C# Essentials

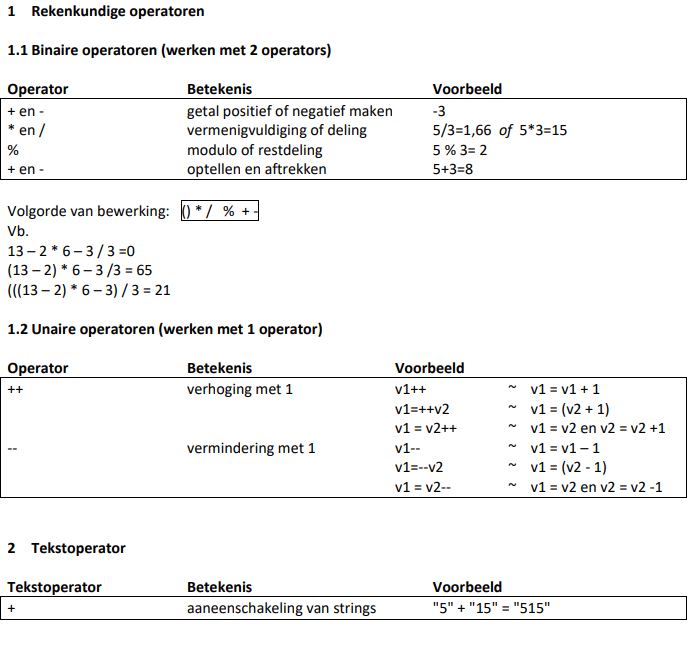
# Hoofdstuk 1: Probleemoplossend denken

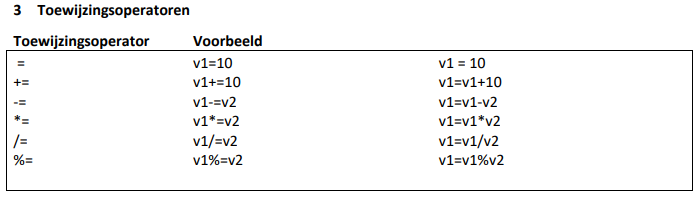
Leren programmeren bestaat vooral uit het leren van een probleem te ontleden en een stapsgewijze oplossing of algoritme op te bouwen. Alles moet goed gedocumenteerd en duidelijk gestructureerd zijn.

