

## Algoritmiek 2009

Erik Tjong Kim Sang  
erikt(at)xs4all.nl

7 december 2009

### Vorige week

- abstract window toolkit (rest)
- applets (deel)

7 december 2009

1

### Deze week

- applets (rest)

Literatuur: King: appendix D

Ook gebruikt voor de slides maar hoeft niet te worden gelezen: Mary Campione & Kathy Walrath, *The Java Tutorial*, hoofdstukken 6-10 en 20

7 december 2009

2

### KING APPENDIX D: APPLETS

### Wat is een applet?

Een Java-applet is een programma dat wordt aangeroepen vanuit een webpagina

Java-applets bevatten een hoofdklasse die een extensie is van de klasse `Applet`

Ze worden meestal gebruikt voor interactie en animatie die niet kan worden aangeboden met HTML

Applets hebben beperkte rechten voor zaken als toegang tot bestanden

7 december 2009

4

### Een Java-applet-voorbeeld

Een Java-applet bevat een verzameling van standaardmethods die gedefinieerd kunnen worden:

```

4 public class ComplexApplet extends Applet {
5     private String text = "";
6     public void init() { text += " init"; }
7     public void start() { text += " start"; }
8     public void stop() { text += " stop"; }
9     public void destroy() { System.out.println("destroy"); }
10    public void paint(Graphics g) {
11        g.drawString(text,10,10);
12    }
13 }

```

7 december 2009

5

### Aanroepen van applets vanuit HTML

Applets worden vanuit HTML aangeroepen vanuit een `APPLET`-tag

Argumenten en parameters kunnen via deze tag worden doorgegeven:

```

<APPLET CODE="ConvertTempApplet.class" WIDTH="160" HEIGHT="50">
  <PARAM NAME="startF" VALUE="0">
</APPLET>

```

De HTML-code plaatst de gecompileerde applet in een rechthoek van 160x50 pixels linksboven in de browserwindow

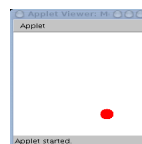
De waarde van de parameter `startF` kan in de applet worden opgevraagd met een standaardmethod: `getParameter("startF")`

7 december 2009

6

### Voorbeeld van een appletanimatie

Plaats een rode cirkel met een diameter van 20 pixels op een willekeurige plek op het scherm. Verplaats de cirkel daarna in stappen van 1 pixel per 50 milliseconden naar de positie (0,0)



7 december 2009

7

## Plan

```
// kies een willekeurige schermpositie
// teken de cirkel
// zolang de cirkelpositie niet gelijk is aan (0,0)
// verlaag x-positie als deze groter is dan 0
// verlaag y-positie als deze groter is dan 0
// teken de cirkel
// wacht 50 milliseconden
```

7 december 2009

8

## Uitwerking in Java (1)

```
// kies een willekeurige schermpositie tussen 100 en 170
private int x = 100 + (int) (Math.random()*71);
private int y = 100 + (int) (Math.random()*71);
// teken de cirkel
public void paint(Graphics g) {
    g.setColor(Color.RED);
    g.fillOval(x,y,20,20);
// zolang de cirkelpositie niet gelijk is aan (0,0)
while (x > 0 || y > 0) {
    // verlaag x-positie als deze groter is dan 0
    if (x > 0) x--;
    // verlaag y-positie als deze groter is dan 0
    if (y > 0) y--;
}
```

7 december 2009

9

## Uitwerking in Java (2)

De herhaalde animatie-instructies staan in de method `run()`:

```
public void run() {
    // zolang de cirkelpositie niet gelijk is aan (0,0)
    // verlaag x-positie als deze groter is dan 0
    // verlaag y-positie als deze groter is dan 0
    // teken de cirkel
    repaint();
    // wacht 50 milliseconden
    try { Thread.sleep(50); }
    catch (InterruptedException e) { break; }
}
}
```

7 december 2009

10

## Threads

De method `run()` draait in een parallel proces, een thread

Deze thread moet in het programma worden gestart:

```
Thread t; // variabele, toegankelijk vanuit alle methods
public void start() {
    t = new Thread(this); // maak nieuwe thread t
    t.start(); // start thread t
}
```

Dit vereist een aanpassing van de klassedefinitie:

```
public class MovingDot extends Applet implements Runnable {
```

7 december 2009

11

## Tips en truuks voor animaties (1)

Het uitrekenen en tekenen van een plaatje kost ook tijd

Hou hiermee rekening bij het bepalen van de exacte wachttijd

```
public void run() {
    // zolang de cirkelpositie niet gelijk is aan (0,0)
    long start = System.currentTimeMillis();
    // bereken en teken plaatje
    ...
    // wacht 50 milliseconden
    int delay = (int) (System.currentTimeMillis()-start);
    try { Thread.sleep(50-delay); }
    catch (InterruptedException e) { break; }
}
```

7 december 2009

12

## Tips en truuks voor animaties (2)

Animaties kunnen erg storend zijn voor een gebruiker

Maak het daarom gemakkelijk voor een gebruiker om een animatie te stoppen, bijvoorbeeld met een muisklik

```
private boolean gestopt = false;
public boolean mouseDown(Event e, int x, int y) {
    gestopt = gestopt ? false : true; // verander waarde gestopt
    return(true);
}
...
while (! gestopt && (x > 0 || y > 0)) {
```

