Alcance de variables

¡Comencemos!

¿Qué aprenderás?

- Aplicar el concepto de alcance de variables para así usarlas correctamente.
- Comprender el ámbito en que las variables actúan.

Introducción

Los tipos de variable y el alcance de estas son conceptos muy importantes, nos permiten entender desde dónde podemos acceder a una variable. A veces podemos enfrentarnos a problemas donde queremos acceder a una variable, pero esta no lo permite. Es por eso que a continuación aprenderemos cuáles son los distintos tipos de variables dentro del mundo de Java y cuál es su alcance.

¡Vamos con todo!

Categorías de variables

Existen varias categorías de variables en Java que definen en parte el alcance de las mismas. Estas categorías son las siguientes:

- Variables locales
- Variables de instancia
- Variables de clase /static

El alcance
El alcance o scope en inglés, define desde donde podemos acceder a una variable.

Variables locales

Una variable definida dentro de un método puede ser accedida solamente dentro del mismo método. No puede ser accedida fuera de este.

```
public static void main(String[] args) {
    System.out.println(aprobado(5,6));
}
static boolean aprobado(float nota1, float nota2) {
    float promedio = (nota1+nota2)/2; //Promedio variable local del método
    return promedio <=5 ? true:false;
}
```

Si en el main tratamos de acceder al promedio.

```
System.out.println(promedio);
```

Obtendremos el siguiente error, donde nos dice que la variable promedio no se puede resolver debido a que no está en el mismo ámbito o scope.

```
Exception in thread "main" java.lang.Error: Unresolved compilation problem:
promedio cannot be resolved to a variable
```

Los parámetros también cuentan como variables locales

Si en el main tratamos de acceder a `nota1`, obtendremos el mismo error que para promedio.

```
System.out.println(nota1);
```

```
Exception in thread "main" java.lang.Error: Unresolved compilation problem:
nota1 cannot be resolved to a variable
```

Variables de instancia

De momento no hemos utilizado ninguna variable de instancia. Hemos tenido variables locales en el método main y variables locales en los métodos que hemos creado.

Para crear una variable de instancia, debemos primero crear una nueva clase. Las variables de instancia hacen referencia a variables dentro de una clase, fuera de los métodos, las cuales pueden ser accedidas desde cualquiera de los métodos de dicha clase.

Veamos esto con un ejemplo.

Creamos una nueva clase a la cual denominaremos Notas, dentro del mismo paquete:

```
//Nueva clase
public class Notas {
    int nota1;
    int nota2;
    /*A los métodos que se llaman igual que la clase, los llamamos
constructores de la clase*/
    public Notas(int n1, int n2) {
        this.nota1 = n1;
        this.nota2 =n2;
    }
    //Otro método de la clase
    public float promedio() {
        return (float)(nota1+nota2)/2.0f;
    }
}
```

Por otro lado, tenemos el main de la siguiente forma.

```
package métodos;
import java.util.Scanner;
public class Metodos {
    public static void main(String[] args) {
        //Se crea la instancia de la clase Notas
        Notas n = new Notas(4,5);
        System.out.printf("%f\n",n.promedio());
    }
}
```

Así como hemos creado Strings (que es una clase), podemos utilizar las clases que nosotros necesitemos personalizar. Podemos ver que creamos una variable de tipo `Notas`, a la cual denominamos `n`.

Para crearla, utilizamos el constructor `Notas`, la cual recibe 2 parámetros y se inicializa el objeto con los valores `notas1 = 4` , y `notas2 = 5` .

En el ejemplo anterior, las variables `notas1` y `notas2` podemos accederlas o usarlas desde el método promedio, sin pasarlas como parámetros.

Variables de clase/static

Para crear una variable de clase, basta con agregar al inicio de la variable la palabra `static`. Pero, ¿qué pasa al hacer dicha acción?.

Al definir una variable de tipo `static`, hará que al modificar por ejemplo la variable en un objeto, se modifique para todos los objetos que estén creados.

