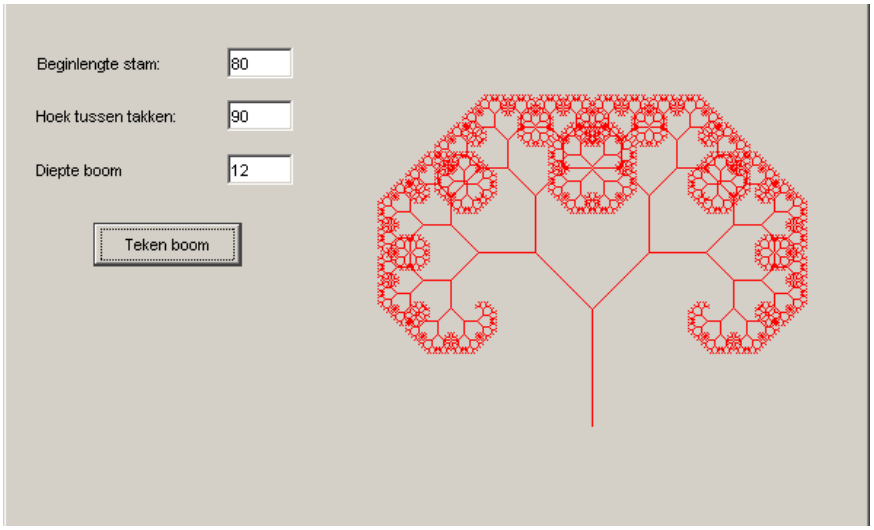
Het tekenen van een boomstructuur

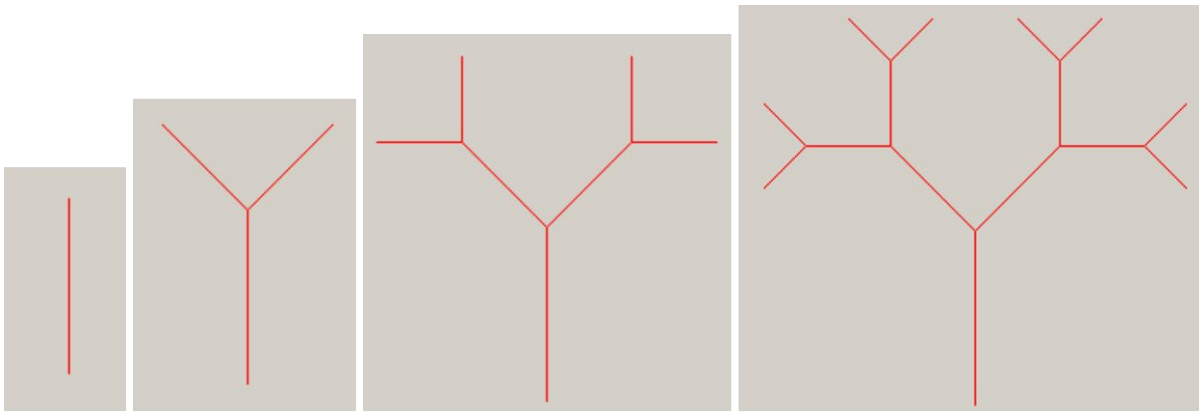
In deze opgave wordt gevraagd een recursieve methode te schrijven die een boomstructuur tekent als getoond in figuur 1.



FIGUUR 1 Een recursieve boomstructuur

De tekening heeft drie parameters, namelijk de diepte van de boom, de hoek tussen de takken, en de lengte van de (onderste) stam.

Om te laten zien hoe de boom is opgebouwd, tonen we in figuur 2 bomen met een diepte van 1 (alleen een stam), 2 (een stam met twee takken), 3 (een stam met twee takken die elk weer twee takken hebben, en 4 (ook die takken hebben weer takken).



FIGUUR 2 Bomen met een diepte van 1, 2, 3 en 4

De takken worden naar boven toe steeds korter, en wel steeds met een factor 0.7.

Bouwstenen

Als bouwstenen voor deze applicatie worden drie klassen meegeleverd.

– Een klasse Pen. In Java wordt normaal gesproken 'absoluut' getekend, dat wil zeggen: als je een lijn tekent, moet je de coördinaten van de eindpunten opgeven. Het alternatief is 'relatief' tekenen, waarbij gebruik gemaakt wordt van een pen die op elk moment op een bepaald punt en in bepaalde richting staat. Door die pen vooruit te sturen en te draaien ontstaat een tekening.
De klasse Pen levert een dergelijke bestuurbare pen. De bijlage toont de documentatie van deze klasse.

– Een klasse BoomFrame, die de gebruikersinterface realiseert. Bij klikken op de Teken-knop worden de waarden in de drie velden ingelezen (zie figuur 1) en worden de volgende controles uitgevoerd:
* diepte moet groter zijn dan 0
* hoek moet groter dan 0 en kleiner dan 360 zijn
* NumberFormatExceptions worden afgevangen
Foutmeldingen verschijnen in een foutlabel onderin het frame.
Als de waarden in orde zijn, wordt een nieuwe instantie van BoomPanel gemaakt en aan het frame toegevoegd (na verwijdering van een eventuele oude).

