

5. enum

H5. Beslissingen

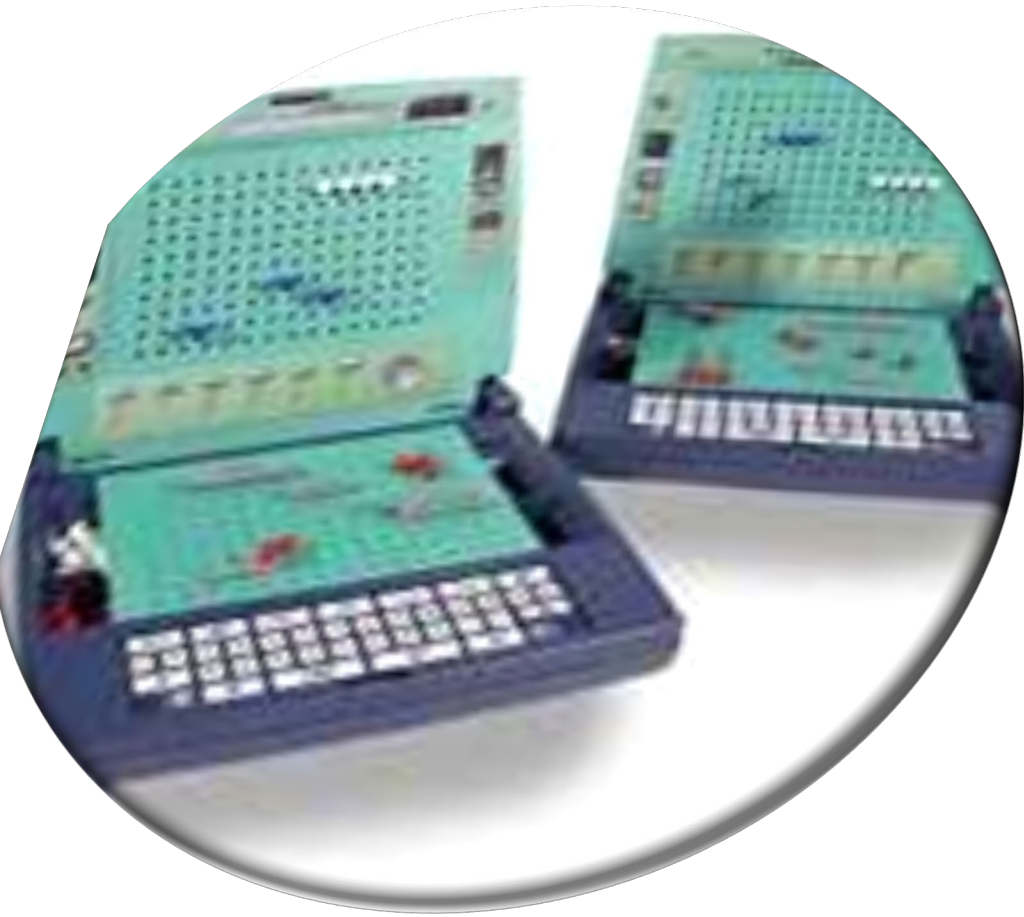


Enumeratie en states

- Enumeration of enumeratie = nummering
- Enumerated types: speciaal type (maar duur woord)

Nut van enum?

- Nummer bewaren:
 - Gebruik int, short, etc
- True/False:
 - Bool
- Maar wat als we specifieke waarden of status van een systeem (states) willen bewaren?



States/status voorbeelden

- Zeeslag:
- Ieder vakje kan een van volgende zaken bevatten:
 - Empty sea
 - Attacked
 - Battleship
 - Cruiser
 - Submarine
 - Rowing boat

States/status voorbeelden

- Eigenlijk willen we dus metadata beschrijven:
- Of
 - Empty sea = 1
 - Attacked = 2
 - Battleship = 3
 - Cruiser = 4
 - Submarine = 5
 - Rowing boat = 6



Moeilijk leesbare/aanpasbare code

Bad coding
example!



```
if(geraaktvakje==1)  
    Console.WriteLine("Plons, gemist");
```

- Wat als we per ongeluk 6 typen?
- Wat als we later liever voor een leeg vakje een andere waarde willen?
- Wat als we 7 typen? Bestaat dit?
- Etc. Etc.