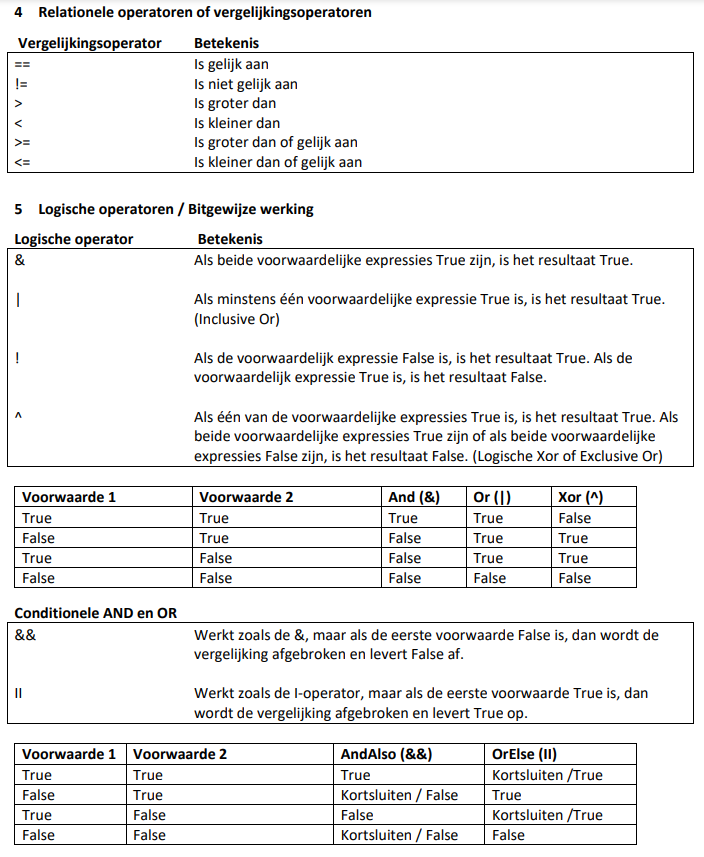
Hier heb je 5 fases voor

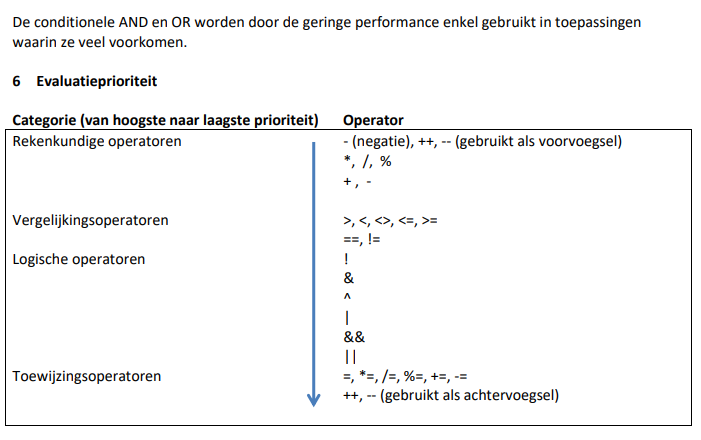
1. Probleemanalyse -> wat moet er gebeuren? Hoe pak ik dit aan? Wat is het probleem, ...
2. Oplossingsstrategie -> Hoe los ik het probleem op? Welke technieken gebruik ik? ...
3. Algoritme -> oplossing gebaseerd op stappen vanaf een beginsituatie tot eindproduct.
4. Programma -> Welk programma/Programmeertaal ga ik gebruiken?
5. Testen van het programma
6. Documentatie (ook deel van 5)

# Hoofdstuk 2: Operatoren



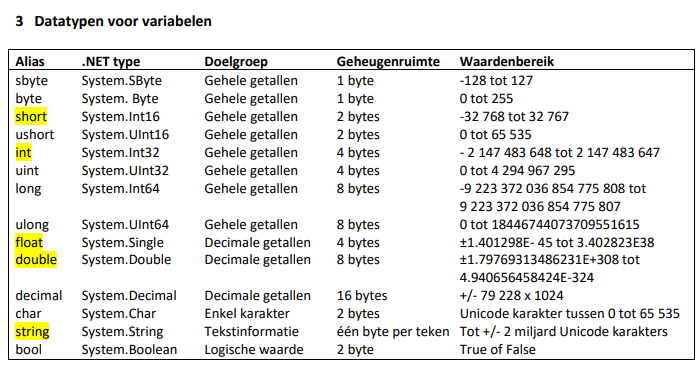


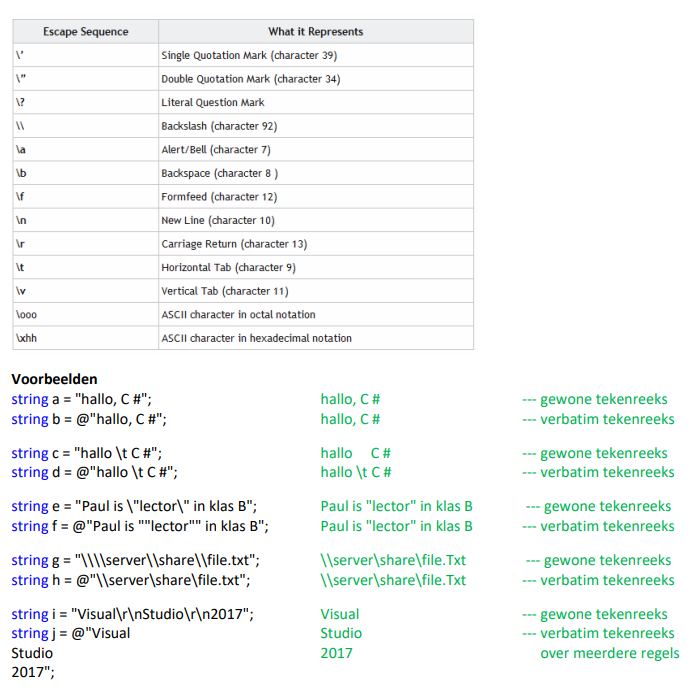




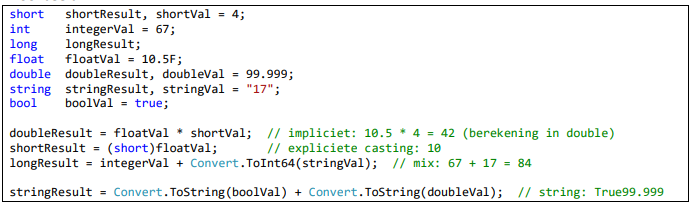
# Hoofdstuk 3: Informatie opslaan in variabelen en constanten

