Hand-out

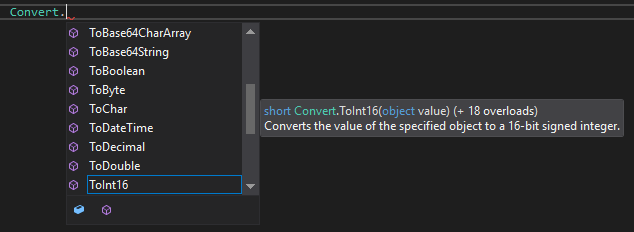
**Programmeren**

03. Variabelen & Datatypes

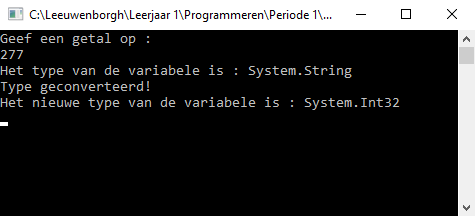
1. **Variabelen**  
     
   Een variabele is een reservering in het geheugen van de computer waar je data in op kunt slaan. Deze data kan van een bepaald type zijn. Een stuk tekst is een ander type dan een getal, en er zijn nog veel meer mogelijkheden.  
     
   Een variabele kun je zien als een doosje waar een label op staat. Dit doosje kun je open doen en je kunt er iets in doen, je kunt hem dicht doen en op de plank leggen, maar ook weer openmaken om iets aan de inhoud toe te voegen, deze te veranderen of zelfs weg gooien zodat het doosje weer leeg is.  
     
   Het label dat op de doos staat is de naam van de variabele, deze is nadat hij is gemaakt niet meer aan te passen. Wel kun je de inhoud van het doosje in een ander doosje plaatsen waar dan weer een ander labeltje op zit.  
     
   Hier volgt een voorbeeld :  
     
   
2. **Naamgeving**  
     
   Een goede naamgeving is essentieel voor je programma, als je variabele aan maakt zorg dan eerst dat je het type ook in de naam verwerkt. Meestal zie je dat gedaan in een afkorting van 3 karakters. Een string word str, een integer int en een boolean bln. Zorg vervolgens dat de naam van je variabele iets zegt over de werking er van. Als je een variabele hebt waarin een naam is opgeslagen zorg dan dat je hem strName noemt. Let hierbij vooral op de manier waarop de hoofdletters gebruikt zijn, dit noemen we camelCase omdat het lijkt op een kamelen bult. We beginnen dus altijd met een kleine letter en bij het eerste nieuwe woord gebruiken we een hoofdletter. Voor classen methodes gebruiken we de PascalCase notatie, in deze notatie beginnen we met een Hoodletter.
3. **Commentaar**Commentaar kan je als programmeur goed helpen, door commentaar in je code te zetten kun je later als je terug moet naar bepaalde onderdelen van je programma altijd een extra beetje uitleg geven over wat de code doet en hoe je er mee om moet gaan. Ook is dit enorm handig als je in een team zit en je collega’s moeten op een dag in jou code werken om een update te schrijven of een aanpassing te maken, dan is het handig als er een extra stukje uitleg bij de code staat. Commentaar pas je als volgt toe :  
     
     
     
     
     
     
     
   Als je net een nieuw programma begint is het vaak lastig om te bedenken waar je moet beginnen. Blijf niet te lang in deze gedachte hangen want er is een oplossing voor, de rede dat dit niet lukt is omdat je het programma eerst moet gaan indelen in kleine stapjes of hapklare brokjes. Deze stapjes moet je in principe een voor een kunnen uitvoeren en tegen de tijd dat alle stapjes zijn uitgevoerd zou je programma moeten werken.   
     
   Probeer voor je ook maar code schrijft voor je programma een blok aan te maken voor je commentaar en ga dan in simpele zinnen punt voor punt de stappen uit schrijven die nodig zijn om tot het eindresultaat te komen. Dit noemen we pseudo code en kan er bijvoorbeeld zo uitzien :  
     
   
4. **Datatypes**  
     
   De data die een variabele vasthoud is van een bepaald type, dit type bepaald wat voor soort data er kan worden opgeslagen in de variabele. Een nummer of getal is van een nader type dan een stuk tekst en hebben dus ook een ander type.  
     
   Zoals we in de vorige hand-out hebben geleerd is het type variabele van een stuk tekst een string. Een getal is van het type int of integer.  
     
   Hieronder vind je een lijst van de meest voorkomende datatypes, het is handig als je deze lijst de komende tijd regelmatig doorneemt zodat je de datatypes straks goed kent.

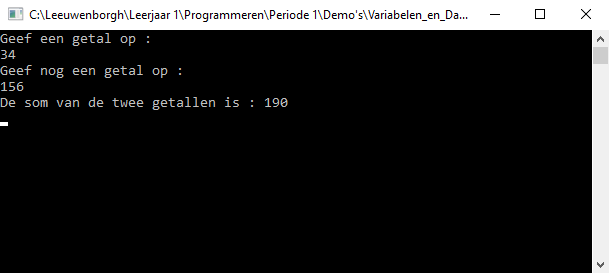
|  |  |  |
| --- | --- | --- |
| **DATATYPES** | | |
| Type | Omschrijving | Voorbeeld |
| Boolean | Waar of niet waar, 1 of 0 | bool blnFoo = true; |
| Byte | Bit waarde | byte bytFoo = 5; |
| Short | Kleiner getal | short shrFoo = 32; |
| Integer | Getal | int intFoo = 255; |
| Long | Groter getal | long lngFoo = 302546; |
| Character | Karakter | char chrFoo = 'v'; |
| Float | Getal met een punt en decimalen | float fltFoo = 1.001F; |
| Double | Getal met een punt en decimalen | double dblFoo = 23.5; |
| Decimal | Getal met een punt en decimalen | decimal dcmFoo = 44.1M; |
| Object | Basis type voor alle andere types | object objFoo = new Object(); |
| String | Een of meerdere karakters, tekst | string strFoo = "een string"; |
| DateTime | Datum en tijd | DateTime dtmFoo = new DateTime(); |

1. **Convertors**Binnen ons programma kunnen we datatypes converteren via de Convert class. De Convert class is onderdeel van de System namespace en bevat een aantal functies om bijvoorbeeld een stuk tekst om te zetten naar een getal.   
     
   Hier volgt een voorbeeld :



1. **Demo**In dit stuk gaan we klassikaal een demo maken. Het is de bedoeling dat je mee doet en zo goed te weten komt hoe we variabelen definiëren en er data aan toe kennen , daarna ga je zelf aan de slag dus let goed op!



1. **Opdracht**  
     
   Maak een programma dat vraagt twee keer vraagt om een getal in te voeren, sla deze getallen beide op in een eigen variabele. Laat vervolgens de som van de twee getallen op het scherm zien door deze bij elkaar op te tellen als volgt :   
     
   “De som van de getallen is <som>”  
     
   De woorden tussen de haakjes moeten natuurlijk vervangen worden met de som van de twee getallen. Als je klaar bent met deze opdracht ga je je werk samen met de docent bekijken. veel succes!  
     
    
2. **Uitwerking**  
   Demo code :   
     
   

De uitwerking van de opdracht volgt de volgende les!