Tarea 6: Diseño de clases

Mario Medina
mariomedina@udec.cl

Programación orientada al objeto — 12 de noviembre de 2018

Introducción

En esta sexta tarea, Uds. deben experimentar con el desarrollo e implementación de clases en un problema de criptografía simulando la máquina Enigma, un dispositivo para codificación de texto usado por Alemania en la 2da. Guerra Mundial.

Info: De más está decirle que esta tarea es individual: puede comentar posibles métodos de solución con sus compañeros, pero se espera que los códigos entregados por todos los alumnos sean diferentes.

Envíeme su código fuente junto con un informe de a lo más 3 planas, detallando su método de solución y las dificultades encontradas a más tardar el día miércoles 28 de noviembre, antes de medianoche a mariomedina@udec.cl.

La máquina Enigma

En esta tarea, se le pide simular una versión simplificada de la máquina Enigma G, un dispositivo para la codificación de texto usado por Alemania en la 2da. Guerra Mundial. La decriptación exitosa de esta máquina por parte de un grupo de criptógrafos en Inglaterra es un hito muy importante en la historia de la criptografía.

Esta máquina consta de varios rotores de cifrado que implementan un algoritmo de encriptación polialfabético. En las siguiente secciones se explica en más detalle su funcionamiento.

Definiendo un rotor de cifrado por substitución de letras

Como primer paso, se le pide implementar una clase llamada *Rotor* para implementar un cifrado por substitución de letras. Este cifrado usa una clave de 26 letras para codificar cada letra de entrada como otra letra de salida, como se muestra en la tabla 1:

Entrada	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z
f Salida	Q	W	E	R	T	Y	U	I	O	P	A	S	D	F	G	H	J	K	L	Z	X	C	V	B	N	M

Tabla 1: Ejemplo de cifrado por substitución de letras

El constructor de esta clase recibe como argumento un objeto `String` de 26 letras que define el cifrado a implementar. El constructor debe verificar que el argumento tenga exactamente 26 letras, y que no aparezcan letras repetidas. Si la clave no cumple con estas condiciones, entonces el constructor debe generar una excepción.

Esta clase debe incluir un método `String encripta(const String& s)` que utiliza la clave del rotor para traducir los caracteres de entradas. Por ejemplo, si la entrada es `HOLA MUNDO!`, la salida de la función debe ser `IGSQ DXFRG!`. Asimismo, su clase debe incluir un método `String decripta(const String& s)` que hace la operación contraria. Ambas funciones no deben modificar los caracteres que no son letras.

Para introducir mayor complejidad en la codificación, el rotor puede ser girado para cambiar su posición inicial. Esto equivale a rotar la clave de manera circular. Por ejemplo, si la posición inicial del rotor es la letra P, entonces la codificación a usar es la que se muestra en la tabla 2:

Entrada	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z
Salida	P	A	S	D	F	G	H	J	K	L	Z	X	C	V	B	N	M	Q	W	E	R	T	Y	U	I	O

Tabla 2: Rotación de la clave de cifrado

Agregue entonces un método `rotaRotor(const char& c)` a su clase que realice esta rotación circular. Finalmente, por razones que se describirán más adelante, agregue además un método `avanzaRotor(void)` que desplaza la clave una posición hacia la izquierda de manera circular.

Modelo de la máquina Enigma

El siguiente paso en esta tarea es modelar la máquina Enigma de tres rotores. Para ello, se da una breve explicación de su operación.

Esta máquina usa un módulo reflector y tres rotores denominados el rotor lento (*slow rotor*), el rotor mediano (*medium rotor*), y el rotor rápido (*fast rotor*), como se muestra en la figura 1.