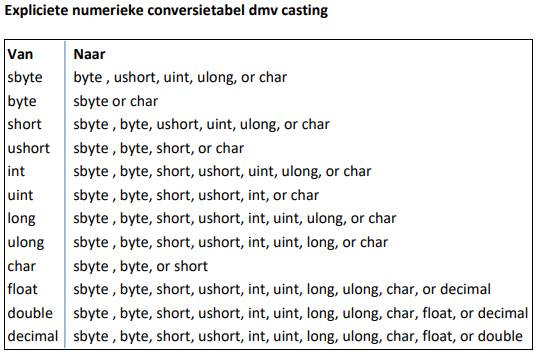
Code conventie voor variabelen:

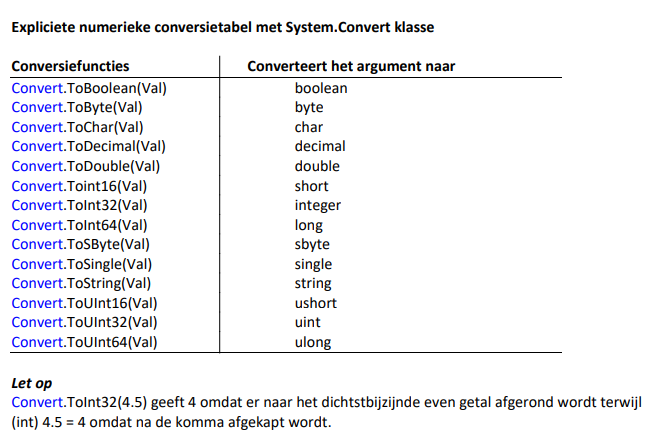
* De naam kan met een letter, \_ of @ beginnen, niet met een cijfer of ander teken.
* De rest van de naam mag uit letters, cijfers en \_ bestaan (geen spatie, punt en puntkomma).
* De naam mag niet langer dan 255 tekens zijn.
* Binnen het bereik van de variabele moet de naam uniek zijn.
* De naam mag niet een van de gereserveerde woorden van Visual C# zijn.
* C# is hoofdlettergevoelig : voornaam, VOORNAAM, Voornaam,… zijn 3 verschillende identifiers!!
* camelCasing -> variabelen, parameters, ...
* PascalCasing -> Functies, methodes, ...

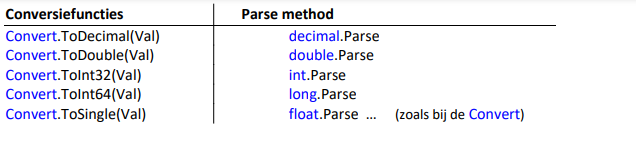
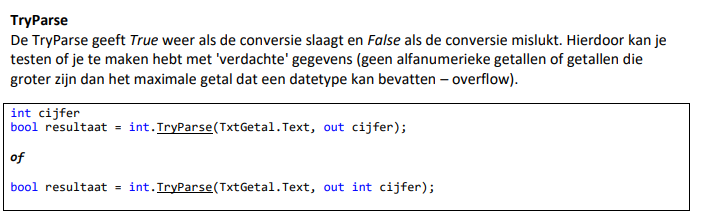