Oplossing: enumerated types

- `enum` keyword

We definiëren een nieuw type "SeaState" dat enkel de waarden kan hebben tussen de accolades

```
public enum SeaState
{
    EmptySea,
    Attack,
    BattleShip,
    Cruiser,
    Submarine,
    RowingBoat
};
```

- Moet gedeclareerd worden buiten Main (als apart onderdeel van Program class bijvoorbeeld)

Gebruik enum

- SeaState is als het ware een nieuw type (zoals int, double) die maar 6 mogelijke waarden kan bevatten.

```
SeaState openSea;  
openSea = SeaState.EmptySea;  
//...  
//verder in code  
if (openSea==SeaState.Attack)  
    Console.WriteLine("You already shot this spot");
```


Slowmotion

- 1° Variabele van het type SeaState aanmaken, genaamd openSea

```
SeaState openSea;
```

Slowmotion

- 2° Variabele openSea een waarde geven, nl. EmptySea

```
SeaState openSea;  
openSea = SeaState.EmptySea;
```

Slowmotion

- 3° iets doen afhankelijk van de huidige waarde van openSea

```
SeaState openSea;  
openSea = SeaState.EmptySea;  
//...  
//verder in code  
if(openSea==SeaState.Attack)  
    Console.WriteLine("You already shot this spot");
```

Nog een voorbeeld

```
public enum TrafficLight
{
    Red,
    Amber,
    Green
};

static void Main(string[] args)
{

    TrafficLight light1, light2;
    light1 = TrafficLight.Red;
    light2 = TrafficLight.Green;
```

Enum voor states

- Enum is ideaal om de staat (state) van iets te bevatten dat een eindig aantal mogelijkheden maar heeft.

- Bijvoorbeeld

- State van game:

```
public enum GameState
{
    Intro,
    Play,
    GameOver,
    HighScore
};
```

- Staat van persoon zijn relatie:

```
public enum RelatieStatus
{
    Vrijgezel,
    Samenwonend,
    Getrouwd,
    Gescheiden,
    Weduwenaar,
    Amish
};
```

Enum to int

- Omzetten tussen int en enum enkel via casting!

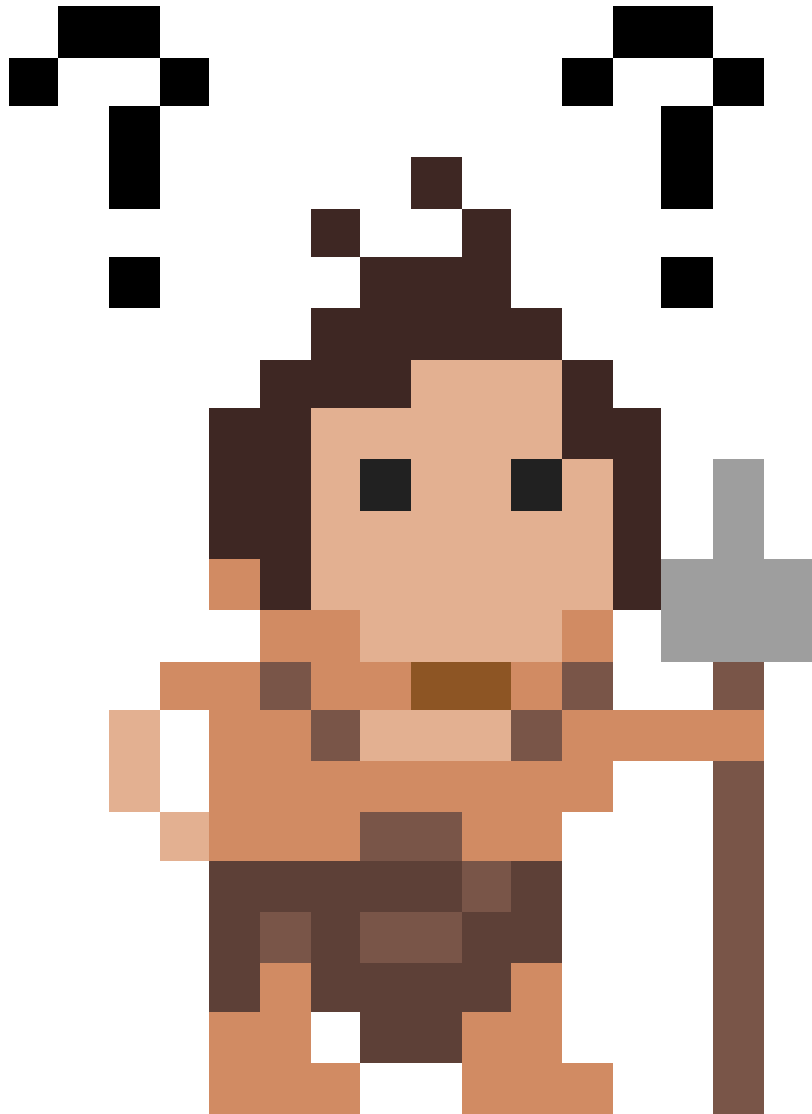
```
enum Weekdagen {Maandag, Dinsdag, Woensdag, Donderdag, Vrijdag, Zaterdag, Zondag}  
  
int keuze= 3;  
  
Weekdagen dagKeuze = (Weekdagen)keuze;  
  
//dagKeuze zal de waarde Weekdagen.Donderdag hebben
```

Programmer's Point

“Use enumerated types”

Voordeel voor alle partijen:

- 1° Makkelijker te schrijven en lezen code
- 2° Verstaanbare code
- 3° Meer resistent tegen bugs



Leer goed werken met enums

They might someday save your life!