ESCUELA SUPERIOR DE COMPUTO

PROGRAMACIÓN ORIENTADA A OBJETOS

PRÁCTICA 5

Tipos, Variables, Métodos y Constructores

Comprender, utilizar y aplicar correctamente los tipos en variables utilizados en el lenguaje de programación Java

Rubio Haro Rodrigo R.

CDMX. MAYO, 2020.

INSTITUTO POLITÉCNICO NACIONAL

Práctica 5: Operadores y Clases de los Tipos de Datos

1. Introducción

Los operadores de Java son muy parecidos en estilo y funcionamiento a los de C. En la siguiente tabla aparecen los operadores que se utilizan en Java, por orden de precedencia:

```
.    []    ()
++    --
!    ~    instanceof
*    /    %
+    -
<<    >>    >>>
<    >    <=    >=    ==    !=
&    ^    |
&&    ||
? :
=    op=    (*=    /=    %=    +=    -=    etc.)    ,
```

2. Desarrollo

2.1 Convertidor

Se creó transformó el programa que convierte cadenas a flotantes, chars y doubles.

2.1.1 Código

```
public class Conversiones {

    public static void main(String args[]) {
        Convertidor convertidor = new Convertidor();
        convertidor.establecerCadena("1234");
        convertidor.imprimirConversionChars();
        convertidor.establecerCadena("1.358");
        convertidor.imprimirConversionFloats();
        convertidor.establecerCadena("102.3654");
        convertidor.imprimirConversionDoubles();
    }
}
```

```
public class Convertidor {
  private String cadena;

  public void establecerCadena(String cadena) {
      this.cadena = cadena;
  }

  public void imprimirConversionChars() {
      // se declara un tipo char
      char c;
      // el for recorre la longitud del atributo cadena e imprime uno por uno sus
      // caracteres en forma de enteros
      for (int n = 0; n < cadena.length(); n++) {
          c = cadena.charAt(n);
          int i = Character.digit(c, 10);
          System.out.println("El numero entero es : " + i);
      }
  }

  public void imprimirConversionFloats() {
      // Se convierte la cadena en tipo float con el metodo valueOf
      // de la clase Float
      Float fo = Float.valueOf(cadena);
      // Se imprime fo de tipo Float
      System.out.println("El objeto Flotante es : " + fo);

      // Se convierte objeto fo de tipo Floar en tipo float
      // con el metodo floatValue de la clase Float
      float f = fo.floatValue();
      // Se imprime f de tipo float
      System.out.println("El numero flotante es : " + f);
  }

  public void imprimirConversionDoubles() {
      // Se convierte la cadena en tipo Double con el metodo valueOf de la clase
      // Double
      Double D = Double.valueOf(cadena);
      // Se imprime D de tipo Double
      System.out.println("El objeto Double es : " + D);

       // Se convierte objeto D de tipo Double en tipo double con el
       //metodo doubleValue
      // de la clase Double
      double d = D.doubleValue();
      // Se imprime d de tipo double
      System.out.println("El numero double es : " + d);
  }
}
```

2.1.2 Compilación y Ejecución

```
chavo@LAPTOP-R4P7ROT5 MINGW64 /f/Roy/programming/POO/escom/De la O/Departamental
fo_Practica05_2CV1_Departamental1/1.Convertidor (master)
$ javac *.java

chavo@LAPTOP-R4P7ROT5 MINGW64 /f/Roy/programming/POO/escom/De la O/Departamental
fo_Practica05_2CV1_Departamental1/1.Convertidor (master)
$ java Conversiones
El numero entero es : 1
El numero entero es : 2
El numero entero es : 3
El numero entero es : 4
El objeto Flotante es : 1.358
El numero flotante es : 1.358
El objeto Double es : 102.3654
El numero double es : 102.3654
```

Resultado de compilar y ejecutar el código.

2.2 Flotantes

Se transformó a paradigma orientado a objetos el programa que sumaba flotantes consecutivamente.