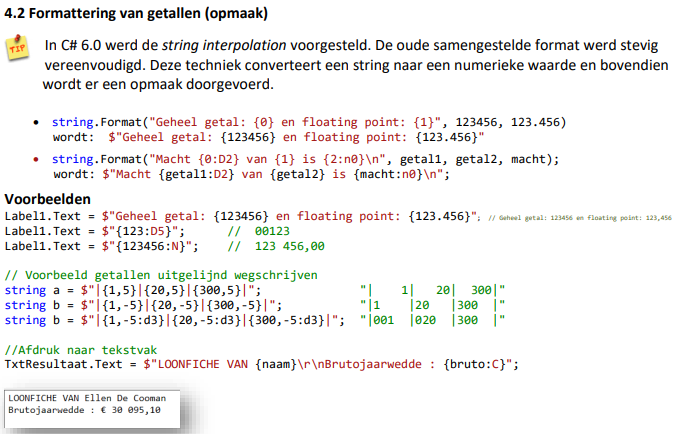
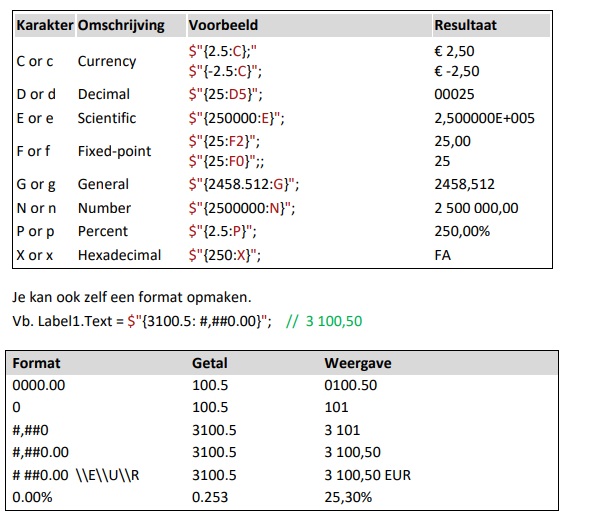
# Hoofdstuk 4: Convensies











# Hoofdstuk 5: Controlestructuren

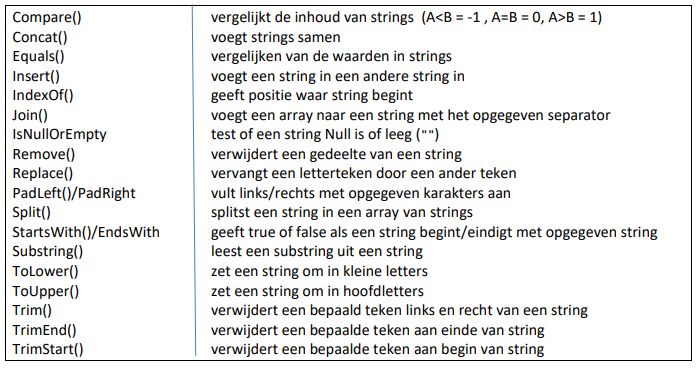
|  |  |
| --- | --- |
| (Geneste) Ifs  If() {  }  else if () {  } |  |
| (Ternary) if |  |
| Switch  Switch() {  Case:  Break;  Default:  Break;  } |  |
| For-loop  For (int i = 0; i < max; i++ {  } |  |
| While-loop |  |
| Do – while loop |  |

# Hoofdstuk 6: Veel gebruike klassen

**System.math**

* Math.pi -> Pi
* Math.e -> E
* Math.abs -> absolute waarde
* Math.min -> minimum waarde
* Math.max -> maximum waarde
* Math.round -> afronden
* Math.ceiling -> afronden naar boven
* Math.floor -> afronden naar beneden
* Math.pow -> macht
* Math.log -> logaritme
* Math.sqrt -> vierkantswortel

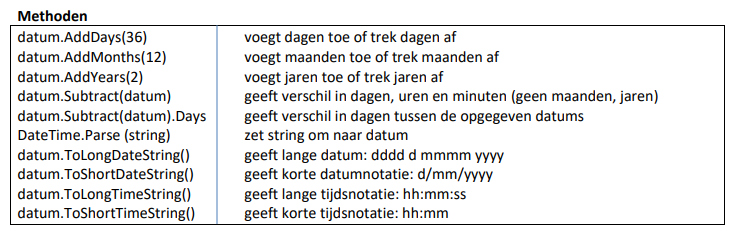
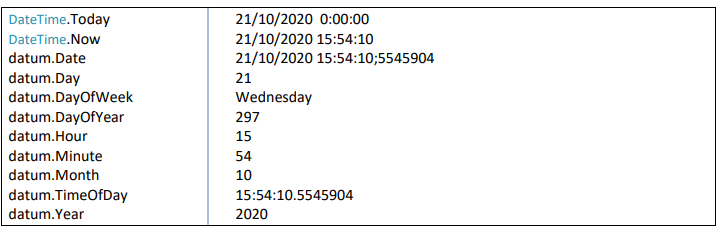
**System.string**



