

4EE11 – Project Programmeren voor W

College 1, 2008–2009, Blok D

Tom Verhoeff, Software Engineering & Technology, TU/e

Betrokkenen

- Circa 100 deelnemers (W, 3e-jaars)
- Joris Remmers (W)
- Patrick Anderson (W)
- Tom Verhoeff (Inf)
- Technische adviseurs (W)
- Studentassistenten (Inf)

Motivatie

- Problemen voor jezelf oplossen met de computer (projecten, afstuderen, aio, etc.), i.p.v. “met de hand doen”
- De meeste producten bevatten software, of ontwikkeling ervan vereist specialistische maatsoftware
- Veelal niet zelf programmeren, maar wel weten wat er speelt uit eigen ervaring

Software is anders

- Niet fysiek
- Kneedbaar (voor- en nadeel)
- Vergt m.n. intellectuele arbeid
- Onvolledig (vereist computerhardware)
- Goedkoop te reproduceren
- Slijt niet (maar ...)
- Makkelijk om ononderhoudbare producten te maken

Software is niet anders

1. Beslissen of software helpt (*business case*)
2. Eisen en domein kennen (*requirements*)
3. Ontwerp bedenken, vastleggen, evalueren
4. Product realiseren (*coding*)
5. Gebruiken, onderhouden, afvoeren
 - Controleren (op vele momenten/manieren)

Wat komt aan bod

- Programmeerconcepten en -jargon
- Programmeertaal
- Programmeergereedschap
- Programmeergewoonten (*best practices*)
- Programmeermethoden en -technieken

Opzet

- In koppels
- Eigen laptop met wat extra software
- Een algemene opgave
- Eén probleem met werktuigbouwkundige achtergrond “oplossen” met de computer
- **Veel zelf uitzoeken en doen**
- Inleveren via peach.win.tue.nl

Programmeertaal

- Syntax: vormleer, wat mag je opschrijven, komt heel precies
- Semantiek: betekenisleer, wat betekent het, dit kan soms lastig te begrijpen zijn
- Pragmatiek: hoe wordt het in de praktijk gebruikt, vele oplossingen voor hetzelfde probleem, welke heeft voorkeur
- Documentatie raadplegen

Concepten

- Abstracte automaat
- (systeem)toestand in (data)variabelen en “opdrachtaanwijzer”
- toestandsverandering via opdrachten
 - data inlezen/uitvoeren
 - nieuwe waarde toekennen aan variabele
 - selectief/herhaald opdrachten uitvoeren

Een C programma

```
#include <stdio.h>
int main(void) {
int n;
scanf("%d",&n);
for(int i=1;i<=10;++i) printf("%d\n",i*n);
return 0;
}
```

- Wat doet dit programma?
- Waar zit 'm dat in?
- Hoe leesbaar is dit?
- Hoe kan het beter?

Standaard geraamte

```
#include <stdio.h>
```

use standard input-output library

```
int main ( void ) {
```

the block with the main program

```
int n;

printf ( "Geef grondtal voor tafel: " );
scanf ( "%d", &n );

for ( int i = 1; i <= 10; ++i ) {
    printf ( "%2d x %2d = %4d\n", i, n, i * n );
}
```

report that everything ends OK

```
return 0;
```

```
}
```

Betekenis

```
#include <stdio.h>

int main ( void ) {
    int n; // declareer integer variabele genaamd n

    printf ( "Geef grondtal voor tafel: " ); // druk af
    scanf ( "%d", &n ); // lees invoerwaarde voor n

    for ( int i = 1; i <= 10; ++i ) { // herhaal 10 keer
        printf ( "%2d x %2d = %4d\n", i, n, i * n );
                format string          formule
    }

    return 0;
}
```

