Why can you use just the alias to declare a enum and not the .NET type?

Asked 4 years, 11 months ago Active 4 years, 11 months ago Viewed 489 times

11

3

This works perfectly..

```
public  enum NodeType : byte
{ Search, Analysis, Output, Input, Audio, Movement}
```

This returns a compiler error...

```
public  enum NodeType : Byte
{ Search, Analysis, Output, Input, Audio, Movement}
```

Same happen when using reflection...

So, does somebody know why the `enum` -base is just an integral-type?

c# .net types enums

edited Nov 11 '14 at 14:33
Patrick Hofman
134k 18 193 258

asked Nov 11 '14 at 14:13
MrVoid
484 4 17

possible duplicate of Difference between byte vs Byte data types in C# – paparazzo Nov 11 '14 at 14:28

See the duplicate I posted. In some cases you are required to use the keyword. – paparazzo Nov 11 '14 at 14:29

3 Answers

Probably it is just a incomplete compiler implementation (while documented).

Technically, this should work too, but it doesn't.

10

```
using x = System.Byte;

public  enum NodeType : x
{ Search, Analysis, Output, Input, Audio, Movement}
```

So the parser part of the compiler just allows the fixed list `byte, sbyte, short, ushort, int, uint, long, or ulong` . There is no technical restriction I am aware of.

answered Nov 11 '14 at 14:21

Patrick Hofman
134k 18 193 258

5

Because the specs say so:

```
enum-declaration:

    attributes_opt
    enum-modifiers_opt enum identifier enum-base_opt
    enum-body ;_opt


enum-base:

    : integral-type


enum-body:

    { enum-member-declarations_opt }

    { enum-member-declarations , }


Each enum type has a corresponding integral type called the underlying type
of the enum type. This underlying type must be able to represent all the
enumerator values defined in the enumeration. An enum declaration may
explicitly declare an underlying type of byte, sbyte, short, ushort, int, uint,
long or ulong. Note that char cannot be used as an underlying type. An enum
declaration that does not explicitly declare an underlying type has an
underlying type of int.


...
```

integral-type is defined as,

```
 integral-type:

     sbyte

     byte

     short

     ushort

     int

     uint

     long

     ulong

     char
```

edited Nov 11 '14 at 15:19
Jodrell
27.9k 3 61 107

answered Nov 11 '14 at 14:22
Selman Genç
87.3k 11 84 149

0

Byte, Int32 etc are objects. Since enums require integral types and these are not, you get a compiler error. C#'s enum definition is much closer to C in that respect.

This is very different from Java, where enums are a misnomer since they are really named singletons.

answered Nov 11 '14 at 14:24
plinth
42.5k 8 72 112