UNIVERSIDAD EL BOSQUE

Fundamentos de Programación

Clase 3

Agenda

- ¿Cómo definir la estructura de un programa?
  - Identificar las entidades
  - Identificar las características
  - Buscar las relaciones
- Sintaxis (resumen)
- Métodos
  - Instrucción de retorno
  - Método main
  - Llamadas a métodos

¿Cómo definir la estructura?

- El objetivo es **observar**, no opinar sobre los elementos observados ni proponer cambiarlos
- Para expresar el modelo se utiliza la sintaxis definida en el diagrama de clases del lenguaje **UML** (*Unified Modeling Language*)
  1. Identificar las entidades (clases)
  2. Identificar sus características (atributos)
  3. Buscar las relaciones

Identificar las entidades

- Elementos del contexto(concretos o abstractos)
- Una pista es identificar los sustantivos del enunciado del problema
- Cada entidad se denomina **Clase**
- Por convención, los nombres de las clases empiezan por mayúsculas y utilizan *camelcase*

Identificar las características

- Tienen un nombre significativo y la descripción del conjunto de valores que puede tomar
- Se denominan **atributos**
- El nombre de un atributo debe empezar con una letra y no tener espacios en blanco
- Por convención los nombres de los atributos empiezan en minúsculas y usan *camelcase*

Buscar las relaciones

- Se puede ver como una característica de una entidad cuyo valor está representado por otra clase
- Se denominan **asociaciones**
- Es posible tener varias relaciones entre dos clases
- Se representa en UML como una flecha, cuya dirección indica la entidad que contiene a la otra

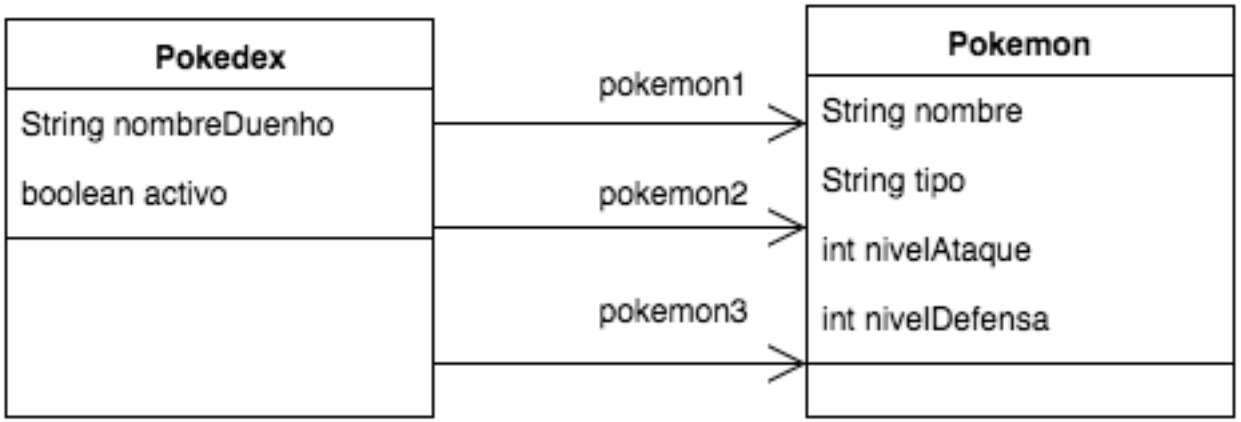
Quiz (20 mins)

- Una empresa de videojuegos lo ha contratado a usted para que desarrolle el juego de **Pokemon**. El juego se compone de un pokedex y 3 pokemones. El **Pokedex**, que es como una agenda de pokemones, tiene el nombre del dueño y si está activo o no. Cada **Pokemon** tiene su nombre, el tipo de pokemon (agua, fuego, eléctrico), un nivel de ataque (número entero) y un nivel de defensa (número entero).
- Su trabajo es diseñar el diagrama de clases del juego y escribir el código de cada clase

Solución

```
public class Pokedex
{
        private String nombreDuenho;
        private boolean activo;
        private Pokemon pokemon1;
        private Pokemon pokemon2;
        private Pokemon pokemon3;
}
```

```
public class Pokemon
{
        private String nombre;
        private String tipo;
        private int nivelAtaque;
        private int nivelDefensa;
}
```



Sintaxis (Resumen)