Nog beter

```
/*
 * (c) 2009, Tom Verhoeff (SET, TUE)
 * Voorbeeldprogramma dat tafel van n (invoer) afdrukt.
 */

#include <stdio.h>

int main ( void ) {
    int grondtal; // tafelgrondtal (invoer)

    printf ( "Geef grondtal voor tafel: " );
    scanf ("%d", &grondtal);

    for ( int regel = 1; regel <= 10; ++regel ) {
        int uitkomst = regel * grondtal; // uitvoer
        printf ( "%2d x %2d = %4d\n", regel, grondtal, uitkomst );
    }

    return 0;
}
```

aanhef met datum, id.nr., auteur, doel

zinnige namen met toelichting

hulpvariabele

In FORTRAN95

```
PROGRAM tafelgenerator
```

```
! (c) 2009, Tom Verhoeff (SET, TUE)
```

```
!
```

```
! Voorbeeldprogramma dat tafel van n (invoer) afdrukt.
```

```
INTEGER :: n ! tafelgrondtal (invoer)
```

```
INTEGER :: i ! doorloopt regels van tafel
```

```
WRITE (*,*) 'Geef grondtal voor tafel: '
```

```
READ (*,*) n
```

```
DO i = 1, 10
```

```
    WRITE (*,100) i, n, i * n
```

```
    100 FORMAT (I2, ' x ', I2, ' = ', I4)
```

```
END DO
```

```
STOP
```

```
END PROGRAM tafelgenerator
```

Nog beter

```
INTEGER :: grondtal ! voor tafel (invoer)
INTEGER :: regel ! doorloopt regels van tafel
INTEGER :: uitkomst ! product (uitvoer)

WRITE (*,*) 'Geef grondtal voor tafel: '
READ (*,*) grondtal

DO regel = 1, 10
  uitkomst = regel * grondtal
  WRITE (*,100) regel, grondtal, uitkomst
  100 FORMAT (I2, ' x ', I2, ' = ', I4)
END DO
```

Programmatekst is communicatiemiddel

- Legt vast hoe automaat zich moet gedragen
- Moet door andere mensen begrepen kunnen worden, m.n. ter controle

Taalelementen

- Sleutelwoorden
- Symbolen (één of meer tekens): != voor ≠
- Wit: inspringen, spatiëring, nieuwe regel
- Commentaar
- Namen (identifiers): **case sensitive**
- Letterlijke waarden (getallen, strings)

Omgaan met data

- Variabele: heeft op elk moment één waarde
- Type: bepaalt verzameling mogelijk waarden
- Constante: waarde is onveranderbaar
- Declaratie: introduceert variabele met type
- Expressie, uitdrukking: formule opgebouwd uit letterlijke waarden, variabelen, constanten, operaties, functies, haakjes

Standaard types, values

- bool, int, float, double, char
- true, false

- LOGICAL, INTEGER, REAL, CHARACTER
- .TRUE., .FALSE.

Basisopdrachten

- Toekenning: wijzigt waarde van variabele
- Inlezen: geeft waarde aan variabele
- Wegschrijven: evalueert formule
- Selectie: keuze uit één of meer opdrachten
- Herhaling: vast of variabel aantal keer
- Samenstelling: rij opdrachten als één blok

Toekenning

- Wijzigt waarde van variabele o.g.v. formule
- *variabele = formule; // C*
- *variabele = formule ! Fortran*
- Vertrouw er niet op dat een variabele na declaratie vanzelf een goede waarde heeft!
Initialiseer variabelen expliciet.
- *i = i + 1; // i rechts betreft 'oude' waarde*

Inlezen

- Van stdin (standaard invoer) of file
- Er is een type-afhankelijke conversie nodig van tekst naar waarde uit het type
- `scanf (format_string, & variabele, ...); // C`
- `READ (*,*) variabele, ... ! Fortran`

Wegschrijven

- Naar stdout (standaard uitvoer) of file
- Er is een type-afhankelijke conversie nodig van waarde uit het type naar tekst
- Formattering is te sturen
- `printf (format_string, formule, ...); // C`
- `WRITE (*,*) formule ! Fortran`

Selectie

```
if ( conditie ) {  
    opdrachten  
}
```

```
if ( conditie_A ) {  
    opdrachten_1  
} else { // conditie_A geldt niet  
    opdrachten_2  
}
```

```
if ( conditie_A ) {  
    opdrachten_1  
} else if ( conditie_B ) { // conditie_A geldt niet, conditie_B wel  
    opdrachten_2  
} else { // conditie_A en conditie_B gelden beide niet  
    opdrachten_3  
}
```

```
if ( conditie_B ) {  
    opdrachten_2  
} else if ( conditie_A ) { // conditie_B geldt niet, conditie_A wel  
    opdrachten_1  
} else { // conditie_A en conditie_B gelden beide niet  
    opdrachten_3  
}
```


