

Ámbito referente a una clase

Dr. José Lázaró Martínez Rodríguez

Introducción

- El ámbito de una variable define su alcance de uso, o lo que es lo mismo, en que secciones de código una variable estará disponible.
- Fuera de este ámbito, una variable **NO** podrá ser accedida (no existe).

Tipos de ámbito en Java

- En Java tenemos tres tipos de ámbito que pueden aplicar a una variable:
 - Local
 - Global
 - Estático

```
public class MiClase {  
    static int variableEstatica;  
    int variableGlobal;  
  
    void miMetodo(int parametro) {  
        int variableLocal;  
  
        // parametro también es local dentro del método  
    }  
}
```

Variables de ámbito local

- Las variables de ámbito local, o de bloque, son aquellas que sólo pueden ser accedidas desde el bloque de código en el que han sido declaradas

```
void miMetodo(int parametro) {  
    int variableLocal = new Random().nextInt();  
    System.out.println("El valor de variableLocal es: " + variableLocal);  
    System.out.println("EL valor de parámetro es: " + parametro);  
}
```

Variables de ámbito global

- Las variables de ámbito global, o de instancia, son aquellas que pertenecen a cada instancia concreta de la clase donde han sido declaradas
- dependiendo del modificador de visibilidad
 - podrían ser sólo accedidas desde la propia instancia a la que pertenecen:

Variables de ámbito global

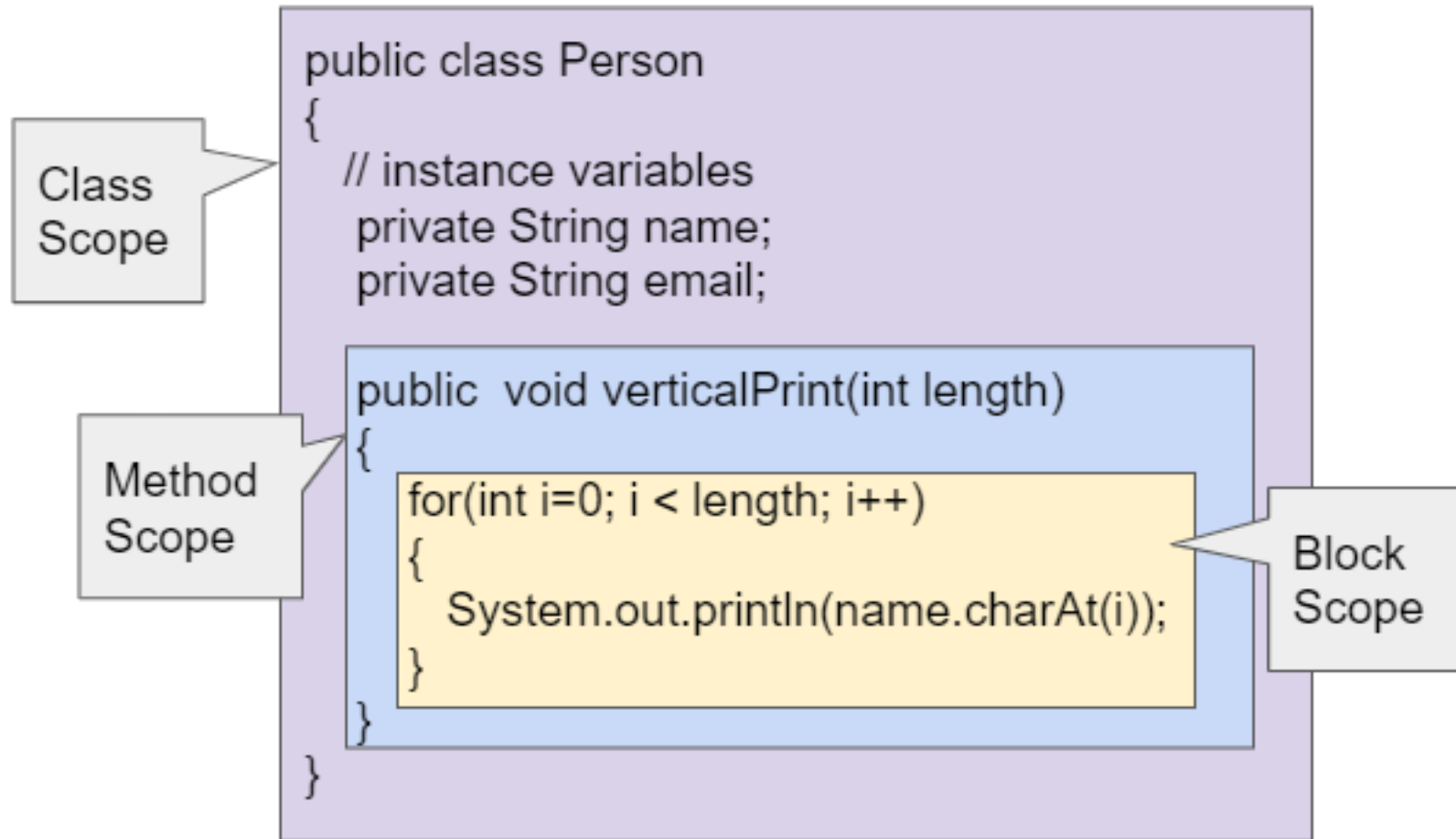
```
public class MiClase {  
    int variableGlobal;  
  
    void miMetodo() {  
        System.out.println("Valor de variableGlobal: " + variableGlobal);  
    }  
}  
  
public class OtraClase {  
    void otroMetodo() {  
        // Error de compilación, no es visible  
        System.out.println("Valor de variableGlobal: " + variableGlobal);  
    }  
}
```

