Síntesis conceptual

Grado: Administración de sistemas informáticos en red
Asignatura: Programación básica
Unidad: 4. Elementos de un programa informático

Resumen

Un programa informático de Java se constituye por un conjunto de elementos, como son:

- El **proyecto**: Consiste en un grupo de archivos y carpetas que se organizan de acuerdo con un criterio que da lugar a un orden lógico, para llevar a cabo el desarrollo de un programa, lo primero que deberemos hacer será crear el proyecto.
    - Solución: Cuando un proyecto o varios que están relacionados entre sí se agrupan, forman una solución, sin embargo, una solución puede contener también archivos sin conexión con los proyectos de esa solución.
- **Identificador**: Un identificador permite que el programador denomine un elemento del programa, ya sean variables, clases, métodos, etc. Y este consiste en una secuencia de uno o varios caracteres.
- **Palabras reservadas**: Palabras que cuentan con su propio significado, por lo que no se pueden utilizar como identificadores en los programas.
- **Clases**: Elemento que encapsula los datos y las operaciones posibles de un objeto.
- **Paquete**: Elemento que permite encapsular entre otras cosas las clases.
- **Variables**: Valores de un objeto o incluso los valores primitivos. Estas variables pueden ser accedidas de diversas maneras, y debemos recordar que estas se eliminarán al salir del ámbito donde fue declarada.
- **Constantes**: Elementos que, a lo largo de la vida del programa, mantienen un mismo valor.
- **Datos primitivos**: El tipo de dato influirá en el tipo de valor posible. Los datos primitivos ya están definidos en el lenguaje por defecto, por lo que, para referenciarlos, hay que utilizar las palabras reservadas para ello.
    - Es posible llevar a cabo conversiones de tipo, *casting*, entre tipos de datos, pero esto puede producir pérdidas en la traducción. Pueden ser de dos tipos:
        - implícitas: se lleva a cabo una operación de asignación, pero no hemos declarado el tipo de conversión a realizar.
        - Explícitas: se llevan a cabo colocando primero el tipo de variable destino y después el dato origen de la conversión.
- **String**: Empleo de objetos de la clase String con el fin de almacenar elementos de tipo *char*.
- **Clases envoltorio**: conocida como Wrapper Class, es un tipo de clase cuya finalidad es la creación de objetos que encierren un dato de tipo primitivo, así como la provisión de métodos para facilitar su manejo. Hay métodos que solo se pueden utilizar en una única clase y otros que están disponibles en cualquiera.
  Las clases envoltorio y los tipos primitivos tienen correspondencia directa, por eso, desde el JDK 5 es posible la conversión automática entre ellos:
    - Autoboxing: Conversión de tipo primitivo a envoltorio.

    - Unboxing: Conversión de tipo envoltorio a primitivo.
- **Secuencias de escape**: Java permite utilizar secuencias de escape que pueden usarse tanto en variables del tipo primitivo char como en String. Las secuencias String hacen que se trate como si no tuviera significado alguno el carácter que esté contenido en la secuencia, o, por el contrario, permiten reemplazarlo por un contenido diferente.
- **Comentarios**: Texto introducido en el código fuente que no afecta a este, sino que esta destinado a ser leído por los desarrolladores a modo de anotaciones. Se puede introducir de tres formas: De una línea, //; de varias líneas, /* y */; y de documentación, /** y */.
- **Operadores**: Permiten realizar operaciones con las variables explicadas anteriormente. Podemos realizar operaciones sobre uno, dos o tres operandos.

Conceptos fundamentales

- **lowerCamelCase**: convención de escritura donde las palabras se juntan para evitar espacios y se distinguen entre ellas con mayúsculas en la primera letra de cada palabra, con la excepción de la primera letra de la primera palabra.

- **lowercase**: convención de escritura donde las palabras se juntan para evitar espacios y todo en minúsculas.

- **Notación**: sistemas de signos destinado a representar símbolos o conceptos especiales por pertenecer a campos específicos y por lo tanto no encontrarse en el lenguaje general.

- **SCREAMING_SNAKE_CASE**: convención de escritura donde las palabras se separan con guiones bajos "_" y completamente en mayúsculas.

- **UpperCamelCase**: convención de escritura donde las palabras se juntan para evitar espacios y se distinguen entre ellas con mayúsculas en la primera letra de cada palabra.