

Oefeningenlijst Gestructureerd programmeren

ICT 1

Bachelor T elektronica-ict

academiejaar 2010 - 2011
semester 1

Ing. Tim Dams
tim.dams@artesis.be
T: +32 486181475

Oefeningenlijst Gestructureerd programmeren 1

ICT 1 1

1	Inleiding	3
2	Denkvraagjes	4
3	Zoek-de-fout	6
4	Schrijf een algoritme	7
5	Vul de code aan	9
6	Praktische oefeningen	10
7	Maak een flowchart	11
8	Casting oefeningen	12

1 Inleiding

Dit document bevat een arsenaal aan oefeningen waarmee je je programmeerkennis kan testen en bijschaven. Dit zijn de soort oefeningen die ook op het schriftelijk examen zullen gesteld worden. Het is dus belangrijk dat je de oefeningen kan oplossen op papier, zonder de hulp van Visual Studio of andere bronnen!

Het is dan ook aan te raden dat je deze oefeningen ook op die manier oplost. De 'zoek-de-fout' oefeningen bijvoorbeeld zijn zeer eenvoudig oplosbaar met een Visual Studio bij de hand, maar je zal ontdekken dat dit niet het geval is wanneer je op papier moet werken en dus zelfstandig moet nadenken over de syntax en logica van de code.

Indien bepaalde oefeningen niet duidelijk zijn, fouten bevatten (en niet bij de zoek-de-fout-categorie horen), etc. aarzel dan niet om dit te melden.

Veel oplosplezier gewenst!

2 Denkvraagjes

1. Welke van de volgende identifiers zijn niet geldig in C#, verklaar telkens waarom
 - a. RESULT
 - b. Result
 - c. 12345
 - d. X12345y
 - e. black&white
 - f. answer_7
 - g. Factorial
 - h. anExtremelyLongIdentifierIfYouAskMe
 - i. 2ndLevel
 - j. level2
 - k. MAX_SIZE
 - l. highest\$
 - m. hook&ladder
2. Geef voorbeelden van de 2 manieren waarop je commentaar in C#-code kan plaatsen. Wat is het verschil tussen beide?
3. Waarom zijn volgende geldige identifiers niet bepaald goed om te gebruiken?
 - a. q
 - b. totVal
 - c. theNextValueInTheList
4. Wat is een literal. Geef een voorbeeld van een dubbele, decimaal, float, char, string en int literal.
5. Wat is het verschil tussen Write() en de WriteLine() methode.
6. Wat is een 'escape sequence', geef hier een voorbeeld van.
7. Welke types ken je om gehele getallen in te bewaren. Waarin verschillen ze?
8. Wat is het resultaat van de expressie $19\%5$? Verklaar dit.
9. Waarom is een widening conversie veiliger dan een narrowing conversie?
10. Welke output zal volgende stuk code genereren? Verklaar:

```
Console.Write("Here we go!");
Console.WriteLine("12345");
Console.WriteLine("Test this if you are not sure.");
Console.Write("Another.");
Console.WriteLine();
Console.WriteLine("\nAll done");
```
11. Wat is er verkeerd met de volgende lijn code? Hoe kan je dit oplossen?

```
Console.WriteLine ("To be or not to
be");
```
12. Welke output wordt gegenereerd door volgende lijn code? Verklaar

```
Console.WriteLine("50 plus 25 is " + 50 + 25);
```
13. Welke output wordt gegenereerd door volgende lijn code? Verklaar

```
Console.WriteLine("He thrusts his fists\n\tagainst" + " the post\n
and still insists\n\tthe sees the\"ghost\");
```
14. Gegeven de volgende declaraties, welke resultaat wordt bewaard in elke van de getoonde statements?

```
int iResult, num1 = 25, num2=40, num3= 17, num4 =5;
double fResult, val1 = 17.0, val2 = 12.78;
```

 - a. `iResult = num1/num4;`
 - b. `fResult = num1/num4;`
 - c. `iResult = num3/num4;`

- d. `fResult = num3/num4;`
- e. `fResult = val1/num4;`
- f. `fResult = val1/val2;`
- g. `iResult = num1/num2;`
- h. `fResult = (double)num1/num4;`
- i. `fResult = num1/(double)num2;`
- j. `fResult = (double)(num1/num2);`
- k. `iResult = (int)(val1/num4);`
- l. `fResult = (int)(val1/num4);`
- m. `fResult = (int)((double)num1/num2);`
- n. `iResult = num3%num4;`
- o. `iResult = num2%num3;`
- p. `iResult = num3%num2;`
- q. `iResult = num2%num4;`

r.

