Sesión 2. Primeros programas - Bibliotecas

Competencias previas

- Explicar las partes básicas e instrucciones de un programa en C++.
- Resolver problemas mediante programas en C++ usando variables de tipos simples y las instrucciones de escritura, lectura y asignación.
- Crear, importar, exportar, compilar y ejecutar un proyecto en Eclipse.

Objetivos

- Practicar la importación, exportación, creación, compilación y ejecución de proyectos en Eclipse.
- Utilizar las bibliotecas estándar de C++.
- Utilizar instrucciones condicionales (if).

Actividad 1: Media ponderada de la asignatura

- Importa el proyecto **notafinal.zip** que encontrarás en el aula virtual.
- Este programa calcula la media ponderada a partir de las notas de los tres bloques de la asignatura.
- Amplía el programa de forma que controle la posibilidad de que el usuario introduzca datos incorrectos, es decir, números que no estén en el rango 0 - 10. En ese caso, el programa mostrará un mensaje indicando que los datos no son correctos.
- Analiza y debate tu solución con tus compañeros.

Bibliotecas estándar de C++

Cuando se escriben programas dentro de un ámbito concreto, al final hay muchos subproblemas concretos que se repiten. No tiene sentido tener que resolver esos subproblemas desde el principio.

Pensemos, por ejemplo, en programas que resuelvan problemas matemáticos. En muchas ocasiones habrá que elevar un número a otro, calcular la raíz cuadrada, el coseno o sumar dos polinomios. Nos interesaría tener ya definidas esas operaciones en algún sitio, de forma que se pudieran usar cuando las necesitáramos, teniendo la seguridad además de que funcionan bien ya que han sido probadas con anterioridad.

Se denomina **biblioteca** (o ***librería,*** mala traducción de ***library*** pero de uso muy extendido) a una recopilación de funciones que resuelven problemas concretos. Una biblioteca está almacenada en ficheros separados al fichero donde se encuentra el programa principal.

C++ cuenta con **bibliotecas predefinidas** que se pueden usar en nuestros programas. Usar funciones de las bibliotecas estándar ayuda a mejorar el rendimiento y la portabilidad del software.

Hay muchas bibliotecas predefinidas en C y en C++ sobre diferentes temas: matemáticas, entrada/salida, cadenas de caracteres, tiempo, etc.

Se puede consultar una descripción completa de las bibliotecas estándar y las funciones que incluyen en:
http://www.cplusplus.com/reference/clibrary/

En C y C++ existen bibliotecas equivalentes para mantener la compatibilidad entre los dos lenguajes. Así, la librería matemática se denomina <math.h> y <cmath> en C++. Ambas se pueden usar en C++.

Para poder usar las funciones de una biblioteca predefinida se debe poner la directiva **#include** seguida del nombre de la biblioteca. Por ejemplo:

```
#include <iostream>
#include <cmath>
```

A partir de ese momento, y en todo el fichero, se pueden usar las funciones definidas en la biblioteca.

Para poder usar una función concreta tenemos que conocer la biblioteca en la que se encuentra y su

cabecera.

La **cabecera de una función** indica el nombre exacto de la función, el número y tipo de la información que hay que pasarle y el tipo del resultado.

Así, por ejemplo, si queremos calcular $\mathbf{a^b}$ podemos usar la siguiente función de la librería <cmath>:

```
double pow (double base, double exponente);
```

`double` es un tipo real, como **`float`**, pero con un rango mayor de valores. Se puede usar también con variables de tipo **`int`** y **`float`**.

De esa cabecera sacamos la siguiente información:

- La función se denomina pow
- Hay que pasarle dos valores reales ***(parámetros)***: el primero será la base y el segundo, el exponente. (Lo importante es el tipo, no el nombre de las variables, que sirven para aclarar el significado.)
- La función devolverá un valor de tipo double con el resultado.

Si, por ejemplo, queremos guardar el resultado de elevar 2 a 4 en una variable **x**, pondríamos la siguiente instrucción en un programa:

```
x = pow ( 2, 4 );
```

Tras ejecutarse, en x tendríamos el valor 16.

La llamada a una función puede ponerse en cualquier lugar en el que se pueda poner un valor del tipo devuelto por la función.

