

Universidad Mariano Gálvez de Guatemala

Campus Villa Nueva, Guatemala

Ingeniería en Sistemas

Ing. Carlos Arias

Curso: Programación I

Laboratorio 6, Programa 4

Sección: “A”

Carné: 5090-23-1407

Nombre: Angelyn Judith Díaz Zeceña

Introducción

Como podemos observar en esta parte del programa, se realizó una clase llamada “CuentaBancaria” que encapsula la información esencial de una cuenta bancaria, como el nombre del titular, el número de cuenta y el saldo. Y así mismo, el programa principal utiliza esta clase para simular las operaciones típicas de una cuenta bancaria, como depósitos y retiros. En donde, mas adelante, podrá observar mas a detalle.

Contenido

```
Explorador de Soluciones  Lab6Pro4.cpp  (Ámbito global)
1  #include <iostream>
2  #include <string>
3
4  using namespace std;
5
6  // Aquí se define la clase CuentaBancaria
7  class CuentaBancaria {
8  private:
9      // Atributos privados de la cuenta: nombre del titular, número de cuenta y saldo
10     string nombreTitular;
11     string numeroCuenta;
12     double saldo;
13
14 public:
15     // Aquí se presenta un Constructor que inicializa la cuenta con información básica
16     CuentaBancaria(const string& nombre, const string& numero, double saldoInicial)
17         : nombreTitular(nombre), numeroCuenta(numero), saldo(saldoInicial) {}
18
19     // Método para realizar un depósito en la cuenta
20     void depositar(double cantidad) {
21         // Aquí Verifica que la cantidad a depositar sea mayor que cero
22         if (cantidad > 0) {
23             saldo += cantidad;
24             cout << "Depósito exitoso. Nuevo saldo: Q" << saldo << endl;
25         }
26         else {
27             cerr << "Error: La cantidad debe ser mayor que cero." << endl;
28         }
29     }
30
31     // Método para realizar un retiro de la cuenta
32     void retirar(double cantidad) {
33         // Verificar que la cantidad a retirar sea mayor que cero y no supere el saldo actual
34         if (cantidad > 0 && cantidad <= saldo) {
35             saldo -= cantidad; // Actualizar el saldo restando la cantidad retirada
36             cout << "Retiro exitoso. Nuevo saldo: Q" << saldo << endl;
37         }
38         else {
39             cerr << "Error: Fondos insuficientes o cantidad no válida." << endl;
40         }
41     }
42 }
```

