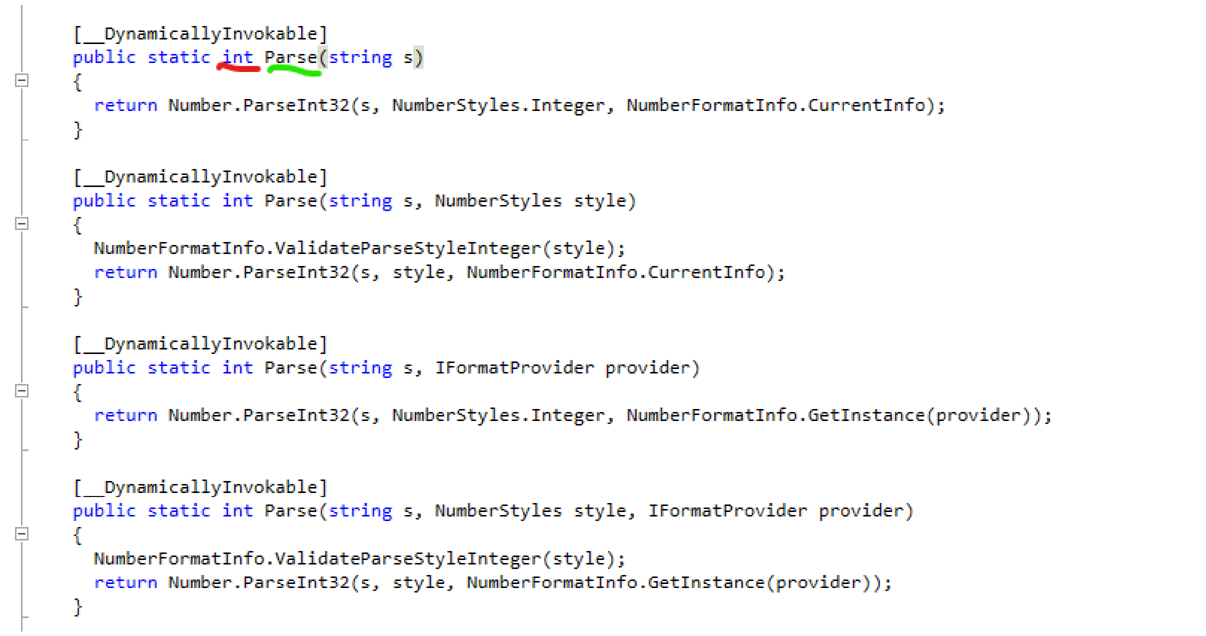
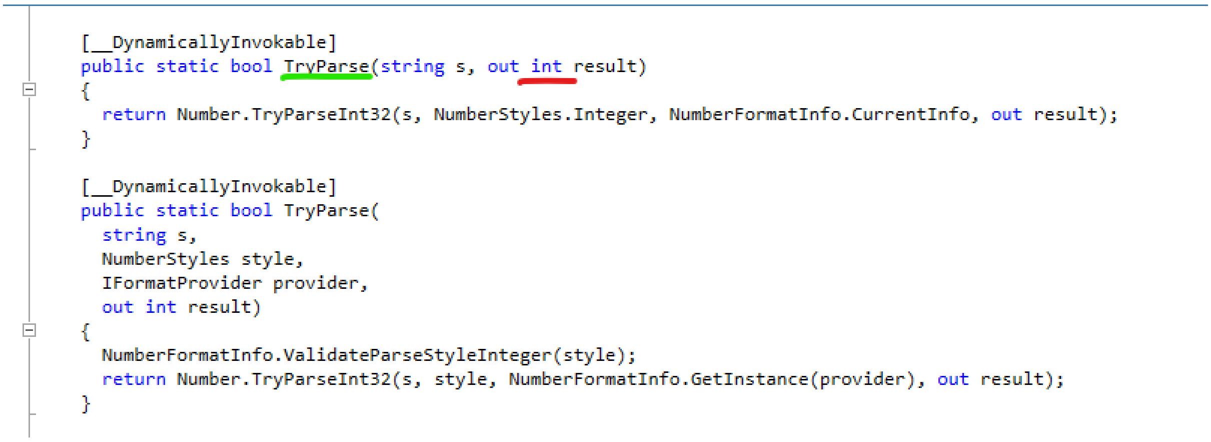
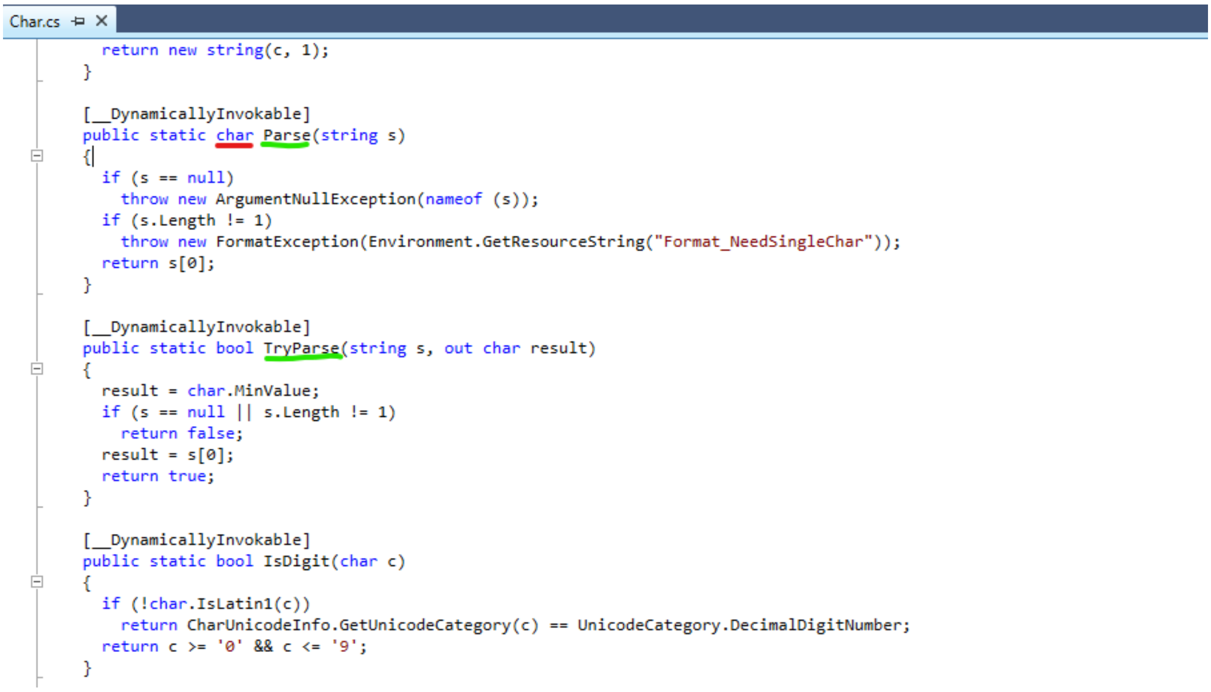


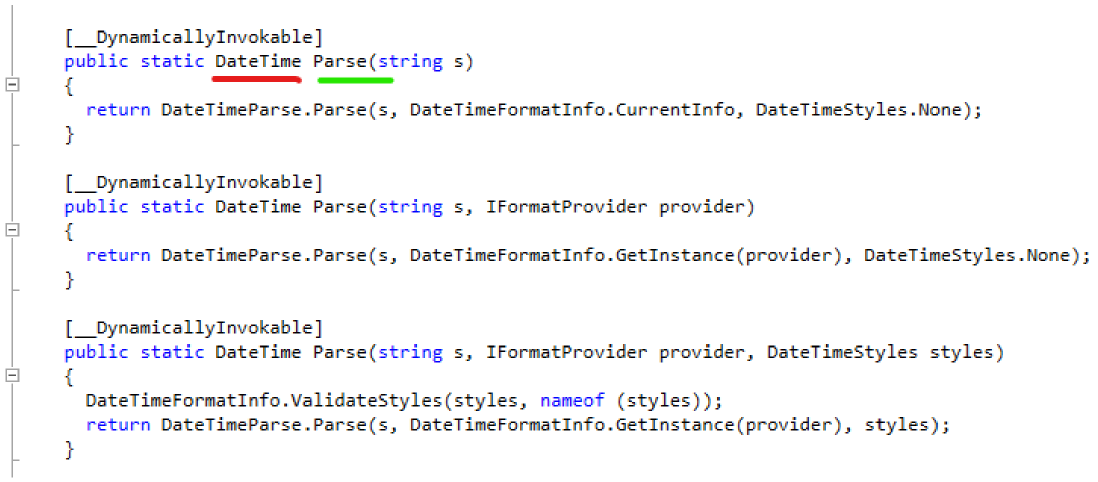
```
[__DynamicallyInvokable]
public static int Parse(string s)
{
  return Number.ParseInt32(s, NumberStyles.Integer, NumberFormatInfo.CurrentInfo);
}

[__DynamicallyInvokable]
public static int Parse(string s, NumberStyles style)
{
  NumberFormatInfo.ValidateParseStyleInteger(style);
  return Number.ParseInt32(s, style, NumberFormatInfo.CurrentInfo);
}

[__DynamicallyInvokable]
public static int Parse(string s, IFormatProvider provider)
{
  return Number.ParseInt32(s, NumberStyles.Integer, NumberFormatInfo.GetInstance(provider));
}

[__DynamicallyInvokable]
public static int Parse(string s, NumberStyles style, IFormatProvider provider)
{
  NumberFormatInfo.ValidateParseStyleInteger(style);
  return Number.ParseInt32(s, style, NumberFormatInfo.GetInstance(provider));
}
```



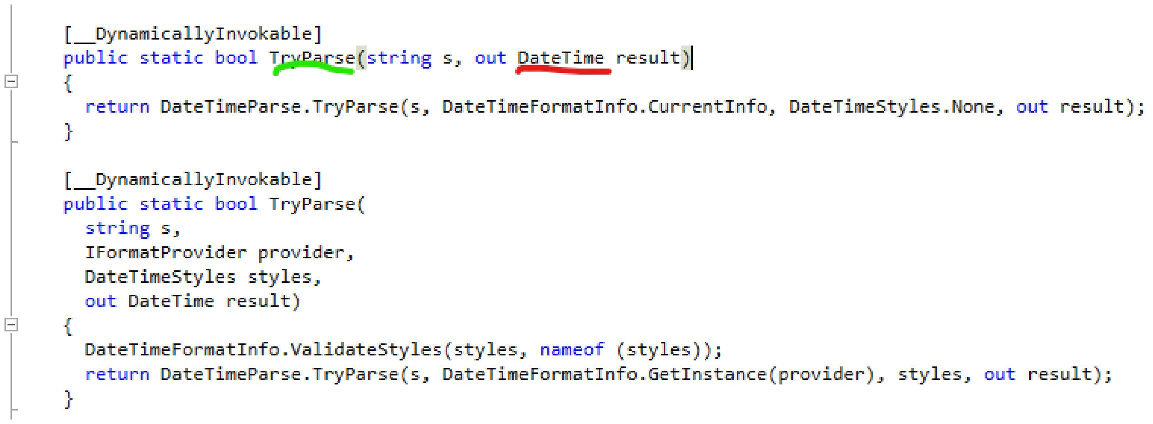
```
[__DynamicallyInvokable]
public static bool TryParse(string s, out int result)
{
