

Clase	Métodos	Descripción
Device		Permanece igual que se definió.
Input	Input::Input(const string & name)	Inicializa un dispositivo de entrada con un nombre.
	Input::connectTo(Processor & P)	Conecta un dispositivo de entrada con un procesador.
	Processor& Input::operator>>(Processor & P)	Conecta un dispositivo de entrada con un procesador.
	void Input::process()	Función virtual pura.
Keyboard	Keyboard::Keyboard(const string & name)	Inicializa un teclado con el nombre especificado.
	void Keyboard::process()	Implementación por defecto de un teclado, lee una cadena de caracteres desde el teclado real utilizando el operador de flujo <code>cin</code> . Después manda esta cadena al procesador llamando al método <code>process()</code> del procesador al cual está conectado.
CharKeyboard	CharKeyboard::CharKeyboard(const string & name)	Inicializa un teclado de carácter con el nombre especificado.
	void CharKeyboard::process()	Igual que <code>Keyboard::process()</code> pero solo lee un carácter (para ello se puede utilizar la función estándar <code>cin.get()</code> ).
LineKeyboard	LineKeyboard::LineKeyboard(const string & name)	Inicializa un teclado de línea con el nombre especificado.
	void LineKeyboard::process()	Igual que <code>Keyboard::process()</code> pero lee una línea completa, es decir, lee caracteres hasta encontrar un <code>'\n'</code> y manda toda la línea directamente al procesador.
Processor	Processor::Processor(const string & name)	Inicializa un procesador con el nombre especificado.
	void Processor::connectTo(Output & O)	Conecta el procesador con una salida.
	void Processor::process(const string & data)	Función miembro virtual que se va a encargar de buscar algunos comandos especiales en los datos que le llegan como puede ser el comando <code>"quit"</code> .
	void Processor::process(const string & data, int n)	Lo mismo que la función anterior pero considerando sólo los <i>n</i> primeros caracteres.
	Output& Processor::operator>>(Output & O)	Conecta un procesador con un dispositivo de salida.

Uppercase	Uppercase::Uppercase(const string & name)	Inicializa un procesador de mayúsculas con el nombre especificado.
	void Uppercase::process(const string & data)	Convierte en mayúsculas toda la cadena de caracteres <i>data</i> y añade un prefijo "PROCESSED: ". Manda el resultado al <i>Display</i> al cual está conectado. Por ejemplo, si <i>data</i> vale "hola", la cadena enviada al <i>Display</i> debe ser "PROCESSED: HOLA".
	void Uppercase::process(const string & data, int n)	Lo mismo que la función anterior pero considerando sólo los <i>n</i> primeros caracteres.
Reverse	Reverse::Reverse(const string & name)	Inicializa un procesador para invertir el orden de caracteres con el nombre especificado.
	void Reverse::process(const string & data)	Invierte la cadena de caracteres <i>data</i> y añade un prefijo "PROCESSED: ". Manda el resultado al <i>Display</i> al cual está conectado. Por ejemplo, si <i>data</i> vale "hola", la cadena enviada al <i>Display</i> debe ser "PROCESSED: aloh".
	void Reverse::process(const string & data, int n)	Lo mismo que la función anterior pero considerando sólo los <i>n</i> primeros caracteres.
Output	Output::Output(const string & name)	Inicializa un dispositivo de salida con el nombre especificado.
	void Output::process()	Función virtual pura.
Display	Display::Display(const string & name)	Inicializa una pantalla con el nombre especificado.
	void Display::process(const string & data)	Muestra la cadena en la pantalla utilizando cout.
Printer	Printer::Printer(const string & name)	Inicializa una impresora con el nombre especificado.
	void Printer::process(const string & data)	Simula la impresión de la cadena recibida mostrando por pantalla la cadena con el prefijo "Imprimiendo...". Por ejemplo, si le llega la cadena "PROCESSED: aloh", se imprimirá por pantalla utilizando cout la cadena "Imprimiendo... PROCESSED: aloh".

