CEU Andalucía
Abel Morillo Sánchez

¿Qué es un algoritmo?

Un algoritmo es un conjunto de instrucciones para realizar una o varias tareas con un objetivo concreto: resolver un problema o necesidad.

Un ejemplo de algoritmo puede ser una receta de cocina o un manual de instrucciones de montaje de un mueble Ikea.

¿Qué es un programa?

Un programa es uno o varios algoritmos interrelacionados que trabajan con instrucciones informáticas para resolver un problema o cubrir una necesidad utilizando medios electrónicos.

Las instrucciones que en una receta de cocina son “Lavar la lechuga”, “Pelar los ajos”, etc., en un programa informático serían instrucciones enviadas a la máquina (al ordenador) para que realice cálculos en el procesado, guarde o recupere cosas de la memoria, o lea o escriba en los dispositivos de entrada/salida.

Dependiendo de la complejidad del programa puede estar compuesto por un algoritmo (un único conjunto de instrucciones) o por muchos que resuelven diferentes tareas.

¿Cuáles son los pasos que seguimos para escribir un programa?

0. Primero partimos de un problema que es lo que tenemos que resolver
1. A continuación, hacemos un análisis de ese problema. En el análisis, entre otras cosas, debemos:
    a. Entender bien el problema, sabiendo concretamente qué hay que resolver
    b. Revisar de qué recursos disponemos para resolverlo
    c. Intentar dividir el problema en otros más pequeños (divide y vencerás)
2. Tras el análisis, procedemos a hacer el diseño de la solución. Consiste en definir nuestro algoritmo. Lo podemos definir de dos modos:
    a. Mentalmente, pensando en todas las instrucciones que tenemos que hacer para solucionar el problema.
    b. Con pseudocódigo, es decir, escribiendo las instrucciones en lenguaje normal, como si fuera un manual.
    c. Con un diagrama de flujo
3. Una vez hecho el diseño, transcribiremos las instrucciones que hemos diseñado al lenguaje de programación elegido. Es decir, codificaremos, escribiremos el programa en código.
4. Terminado de escribir el código, el último paso será probar que efectivamente resuelve el problema. Es importante probar muy bien nuestro programa. Siempre hay errores ocultos. Si se encuentran errores, dependiendo de su gravedad, es posible que sea necesario rehacer el código, el diseño o incluso el análisis.

¿Qué lenguajes de programación podemos utilizar?

Hay muchos lenguajes de programación. Nosotros vamos a aprender en este curso a utilizar JAVA. Alguna de sus características:

- Es un lenguaje compilado (el código escrito necesita ser compilado a lenguaje máquina para poder ser ejecutado)
- Es uno de los lenguajes más utilizados en el mundo en diferentes ámbitos tecnológicos (programación web, aplicaciones de escritorio, etc)
- Lenguaje orientado a objetos

¿Cómo escribimos nuestro primer programa en Java?

Utilizaremos Eclipse IDE para programar (ver diapositivas que explican como descargarlo y utilizarlo).

Para crear un programa que podamos ejecutar, crearemos una clase con el nombre que queramos. Dentro de esa clase, creamos un método main. Ahí dentro podemos poner nuestras instrucciones.

```
public class MiPrograma {

    public static void main(String[] args) {
        // aquí irían mis instrucciones
    }

}
```

¿Qué es una variable?

Una variable me permite almacenar datos de diferente tipo. Para declarar una variable basta con escribir el tipo de dato y el nombre de la variable.

```
Tipo nombreVariable;
```

¿Qué tipos de datos podemos usar?

Muchos tipos de datos dentro del paradigma de la orientación a objetos. Pero esto lo veremos en próximos temas. Por ahora, los datos básicos que vamos a manejar son estos:

- Texto (cadena): String
- Números enteros cortos: Integer o int
- Números enteros largos: Long o long
- Números decimales: Float o float

- Números decimales largos: Double o double

Por ejemplo, para declarar una variable de tipo Texto con nombre “mensaje” lo haríamos así:

```
String mensaje;
```

¿Cómo le damos valor a una variable?

Basta con poner el nombre de la variable, luego el operador igual (=) y a continuación el valor que le queremos dar. Este valor puede ser un valor directo, u otra variable, o una operación entre variables/valores.

El valor de una variable puede cambiar cuando queramos.

Ejemplos:

```
public static void main(String[] args) {
    String mensaje;
    mensaje = "Hola";
    mensaje = "Hola " + "qué tal";

    Integer numero1;
    numero1 = 84;
    Integer numero2;
    numero2 = 65 + numero1;
}
```

También podemos darle valor a una variable a la vez que la declaramos, en la misma línea:

```
String mensaje = "Hola mundo";
```

¿Qué es una constante?

Una constante funciona igual que una variable, pero el primer valor que recibe cuando la declaramos ya no puede cambiar. Se utilizan para registrar valores fijos.

Se debe declarar fuera del método main arriba del todo. Se hace de este modo:

```
public static final Tipo NOMBRE_CONSTANTE = valor;
```

Se utilizan igual que las variables.