  return Number.TryParseInt32(s, NumberStyles.Integer, NumberFormatInfo.CurrentInfo, out result);
}

[__DynamicallyInvokable]
public static bool TryParse(
  string s,
  NumberStyles style,
  IFormatProvider provider,
  out int result)
{
  NumberFormatInfo.ValidateParseStyleInteger(style);
  return Number.TryParseInt32(s, style, NumberFormatInfo.GetInstance(provider), out result);
}
```



```
  return new string(c, 1);
}

[__DynamicallyInvokable]
public static char Parse(string s)
{
  if (s == null)
    throw new ArgumentNullException(nameof (s));
  if (s.Length != 1)
    throw new FormatException(Environment.GetResourceString("Format_NeedSingleChar"));
  return s[0];
}

[__DynamicallyInvokable]
public static bool TryParse(string s, out char result)
{
  result = char.MinValue;
  if (s == null || s.Length != 1)
    return false;
  result = s[0];
  return true;
}

[__DynamicallyInvokable]
public static bool IsDigit(char c)
{
  if (!char.IsLatin1(c))
    return CharUnicodeInfo.GetUnicodeCategory(c) == UnicodeCategory.DecimalDigitNumber;
  return c >= '0' && c <= '9';
}
```



```
[__DynamicallyInvokable]
