ENUMS- wyliczenia pozwalają zadeklarować zestaw nazwanych stałych, tj. zbiór powiązanych wartości, które mogą być numeryczne lub łańcuchowe. Istnieją 3 typy enums:

1. **liczbowe 2. Ciągu (string) 3. Heterogeniczne**

1. Wyliczenia liczb (number)- oparte na liczbach, tj. przechowują wartości łańcuchowe jako liczby. Wyliczeniom zawsze przypisuje się liczby. Pierwsza wartość przyjmuje wartość równą 0, a pozostałe zwiększane o 1(**A**). Mamy możliwość samodzielnego zainicjowania pierwszej wartości(**B**). Nie jest konieczne przypisywanie kolejnych wartości(**C**).Wyliczenia można użyć jako parametru funkcji lub typu zwracanego, jak pokazano poniżej:

enum PrintMedia {

function getMedia(mediaName: string): PrintMedia {

if ( mediaName === 'Forbes' || mediaName === 'Outlook') {

return PrintMedia.Magazine;

}

}

Newspaper = 1,

Newsletter,

Magazine,

Book

}

let mediaType: PrintMedia = getMedia('Forbes'); // returns Magazine

Enum Direction **A** {

Up, //0

Down, //1..

Left,

Right

}

Enum Direction **B** {

Up=1 //1

Down, //2…

Left,

Right

}

Enum Direction **C** {

Up, //1

Down, //5.

Left, //5

Right //10

}

**COMPUTED (obliczone)enumy liczbowe** mogą zawierać ele. z obliczaną/stałą wartością Np:

enum PrintMedia {

Newspaper = 1,

Newsletter = getPrintMediaCode('newsletter'),

Magazine = Newsletter \* 3,

Book = 10

}

function getPrintMediaCode(mediaName: string): number {

if (mediaName === 'newsletter') {

return 5;

}

} PrintMedia.Newsetter; // returns 5 PrintMedia.Magazine; // returns 15

Gdy enum ma obliczone i stałe elementy, to nieinicjowane muszą być pierwsze lub muszą następować po innych zainicjowanych elementach ze stałymi wartościami. Przykład błędu:

enum PrintMedia {

Newsletter = getPrintMediaCode('newsletter'),

Newspaper, // Error: Enum member must have initializer

Book,

Magazine = Newsletter \* 3,

}

Powyższe wyliczenie można zadeklarować jak poniżej.

enum PrintMedia {

Newspaper,

Book,

Newsletter = getPrintMediaCode('newsletter'),

Magazine = Newsletter \* 3

}

// or

enum PrintMedia {

Newsletter = getPrintMediaCode('newsletter'),

Magazine = Newsletter \* 3,

Newspaper = 0,

Book,

}

2. Wyliczenia ciągów (stringów)

Niżej zdefiniowaliśmy enum string, z tą różnicą, że tu wartości enum są inicjowane literałami łańcuchowymi. Różnica między enum numerycznymi i łańcuchowymi to, to że wartości liczbo-we są automatycznie zwiększane, a wartości stringów muszą być inicjowane indywidualnie.

enum PrintMedia {

// Access String Enum

PrintMedia.Newspaper; //returns NEWSPAPER

PrintMedia['Magazine'];//returns MAGAZINE

Newspaper = "NEWSPAPER",

Newsletter = "NEWSLETTER",

Magazine = "MAGAZINE",

Book = "BOOK"

}

3. Wyliczenia heterogeniczne- zawierające i wartości łańcuchowe i liczbowe.

enum Status {

Active = 'ACTIVE',

Deactivate = 1,

Pending

}

Mapowanie odwrotne- W TS możemy uzyskać dostęp do wartosci z elementu, a także nazwe z jego wartości.

enum PrintMedia { PrintMedia.Magazine; // returns 3

Newspaper = 1, PrintMedia["Magazine"];// returns 3

Newsletter, PrintMedia[3]; // returns Magazine

Magazine,

Book

}

**Uwaga!** Mapowanie odwrotne nie jest obsługiwane w przypadku elementów stringów. W przypadku heterogenicznego jest obsługiwane tylko dla elementów typu liczbowego.

## Obliczone i stałe elementy –Element wyliczenia jest stały, jeśli:

* Jest to pierwszy element w wyliczeniu i nie ma inicjatora, w takim przypadku jest przypisywana wartość 0:

enum Elf {

X, // Elf.X jest stala = 0:

}

* Nie ma inicjatora, a poprzedzający element wyliczeniowy był stałą numeryczną . W tym przypadku wartość bieżącego elementu enum będzie wartością poprzedniego elementu enum plus jeden.

enum E1 { enum E2 {

X, //staly,bo pierwszy i bez inicjatora,tj=0 A = 1,

Y, //staly, bo poprz. jest staly = 0; tj.=1 B,

Z, //stały, bo poprz. jest stały = 1;tj.= 2 C, //wszystkie sa stale

} }

* Element jest inicjowany za pomocą stałego wyrażenia wyliczenia, które jest podzbiorem wyrażeń TypeScript, które można w pełni ocenić w czasie kompilacji. Wyrażenie jest stałym wyrażeniem wyliczeniowym, jeśli jest:
  1. dosłowne wyrażenie enum (w zasadzie literał łańcuchowy lub literał liczbowy)
  2. odniesienie do wcześniej zdefiniow. stałego członka enum (który może pochodzić z innego enum)
  3. stałe wyliczenie w nawiasach
  4. jeden z operatorów jednoargumentowych +, -, ~zastosowany do wyrażenia stałego wyliczenia
  5. +, -, \*, /, %, <<, >>, >>>, &, |, ^operatory binarne ze stałymi wyrażeniami enum jako argumentami

Jest to błąd czasu kompilacji dla stałych wyrażeń wyliczeniowych, które mają być oceniane na NaNlub Infinity.

We wszystkich innych przypadkach element enum jest uważany za obliczony.

enum FileAccess {

// stale elementy

None,

Read = 1 << 1,

Write = 1 << 2,

ReadWrite = Read | Write,

// obliczone elementy

G = "123".length, }