

***SOFTWARE ONTWIKKELEN***



# ***AGENDA VOOR VANDAAG***

- Lesdoelen
- Leerstrategie: concrete voorbeelden
- Lists
- Diverse opdrachtjes

# ***LESDOELEN***

Na de les van vandaag:

- Heb je nieuwe *concrete voorbeelden* gevonden over het gebruik van methodes.
- Weet je wat een list is en kun je deze aanmaken in C#
- Kun je elementen beheren in een lijst: toevoegen, aanpassen en verwijderen!
- Weet je wat de index van een lijst is!

# ***VOORBEELDEN ZOEKEN (1)***

- In tweetallen
- Zoek drie voorbeelden in drie bronnen van verschillende methoden:
  - 1 in de lesmaterialen -> kijk eens in het project.
  - 1 op het internet -> zoek op internet naar opensource projecten in Github.
  - 1 zelfbedacht voorbeeld -> waar zou jij een methode voor schrijven in je project?
- Vraag de docent of je voorbeelden kloppen (indien nodig)
- Beschrijf aan elkaar hoe het voorbeeld past bij het concept dat je bestudeert.

# VOORBEELDEN ZOEKEN (2) (15 MINUTEN)

- Zoek een voorbeeld van een:
  - Static methode.
  - Een void methode.
  - Methode die iets teruggeeft wat je niet direct begrijpt en probeer uit te zoeken wat dat dan is.

```
2 references  
public Task<List<Measurement>> GetEnergyConsumed(int meterId, int daysToRetrieve, string aggregationWindow)  
{
```

Focus op: wat herken ik wel?

- naam methode,
- parameters (3) met types,
- access modifier (public)!,
- List? Als return type!
- Maar: wat is een Task? stel jezelf dan de vraag: is dat erg belangrijk om te weten?  
Zo ja: zoek dat dan op!

# ***VOORBEELDEN***

- Zoek voorbeelden van alle voorgaande concepten in het lesmateriaal en op internet.
  - Raadpleeg je docent als je niet zeker weet of je voorbeeld wel klopt
  - Leg aan je eendje (of een medestudent, of een huisgenoot) uit hoe het voorbeeld verband houdt met je concept
  - Verzin ook nieuwe voorbeelden
- 
- Overleg dat wat je gevonden hebt met de medestudent naast je!



# ***LISTS - LIJSTEN***

- Onderdeel van C# collections.
- Lijken veel op array, maar flexibeler qua grootte en toepassing
- Verzameling van gelijksoortige gegevens.



# ***LISTS – WANNEER GEBRUIK IK ZE?***

- Om meerdere gelijksoortige gegevens in op te slaan.
- Krijg je de neiging om een 2<sup>e</sup> of 3<sup>e</sup> variabele te maken: overweeg dan: een lijst!

```
string aPhonetic = "Alfa";  
string bPhonetic = "Bravo";  
  
List<string> phoneticAlphabet = ["Alfa", "Bravo", "Charlie", "Delta", "Echo"];
```



# LISTS – C# SYNTAX

```
List<int> leeftijden = [12];
```

- Nieuwe lijst:

```
List<int> leeftijden = [];
```

- Voeg een leeftijd toe:

```
leeftijden.Add(12);
```

- Verwijder de 'eerste' 12:

```
leeftijden.Remove(12);
```



Zeg dan!  
eerstvoorkomende

# ***LISTS – NU ZELF DOEN***

Verzamel namen totdat de gebruiker 'stop' invult.

Nadat je de namen hebt verzameld stuur je naar de console:

“Naam 0 is William en bevat 7 karakters”

“Naam 1 is Marc en bevat 4 karakters”

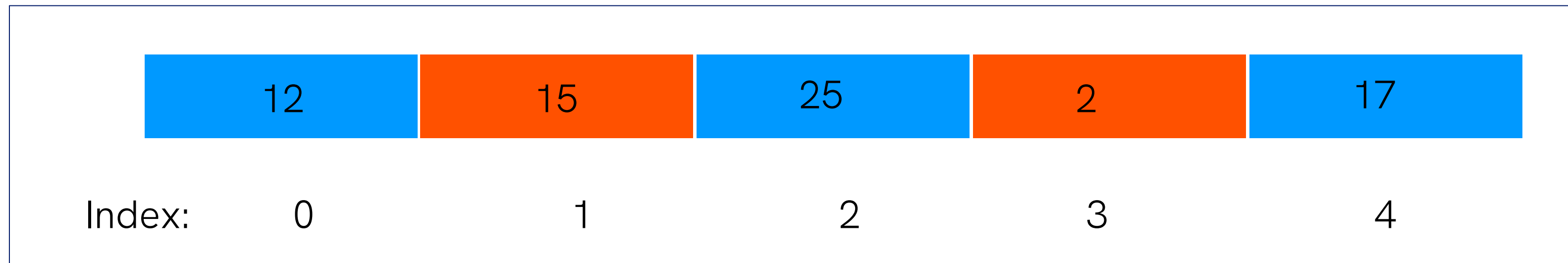
Etc...

```
Naam 0 is Marc en bevat 4 karakters  
Naam 1 is Alex en bevat 4 karakters  
Naam 2 is Jan en bevat 3 karakters  
Naam 3 is Lisa en bevat 4 karakters
```



# LISTS – INDEX

```
List<int> leeftijden = [12, 15, 25, 2, 17];
```



Een index geeft de positie in de lijst aan.

- Je kunt die ook handig gebruiken bij het uitlezen:

```
Console.WriteLine(leeftijden[2]);
```

- Of bij het veranderen:

```
leeftijden[0] += 1;
```

- Of bij het verwijderen:

```
leeftijden.RemoveAt(2);
```

# ***LISTS – NU ZELF DOEN***

- Vul een lijst met alle veelvouden van 5 tussen 0 en 2000.
- Verwijder daarna alle getallen die een index hebben met een veelvoud van 10! Bewaar deze getallen wel in een nieuwe lijst!

Ok, vooruit een hint: `List<int> veelvoudenIndexTien = [];`





# ***LISTS – HANDIGHEIDJES***

```
foreach (int getal in veelvoudenVijf)
{
    Console.Write(getal + " ");
}
// verkort:
Console.WriteLine(string.Join(" ", veelvoudenVijf));
```

# ***LESDOELEN BEHAALD?***

- Heb je nieuwe concrete voorbeelden gevonden over het gebruik van methodes.
- Weet je wat een list is en kun je deze aanmaken in C#
- Kun je elementen beheren in een lijst: toevoegen, aanpassen en verwijderen!
- Weet je wat de index van een lijst is!



# ***VOORBEREIDEN***

- Neem een nieuw voorbeeld van het gebruik van lists mee naar de eerstvolgende les!