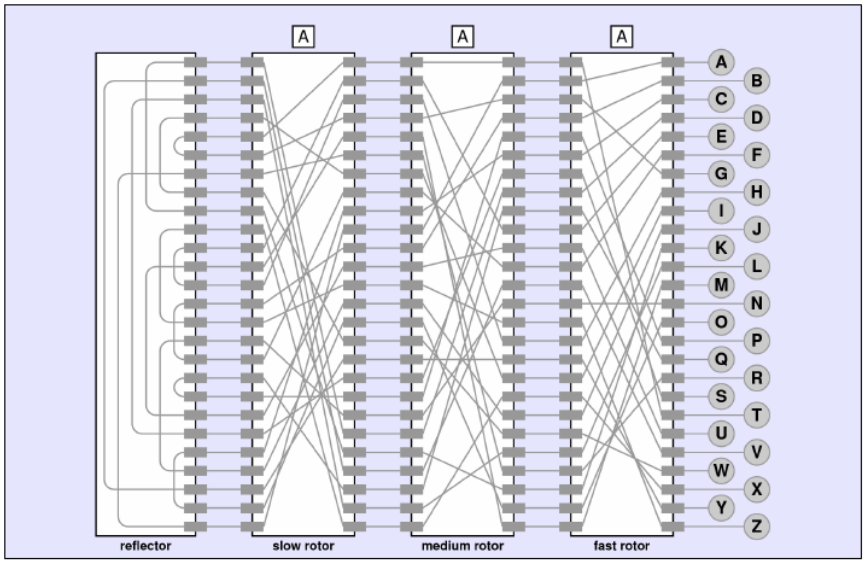

Figura 1: Modelo de la máquina Enigma

Los usuarios de esta máquina Enigma tenían un banco de 5 rotores distintos de donde escoger los 3 rotores a usar. Este banco de rotores se muestra en la tabla 3.

Rotor 1	E	K	M	F	L	G	D	Q	V	Z	N	T	O	W	Y	H	X	U	S	P	A	I	B	R	C	J
Rotor 2	B	D	F	H	J	L	C	P	R	T	X	V	Z	N	Y	E	I	W	G	A	K	M	U	S	Q	O
Rotor 3	A	J	D	K	S	I	R	U	X	B	L	H	W	T	M	C	Q	G	Z	N	P	Y	F	V	O	E
Rotor 4	Q	A	Z	W	S	X	E	D	C	R	F	V	T	G	B	Y	H	N	U	J	M	I	K	O	L	P
Rotor 5	H	G	T	Y	U	J	M	N	B	V	F	R	I	K	C	D	E	O	L	X	S	W	P	Z	A	Q

Tabla 3: Banco de rotores

El primer paso es, entonces, escoger qué rotores usar, mientras que el segundo paso es escoger la posición inicial de cada uno de los rotores. Esto se especificará como la letra inicial de cada rotor. Por ejemplo, escoger los rotores 1, 3 y 2 y la combinación inicial `EAB` define la codificación de la tabla 4

Rotor lento	No. 1	E	K	M	F	L	G	D	Q	V	Z	N	T	O	W	Y	H	X	U	S	P	A	I	B	R	C	J
Rotor mediano	No. 3	A	J	D	K	S	I	R	U	X	B	L	H	W	T	M	C	Q	G	Z	N	P	Y	F	V	O	E
Rotor rápido	No. 2	B	D	F	H	J	L	C	P	R	T	X	V	Z	N	Y	E	I	W	G	A	K	M	U	S	Q	O

Tabla 4: Ejemplo de codificación de la máquina Enigma

Se ilustrará la operación de la máquina Enigma encriptando la letra `A` usando la codificación mostrada.

Primero, se usa el rotor rápido para codificar la letra la letra `A` en la salida `B`. Luego, esta salida pasa por el rotor mediano para obtener la salida `J`. Finalmente, esta salida pasa por el rotor lento para obtener la salida `Z`.

Rotor rápido	Entrada	**A**	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z
	Salida	**B**	D	F	H	J	L	C	P	R	T	X	V	Z	N	Y	E	I	W	G	A	K	M	U	S	Q	O
Rotor mediano	Entrada	A	**B**	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z
	Salida	A	**J**	D	K	S	I	R	U	X	B	L	H	W	T	M	C	Q	G	Z	N	P	Y	F	V	O	E
Rotor lento	Entrada	A	B	C	D	E	F	G	H	I	**J**	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z
	Salida	E	K	M	F	L	G	D	Q	V	**Z**	N	T	O	W	Y	H	X	U	S	P	A	I	B	R	C	J

Tabla 5: Codificación directa usando los tres rotores

La salida del rotor lento ingresa en el reflector, que corresponde a un disco de codificación fijo. En este caso, supondremos que el reflector usa el `String IXUHFEZDAOMTKQJWNSRLCYPBVG` como clave de codificación. Entonces, el reflector recibe la letra `Z` que emerge del rotor lento y la convierte en la letra `G`.

La señal luego pasa por los tres rotores nuevamente, pero en sentido inverso. Primero, se usa el rotor lento para codificar la letra `G` en la salida `F`. Luego, esta salida pasa por el rotor mediano para obtener la salida `W`. Finalmente, esta salida pasa por el rotor rápido para para obtener la salida `R`.