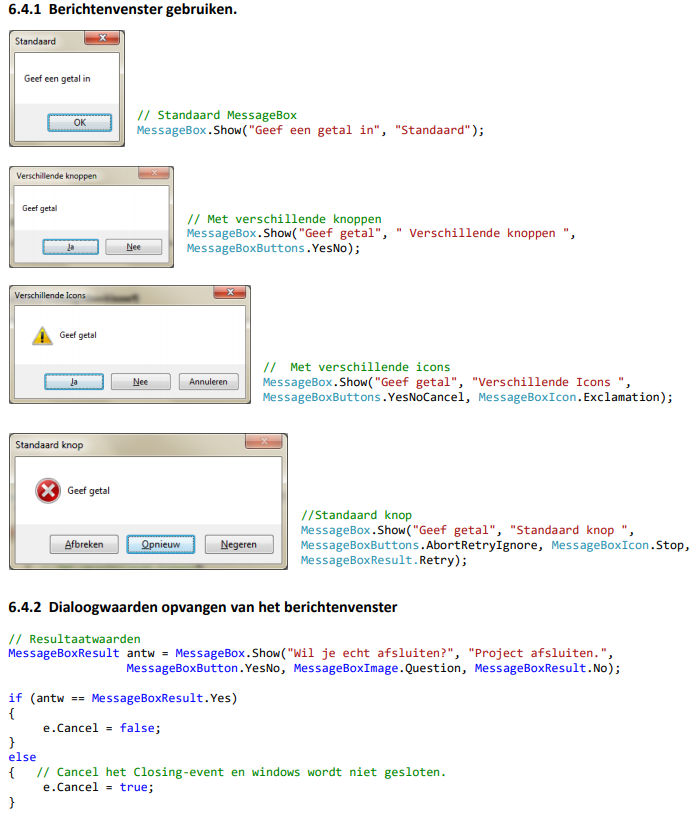




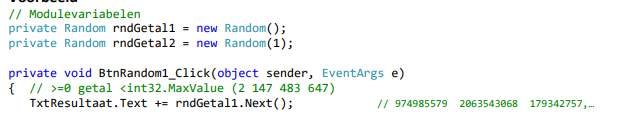
System.dateTime



**Visual basic interaction klasse**

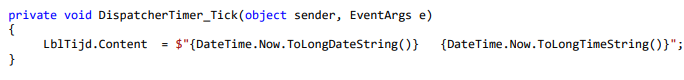
**Berichtvenster**

**Random klasse**



**TimerDispatcher klasse**





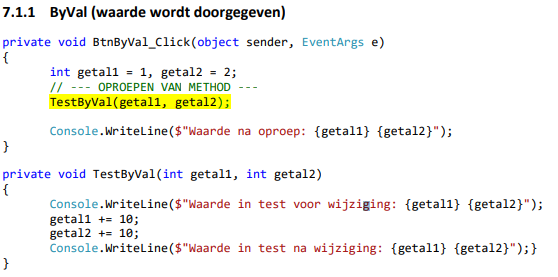
**System.char**

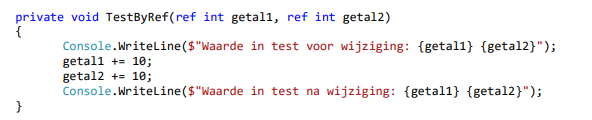
# Hoofdstuk 7: Methoden en parameters

3 verschillende soorten methoden/procedures

* Eventprocedures: click(), keydown(), ...
* Voidprocedures: Messagebox, focus, ...
* Functieprocedures: round(), int.parse(), ..

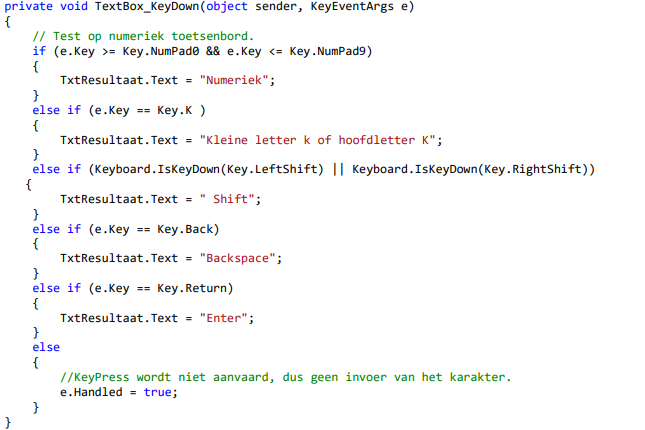
Method overloading: methoden met dezelfde naam maar andere parameters