public static DateTime Parse(string s)
{
  return DateTimeParse.Parse(s, DateTimeFormatInfo.CurrentInfo, DateTimeStyles.None);
}

[__DynamicallyInvokable]
public static DateTime Parse(string s, IFormatProvider provider)
{
  return DateTimeParse.Parse(s, DateTimeFormatInfo.GetInstance(provider), DateTimeStyles.None);
}

[__DynamicallyInvokable]
public static DateTime Parse(string s, IFormatProvider provider, DateTimeStyles styles)
{
  DateTimeFormatInfo.ValidateStyles(styles, nameof (styles));
  return DateTimeParse.Parse(s, DateTimeFormatInfo.GetInstance(provider), styles);
}
```



```
[__DynamicallyInvokable]
public static bool TryParse(string s, out DateTime result)
{
  return DateTimeParse.TryParse(s, DateTimeFormatInfo.CurrentInfo, DateTimeStyles.None, out result);
}

[__DynamicallyInvokable]
public static bool TryParse(
  string s,
  IFormatProvider provider,
  DateTimeStyles styles,
  out DateTime result)
{
  DateTimeFormatInfo.ValidateStyles(styles, nameof (styles));
  return DateTimeParse.TryParse(s, DateTimeFormatInfo.GetInstance(provider), styles, out result);
}
```