Veamos el ejemplo anterior, definiendolas como static

```
package metodos;
public class Notas {
    static int nota1;
    static int nota2;
    public Notas(int n1, int n2) {
        this.nota1 = n1;
        this.nota2 =n2;
    }
    public float promedio() {
        return (float)(nota1+nota2)/2.0f;
    }
}
```

Y al implementar la clase `Notas` en el método `Main`:

```
package metodos;
import java.util.Scanner;
public class Metodos {
    public static void main(String[] args) {
        Notas n = new Notas(4,5);
        Notas n2 = new Notas(4,6);
        System.out.printf("%f %f\n",n.promedio(),n2.promedio());
    }
}
```

El resultado para ambos promedios será,

```
5,000000 5,000000
```

Ya que al instanciar `n2`, asigna las variables `notas1` y `notas2` para ambos en 4 y 6.

Ejercicio guiado: Calculadora Suma y Resta

Contexto

Se necesita crear una calculadora que pida números al usuario de forma infinita hasta que el usuario ingrese el operador igual.

Requerimientos

1. Los números a sumar o restar deben ser de tipo `float`.
2. Se debe indicar al usuario que debe ingresar, si un número, o un operador.
3. Como operador solo se debe admitir el ingreso del signo más (+) o menos (-).
4. Cuando el usuario ingrese el signo igual (=), se mostrará el resultado.

Solución

Requerimiento 1:

a. Se crea el proyecto, con una clase `Calculadora.java` dentro del package cl.desafiolatam

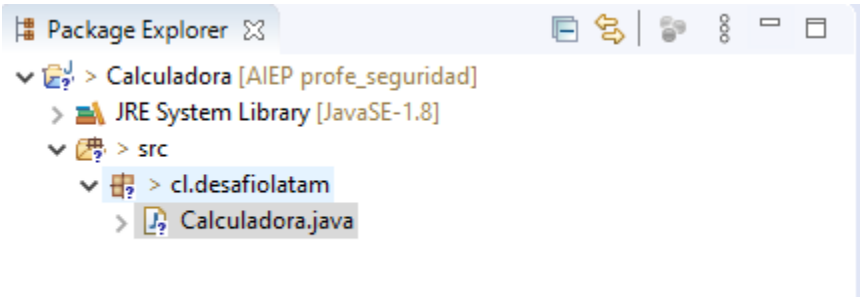

Imagen 1. Crear clase Calculadora.
Fuente: Desafío Latam.

```
package cl.desafiolatam;

public class Calculadora {

	public static void main(String[] args) {

	}
}
```

b. Se declaran las variables

```
//Se declaran las variables a utilizar
//El número que ingrese el usuario debe ser float
float numero = 0f;
float resultado = 0;
//Variable tipo String, ya que puede ser un operador o un número
String ingreso = null;
//Variable que guardará el código ascii de cada ingreso del usuario
int ascii = 0;
/*
 * Se crea variable contador, para identificar si es el primer ingreso
 * de usuario,el cual debe ser un número, y luego un operador, número
 * así sucesivamente. Si contador es cero es inicio y debe ser un
 * número. Si es impar, debe ser un operador +, -, =
*/
int contador = 0;
Scanner sc = new Scanner(System.in); //Se crea objeto Scanner
```

Requerimiento 2:

a. Utilizando el contador, se evalúa si el módulo es cero (par) o impar (distinto de cero) para saber si debe ingresar el usuario un número o bien un operador.

```
//Si contador es par el uuario debe ingresar un número, de lo contrario
debe ingresar un operador
if((contador % 2) == 0) {
      System.out.println("Ingrese un número: ");
}else {
      System.out.println("Ingrese un operador (+, -, =): ");
      contador++;
}
```

Requerimiento 3:

