

```
1 using System;
2 using System.Collections.Generic;
3 using System.Text;
4 namespace Entities {
5     class BanckAccount {
6         public int AccountNumber { get; private set; }
7         public string Holder { get; set; }
8         public double Balance { get; private set; }
9         public BanckAccount() { }
10        public BanckAccount(int accountNumber, string holder) {
11            AccountNumber = accountNumber;
12            Holder = holder;
13        }
14        public BanckAccount(int accountNumber, string holder, double balance) :
15            this(accountNumber, holder) {
16            Balance = balance;
17        }
18        public void Deposit(double amountDeposit) {
19            Balance += amountDeposit;
20        }
21        public void Withdraw(double amountWithdraw) {
22            Balance -= amountWithdraw + 5;
23        }
24        public override string ToString() {
25            return "Conta : " + AccountNumber + ", Titular: " + Holder + ",
26                Saldo: $" + Balance.ToString("F2");
27        }
28    }
29 }
```

Handwritten annotations:

- Soma* (green) with an arrow pointing to the `Balance += amountDeposit;` line (line 19).
- Diminuir* (orange) with an arrow pointing to the `Balance -= amountWithdraw + 5;` line (line 22).

```

1 using System;
2 using Entities;
3 namespace _05__Manual__Exercise_02 {
4     class Program {
5         static void Main(string[] args) {
6             BanckAccount banckAccount;
7             Console.Write("Entre o número da conta: "); int accountNumber =
8             (int.Parse(Console.ReadLine()));
9             Console.Write("Entre o titular da conta: "); string holder =
10             Console.ReadLine();
11
12             Console.WriteLine("Haverá depósito inicial (s/n)? ");
13             char resp = char.Parse(Console.ReadLine());
14             if (resp == 's' || resp == 'S') {
15                 Console.WriteLine("Entre o valor de depósito inicial: ");
16                 double initialDeposit = double.Parse(Console.ReadLine());
17                 banckAccount = new BanckAccount(accountNumber, holder,
18                 initialDeposit);
19             }
20             else {
21                 banckAccount = new BanckAccount(accountNumber, holder);
22             }
23             Console.WriteLine();
24             Console.WriteLine("Dados da conta:");
25             Console.WriteLine(accountNumber);
26
27             Console.WriteLine("");
28             Console.WriteLine("Entre um valor para depósito: ");
29             double amountDeposit = double.Parse(Console.ReadLine());
30             banckAccount.Deposit(amountDeposit);
31             Console.WriteLine("Dados da conta atualizados:");
32             Console.WriteLine(banckAccount);
33
34             Console.WriteLine();
35             Console.WriteLine("Entre um valor para saque: ");
36             double amountWithdraw = double.Parse(Console.ReadLine());
37             banckAccount.Withdraw(amountWithdraw);
38             Console.WriteLine("Dados da conta atualizados:");
39             Console.WriteLine(banckAccount);
40         }
41     }
42 }
43

```

Chamada entidade

Chamada método

Chamada método