

```

1  /* bankaccount.cpp
2  *
3  * Ejemplo de una clase de cuenta bancaria simple.
4  *
5  * para compilar escriba en una consola en el mismo directorio donde
6  * está este archivo:
7  *
8  * g++ -Wall -std=c++11 -o bankaccount.out bankaccount.cpp
9  */
10
11 #include <iostream>
12 // para usar cin y cout
13
14
15 /*-----
16 *                      class BankAccount
17 *-----*/
18 class BankAccount
19 {
20 public:
21     static constexpr double PENALIZACION {10.0};
22
23     BankAccount ();
24     BankAccount (double nuevoSaldo, double nuevoInteres);
25
26     void Depositar (double cantidad);
27     void Imprimir ();
28     bool Debitar (double cantidad);
29     void AgregarInteres ();
30
31     void SetSaldo (double nuevoSaldo);
32     double GetSaldo ();
33
34     void SetInteres (double nuevoInteres);
35     double GetInteres ();
36
37     double GetPenalizacion ();
38
39 private:
40     double saldo {0.0};
41     double interes {0.0};
42 };
43
44 BankAccount::BankAccount () : saldo (0.0), interes (0.0)
45 {
46     // nada
47 }
48
49 BankAccount::BankAccount (double nuevoSaldo, double nuevoInteres) :
50     saldo (nuevoSaldo), interes (nuevoInteres)
51 {
52     // hacer saldo >= 0.0
53
54     // hacer 0.0 <= interes <= 0.1
55 }
56
57 void BankAccount::Depositar (double cantidad)
58 {
59     // sumar al saldo actual la nueva cantidad, si y únicamente si la nueva
60     // cantidad es positiva
61 }
62

```

```

67
68 bool BankAccount::Debitar (double cantidad)
69 {
70     // SI la cantidad a debitar es menor que el saldo actual, ENTONCES
71     //     debita la cantidad del saldo actual
72     //     devuelve true para avisar que la operación se realizó con éxito
73     // SINO
74     //     quita la cantidad PENALIZACION del saldo actual
75     //     SI el saldo actual fuera menor que cero, ENTONCES
76     //         hacer saldo = 0.0
77     //     FIN SI
78     //     devuelve false para avisar que la operación no se llevó a cabo
79     // FIN SI
80 }
81
82 void BankAccount::AgregarInteres ()
83 {
84     // saldo = saldo + saldo * interes
85 }
86
87 void BankAccount::SetSaldo (double nuevoSaldo)
88 {
89     // antes de asignar nuevoSaldo a la variable miembro saldo, verificar que se
90     // cumpla nuevoSaldo >= 0.0 (igual que en el constructor)
91 }
92
93 double BankAccount::GetSaldo ()
94 {
95     // devuelve el saldo asociado esta cuenta
96 }
97
98 void BankAccount::SetInteres (double nuevoInteres)
99 {
100     // antes de asignar nuevoInteres a la variable miembro interes, verificar que se
101     // cumpla 0.0 <= interes <= 0.1 (igual que en el constructor)
102 }
103
104 double BankAccount::GetInteres ()
105 {
106     // devuelve el interés asociado a esta cuenta
107 }
108
109 double BankAccount::GetPenalizacion ()
110 {
111     // devuelve la penalización asociada a esta clase
112 }
113
114
115 /*-----
116 *                               Driver program
117 *-----*/
118 int main()
119 {
120     BankAccount miCuenta {5000.0, 7.0};
121     // nueva cuenta bancaria, con $5000.00 de saldo a 7% de interés
122
123     miCuenta.Imprimir ();
124
125     double cantidadADepositar {0.0};
126     std::cout << "Por favor indique la cantidad a depositar: ";
127     std::cin >> cantidadADepositar;
128

```

```

133     std::cin >> cantidadADebitar;
134
135     if (miCuenta.Debitar (cantidadADebitar) == true) {
136         std::cout << "La operación se realizó con éxito!\n";
137         std::cout << "Su saldo actual es: $" << miCuenta.GetSaldo () << std::endl;
138     }
139     else {
140         std::cout << "Ups, saldo insuficiente\n";
141         std::cout << "A su cuenta se le han quitado $" << miCuenta.GetPenalizacion ();
142         std::cout << " por penalización de saldo insuficiente\n";
143         std::cout << "Su saldo actual es: $" << miCuenta.GetSaldo () << std::endl;
144     }
145
146     miCuenta.AgregarInteres ();
147     // llegó el fin de mes y hay que sumarle los intereses generados al saldo
148     // actual de esta cuenta
149
150     miCuenta.Imprimir ();
151
152     return 0;
153 }
154

```