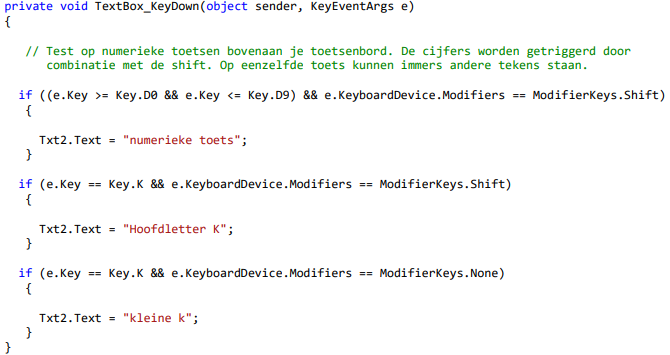


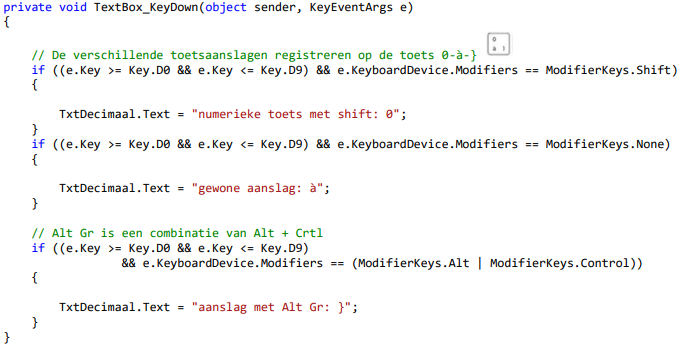


# Hoofdstuk 8: Eventprocedures en parameters

Keydown event

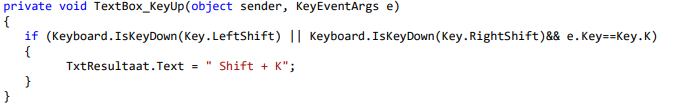






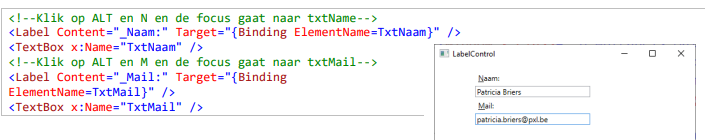
KeyUp event:

Voor 2 knoppen tegelijk te testen gebruik je de keyup methode aangezien je niet 2 dezelfde toetsen kan gebruiken binnen 1 seconde. Zo registreer je welke keys losgelaten zijn.



# Hoofdstuk 9: WPF Besturingselementen

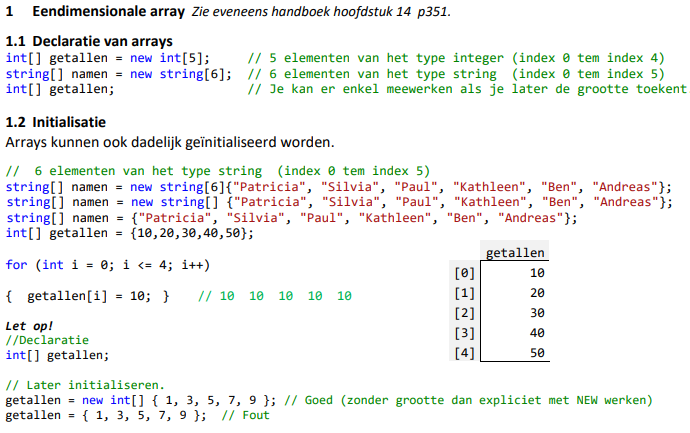


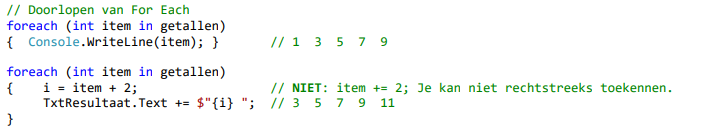
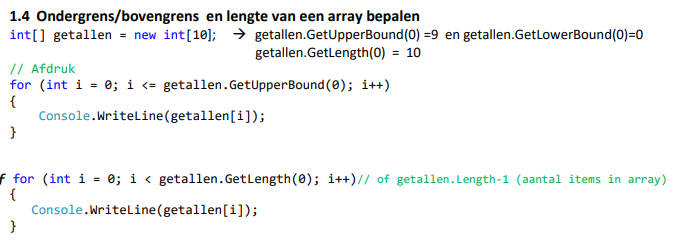


