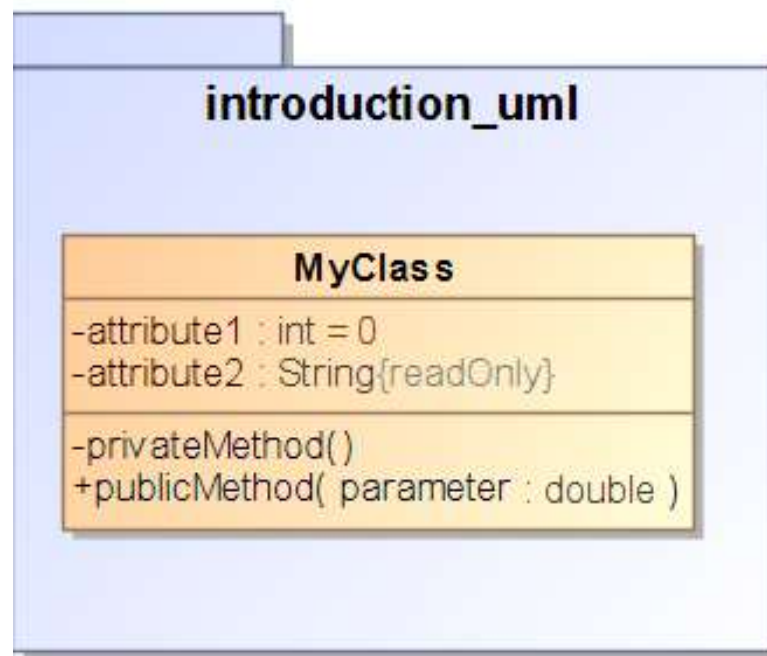
Lenguaje Unificado de Modelado

UML: Unified Modeling Language

- Es un lenguaje gráfico para visualizar, especificar, construir y documentar un sistema software.
- Diagrama de clases. Muestra las clases en un sistema, atributos y operaciones de cada clase y la relación entre cada clase.
- Diagrama de objetos. Muestran la relación entre los objetos en un momento dado.
- Diagrama de secuencia. Muestran cómo los objetos interactúan entre sí y el orden en que se producen esas interacciones.
- Diagrama de componentes, diagramas de despliegues, diagramas de casos de uso...
- <https://www.magicdraw.com>, <https://createlly.com/>, <https://www.diagrams.net/> ...

Lenguaje Unificado de Modelado

UML: Unified Modeling Language



Modelo del dominio

- Es un modelo conceptual de todos los temas relacionados con un problema específico.
- Ejemplos...
 - Biblioteca.