Selectie

```
if ( conditie_A ) {  
    opdrachten_1  
} else { // conditie_A geldt niet  
    opdrachten_2  
}
```

// versus

```
if ( conditie_A ) {  
    opdrachten_1  
}  
if ( ! conditie_A ) {  
    opdrachten_2  
}
```

Selectie

```
#include <stdbool.h>

...

bool conditie; // conditie ...

conditie = formule; // bepaal conditie
if ( conditie ) {
    opdrachten_1
} else { // conditie geldt niet
    opdrachten_2
}

//=====

bool v; // ...

if ( conditie ) {
    v = true;
} else {
    v = false;
}

versus

v = conditie
```

Herhaling

```
// vast aantal keer  
for ( variabele = formule; stopconditie; opdracht_volgende ) {  
    opdrachten  
}
```

```
// nul of meer keer  
while ( conditie ) {  
    opdrachten  
}
```

```
// één of meer keer  
do {  
    opdrachten  
} while ( conditie )
```

C Gereedschap

- Pelles C: www.smorgasbordet.com/pellec
- Download & execute: `Setup, 32-bit edition`
- `poide.exe` in Program Files\Pelles C
- Ingebouwde helpvoorziening
- Documentatie: www2.its.strath.ac.uk/courses/c/
- Naslag: www.acm.uiuc.edu/webmonkeys/book/c_guide/

Fortran gereedschap

- **Silverfrost FTN95** w3.tue.nl/nl/diensten/dienst_ict/organisatie/groepen/wins/campus_software/salford_fortran/
- **Boek: *Fortran 90/95 for Scientists and Engineers*** door Stephen J. Chapman (lenen)

Werkwijze

- Lees en analyseer het probleem
- Ontwerp oplossing
- Edit programmateksten; altijd netjes werken
- Review (nalezen)
- Compile (vertalen, syntaxfouten)
- Build (samenvoegen)
- Execute (uitvoeren: test, debug, use)

Gereedschap (algemeen)

- IDE: Integrated Development Environment
- Editor toont/bewerkt programmateksten, liefst met **syntax highlighting**
- Compiler vertaalt ze naar machinetaal
- Linker voegt alles samen tot executable
- Debugger voert uit met vinger aan de pols

Omgaan met fouten

- Erkennen dat ze gemaakt worden
- Zoveel mogelijk voorkomen: **netjes werken**
- Zo snel mogelijk zoeken: detecteren, localiseren, elimineren
- **Nalezen** en **testen**
- Debugging: diagnostiseren, localiseren; **tijdrovend en onvoorspelbaar proces**

Codeerstandaard

- Opmaak: inspringen, spaties, lege regels
- Aanhef met datum, id.nr., auteurs, doel
- Variabelen bij declaratie toelichten
- Zinvolle namen geven
- Maximaal één opdracht per regel

Waarom standaarden

- Je maakt zo minder fouten
- Als je ze toch maakt, zie je ze sneller en zijn ze makkelijker te verbeteren
- Als je ze zelf niet kan vinden, dan kunnen anderen beter helpen
- In geval van een rechtzaak sta je sterker

Soorten fouten

- Syntaxfouten: compiler detecteert, maar
 - feitelijke fout kan elders zitten
 - één fout kan tot meer meldingen leiden
- Run-time fouten, “**crash**”, bijv. delen door 0
- Afwijking van requirements:
 - verkeerde uitkomst
 - verkeerde uitvoerformaat
 - onnauwkeurig, te langzaam

Veelgemaakte C fouten

- Vergeten of extra puntkomma
- **if** (i = 0) i.p.v. **if** (i == 0)
- geen accolades na **for** (...) e.d.

Projectopdrachten

1	Chaotisch mengen	Fortran	Hulsen, Anderson
2	Multi-particle dynamics I	C	Nedea
3	Stroming en warmte	Fortran	Kuerten
4	Systems Engineering	C	Hofkamp
5	Audio sequencer	C	van de Molengraft
6	Multi-particle dynamics II	C	Remmers
7	Test environment for constitutive models	C	Hirschberger

peach.win.tue.nl

- **Registreren** (emailadres vereist)
- **Aanmelden**
- **Koppelen** aan 4EEI I in juiste cluster
- Werk **inzenden** als koppel bij juiste opgave
- Werk, terugkoppeling, resultaat inzien