2.2.1 Código

```
class InterfazSumador {
    public static void main(String args[]) {
        SumadorFlotantes menu = new SumadorFlotantes();
        menu.sumarFlotantes();
        menu.imprimirRespuesta();
    }
}
```

```java
import java.io.*;

public class SumadorFlotantes {

    private float respuesta;

    public String leer() {
        String s = "";
        DataInputStream sd = new DataInputStream(System.in);
        System.out.flush();
        try {
            s = sd.readLine();
        } catch (IOException e) {
            System.out.println("ERROR: Se introdujo el dato");
            System.out.println(e);
        }
        return s;
    }

    public void sumarFlotantes() {
        Boolean activo = true;
        String cadena;

        respuesta = (float) 0.0;// , b;
        Float respuestaEnvoltorio;// , bf;

        while (activo) {
            cadena = pedirNumero();
            if (cadena.equals("") || cadena == null)
                activo = false;
            else {
                respuestaEnvoltorio = Float.valueOf(cadena);
                respuesta += respuestaEnvoltorio.floatValue();
            }
        }
    }

    public String pedirNumero() {
        System.out.print("Introduzca un flotante:");
        return leer();
    }

    public void imprimirRespuesta() {
        System.out.print("La suma de sus numeros es: " + respuesta);
    }
}
```

2.2.2 Compilación y Ejecución

```
chavo@LAPTOP-R4P7ROT5 MINGW64 /f/Roy/programming/POO/escom/De la O/Departamental1
fo_Practica05_2CV1_Departamental1/2.Flotantes (master)
$ javac *.java
Note: SumadorFlotantes.java uses or overrides a deprecated API.
Note: Recompile with -Xlint:deprecation for details.

chavo@LAPTOP-R4P7ROT5 MINGW64 /f/Roy/programming/POO/escom/De la O/Departamental1
fo_Practica05_2CV1_Departamental1/2.Flotantes (master)
$ java interfazSumador
Suma de Flotantes, ingresa enter vacio si quieres finalizar
Introduzca un flotante:2
Introduzca un flotante:2
Introduzca un flotante:3
Introduzca un flotante:
La suma de sus numeros es: 7.0
```

El programa sumo, y mostró el resultado de los flotantes sin ningún problema.

2.3 Calculadora

Se desarrolló el programa que realiza operaciones dependiendo el operador que sea usado y de manera consecutiva al paradigma Orientado a Objetos.

2.3.1 Compilación y Ejecución

```
chavo@LAPTOP-R4P7ROT5 MINGW64 /f/Roy/programming/POO/escom/De la O/Departamental1/P
fo_Practica05_2CV1_Departamental1/3.Calculadora (master)
$ javac *.java
Note: Calculadora.java uses or overrides a deprecated API.
Note: Recompile with -Xlint:deprecation for details.

chavo@LAPTOP-R4P7ROT5 MINGW64 /f/Roy/programming/POO/escom/De la O/Departamental1/P
fo_Practica05_2CV1_Departamental1/3.Calculadora (master)
$ java Calcular
Ingresa una operacion, por ejemplo: +6.4
operacion:+5.5
x += 5.5
Ingresa una operacion, por ejemplo: +6.4
operacion:-3.5
x -= 3.5
Ingresa una operacion, por ejemplo: +6.4
operacion:*-10
x *= -10.0
Ingresa una operacion, por ejemplo: +6.4
operacion:*-1
x *= -1.0
Ingresa una operacion, por ejemplo: +6.4
operacion:
El resultado de tus operaciones
x += 5.5,x -= 3.5,x *= -10.0,x *= -1.0 es: X = 20.0
```

2.4 Envoltorios

Se implementó un programa orientado a objetos con las clases de envoltorio para los tipos primitivos, un sencillo despliegue de tipos de datos.

2.4.1 Compilación y Ejecución

```
chavo@LAPTOP-R4P7ROT5 MINGW64 /f/Roy/programming/POO/escom/De la O/Departamental1/
fo_Practica05_2CV1_Departamental1/4.Datos (master)
$ javac *.java

chavo@LAPTOP-R4P7ROT5 MINGW64 /f/Roy/programming/POO/escom/De la O/Departamental1/
fo_Practica05_2CV1_Departamental1/4.Datos (master)
$ java TiposDeDatos
Imprimiendo datos haciendo uso de envoltorios
dato Integer: 3586
dato Float: 8244.879
dato Byte: 101
dato Character: c
dato Short: 275
dato Long: 1907
dato Double: 5903.865166624499
dato Boolean: true
```

Compiló y ejecutó sin problemas.

3. Conclusión

Se hicieron conversiones explícitas de tipos de datos, creo que java está limitado por las clases al momento de hacer programas inteligentes, al no terminar de entender el concepto de programas INTELIGENTES, así como la poco experiencia en otros lenguajes, me veo limitado a decidir si Java es la mejor opción para este tipo de programas. Más que algún operador, me costo un poco la estructura para distinguir las diferentes formas de los operadores. Considero un útil esfuerzo la creación de estas clases, sin embargo, en mi poca experiencia en el lenguaje, son poco usadas; probablemente los programadores prefieran usar primitivos y no seguir completamente el paradigma Orientado a Objetos.