Variables estáticas

- Las variables estáticas, o de clase, son aquellas que pertenecen a la propia clase donde han sido declaradas, y dependiendo del modificador de visibilidad usado podrían ser sólo accedidas desde la propia clase en la que han sido declaradas:

```
public class UnaClase {  
    public static int variableEstatica;  
}  
  
public class OtraClase {  
    void metodo() {  
        System.out.println("Valor de UnaClase.variableEstatica:" +  
UnaClase.variableEstatica);  
    }  
}
```

Niveles de alcance (scope)



Modificadores de acceso

- `public`: Todo el mundo puede acceder al elemento. Si es un atributo (dato miembro) todo el mundo puede ver el dato miembro, es decir, usarlo y asignarlo. Si es un método todo el mundo puede invocarlo.
- `private`: Sólo se puede acceder al dato miembro desde métodos de la clase, o sólo puede invocarse el método desde métodos de la clase.
- `protected`: Sus datos/métodos miembro tienen acceso a nivel de clase, paquete y subclase.
- `sin modificador`: Se pueden acceder o invocar a sus elementos desde cualquier clase del paquete donde se define la clase.

Modificadores de acceso

Modificador	Clase	Package	Subclase	Otros
public	Y	Y	Y	Y
protected	Y	Y	Y	N
sin modificador	Y	Y	N	N
private	Y	N	N	N

- Y = si
- N = no

Ejemplo

- ¿Cuáles declaraciones son a nivel de clase?

```
public class Name {  
  
    private String first;  
    public String last;  
  
    public Name(String theFirst, String theLast) {  
        String firstName = theFirst;  
        first = firstName;  
        last = theLast;  
    }  
}
```

Ejemplo

- ¿Cuáles variables son a nivel de método?

```
public class Name {  
  
    private String first;  
    public String last;  
  
    public Name(String theFirst, String theLast) {  
        String firstName = theFirst;  
        first = firstName;  
        last = theLast;  
    }  
}
```

Ejemplo

- Indique si hay errores en el siguiente código

```
public class A {  
{  
    System.out.println(i);  
}  
int i = 10;  
  
    public static void main(String E) {  
        A a = new A;  
    }  
}
```

Ejemplo

- ¿Hay algún error aquí?

```
public class A {  
    public static void main(String[] args)  
    {  
        int i;  
        System.out.println(i);  
    }  
}
```