7 december 2009

13

## Tips en truuks voor animaties (3)

De schermopbouw kan zoveel tijd kosten dat de animatie schokkend verloopt

Dit kan worden opgelost door het nieuwe beeld eerst uit te rekenen en dan pas te tekenen

Om dit voor mekaar te krijgen moeten we in plaats van `paint()` de appletmethod `update()` gebruiken

Deze method wordt door `paint()` aangeroepen en maakt steeds het hele scherm leeg

7 december 2009

14

## Apart opbouwen van het scherm in Java

```
// public void paint(Graphics g) { // niet meer nodig

public void update(Graphics g) {
    // definieer nieuw plaatje van dezelfde grootte
    Dimension d = getSize();
    Image newI = createImage(d.width,d.height);
    // haal graphics-element uit plaatje
    Graphics newG = newI.getGraphics();
    // teken het nieuwe plaatje
    newG.setColor(Color.RED);
    newG.fillOval(x,y,DIAMETER,DIAMETER);
    // plaats het in de applet
    g.drawImage(newI,0,0,this);
}
```

7 december 2009

15

## Applet-animaties met plaatjes

<http://www.ira.cnr.it/manuals/tutorial/ui/drawing/imageSequence.html>

<http://www.ira.cnr.it/manuals/tutorial/ui/drawing/movingImage.html>

7 december 2009

16

## TENSLOTTE

## Leesmateriaal voor deze week

K.N. King, *Java Programming from the Beginning*, appendix D

Ook gebruikt voor de slides maar hoeft niet te worden gelezen: Mary Campione & Kathy Walrath, *The Java Tutorial*, hoofdstukken 6-10 en 20

7 december 2009

18

## Leesmateriaal voor het tentamen

K.N. King, *Java Programming from the Beginning*: hoofdstukken 1–12 en appendix D

J. Glenn Brookshear, *Computer Science: An Overview*, hoofdstukken 0, 1, 2, 4, 5.1-5.4, 6 en 7

7 december 2009

19

## Tentamen Algoritmiek

Het tentamen voor het vak Algoritmiek is op maandag 18 januari 2010 in zaal 12 van het Harmoniegebouw

Je kan je vanaf 23 november tot en met 18 december inschrijven voor dit tentamen via <https://progresswww.nl/rug> en daarna klikken op *intekenen*

Als je wil deelnemen aan het tentamen dan moet je je vooraf inschrijven

Bureau Studentzaken accepteert geen cijfers van studenten die niet voor het tentamen zijn ingeschreven

7 december 2009

20

## Oprichtcijfers op Nestor (klik op Details)

Exam Name	Details	Due Date	Last Submitted, Edited or Graded	Grade	Points Possible	Average	Median
opdracht 1	<a href="#">Details</a>	Sep 7, 2009	-	10	0.0	10	
opdracht 2	<a href="#">Details</a>	Sep 14, 2009	-	10	7.66	9.15	
opdracht 3	<a href="#">Details</a>	Sep 21, 2009	-	10	7.66	8.5	
opdracht 4	<a href="#">Details</a>	Sep 28, 2009	-	10	6.93	7.25	
opdracht 5	<a href="#">Details</a>	Oct 5, 2009	-	10	7.66	8.2	
opdracht 6	<a href="#">Details</a>	Oct 12, 2009	-	10	6.69	7.5	
opdracht 7	<a href="#">Details</a>	Oct 26, 2009	-	10	4.27	3.5	
opdracht 8	<a href="#">Details</a>	Nov 2, 2009	-	10	6.18	5	
opdracht 9	<a href="#">Details</a>	Nov 9, 2009	-	10	7.17	7	
opdracht 10	<a href="#">Details</a>	Nov 16, 2009	-	10	3.83	3.55	
opdracht 11	<a href="#">Details</a>	Nov 23, 2009	-	10	5.15	5.5	
opdracht 12	<a href="#">Details</a>	Dec 7, 2009	-	10			
opdracht 13	<a href="#">Details</a>	Dec 14, 2009	-	100	43.056	49%	
total	<a href="#">Details</a>				2		

[View Legend](#)

7 december 2009

21

## Bepaling eindcijfer vak

Het eindcijfer voor het vak is de helft van de som van het opdrachtencijfer (zonder afronding) en het tentamencijfer. Het eindcijfer wordt afgerond op het dichtsbijzijnde gehele cijfer

Voorbeeld: opdrachtencijfer = 4.385; tentamencijfer = 6.7; eindcijfer =  $11.085/2 = 5.5425$ ; afgerond 6

7 december 2009

22

## Na Algoritmiek...

Behandelde onderwerpen: programmeren in Java, variabelen, methods, klassen, controlestructuren, lijsten, tekenen in Java, klassemethods, basisdatatypen, subklassen, de abstract window toolkit en applets

Niet behandeld: meerdimensionale lijsten, omgaan met bestanden, operating systemen en recursief programmeren

Deze laatste onderwerpen komen aan bod in het vak Datastructuren dat in februari begint

7 december 2009

23

**Bachelorstudie Informatiekunde Jaar 1**

Ia	Algoritmiek Tjong Kim Sang	Computer- communicatie Welling		Inleiding Informatiekunde Tjong Kim Sang
Ib			Telst- manipulatie Plank	Inleiding Logica KI
IIa	Datastructuren Bosveld	Webdesign Meindersma		Minor
IIb	Corputaalkunde Bouma & Tjong Kim Sang			

**THE END**