

```
151 // Use a diretiva @ para criar um literal de cadeia de caracteres que mantenha toda a formatação de espaço em branco...
152 // ... e caracteres de barra invertida em uma cadeia de caracteres;
153 // Use o \u mais um código de quatro caracteres para representar caracteres Unicode (UTF-16) em uma cadeia de caracteres;
154 // Os caracteres Unicode podem não ser impressos corretamente dependendo do aplicativo;
155
156 //-----
157
158 // Exercício - Concatenação de cadeias de caracteres
159
160 // Etapa 1: Concatenar uma cadeia de caracteres literal e uma variável
161 // Concatenação de cadeia de caracteres: "fala do programador", combinar dois ou mais valores em um novo valor;
162 // Operador de concatenação: símbolo de adição "+";
163
164 // string firstName = "Bob";
165 // string message = "Hello " + firstName;
166 // Console.WriteLine(message);
167 /*
168     Hello Bob
169 */
170 //-----
171 // Etapa 2: Concatenar diversas variáveis e cadeias de caracteres literais
172
173 // string firstName = "Bob";
174 // string greeting = "Hello";
175 // string message = greeting + " " + firstName + "!";
176 // Console.WriteLine(message);
177 /*
178     Hello Bob!
179 */
180 //-----
181 // Etapa 3: Como evitar variáveis intermediárias
182
183 // Evitar usar variáveis intermediárias executando a operação de concatenação;
184 // Simplificando o código;
185
186 // string firstName = "Bob";
187 // string greeting = "Hello";
188 // Console.WriteLine(greeting + " " + firstName + "!");
189 /*
190     Hello Bob!
191 */
192
193 //-----
194
195 // Exercício - Interpolação de cadeia de caracteres
196
197 // Interpolação de cadeia de caracteres: combina vários valores em uma única cadeia de caracteres literal...
198 // ... usando um "modelo" e uma ou mais expressões de interpolação;
199 // Expressão de interpolação: variável cercada por um símbolo de chave de abertura e fechamento { };
```