a. Se agregan las condiciones para evaluar qué es lo que está ingresando el usuario, si un signo =, + o -, para así saber qué cálculo se debe realizar.

```
//Se lee por pantalla el ingreso del usuario, operador o número
ingreso = sc.next();
/*
 * Códigos ascii para operadores
 * + --> 43
 * - --> 45
 * = --> 61
 */
ascii = (int) ingreso.charAt(0);
/*
 *Si el programa inicio, es decir el contador esta en cero y el usuario
 * no ingreso un operador (+, -, =)
*/
if(contador == 0 && ascii != 43 && ascii != 45 && ascii != 61) {
      //Se hace un cast de ingreso (String) a float
      /*
       * Todas las clases tienen un parse, Integer.parseInteger,
       * Double.parseDouble etc. SE hace de esta manera, porque un
       * String no se puede transformar en un número, sin embargo si
       * el String no es un número, el programa dará un error.
      */
      numero = Float.parseFloat(ingreso);
      //Se asigna a resultado el primer numero ingresado
      resultado = numero;
      contador++;
}else { // De lo contrario, es el segundo ingreso de usuario
      //Suma
      if(ascii == 43) {
            System.out.println("Ingrese un número: ");
            ingreso = sc.next(); //Debiese ser un número o si no error
            numero = Float.parseFloat(ingreso);
            resultado = resultado + numero;
            contador++;
      }
      //Resta
```

```
      if(ascii == 45) {
            System.out.println("Ingrese un número: ");
            ingreso = sc.next(); //Debiese ser un número o si no error
            numero = Float.parseFloat(ingreso);
            resultado = resultado - numero;
            contador++;
      }
}
```

Requerimiento 4:

El algoritmo anterior del requerimiento 2 y 3 debe quedar dentro de un loop `do-while`, para que se repita el proceso hasta que el usuario ingrese el operador =.

```
do{
.
.
.
//Repite mientras el ingreso no sea =
}while((int)ingreso.charAt(0) != 61);
System.out.printf("El Resultado es: %f", resultado);
```

Solución completa

```
package cl.desafiolatam;

import java.util.Scanner;

public class Calculadora {

	public static void main(String[] args) {
		//Se declaran las variables a utilizar
		//El número que ingrese el usuario debe ser float
		float numero = 0f;
		float resultado = 0;
		//Variable tipo String, ya que puede ser un operador o un
número
		String ingreso = null;
		//Variable que guardará el código ascii de cada ingreso del
usuario
		int ascii = 0;
		/*
		 * Se crea variable contador, para identificar si es el primer
ingreso
		 * de usuario,el cual debe ser un número, y luego un operador,
número
		 * así sucesivamente. Si contador es cero es inicio y debe ser
un
		 * número. Si es impar, debe ser un operador +, -, =
		*/
		int contador = 0;
		Scanner sc = new Scanner(System.in); //Se crea objeto Scanner
		do {
			//Si contador es par el usuario debe ingresar un número,
de lo contrario debe ingresar un operador
			if((contador % 2) == 0) {
				System.out.println("Ingrese un número: ");
			}else {
				System.out.println("Ingrese un operador (+, -, =):
");
				contador++;
			}
```

```
                  //Se lee por pantalla el ingreso del usuario, operador o
número
                  ingreso = sc.next();
                  /*
                   * Códigos ascii para operadores
                   * + --> 43
                   * - --> 45
                   * = --> 61
                   */
                  ascii = (int) ingreso.charAt(0);
                  /*
                   *Si el programa inicio, es decir el contador esta en
cero y el usuario
                   * no ingreso un operador (+, -, =)
                  */
                  if(contador == 0 && ascii != 43 && ascii != 45 && ascii
!= 61) {
                        //Se hace un cast de ingreso (String) a float
                        /*
                         * Todas las clases tienen un parse,
Integer.parseInteger,
                         * Double.parseDouble etc. SE hace de esta manera,
porque un
                         * String no se puede transformar en un número, sin
embargo si
                         * el String no es un número, el programa dará un
error.
                        */
                        numero = Float.parseFloat(ingreso);
                        //Se asigna a resultado el primer numero ingresado
                        resultado = numero;
                        contador++;
                  }else { // De lo contrario, es el segundo ingreso de
usuario
                        //Suma
                        if(ascii == 43) {
                              System.out.println("Ingrese un número: ");
                              ingreso = sc.next(); //Debiese ser un número
o si no error
```

```
                              numero = Float.parseFloat(ingreso);
                              resultado = resultado + numero;
                              contador++;
                        }
                        //Resta
                        if(ascii == 45) {
                              System.out.println("Ingrese un número: ");
                              ingreso = sc.next(); //Debiese ser un número
o si no error
                              numero = Float.parseFloat(ingreso);
                              resultado = resultado - numero;
                              contador++;
                        }
                  }
            //Repite mientras el ingreso no sea =
            }while((int)ingreso.charAt(0) != 61);
            System.out.printf("El Resultado es: %f", resultado);
      }

}
```