- Clases

```
public class MiClase {…}
```

- Atributos

```
private int miAtributo;
```

- Asociaciones

```
private TipoDeDato otroAtributo;
```

Métodos

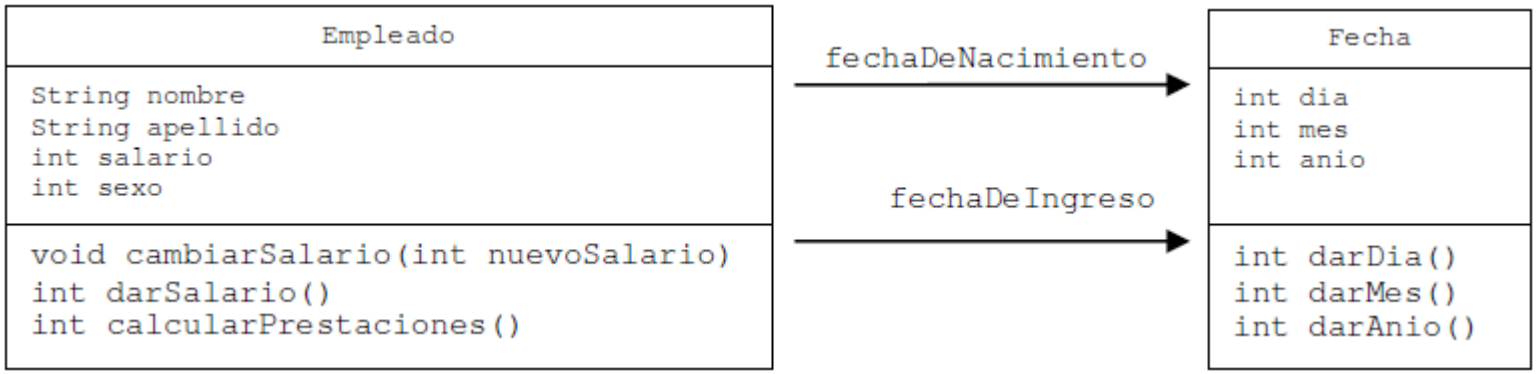
- Son algoritmos los cuales pretenden resolver un problema puntual dentro del contexto del problema global.
- Elementos
  - Nombre
  - Lista de parámetros
  - Tipo de respuesta
  - Cuerpo del método

Métodos

- Un método tiene una Firma o Signatura

  <visibilidad> <tipo_de_retorno> <nombre> (<tipo> <nombre_parametro>, …)

```
public void cambiarSalario(int nuevoSalario)
{
// Aquí va el cuerpo del método
}
```



Métodos

- **Visibilidad**: **public**
- **Tipo de retorno**: Tipo de dato que el método va a retornar. Si no hay retorno el tipo de dato es **void**
- **Nombre**: debe ser un verbo, empieza con minúscula y usa *camelcase*
- **Parámetros**: lo que necesita el método para funcionar

Instrucción de retorno

- Todos los métodos declarados para devolver un resultado (todos los métodos que NO son de tipo **void**) deben tener en su cuerpo una instrucción de retorno

```
        //-------------------------------
        // Metodos
        //-------------------------------
        public int darSalario()
        {
                return salario;
        }
}
```

Método main

- Es el punto de entrada de un programa
- A partir de este método, se crean todos los objetos del programa

```
public static void main(String[] args)
{
    //Aquí va el código del programa
}
```

Llamadas a métodos

- Un método se “llama” utilizando su nombre dentro de una instrucción.
- Se pueden llamar métodos que se encuentren en la misma clase o métodos públicos de las clases asociadas.
- Si se quieren llamar métodos de clases asociadas, DEBE hacerse a través de una instancia (objeto de la clase)

Llamadas a métodos

- El proceso de llamar métodos de otro objeto se denomina invocación

<nombre_de_instancia>.<nombre_de_metodo>

```
public class Empleado {
...

public void miProblema()
{
        ...
        int valor = fechaDeIngreso.darAnio();
        ...
}
```

```
public class Fecha {

private int anio;
        ...

public int darAnio()
{
        return anio;
}
...
```

Llamadas a métodos con parámetros

1. ¿Cuándo necesita parámetros un método?
   - Cuando la información que contiene el objeto en sus atributos no es suficiente para resolver el problema
2. ¿Cómo se declara un parámetro?
   - En la signatura del método se define el tipo del parámetro y se le asigna un nombre
3. ¿Cómo se utiliza el valor del parámetro?
   - Basta con utilizar su nombre
4. ¿Se puede utilizar el parámetro por fuera del método?
   - No

Llamadas a métodos con parámetros

5. ¿Cómo se hace para definir los valores de los parámetros?
   - El objeto que llama el método debe dar los valores de los parámetros del método.
6. ¿Cómo se hace la relación entre esos valores y los parámetros?
   - Los valores deben tener en cuenta el orden de los parámetros
7. ¿Qué sucede si hay más o menos valores que parámetros?
   - El compilador informa que hay un error en la llamada.

¿PREGUNTAS?