3 Zoek-de-fout

Deze soort oefening schotelt steeds een stuk code aan. De code echter 2 soorten fouten:

- 1) Codesyntax-fouten: dus fouten zoals spellingfouten in de variabele namen, verkeerd gebruik van keywords, vergeten van komma's, etc. Kortom, fouten die ervoor zullen zorgen dat je code zelfs niet gecompileerd kan worden door een compiler.
- 2) Logicafouten: dus fouten zoals verkeerde grenzen in een if-statement, verkeerde uitvoering van bepaalde bewerkingen. Kortom fouten die wel een compileerbaar programma zullen geven, maar waarvan de uitvoer niet correct zal zijn.

```
//1. Declaratie van verschillende variabelen (2 fouten)
int getal = 1;
double getal2 = 4.3;
float getal3 = 5.222;
ushort getal4 = 16;
ushort getal5 = -16;

//2. Casting van verschillende variabelen (2 fouten)
int getal;
double getal2;
getal = 4;
getal2 = getal;
getal = getal2;
getal2 = getal2 + getal;
getal = getal*getal2;

//3. Gelijkheid van 2 getallen testen (1 fout)
int getal1 = 2;
int getal2 = 4;
if(getal1=getal2)
    Console.WriteLine("Ze zijn gelijk");
else
    Console.WriteLine("Ze zijn niet gelijk");

//4. Grootste getal van de twee tonen, indien gelijk boodschap scherm
//(6 fouten)
int getal = 4;
int getal2 = 5.4;
if(getal1 > getal 2)
    Console.WriteLine(getal2);
else if(getal < getal2) ;
    Console.WriteLine(getal);
else(getal=getal2)
    Console.WriteLine("beide gelijk");

//5. Uitlezen en tonen van de naam van een gebruiker (4 fouten)
string name=Console.ReadLine("Gelieve uw naam in te voeren")
Console.WriteLine();
Console.WriteLine("Uw naam is:{1}",name);
```

4 Schrijf een algoritme

Dit soort oefeningen zal steeds uitgaan van een stukje geschreven tekst dat beschrijft wat je algoritme moet doen. Probeer in zo weinig mogelijk code een correct algoritme te beschrijven. Je algoritme moet enkel doen wat er beschreven staat: dus niet meer (en ook niet minder).

- 1) Maak een variabele Name aan van het type string. Wijs aan deze variabele de waarde "Jef Vermassen" toe.
- 2) Maak 3 variabelen Mik, Mak en Mon van het type short. Wijs aan Mik de waarde 1 toe, aan Mak de waarde 2 en aan Mon de waarde 3
- 3) Maak 3 variabelen Mik, Mak en Mon van het type short. Wijs aan Mik de waarde 1 toe, aan Mak de waarde 2. Wijs aan Mon de som van Mik en Mak toe.
- 4) Schrijf een algoritme dat aan de gebruiker 2 getallen vraagt om in te voeren. Toon vervolgens op het scherm het product van beide ingevoerde getallen. Je moet niet controleren op juiste invoer.
- 5) Schrijf een algoritme dat aan de gebruiker 2 getallen vraagt die groter dan 0 en kleiner dan 100 zijn, om in te voeren. Toon vervolgens op het scherm het product van beide ingevoerde getallen. Indien de gebruiker een verkeerd getal invoert toon je op het scherm 'foute invoer' en vraag je de gebruiker om nogmaals te proberen. Blijf dit doen tot je 2 getallen hebt verkregen die binnen de grenzen liggen.
- 6) Schrijf een programma dat de gebruiker om z'n naam vraagt. Vervolgens groet het programma dat gebruiker met zijn naam (bv "Dag Tim Dams", hoe gaat het ermee?")
- 7) Vul het programma aan als volgt: Na de groet kan de gebruiker goed of slecht invullen. Naargelang de invoer zal de computer zeggen: bij goed: "Dat is leuk om horen", bij slecht: "Oei, hopelijk gaat het morgen beter".
- 8) Toon alle natuurlijke getallen van 1 tot 100 onder mekaar op het scherm, gebruikmakende van een while.
- 9) Toon alle gehele getallen van -100 tot 100 onder mekaar op het scherm, gebruikmakende van een while.
- 10) Toon alle natuurlijke getallen van 1 tot 100 onder mekaar op het scherm, gebruikmakende van een for.
- 11) Toon alle gehele getallen van -100 tot 100 onder mekaar op het scherm, gebruikmakende van een for.
- 12) Toon alle gehele getallen tussen 1 en 100 onder mekaar op het scherm die een veelvoud van 6 en 8 zijn. Gebruik een while.
- 13) Toon alle getallen tussen 1 en 100 onder mekaar op het scherm die een veelvoud van 6 en 8 zijn. Gebruik een for.
- 14) Schrijf een methode die twee gehele getallen als invoer krijgt en vervolgens een bool teruggeeft die true is indien beide getallen even groot zijn, false indien ze dat niet zijn.
- 15) Schrijf een methode die twee doubles als invoer krijgt en deze vervolgens onderling verwisselt. De uitvoer zijn de 2 omgewisselde doubles.
- 16) Schrijf een methode die het volume van een kubus berekend (opp kubus= zijde*zijde*zijde) . Als parameter wordt de zijde meegegeven en het resultaat van de methode is de oppervlakte van de kubus.
- 17) Schrijf een methode die het volume van een kubus berekend (opp kubus= zijde*zijde*zijde) . Als parameter wordt de zijde meegegeven, die niet negatief mag zijn, en het resultaat van de methode is de oppervlakte van de kubus.
- 18) Schrijf een methode 'RolDobbelSteen' die telkens een willekeurig getal tussen 1 en 6 teruggeeft (gebruik Math.Random(!))
- 19) Gebruik de methode RolDobbelsteen om een programma te schrijven dat 3 keer met de dobbelsteen rolt en de som van de 3 rols optelt en toont.
- 20) Gebruik de RolDobbelsteen methode om te testen hoe vaak 2 dobbelstenen samen moeten gerold worden (gemiddeld gezien) tot er een snake eyes wordt getoont (= beide dobbelstaan 1 als resultaat).

