

Енумерације

Сваки до сада коришћени тип података је имао дефинисан сет дозвољених вредности, иако су неки типови (нпр `int`, `double`...) изгледали као да баратају са бесконачним сетом вредности.

Постоје многе ситуације где је корисно имати промењиве које могу користити само фиксан сет вредности (нпр стране света везано за орјентацију у простору).

У таквим ситуацијама се користе енумерације које допуштају дефинисање новог типа података над коначним сетом вредности.

У поступку коришћења енумерација се дешава и декларисање новог типа података са дефинисањем могућих вредности и декларисање промењиве тог новог типа података.

Дефинисање енумерације

Да би се дефинисала енумерација користи се кључна реч `enum`:

```
enum <imeTipa>
{
    <vrednost1>,
    <vrednost2>,
    ...
    <vrednostN>
}
```

За декларисање промењивих овог новог типа се користи следећа синтакса:

```
<imeTipa> <imePromenjive>;
```

За доделу вредности се користи следећа синтакса:

```
<imePromenjive> = <imeTipa>.<vrednost>;
```

Енумерације користе један подтип (`underlying type`) који се користи за смештање података.

Свака од вредности коју може имати енумерација је смештен као вредност овог подтипа, који је по дифолту типа `int`.

Може се изабрати и други подтип додавањем типа током декларације енумерације:

```
enum <imeTipa> : <podtip>
{
    <vrednost1>,
    <vrednost2>,
    ...
    <vrednostN>
}
```

Подтипови који се могу користити у овом контексту су `byte`, `sbyte`, `short`, `ushort`, `int`, `uint`, `long`, `ulong`.

По дифолту, свакој вредности је аутоматски додељена одговарајућа вредност у подтипу према редоследу по којем су дефинисане, почевши од 0.

То значи да `<vrednost1>` добија вредност 0, `<vrednost2>` добија вредност 1...

Може да се изврши преклапање (`override`) оваквих израза коришћењем оператора доделе и специфицирањем одређене вредности за сваку вредност енумерације:

```
enum <imeTipa> : <podtip>
{
    <vrednost1> = <odredjenaVrednost1>,
    <vrednost2> = <odredjenaVrednost2>,
    ...
    <vrednostN> = <odredjenaVrednostN>
}
```

Уз то, може се специфицирати идентична вредност за више вредности енумерације коришћењем једне вредности као подвредности друге вредности:

```
enum <imeTipa> : <podtip>
{
    <vrednost1> = <odredjenaVrednost1>,
    <vrednost2> = <vrednost1>,
    <vrednost3>,
    ...
    <vrednostN> = <odredjenaVrednostN>
}
```

Све вредности које су преостале недодељене добијају подвредност аутоматски, пошто вредности у секвенци започињу са вредности за 1 већој од последње експлицитно декларисане.

У претходном коду, <vrednost3> добија аутоматски вредност <vrednost1> + 1.

Овакав начин доделе вредности може креирати проблеме, нпр у следећем коду, <vrednost4> ће имати исту вредност као <vrednost2>:

```
enum <imeTipa> : <podtip>
{
    <vrednost1> = <odredjenaVrednost1>,
    <vrednost2>,
    <vrednost3> = <vrednost1>,
    <vrednost4>,
    ...
    <vrednostN> = <odredjenaVrednostN>
}
```

Али зато следећи код ће изазвати грешку:

```
enum <imeTipa> : <podtip>
{
    <vrednost1> = <vrednost2>,
    <vrednost2> = <vrednost1>
}
```