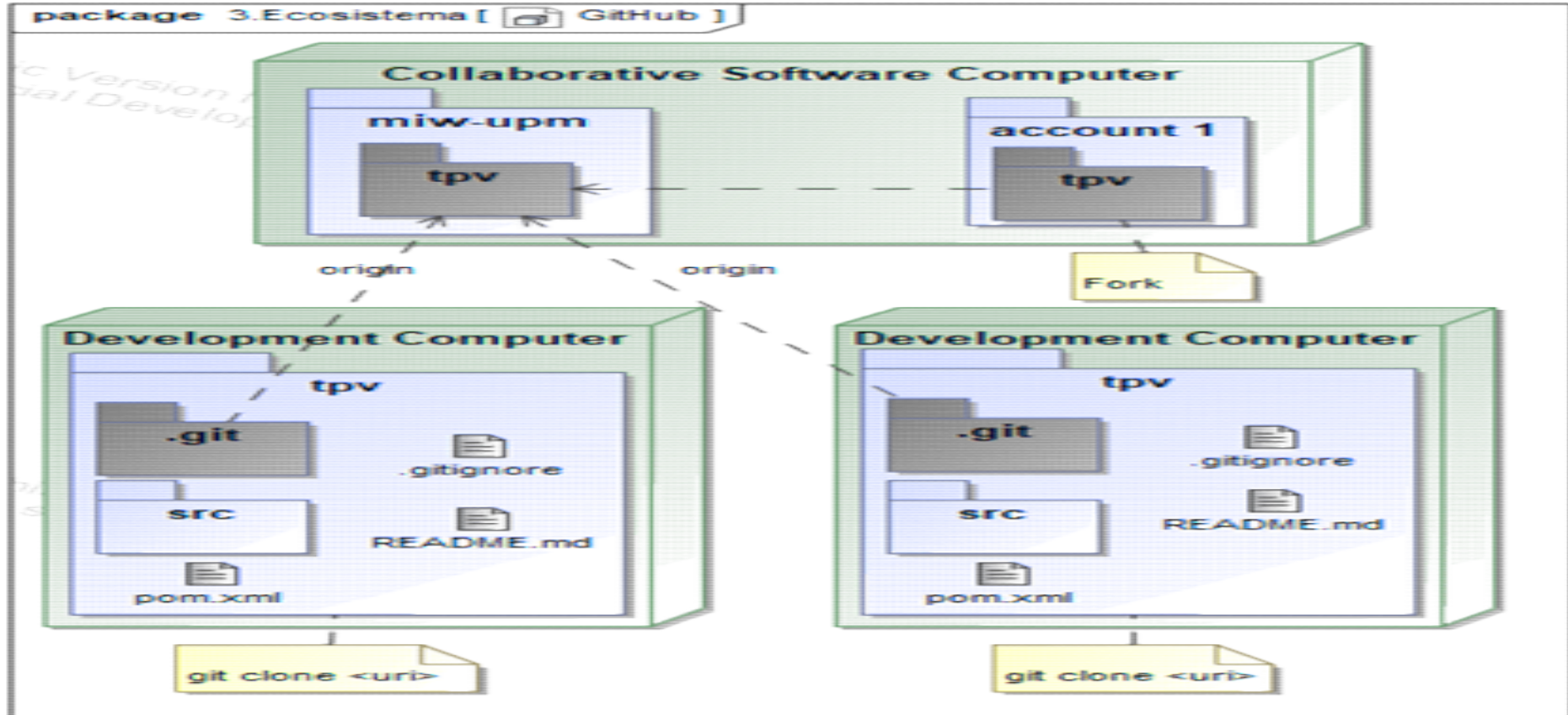
JAVA

- Entorno de Desarrollo Integrado (IDE).
 - IntelliJ IDEA: <https://www.jetbrains.com/es-es/idea/>
- Maven
 - Es una herramienta de gestión y construcción de proyectos de software con Java.
 - Funcionalidad: Identificar el componente, resolver dependencias, lanzar los test, empaquetar...
 - Se basa en un modelo de objetos del proyecto (POM)
 - Fichero pom.xml situado en la raíz del proyecto.
 - Concepto
 - Artefacto. Componente software que es la unidad mínima con la que trabaja Maven
 - Coordenadas. Sistema donde se determina de forma única a cada uno de los artefactos en Internet.
 - Group Id. Identificación del grupo. Normalmente se utiliza el nombre del dominio, al revés: es.upm.etsisi
 - Artifact Id. Identificación del artefacto: poo.
 - Version. 1.0.0-SNAPSHOT, 1.3.4-RC (Release Candidate), 1.4.5-Release
 - Empaquetado
 - Tipo de artefacto: JAR, POM, WAR, EAR, RAR...

Git

- Control de versiones distribuido.
- Muy fiable, imposible perder el proyecto.
- Trabaja sin necesidad de conexión al remoto, muy rápido. Se podrá sincronizar con el remoto, pero con asistencia...
- Snapshot: instantánea (commits)
- CLI: *<https://git-scm.com>*
- *<https://gitlab.etsisi.upm.es/>*

Git



Git

Comandos CLI

- `git config --global user.name "???"`
- `git config --global user.email "???"`
- `git init`
- `git add --all`
- `git commit -m "Initial commit"`
- `git remote add origin https://gitlab.etsisi.upm.es/???/???.git`
- `git push origin mater`
- `git clone https://gitlab.etsisi.upm.es/???/???.git`
- `git fetch origin`
- `git merge origin/master`