Het programma toont het aantal keer dat moest gerold worden met beide stenen. Telkens je het programma uitvoert zal je dus een andere uitvoer mogelijk hebben.

- 21) Schrijf een programma dat aan de gebruiker het aantal 1-cent, 5-cent en 10-cent muntjes vraagt dat die in zijn zak heeft. Nadien toont het programma het totaal aantal euro dat de gebruiker heeft.
- 22) Schrijf een programma dat aan de gebruiker vraagt hoeveel eieren deze heeft. Het programma toont vervolgens hoeveel gros, dozijn en enkele eieren dit zijn (1dozijn = 12 eieren, 1 gros = 12 dozijn). Indien dus de gebruiker bijvoorbeeld 1342 invoert dan verschijnt er op het scherm: "Je hebt 9 gros, 3 dozijn en 10 eieren). (gebruik de modulo-operator!).

5 Vul de code aan

Dit soort oefeningen schotelt een stuk code aan met 'gaten' in. Bedoeling is dat je deze gaten invult met de juiste stukjes code. Dit kunnen 2 soorten zaken zijn:

- 1) Woorden die ontbreken (variabele types, identifiers, keywords, etc)
- 2) Stukjes code: het algoritme zou niet werken zonder aanvulling van 1 of meerdere lijnen code

6 Praktische oefeningen

1. Maak een nieuwe console applicatie genaamd Test en voer de volgende code in binnen de Main-methode:

```
Console.WriteLine("An emergency broadcast");
```

Introduceer nu telkens de volgende error, een per een aan het eerder programma. Noteer welke fouten Visual studio vervolgens rapporteert. Maak de aanpassing ongedaan en voer vervolgens de volgende fout in, enz. Indien er geen fouten door Visual Studio gerapporteerd worden, tracht te verklaren waarom dat zo is. Probeer telkens te voorspellen welke fouten zullen optreden:

- a. Verander `Test` naar `test`
 - b. Verander `Emergency` naar `emergency`
 - c. Verwijder het eerste aanhalingsteken in de string
 - d. Verwijder het laatste aanhalingsteken in de string
 - e. Verander `main` naar `man`
 - f. Verander `WriteLine` naar `bogus`
 - g. Verwijder de punt-komma achtereen de `WriteLine`-statement
 - h. Verwijder de allerlaatste accolade in je programma
2. Schrijf een applicatie dat 3 gehele getallen aan de gebruiker vraagt en vervolgens het gemiddelde daarvan op het scherm toont.

7 Maak een flowchart

Dit soort oefeningen gaat uit van een stuk code dat werkt. Bedoeling is dat je de code omzet naar een flowchart. Om te controleren of je flowchart werkt doe je best vervolgens de omgekeerde operatie. Je zet je flowchart om naar code, zonder naar de originele code van de oefeningen te kijken.

8 Casting oefeningen

Gebruik casting om volgende oefeningen zonder fouten te doen compileren: je mag echter geen Visual Studio gebruiken en dient dus alles op papier op te lossen. Je mag de types bij declaratie niet aanpassen. ENKEL CASTEN.

Oefening 1

```
int var1 = 10;
short var2 = var1;
```

Oefening 2

```
float x;
int j;
x = 3.5F;
j = x;
```

Oefening 3

```
double temperatuur = 18;
int vorigeTemperatuur = 20;
int temperatuursverschil;
temperatuursverschil = temperatuur - vorigeTemperatuur;
```

Oefening 4

```
int temperatuur = 18;
int vorigeTemperatuur = 20;
double temperatuursverschil;
temperatuursverschil = temperatuur - vorigeTemperatuur;
```

Oefening 5

```
int a = 5;
int b = 10;
float c = 15.5F;
int resultaat = a + b + c;
```

Oefening 6

```
int a = 5;
double b = 10;
float c = 15.5F;
short d;
d = c * b;
a = a - d;
```

Oefening 7

```
int a = 5;
double b = 10;
float c = 15.5F;
short d;
d = b * b;
a = (a - d) * c;
```