Entrada	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	**Z**
Salida	I	X	U	H	F	E	Z	D	A	O	M	T	K	Q	J	W	N	S	R	L	C	Y	P	B	V	**G**

Tabla 6: Operación del reflector

Rotor lento	Entrada	E	K	M	F	L	**G**	D	Q	V	Z	N	T	O	W	Y	H	X	U	S	P	A	I	B	R	C	J
	Salida	A	B	C	D	E	**F**	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z
Rotor mediano	Entrada	A	J	D	K	S	I	R	U	X	B	L	H	W	T	M	C	Q	G	Z	N	P	Y	**F**	V	O	E
	Salida	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	**W**	X	Y	Z
Rotor rápido	Entrada	B	D	F	H	J	L	C	P	R	T	X	V	Z	N	Y	E	I	**W**	G	A	K	M	U	S	Q	O
	Salida	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	**R**	S	T	U	V	W	X	Y	Z

Tabla 7: Codificación inversa usando los tres rotores

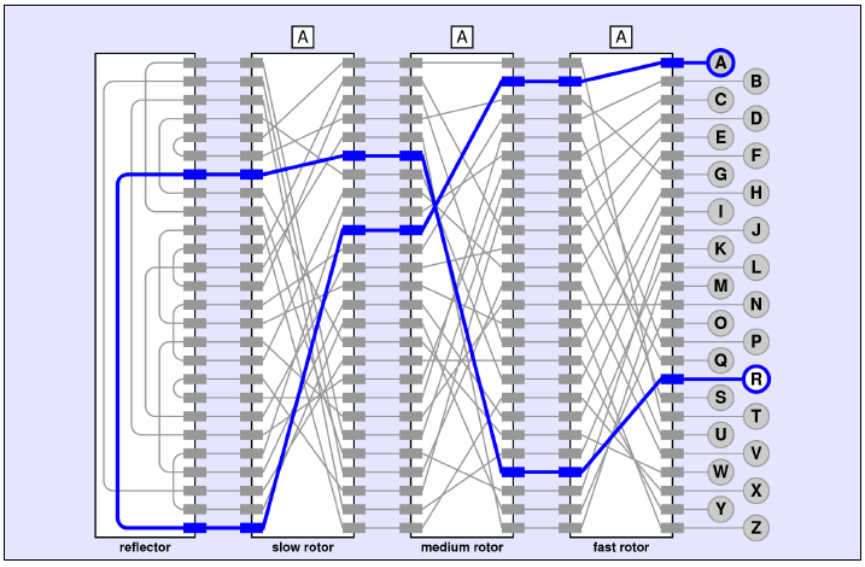

Figura 2: Codificación usando la máquina Enigma

El proceso completo de codificación se muestra en la figura 2.

Como ejercicio, repita el proceso anterior, pero usando la letra `R` como entrada. Ud. debiese ver que este proceso de codificación es reversible!

Avance de los rotores

Para aumentar aún más la complejidad de la encriptación, los diseñadores de la máquina Enigma agregaron la capacidad de hacer permutaciones de los rotores. En efecto, la acción de presionar una letra causa que el rotor rápido rote una posición a la izquierda en forma circular antes de convertir la letra. Además, el rotor mediano rota una posición a la izquierda en forma circular cada 26 rotaciones del rotor rápido. Finalmente, el rotor lento rota una posición a la izquierda en forma circular cada 26 rotaciones del rotor mediano. En otras palabras, el rotor mediano rota cada 26 letras, y el rotor lento rota cada 676 letras. De esta manera, si la primera letra `A` se codifica como una letra `R`, la segunda vez que se ingrese la letra `A`, las posiciones de los rotores habrán cambiado y la salida correspondiente será distinta.

Simulación de la máquina Enigma

Escriba, entonces, una clase en C++ llamada `Enigma` que simule la máquina descrita. Esta clase debe contener al menos seis objetos de la clase `Rotor` que representen a los 5 rotores y al reflector. El constructor de esta clase debe recibir como argumentos los tres rotores a usar, así como sus posiciones iniciales.

Esta clase debe implementar el método `convierte(const String& s)` que procesa el `String s` de acuerdo a lo descrito. Nótese que, dada la simetría del proceso de codificación, puede usarse el mismo método anterior para encriptar y desencriptar mensajes. Esta debilidad de la máquina fue explotada por el proyecto *Ultra* británico para analizar el proceso de encriptación.

Tarea 6

Como última tarea del curso, escriba un programa en C++ que genere una instancia de la clase Enigma y la use para codificar un archivo de texto, generando otro archivo con el texto encriptado. Su programa debe presentar al usuario los 5 rotores programados, y solicitarle cuáles usar como rotores rápido, mediano y lento. Además, debe preguntar al usuario por las posiciones iniciales de los 3 rotores. Pruebe su código con el archivo de texto `MobyDick_Ch01.txt`, que contiene el primer capítulo del libro *Moby Dick*, de Herman Melville.