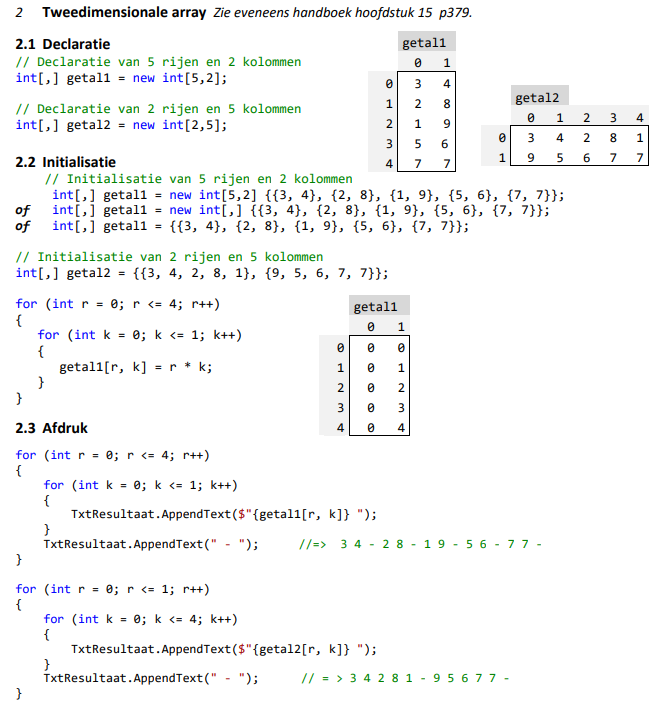




# Hoofdstuk 10: Gegevens beheren met een array







short[] getallen = new short[10];

string getalInput;

for (int i = 0; i <= 9; i++)

getalInput = Interaction.InputBox("Geef een cijfer tussen 0 en 10: ");

getallen[i] = short.Parse(getalInput);

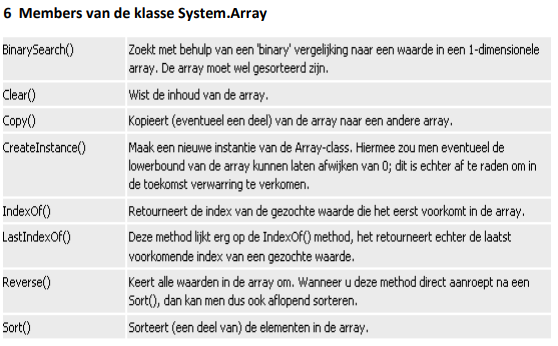
int intSom = 0;

foreach (short getal in arrGetal)

intSom += getal;

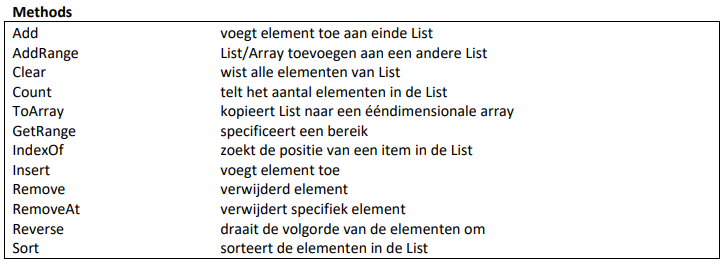
foreach (short getal in arrGetal)

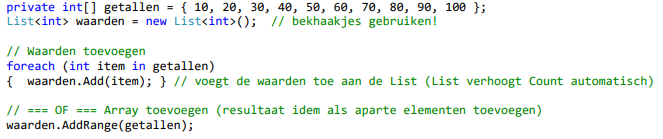
lstCijfer.Items.Add(getal);



# Hoofdstuk 11: Generieke listcollectie

Using System.Collections.Generic





## Dictionary

Een sleutelpaar van 2 elementen

Dictionary punten = new Dictionary();

punten.Add("Anse", 17);

punten.Add("Lise", 12);

punten.Add("Bram", 16);

foreach (var item in punten)

{

Console.WriteLine($"{item.Key}: {item.Value}/20");

}

// Of een element opzoeken met de TryGetValue().

punten.TryGetValue(naam, out int resultaat);

// Onderzoeken of element bestaat

string resultaat=( punten.ContainsKey(naam))?"niet verwijderd":"verwijderd";