– Een onvolledige klasse BoomPanel. Deze klasse bevat een constante FACTOR (de verkortingsfactor van de takken), attributen hoek, diepte, stamlengte en pen, een constructor die vanuit het frame wordt aangeroepen, en een methode paintComponent, die aan pen een nieuwe Pen toekent die op het juiste punt begint en tekent in de juiste context.
Wat u zelf nog moet toevoegen, is
* de gevraagde recursieve methode tekenBoom
* een aanroep naar deze methode vanuit paintComponent().

Aanwijzingen
– Voor de duidelijkheid: aan Pen en BoomFrame hoeft niets gewijzigd te worden.
– In figuur 2 kunt u zien, dat een boom van diepte *n* is opgebouwd uit een stam, plus twee bomen van diepte *n*-1. Die constatering vormt de basis voor de recursie.
– Om te tekenen, hoeft u in methode tekenBoom dus alleen maar methoden van pen aan te roepen. Met de grafische context heeft u niets meer te maken. Zou u bijvoorbeeld een vierkant willen tekenen, dan zou u schrijven:

```
for (int i=1; i<=4; i++) {
   pen.vooruit(100);
   pen.draai(90);
}
```

– In de methode paintComponent wordt een pen gemaakt die onderaan de stam van de boom staat en naar boven wijst.
– Zorg er voor, dat aan het eind van de methode tekenBoom de pen weer teruggezet wordt op zijn oude positie. Zet de pen eerst uit met pen.uit(). Draai hem dan even ver terug als u hem heen hebt laten draaien, en beweeg hem evenveel pixels achteruit als u hem vooruit hebt laten gaan (u kunt draai en vooruit aanroepen met negatieve parameters). Zet hem tot slot weer aan (pen.aan()).

U hoeft zich dan nooit af te vragen, waar de pen is gebleven na voltooiing van een recursieve aanroep: die staat namelijk precies waar die voor de aanroep stond.

Inleveren:

Lever alleen het bestand BoomPanel.java in.

BIJLAGE: De klasse Pen

Package

Class **Tree** **Deprecated** **Index** **Help**

PREV CLASS NEXT CLASS **FRAMES** **NO FRAMES** **All Classes**

SUMMARY: NESTED | FIELD | CONSTR | METHOD DETAIL: FIELD | CONSTR | METHOD

CLASS PEN

```
java.lang.Object
   |
   +--Pen
```

public class **Pen**
extends java.lang.Object

Constructor Summary

Pen(java.awt.Graphics g) Maak een nieuwe pen die tekent in een gegeven grafisch context.
Pen(java.awt.Graphics g, int x, int y, double r) Maak een nieuwe pen die tekent in een gegeven grafisch context en die begint op een gegeven punt en in een gegeven richting.

Method Summary

void	**aan**() Zet de pen op het tekenblad: penbewegingen leiden tot zichtbare lijnen
void	**draai**(double hoek) Draai de pen over de gegeven hoek naar rechts.
void	**setKleur**(java.awt.Color k) Wijzig de tekenkleur
void	**uit**() Haal de pen van het tekenblad: penbewegingen zijn onzichtbaar
void	**vooruit**(double d) Beweeg de pen d pixels in de huidige tekenrichting.

Methods inherited from class java.lang.Object
clone, equals, finalize, getClass, hashCode, notify, notifyAll, toString, wait, wait, wait

Constructor Detail

Pen

```
public Pen(java.awt.Graphics g)
```

Maak een nieuwe pen die tekent in een gegeven grafisch context. De pen staat na creatie in de oorsprong, wijst naar rechts en heeft tekenkleur zwart.

Parameters:

`g` - de grafische context van de pen

Pen

```
public Pen(java.awt.Graphics g,
           int x,
           int y,
           double r)
```

Maak een nieuwe pen die tekent in een gegeven grafisch context en die begint op een gegeven punt en in een gegeven richting. De pen heeft tekenkleur zwart.

Parameters:

`g` - de grafische context van de pen
`x` - de x-coördinaat van de penpositie
`y` - de y-coördinaat van de penpositie
`r` - de richting waarin de pen wijst, in graden.

Method Detail

setKleur

```
public void setKleur(java.awt.Color k)
```

Wijzig de tekenkleur

Parameters:

`k` - de nieuwe tekenkleur

---

aan

```
public void aan()
```

Zet de pen op het tekenblad: penbewegingen leiden tot zichtbare lijnen

---

uit

```
public void uit()
```

Haal de pen van het tekenblad: penbewegingen zijn onzichtbaar

---

vooruit

```
public void vooruit(double d)
```

Beweeg de pen d pixels in de huidige tekenrichting. Als d < 0, gaat de pen achteruit.

Parameters:

`d` - het aantal pixels dat de pen moet bewegen (double om bij schuine lijnen zo nauwkeurig mogelijk te zijn)

---

draai

```
public void draai(double hoek)
```

Draai de pen over de gegeven hoek naar rechts. Als hoek < 0, draait de pen naar links.

Parameters:

`hoek` - de hoek waarover gedraaid wordt, in graden.

Package

Class **Tree** **Deprecated** **Index** **Help**

PREV CLASS NEXT CLASS **FRAMES** **NO FRAMES** **All Classes**

SUMMARY: NESTED | FIELD | CONSTR | METHOD DETAIL: FIELD | CONSTR | METHOD