Метод Parse() в качестве параметра принимает строку и возвращает объект текущего типа.

```
int a = int.Parse("10");
double b = double.Parse("23,56");
decimal c = decimal.Parse("12,45");
byte d = byte.Parse("4");
```

```
a=10  b=23,56  c=12,45  d=4,
```

```
// [24 1 - 24 25]
IL_0000: ldstr      "10"
IL_0005: call       int32 [System.Runtime]System.Int32::Parse(string)
IL_000a: stloc.0    // a

// [25 1 - 25 34]
IL_000b: ldstr      "23,56"
IL_0010: call       float64 [System.Runtime]System.Double::Parse(string)
IL_0015: stloc.1    // b

// [26 1 - 26 36]
IL_0016: ldstr      "12,45"
IL_001b: call       valuetype [System.Runtime]System.Decimal [System.Runtime]System.Decimal::Parse(string)
IL_0020: stloc.2    // c

// [27 1 - 27 26]
IL_0021: ldstr      "4"
IL_0026: call       unsigned int8 [System.Runtime]System.Byte::Parse(string)
IL_002b: stloc.3    // d
```

```
// [29 1 - 29 50]
IL_002c: ldloca.s   V_7
IL_002e: ldc.i4.s   14 // 0x0e
IL_0030: ldc.i4.4
IL_0031: call       instance void [System.Runtime]System.Runtime.CompilerServices.DefaultInterpolatedStringHandler::.ctor(int32, int32)
IL_0036: ldloca.s   V_7
IL_0038: ldstr      "a="
IL_003d: call       instance void [System.Runtime]System.Runtime.CompilerServices.DefaultInterpolatedStringHandler::AppendLiteral(string)
IL_0042: nop
IL_0043: ldloca.s   V_7
IL_0045: ldloc.0    // a
IL_0046: call       instance void [System.Runtime]System.Runtime.CompilerServices.DefaultInterpolatedStringHandler::AppendFormatted<int32>(!!0/*int32*/)
IL_004b: nop
IL_004c: ldloca.s   V_7
IL_004e: ldstr      "  b="
IL_0053: call       instance void [System.Runtime]System.Runtime.CompilerServices.DefaultInterpolatedStringHandler::AppendLiteral(string)
IL_0058: nop
IL_0059: ldloca.s   V_7
IL_005b: ldloc.1    // b
IL_005c: call       instance void [System.Runtime]System.Runtime.CompilerServices.DefaultInterpolatedStringHandler::AppendFormatted<float64>(!!0/*float64*/)
IL_0061: nop
IL_0062: ldloca.s   V_7
IL_0064: ldstr      "  c="
IL_0069: call       instance void [System.Runtime]System.Runtime.CompilerServices.DefaultInterpolatedStringHandler::AppendLiteral(string)
IL_006e: nop
IL_006f: ldloca.s   V_7
IL_0071: ldloc.2    // c
IL_0072: call       instance void [System.Runtime]System.Runtime.CompilerServices.DefaultInterpolatedStringHandler::AppendFormatted<valuetype [System.Runtime]System.Decimal>(!!0/*valuetype [System.Runtime]System.Decimal*/)
IL_0077: nop
IL_0078: ldloca.s   V_7
IL_007a: ldstr      "  d="
IL_007f: call       instance void [System.Runtime]System.Runtime.CompilerServices.DefaultInterpolatedStringHandler::AppendLiteral(string)
IL_0084: nop
IL_0085: ldloca.s   V_7
IL_0087: ldloc.3    // d
IL_0088: call       instance void [System.Runtime]System.Runtime.CompilerServices.DefaultInterpolatedStringHandler::AppendFormatted<unsigned int8>(!!0/*unsigned int8*/)
IL_008d: nop
IL_008e: ldloca.s   V_7
IL_0090: call       instance string [System.Runtime]System.Runtime.CompilerServices.DefaultInterpolatedStringHandler::ToStringAndClear()
IL_0095: call       void [System.Console]System.Console::WriteLine(string)
IL_009a: nop
```

Однако тем не менее потенциально при использовании метода Parse мы можем столкнуться с ошибкой, например, при передачи алфавитных символов вместо числовых. И в этом случае более удачным выбором будет применение метода TryParse(). Он пытается преобразовать строку к типу и, если преобразование прошло успешно, то возвращает true. Иначе возвращается false:

```
Введите строку:
выфвф
Преобразование завершилось неудачно
```

```
Введите строку:
22
Преобразование прошло успешно. Число: 22
```

```
// [34 1 - 34 51]
IL_00ad: ldloc.s      input
IL_00af: ldloca.s     number
IL_00b1: call         bool [System.Runtime]System.Int32::TryParse(string, int32&)
IL_00b6: stloc.s      result
```

```
// [36 5 - 36 74]
IL_00c0: ldloca.s     V_7
IL_00c2: ldc.i4.s     38 // 0x26
IL_00c4: ldc.i4.1
IL_00c5: call         instance void [System.Runtime]System.Runtime.CompilerServices.DefaultInterpolatedStringHandler::.ctor(int32, int32)
IL_00ca: ldloca.s     V_7
IL_00cc: ldstr        "Преобразование прошло успешно. Число: "
IL_00d1: call         instance void [System.Runtime]System.Runtime.CompilerServices.DefaultInterpolatedStringHandler::AppendLiteral(string)
IL_00d6: nop
IL_00d7: ldloca.s     V_7
IL_00d9: ldloc.s      number
IL_00db: call         instance void [System.Runtime]System.Runtime.CompilerServices.DefaultInterpolatedStringHandler::AppendFormatted<int32>(!!0/*int32*/)
IL_00e0: nop
IL_00e1: ldloca.s     V_7
IL_00e3: call         instance string [System.Runtime]System.Runtime.CompilerServices.DefaultInterpolatedStringHandler::ToStringAndClear()
IL_00e8: call         void [System.Console]System.Console::WriteLine(string)
IL_00ed: nop
```

Вывод

Метод TryParse лучше использовать, когда:

Тип входных данных неизвестен (консольный ввод), чтобы избежать ошибки неправильных входных данных. Это позволит нам воспользоваться булевой переменной, которая даст результат true/false, для того, чтобы узнать или сообщить пользователю, что данные введены некорректно.

Метод Parse удобно использовать, когда мы знаем, каким будет тип входных данных и не нужно проверять результат преобразований.