

## 2IP05: Programmeren – Blok A

<http://www.win.tue.nl/~wstomv/edu/2ip05/>

### College 1

Tom Verhoeff

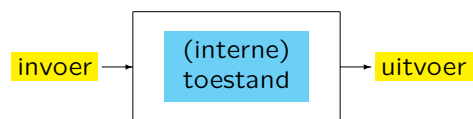
Technische Universiteit Eindhoven  
Faculteit Wiskunde en Informatica  
Software Engineering & Technology

Opmerkingen en vragen aan T.Verhoeff@TUE.NL

## Probleem met dobbelspel

- 5 spelers, 2 dobbelstenen
- Per ronde werpt elke speler 1x
- Speler met hoogste worp wint ronde, mits uniek (anders geen winnaar).
- Speler 1 verzoekt om een dodecaeder te mogen werpen met vlakken genummerd van 1 t/m 12.
- Uitieme vraag: Hoe beïnvloedt dit de winstkansen?

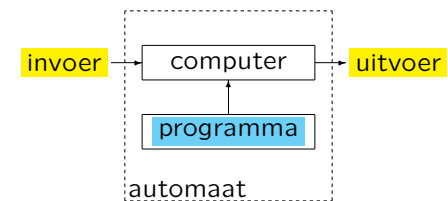
## Abstracte automaat



“Berekenings” stappen:

- Invoer lezen
- Toestand veranderen
- Uitvoer schrijven

## Computer als universele automaat



- Automaat = Programma

## Voorbeeld Pascal programma

```
program Maximum;  
  { Schrijf van twee ingelezen getallen het maximum }  
  
var  
  a,b: Integer; { invoer }  
  m: Integer; { uitvoer }  
  
begin  
  readln ( a, b )      ← getallen invoeren met spatie ertussen  
; if a < b then  
  m := b  
else  
  m := a  
; writeln ( m )  
end.
```

## Levensloop van een programma

1. **Probleem**, toepassing
2. **Specificatie**, eisenpakket
3. **Ontwerp**, oplossing
4. **Productie**, coderen, intikken
5. **Controle**, testen
6. **Gebruik en onderhoud**

## Specificatie van een dobbelprogramma

- **Lees** het aantal te simuleren rondes in: natuurlijk getal  $n$
- **Schrijf** hoe vaak elk der spelers heeft gewonnen wanneer  $n$  rondes van het spel gesimuleerd worden

In het algemeen betreft een programmaspecificatie:

- **invoer**
- **uitvoer**, m.n. ook verband met de invoer
- eventuele overige **restricties**

## Programmeren Blok A

- Taalkennis: Programmeertaal **Object Pascal**
- 'Tool'kennis: Programmeerpakket **Lazarus**
- Vaardigheden: **Software realiseren als ingenieursproduct**
- Terugkoppeling: Programma's inzenden via **peach<sup>3</sup>**
- Afrekening: 2 toetsen en tentamen, 3x met laptop  
Eindcijfer: 25% A + 25% B + 50% C
- Herkansing: tentamen na Blok F

## Alternatief gereedschap

- Borland Delphi xyz, maar . . .
- IDE = Integrated Development Environment versus  
separate editor – compiler – debugger

## Console Application versus GUI Application

- Programmeren Blok A: Console/Text-Filter Applications
- Programmeren Blok B: Opdelen in gespecificeerde routines
- Programmeren Blok C: Graphical User Interface Applications

Motivatie: In- en uitvoer brengen allerlei problemen met zich mee.

In Blok A abstraheren we van de meeste I/O aspecten.

## Voorbeeld Pascal programma

```
program Maximum;  
  { Schrijf van twee ingelezen getallen het maximum }  
  
var  
  a,b: Integer; { invoer }  
  m: Integer; { uitvoer }  
  
begin  
  readln ( a,b )      ← getallen invoeren met spatie ertussen  
; if a < b then  
  m := b  
else  
  m := a  
; writeln ( m )  
end.
```

## Pascal: Structuur van programma

```
program <prograam> ;  
  { <auteur>, <toelichting> }  
  
var { toestandsruimte }  
  <varnaam> : <type> ; { <toelichting> }  
  ...  
  
begin { toestandsveranderingen }  
  <lijst opdrachten>  
end.
```

## Pascal: Namen

Een `<... naam>` is

een rijtje van één of meer hoofd- of kleine letters of cijfers, beginnend met een letter.

Gereserveerde woorden zijn niet toegestaan als naam.

`A` `alpha` `MaxLineLength` `K2`

Niet goed: `2x` `program`

## Pascal: Type (verzameling waarden)

`Integer` Gehele getallen: `...`, `-1`, `0`, `1`, `...`

`Char` Karakters (tekens): `'a'`, `'z'`, `'0'`, `' '`, `'!'`, `'\"'`, `...`

`String` Rijtjes karakters: `'Dit is een string!'`, `...`

`Real` Drijvende-komma getallen: `3.14`, `6.0221415e23`, `...`

`Boolean` Waarheidswaarden: `False` en `True`

## Pascal: Lijst opdrachten

Puntkomma is opeenvolgings-operator:

`<opdracht>` ; `<opdracht>` ; ... ; `<opdracht>`

We schrijven dit meestal als:

`<opdracht>`  
; `<opdracht>`  
; ...  
; `<opdracht>`

of

`<opdracht>` ;  
`<opdracht>` ;  
... ;  
`<opdracht>`

## Pascal: Opdrachten

toekenning: `<varnaam>` := `<formule>`

invoer: `readln` ( `<varnaam>` )

uitvoer: `writeln` ( `<formule>` )

selectie: `if` `<conditie>` `then` `<opdracht>` `else` `<opdracht>`

herhaling: `while` `<conditie>` `do` `<opdracht>`

samenstelling: `begin` `<lijst opdrachten>` `end`

## Pascal: Formule, Conditie

---

**<formule>**: Uitdrukking die tot een waarde geëvalueerd kan worden

**<conditie>**: Formule resulterend in een waarheidswaarde

$a + 1$     $n <> 0$     $(\text{sqr}(r) \leq x) \text{ and } (x < \text{sqr}(r + 1))$

Niet goed:  $\text{sqr}(r) \leq x \text{ and } x < \text{sqr}(r + 1)$     $\text{sqr}(r) \leq x < \text{sqr}(r + 1)$

## Fouten maken

---

- Fout in programmatekst: **compiler**
- Fout tijdens programma-executie: **run-time**
- Fout bij inzenden naar **peach<sup>3</sup>**
- **Programma zelf goed controleren**: alle details tellen!

## Voorzieningen

---

- Web: <http://www.win.tue.nl/~wstomv/edu/2ip05/>
- Lazarus: Getting started en FAQ
- FreePascal: Documentatie, m.n. **Reference Guide**
- Assistenten bij practica en per e-mail

## Weekrooster (cyclisch)

---

- **Dinsdag 1e + 2e uur**: College
- **Donderdag 5e + 6e uur**: Instructie met instructeurs+assistenten
- **Vrijdag 1e + 2e uur**: Practicum met assistenten
- **Zondag 23:00**: Sluiting inzendtermijn (peach<sup>3</sup>)

## Wat te doen vóór het eerste practicum?

---

- Registreren bij peach<sup>3</sup>, koppelen aan juiste groep (ABE of CDF)
- Reeks1.zip ophalen

## Spelregel Nummer 1

---

Dit vak wordt **individueel** afgerekend.

- Zend uw **eigen** werk in.
- Niet dat van een ander.
- peach<sup>3</sup> controleert hierop.

**Plagiaat** wordt niet getolereerd.