CONTROL DE FLUJOS

CADA APARTADO PRINCIPAL CORRESPONDE CON EL NOMBRE DEL PROYECTO CON EJEMPLOS DE LO APUNTADO AQUÍ

1. INTRODUCCIÓN A LA PROGRAMACIÓN CONDICIONAL

- Gracias a la programación condicional podremos hacer diferentes cosas basadas en la condición establecida previamente en nuestro código.
- Hay ciertas palabras clave que nos permiten hacer programación condicional:
  - `if`
  - `else`
  - `switch`
  - Operadores ternarios.

2. IF

- La sentencia `if` permite hacer cosas basadas en una condición que sea cierta o falsa (`true` o `false`).
- Si la condición establecida no es exitosa, podemos usar la sentencia `else` para "atrapar" el caso en el que la condición no sea exitosa.
- Puedes usar expresiones como condiciones de un `if`, mientras que la expresión se evalúe como un `bool`, ya que las condiciones siempre tienen que ser `bool`, tienen que ser `true` o `false`.
- Las condiciones se pueden anidar, es decir puedo meter dentro de un `if`, otro `if` para que se evalúe más de una condición.
  Otra opción para evitar usar condiciones anidadas, y comprobar más de una condición, es mediante el uso de operadores lógicos dentro de una sola condición.

3. ELSE IF

- Se usa junto al `if` normal y sirve para hacer cosas basadas en varias condiciones.
- En un `else if` solo un bloque de condición de toda la sentencia es el que se ejecutará, a diferencia de usar muchos `if`, que se verificarán todos y harán que el código se vea más feo y difícil de leer. En este caso en cuando se cumpla una condición, el programa saltará a ese bloque directamente.

4. SWITCH

- Se usa como alternativa al `else if`, sirve para comprobar varias condiciones. Es una forma más compacta que el `else if`.
- En los `switch` además de todas las condiciones que queramos añadir, hay una condición *default* para el caso en el que no se cumpla ninguna de las indicadas.
- Una vez se cumple la condición y se realiza lo que deseemos, se ejecuta la sentencia *break*, que hace que el programa "salga" de la condición y pueda continuar con la ejecución del resto del código. Si no se añade el *break*, una vez terminado lo que hayamos querido hacer el programa continuará ejecutándose hacia la siguiente condición del `switch` que hayamos especificado, por eso es importante la inclusión de esta sentencia.
- La condición del `switch`, solo puede ser tipos de dato `integer` o `enum` (más adelante en el curso los veremos).
- Las condiciones en los `switch`, tienen que ser variables constantes, es decir variables que no se pueden modificar, para ello se les aplica el modificador `const` delante del tipo de dato en la definición de estas y se definen fuera del `main`. Ejemplo:
  `const double` constanteDeGravitacionUniversal = 9.8

5. OPERADORES TERNARIOS

- Usar un operador ternario es una forma alternativa de hacer comprobaciones con if.

- La sintaxis de una expresión ternaria es la siguiente:
  resultado = (condición) ? `opcion1` : `opción2`;
  En un if normal, la expresión sería así:

```
if (condición)
{
      resultado = opcion1;
}
else
{
      resultado = opcion2;
}
```

- Los tipos de datos de `opcion1` y `opcion2` deben de ser iguales (o que se puedan convertir al mismo tipo de dato) para que no se arroje un error de compilador.
- Se puede hacer una inicialización ternaria, por ejemplo:
  `bool` esRapido = `false`;
  `int` velocidad = esRapido ? 300 : 150; `//int` velocidad = 150;

6. RESUMEN

- Mediante ciertas estructuras de control de flujos podemos realizar programación condicional en la que mediante una o varias condiciones podemos hacer ciertas cosas en nuestro código basándonos constantemente en si las condiciones son ciertas o falsas (`true` o `false`).
- Estas estructuras de control de flujos son:
  - `if`
  - `else if`
  - `switch`
  - Operadores ternarios