Algunos ejemplos de funciones en bibliotecas estándar:

Librería **cmath**: Funciones matemáticas

- `double` **`ceil`**`(double x)` Redondea x al entero más cercano hacia arriba.
- `double` **`cos`**`(double x)` Devuelve el coseno de x, donde x está dado en radianes.
- `double` **`exp`**`(double x)` Devuelve el valor de e (la base de los logaritmos naturales) elevado a la potencia x.
- `double` **`fabs`**`(double x)` Devuelve el valor absoluto del número x.
- `double` **`floor`**`(double x)` Redondea x hacia abajo al entero más cercano hacia abajo.
- `double` **`log`**`(double x)` Devuelve el logaritmo neperiano de x.
- `double` **`log10`**`(double x)` Devuelve el logaritmo decimal de x.
- `double` **`pow`**`(double x, double y)` Devuelve el valor de x elevado a y.
- `double` **`sin`**`(double x)` Devuelve el seno de x.
- `double` **`sqrt`**`(double x)` Devuelve la raíz cuadrada no negativa de x.
- `double` **`tan`**`(double x)` Devuelve la tangente de x.

Librería **cstdlib**: Funciones generales

- `int rand (void)` Devuelve un número pseudoaleatorio entre 0 y la constante RAND_MAX (definida también en esta librería).
- `void srand (unsigned int seed)` Inicia la semilla de una secuencia de números pseudoaleatorios.
- `int abs (int n)` Valor absoluto de n

Actividad 2: Ejemplos de funciones predefinidas

- Descarga del aula virtual e importa en Eclipse el proyecto **funciones.tar.gz** que contiene varios ejemplos de uso de funciones predefinidas.
- Analiza los ejemplos presentados.

Actividad 3: Lista de errores frecuentes y convenciones de programación

- Descarga del aula virtual el fichero **ListaErroresFrecuentes.odt**.
- Cada vez que cometamos un error al escribir un programa (durante el proceso de compilación o de ejecución), debemos escribir en este documento el mensaje de error obtenido o el problema observado y una breve explicación de cómo se solucionó.
- Cuando nos encontremos con errores debemos echar un vistazo a esta lista para comprobar si ya ha aparecido antes y si la solución de entonces sirve para este nuevo caso.
- En el documento IP_Convenciones_de_programación.pdf que está en el aula virtual puedes ver cómo vamos a escribir el código de los ejemplos de la asignatura. Lee el documento e intenta aplicarlo en tus programas.

Actividad 4: Raíces de una ecuación de segundo grado

Escribir un programa que, dados los coeficientes de una ecuación de segundo grado, devuelva el valor de las dos raíces reales.
Recordemos que las raíces de una ecuación $\mathbf{ax^2 + bx + c = 0}$ se calculan: $x = \frac{-b \pm \sqrt{b^2 - 4ac}}{2a}$

En la ejecución del programa deberíamos ver algo similar a esto:

```
Para la ecuación de segundo grado ax² + bx + c = 0
Introduce el coeficiente a:
4.5
Introduce el coeficiente b:
3
Introduce el coeficiente c:
-2
Raíz: 0.412023
Raíz: -1.07869
```

Realizar pruebas suficientes para asegurarnos de que funciona bien en todos los casos. (Hay que tener en cuenta que no todas las ecuaciones tienen dos raíces reales.)

En la siguiente página web se pueden calcular los resultados de las pruebas:

http://www.wolframalpha.com

En la página web anterior, directamente se puede escribir en el cuadro de texto el valor de la ecuación. Para el ejemplo anterior se podría escribir lo siguiente y se obtendría una representación gráfica, las raíces y más información:

4.5 x^2 + 3 x – 2 = 0

Ejercicios adicionales

- Calcular la hipotenusa de un triángulo rectángulo a partir de la longitud de los dos catetos.
- **Bufete de abogados**
  Un bufete de abogados cobra a sus clientes por la cantidad de tiempo que les dedican sus abogados. No cobra tiempos inferiores a un cuarto de hora. Así, si un abogado emplea 1 hora y 14 minutos con el cliente, el bufete solo cobra por 1 hora. Cada cuarto de hora se cobra a 60€. Escribir un programa que, dadas las horas y minutos dedicadas a un cliente, calcule el total de la factura.