

```
1  ///  
    Exemplo de como utilizar o polimorfismo de sobreescrita em  
    linguagem C#    */  
2  
3  //Suponhamos que temos uma classe base chamada 'Animal' e duas classes    
    derivadas,  
4  //'Cachorro' e 'Gato'. A classe 'Animal' possui um método chamado  
5  //'FazerBarulho()', que será sobrescrito nas classes derivadas.  
6  
7  //exemplo01  
8  
9  //class animal  
10 //{  
11 //    public virtual void fazerbarulho()  
12 //    {  
13 //        console.WriteLine("o animal faz algum barulho.");  
14 //    }  
15 //}  
16  
17 //class cachorro : animal  
18 //{  
19 //    public override void fazerbarulho()  
20 //    {  
21 //        console.WriteLine("o cachorro faz um latido.");  
22 //    }  
23 //}  
24  
25 //class gato : animal  
26 //{  
27 //    public override void fazerbarulho()  
28 //    {  
29 //        console.WriteLine("o gato faz um miado.");  
30 //    }  
31 //}  
32  
33 //class program  
34 //{  
35 //    static void main(string[] args)  
36 //    {  
37 //        animal meuanimal = new animal();  
38 //        animal meucachorro = new cachorro();  
39 //        animal meugato = new gato();  
40  
41 //        meuanimal.fazerbarulho();        // chama o método da classe    
        animal  
42 //        meucachorro.fazerbarulho();        // chama o método sobrescrito    
        da classe cachorro  
43 //        meugato.fazerbarulho();        // chama o método sobrescrito    
        da classe gato  
44 //    }  
45 //}  
46  
47 //Em C#, um método virtual é um método de uma classe que pode ser    
    substituído
```

```
48 //por classes derivadas (subclasses) que herdaram da classe onde o método foi definido.
49 //Isso permite que as subclasses forneçam uma implementação específica para o método
50 //Aqui, o método `FazerBarulho` na classe `Animal` é marcado como virtual.
51 //Isso significa que as classes derivadas podem substituir esse método com suas próprias
52 // implementações, usando a palavra-chave `override`.
53 //Neste exemplo, a classe `Animal` tem um método virtual `FazerBarulho()` , que é sobrescrito
54 //nas classes `Cachorro` e `Gato`. Quando você cria instâncias dessas classes e chama o
55 //método `FazerBarulho()` , o método apropriado da classe derivada é chamado,
56 //permitindo que cada tipo de animal faça seu próprio barulho.

57 //Isso demonstra como o polimorfismo de sobrescrita permite que diferentes classes
58 //forneçam implementações específicas para o mesmo método em uma classe base.
59
60
61
62 //Exemplo 02
63
64 //class Veiculo
65 //{
66 //    public virtual void Mover()
67 //    {
68 //        Console.WriteLine("O veículo está se movendo.");
69 //    }
70 //}
71
72 //Classe derivada(subclasse)
73 //class Carro : Veiculo
74 //{
75 //    public override void Mover()
76 //    {
77 //        Console.WriteLine("O carro está se movendo rapidamente.");
78 //    }
79 //}
80
81 //Outra classe derivada (subclasse)
82 //class Bicicleta : Veiculo
83 //{
84 //    public override void Mover()
85 //    {
86 //        Console.WriteLine("A bicicleta está se movendo lentamente.");
87 //    }
88 //}
89
90 //class Program
```

```
91 //{
92 //    static void Main()
93 //    {
94 //        Veiculo veiculo = new Veiculo();
95 //        Veiculo carro = new Carro();
96 //        Veiculo bicicleta = new Bicicleta();
97
98 //        veiculo.Mover(); // Saída: O veículo está se movendo.
99 //        carro.Mover();   // Saída: O carro está se movendo
100 //        bicicleta.Mover(); // Saída: A bicicleta está se movendo
101 //    }
102 //}
103
104
105 class Program
106 {
107
108     class Mercado
109     {
110
111         public float valor { get; set; }
112         public int quantidade { get; set; }
113
114         public virtual void Preco()
115         {
116             Console.WriteLine("O preço dos itens é");
117             Console.WriteLine();
118         }
119     }
120
121
122
123
124     class Doces : Mercado
125     {
126         public override void Preco()
127         {
128             valor = 7.50f;
129
130             Console.WriteLine("Doce: R$ 7,50");
131
132             Console.WriteLine("Digite a quantidade: ");
133             quantidade = int.Parse(Console.ReadLine());
134
135             float resultado = valor * quantidade;
136
137             Console.WriteLine($"Doce: { resultado:F2}");
138
139             Console.ReadKey();
140             Console.WriteLine();
141
```

```
142
143     }
144 }
145
146 class Salgados : Mercado
147 {
148
149     public override void Preco()
150     {
151         valor = 3.50f;
152
153         Console.WriteLine("Salgado: R$ 3,50");
154
155         Console.WriteLine("Digite a quantidade: ");
156         quantidade = int.Parse(Console.ReadLine());
157
158         float resultado = valor * quantidade;
159
160         Console.WriteLine($"Salgado: {resultado:F2}");
161     }
162
163 }
164
165 static void Main(string[] args)
166 {
167
168     Mercado mercado = new Mercado();
169     Mercado doce = new Doces();
170     Mercado salgado = new Salgados();
171
172     mercado.Preco();
173     doce.Preco();
174     salgado.Preco();
175
176 }
177
178 }
179
180
```