Ejemplos de dos constantes de tipo String y Integer:

```
public class MiPrograma {

    public static final Integer MESES_DEL_AÑO = 12;
    public static final String SALUDO = "hola";

    public static void main(String[] args) {
        System.out.println(SALUDO + " los meses del año son " + MESES_DEL_AÑO);
    }

}
```

¿Cómo podemos hacer conversiones de un tipo de dato a otro?

Desde / Hasta	String	Integer	Long	Float	Double
String		Integer.parseInt()	Long.parseLong()	Float.parseFloat()	Double.parseDouble()
Integer	toString()		longValue()	floatValue()	doubleValue()
Long	toString()	intValue()		floatValue()	doubleValue()
Float	toString()	intValue()	longValue()		doubleValue()
Double	toString()	intValue()	longValue()	floatValue()	

Cuidado, algunos pueden generar un error si no se puede hacer la conversión correctamente.

Ejemplos:

```
public static void main(String[] args) {
    String cadena = "49";
    Integer numeroEntero = Integer.parseInt(cadena);
    Double numeroDecimal = Double.parseDouble(cadena);
    Long numeroLargo = Long.parseLong(cadena);
    Float numeroFlotante = Float.parseFloat(cadena);

    cadena = numeroEntero.toString();
    cadena = numeroDecimal.toString();
    cadena = numeroLargo.toString();
    cadena = numeroFlotante.toString();

    numeroLargo = numeroEntero.longValue();
}
```

¿Cómo mostramos algo por consola?

Para imprimir mensajes por consola utilizamos esta instrucción:

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

Entre paréntesis tenemos que indicar qué queremos imprimir. Puede ser una cadena, o una variable de tipo String, o varias variables/valores que concatenados sean una cadena

Ejemplo:

```
public static void main(String[] args) {
    String mensaje = "Hola mundo";
    System.out.println(mensaje);
    System.out.println("Hola qué tal");
    System.out.println(mensaje + "!!");
}
```

¿Cómo obtenemos algo que el usuario escriba por teclado?

Lo hacemos con Scanner. Scanner es una clase especial de Java que debemos inicializar en una variable en primer lugar. Luego lo podemos utilizar todas las veces que queramos. Cada vez que usamos el Scanner para leer un dato por el teclado, la ejecución del programa se pausa hasta que el usuario escribe.

```
public static void main(String[] args) {
    // Inicializamos Scanner en una variable
    Scanner scanner = new Scanner(System.in);

    // Mostramos mensaje al usuario
    System.out.println("Indica un texto por favor");

    // Lo utilizamos
    String mensaje = scanner.nextLine();

    // Mostramos mensaje al usuario
    System.out.println("Indica un número por favor");

    // Volvemos a utilizar el scanner
    Integer numero = scanner.nextInt();

    // Mostramos lo que hemos leído
    System.out.println("He leido esto: " + mensaje + " y esto " + numero);
}
```

Cuando declaramos el Scanner es necesario importar la clase Scanner. Eclipse nos lo hace automáticamente añadiendo este código al principio de nuestra clase:

```
1 package miprograma;
2
3 import java.util.Scanner;
4
5 public class MiPrograma {
6
```

¿Cómo podemos incluir en una cadena un carácter especial?

En las cadenas (String) no podemos incluir un salto de línea o unas dobles comillas como sin nada. Se tratan de caracteres especiales. Para incluirlos se deben escribir de este modo:

Secuencia de escape	Valor
\b	Retroceso o *backspace* (equivalente a \u0008)
\t	Tabulador (equivalente a \u0009)
\n	Nueva línea (equivalente a \u000A)
\f	Salto de página (equivalente a \u000C)
\r	Retorno de carro (equivalente a \u000D)
\"	Doble comilla (equivalente a \u0022)
\'	Comilla simple (equivalente a \u0027)

¿Cómo podemos escribir comentarios en el código?

Los comentarios en el código no se ejecutan, se ignoran. Nos sirven para aclarar qué es lo que estamos programando, para que en el futuro sepamos qué significa cada instrucción. O para explicar por qué hemos tomado determinadas decisiones.

Hay dos formas de escribir comentarios:

- En una línea: basta con poner "//" antes del comentario
- En varias líneas: Hay que empezar con "/*" y terminar con "*/"

Ejemplos:

```
public static void main(String[] args) {
    // Comentario de una sólo línea

    /* Comentario
     de muchas
     líneas */
}
```

¿Cómo debemos escribir los nombres de variables, paquetes y clases?

Hay una serie de normal a la hora de escoger nombres a las cosas que vamos creando cuando programamos. Algunas normas son obligatorias. Otras son por convención. Ambas deben seguirse:

- Nombres de paquetes: todo en minúscula y sin espacios. No pueden empezar por números. Ejemplo: nombredepaquete
- Nombres de clases: La primera letra SIEMPRE en mayúscula. Los espacios se indican con camelCase. No pueden empezar por números. Ejemplo: NombreDeClase
- Nombres de variables: La primera letra SIEMPRE en minúscula. Los espacios se indican con camelCase. Ejemplo: nombreDeVariable
- Nombres de constantes: Todas las letras en mayúscula. Los espacios se indican con guiones bajos. Ejemplo: NOMBRE_DE_CONSTANTE