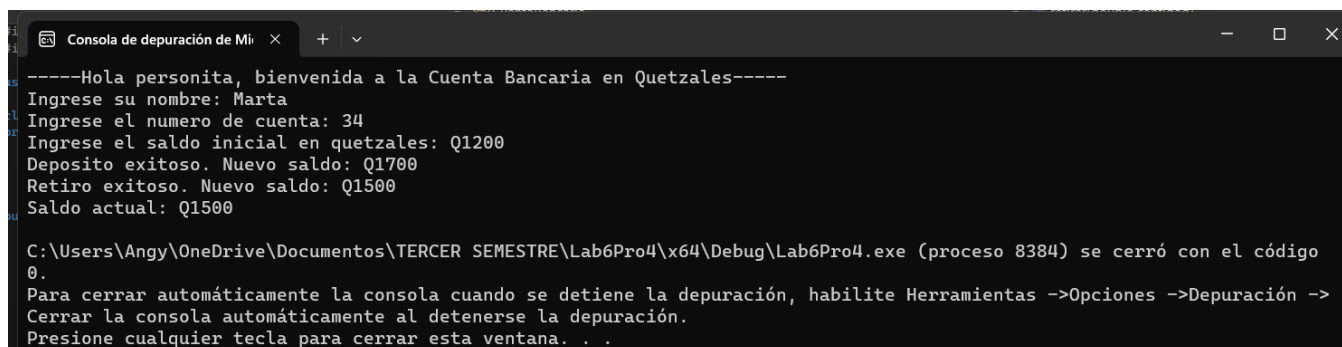
Como podemos observar en esta parte del código, se realiza una definición de la clase con una “CuentaBancaria” y atributos privados, juntamente con la utilización de un constructor, en donde se inicializa los atributos de la clase con los valores. Los métodos que se utilizan en esta clase de código: son, retirar y depositar.

```

46 // Método para consultar el saldo actual de la cuenta
47 double consultarSaldo() const {
48     return saldo;
49 }
50 };
51
52 int main() {
53
54     cout << "-----Hola personita, bienvenida a la Cuenta Bancaria en Quetzales-----" << endl;
55
56     string nombre, numeroCuenta;
57     double saldoInicial;
58
59     // Solicitar al usuario información para crear la cuenta
60     cout << "Ingrese su nombre: ";
61     getline(cin, nombre);
62
63     cout << "Ingrese el número de cuenta: ";
64     getline(cin, numeroCuenta);
65
66     cout << "Ingrese el saldo inicial en quetzales: Q";
67     cin >> saldoInicial;
68
69     // Crear una instancia de la clase CuentaBancaria con los datos proporcionados por el usuario
70     CuentaBancaria miCuenta(nombre, numeroCuenta, saldoInicial);
71
72     // Realizar operaciones y mostrar resultados
73     miCuenta.depositar(500.0);
74     miCuenta.retirar(200.0);
75
76     // Consultar y mostrar el saldo actual
77     cout << "Saldo actual: Q" << miCuenta.consultarSaldo() << endl;
78
79     return 0;
80 }
81
82

```

En esta parte del código, podemos observar que se da un “Mensaje de bienvenida” en donde se Imprime un mensaje de bienvenida al usuario para informarle que está utilizando una cuenta bancaria en quetzales. Solicitando información al usuario, gracias a la utilización de un `getline(cin, nombre);` que Permite que el usuario ingrese su nombre y lo almacena en la variable `nombre` y un `getline(cin, numeroCuenta);` para el número de cuenta.



```

Consola de depuración de Mi
-----Hola personita, bienvenida a la Cuenta Bancaria en Quetzales-----
Ingrese su nombre: Marta
Ingrese el numero de cuenta: 34
Ingrese el saldo inicial en quetzales: Q1200
Deposito exitoso. Nuevo saldo: Q1700
Retiro exitoso. Nuevo saldo: Q1500
Saldo actual: Q1500

C:\Users\Angy\OneDrive\Documentos\TERCER SEMESTRE\Lab6Pro4\x64\Debug\Lab6Pro4.exe (proceso 8384) se cerró con el código 0.
Para cerrar automáticamente la consola cuando se detiene la depuración, habilite Herramientas ->Opciones ->Depuración ->
Cerrar la consola automáticamente al detenerse la depuración.
Presione cualquier tecla para cerrar esta ventana. . .

```

En esta parte de la captura, se saluda primero al usuario, y luego ya le solicita al usuario que ingrese todos los datos, por ejemplo: el nombre de la persona y su cuenta, entre otras. Después muestra el resultado del calculo de la cuenta ingresada.

Conclusión

En conclusión, podemos detallar que, este fue un ejercicio que destacó la utilidad de las clases en C++ para modelar entidades del mundo real de manera efectiva. Y que, a su misma vez, se pudo observar el procedimiento sobre el calculo de una cuenta de banco.

Referencias

<https://github.com/Ashe122/LABORATORIO-6-PROGRAMA-4-.git>

TylerMSFT. (n.d.-a). *funciones*. Microsoft.com. Retrieved March 11, 2024, from

[https://learn.microsoft.com/es-es/cpp/standard-library/string-
functions?view=msvc-170](https://learn.microsoft.com/es-es/cpp/standard-library/string-functions?view=msvc-170)