

```
1 using System;
2
3 namespace EX1
4 {
5     class Calculadora
6     {
7         public float X { get; private set; }
8         public float Y { get; private set; }
9         public float Resultado { get; private set; }
10
11         public void SetValores()
12         {
13             Console.Write("Entre com o 1º Valor: ");
14             X = float.Parse(Console.ReadLine());
15
16             Console.Write("Entre com o 2º Valor: ");
17             Y = float.Parse(Console.ReadLine());
18
19             Console.Clear();
20         }
21         public void Soma()
22         {
23             SetValores();
24             Resultado = X + Y;
25             Console.WriteLine($"Resultado: {Resultado}");
26         }
27         public void Subtracao()
28         {
29             SetValores();
30             Resultado = X - Y;
31             Console.WriteLine("Resultado: {0}", Resultado);
32         }
33         public void Multiplicacao()
34         {
35             SetValores();
36             Resultado = X * Y;
37             Console.WriteLine("Resultado: " + Resultado);
38         }
39         public void Divisao()
40         {
41             SetValores();
42             Resultado = X / Y;
43             Console.WriteLine($"Resultado: {Resultado}");
44         }
45     }
46
47     class Program
48     {
49         static void Main(string[] args)
50         {
51             char opcao;
52             bool funcionando = true;
53
54             Calculadora objeto = new Calculadora();
55
56             Console.WriteLine("Bem vindo á Calculadora!");
```

```
57         do
58         {
59             Console.WriteLine("Qual função deseja executar:\n1 - Soma\n2 - ↗
              Subtração\n3 - Multiplicação\n4 - Divisão\n5 - Sair");
60             opcao = Char.Parse(Console.ReadLine());
61
62             Console.Clear();
63
64             switch (opcao)
65             {
66                 case '1':
67                     objeto.Soma();
68                     break;
69
70                 case '2':
71                     objeto.Subtracao();
72                     break;
73
74                 case '3':
75                     objeto.Multiplicacao();
76                     break;
77
78                 case '4':
79                     objeto.Divisao();
80                     break;
81
82                 case '5':
83                     Console.WriteLine("Encerrando a aplicação...");
84                     funcionando = false;
85                     break;
86
87                 default:
88                     Console.WriteLine("Por favor, digite somente opções ↗
              válidas!");
89                     break;
90             }
91         }while (funcionando == true);
92
93         Console.ReadKey();
94     }
95 }
96 }
97
```

```
1 using System;
2 using System.Collections.Generic;
3 using System.Linq;
4 using System.Text;
5 using System.Threading.Tasks;
6
7 namespace EX2
8 {
9     class Program
10     {
11         class ContaCorrente
12         {
13             //Campos
14             private float _saldo, _credito, _debito;
15             public float SaldoAnterior { get; private set; }
16
17             static float[] totCred = new float[100];
18             string[] descCred = new string[totCred.Length];
19             string[] horaCred = new string[totCred.Length];
20
21             static float[] totDeb = new float[100];
22             string[] descDeb = new string[totDeb.Length];
23             string[] horaDeb = new string[totDeb.Length];
24
25             int[] limite = { 0, 0 };
26
27             //Construtor
28             public ContaCorrente()
29             {
30                 _credito = _debito = SaldoAnterior = _saldo = 0;
31             }
32
33             //Métodos para Crédito
34             public float Credito
35             {
36                 get { return _credito; }
37             }
38             public void AdicionarCredito()
39             {
40                 if (limite[0] > 99)
41                 {
42                     Console.WriteLine("Função inválida, você atingiu o limite de lançamentos.");
43                     Console.ReadKey();
44                 }
45                 else
46                 {
47                     Console.Write("Lançamento a crédito: ");
48                     _credito = float.Parse(Console.ReadLine());
49
50                     Console.Write("Descrição: ");
51                     descCred[limite[0]] = Console.ReadLine();
52
53                     Console.Write("Data do lançamento(dd/mm/aa): ");
54                     horaCred[limite[0]] = Console.ReadLine();
55                 }
46
```

```
56         SaldoAnterior = Saldo;
57
58         //Adicionando crédito ao saldo e posteriormente salvando no histórico de créditos.
59         _saldo += Credito;
60         totCred[limite[0]] = _credito;
61
62         limite[0]++;
63     }
64 }
65
66 //Métodos para Débito
67 public float Debito
68 {
69     get { return _debito; }
70 }
71 public void AdicionarDebito()
72 {
73     if (limite[1] > 99)
74     {
75         Console.WriteLine("Função inválida, você atingiu o limite de lançamentos.");
76         Console.ReadKey();
77     }
78     else
79     {
80         Console.Write("Lançamento a débito: ");
81         _debito = float.Parse(Console.ReadLine());
82
83         Console.Write("Descrição: ");
84         descDeb[limite[1]] = Console.ReadLine();
85
86         Console.Write("Data do lançamento(dd/mm/aa): ");
87         horaDeb[limite[1]] = Console.ReadLine();
88
89         SaldoAnterior = Saldo;
90
91         //Debitando do saldo e posteriormente salvando a ação no histórico de débitos.
92         _saldo -= Debito;
93         totDeb[limite[1]] = _debito;
94
95         limite[1]++;
96     }
97 }
98
99 //Método para Saldo
100 public float Saldo
101 {
102     get { return _saldo; }
103 }
104
105 //Método para mostrar Extrato
106 public void MostraExtrato()
107 {
108     Console.WriteLine("Créditos totais");
```

```
109
110         for (int i = 0; i < limite[0]; i++)
111         {
112             Console.WriteLine("{0}: {1:C} {2}", descCred[i], totCred
113                                     [i], horaCred[i]);
114         }
115
116         Console.WriteLine();
117
118         Console.WriteLine("Débitos totais");
119
120         for (int i = 0; i < limite[1]; i++)
121         {
122             Console.WriteLine("{0}: {1:C} {2}", descDeb[i], totDeb[i],
123                                     horaDeb[i]);
124         }
125
126         Console.ReadKey();
127     }
128
129     static void Main(string[] args)
130     {
131         char opcao;
132         bool sair = false;
133
134         ContaCorrente conta = new ContaCorrente();
135
136         do
137         {
138             Console.Clear();
139
140             Console.WriteLine("1 - Lançar crédito\r2 - Lançar débito\r3 -
141                                     Saldo anterior\r4 - Saldo atual\r5 - Extrato\r6 - Sair");
142
143             opcao = Char.Parse(Console.ReadLine());
144
145             switch (opcao)
146             {
147                 case '1':
148                     conta.AdicionarCredito();
149                     break;
150
151                 case '2':
152                     conta.AdicionarDebito();
153                     break;
154
155                 case '3':
156                     Console.WriteLine($"Saldo anterior: {string.Format
157                                     ("0:C", conta.SaldoAnterior)}");
158                     Console.ReadKey();
159                     break;
160
161                 case '4':
162                     Console.WriteLine($"Saldo: {0:C} ", conta.Saldo);
163                     Console.ReadKey();
```

```
161         break;
162
163         case '5':
164             conta.MostraExtrato();
165             Console.ReadKey();
166             break;
167
168         case '6':
169             sair = true;
170             break;
171     }
172
173     } while (sair == false);
174
175     Console.WriteLine("Encerrando o programa...");
176     Console.ReadKey();
177 }
178 }
179 }
180
```