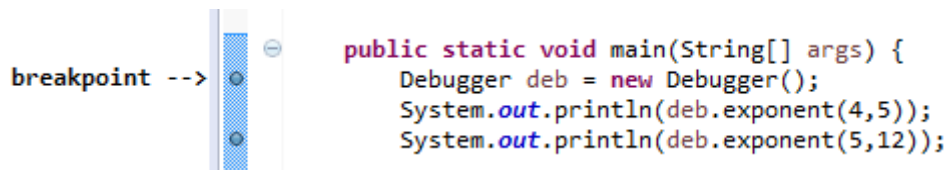





## Java Oefeningen – Reeks 3

### Vraag 0

Voor deze vraag gaan we gebruik maken van de debugger. Download de file “Debugger.java” en maak deze aan in Eclipse. Om gebruik te maken van de debugger moet je eerst minstens één breakpoint maken. Doe dit door ergens (bv. in het begin van de code) op de balk aan de linkerkant van de editor te dubbelklikken (zie hieronder).

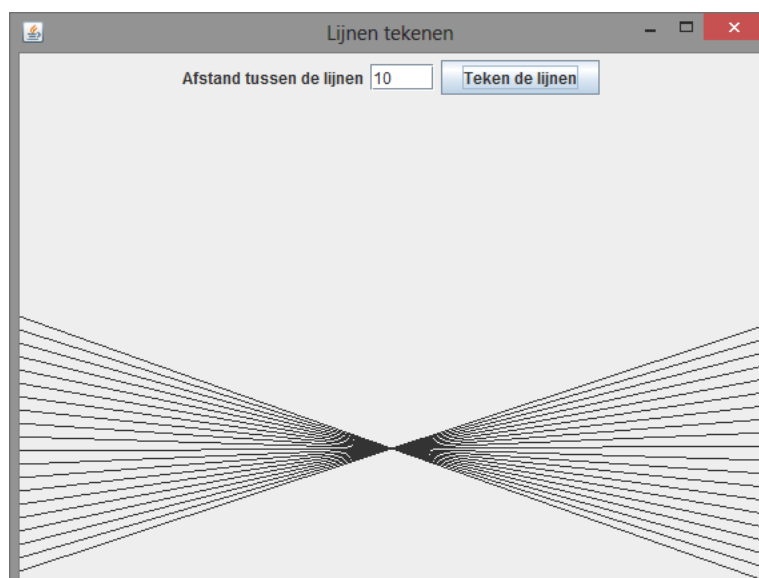


Klik vervolgens op het debugger-icoontje . Je kan stapsgewijs over de code lopen m.b.v. deze twee commando's:

-  Step in (F5): voert een lijn code uit. Gaat mee in de functie indien van toepassing.
-  Step over (F6): voert een lijn code uit, maar stapt niet in de functie. Dit kies je wanneer je zeker weet dat een lijn code geen bugs bevat.

### Vraag 1

Maak een programma dat schuine lijnen plaatst op de onderste helft van het venster. De afstand tussen de lijnen is variabel en door de gebruiker in te stellen via een tekstvak aan de bovenkant van het venster. De eerste lijn loopt van het midden aan de linkerkant naar onderaan rechts, elke volgende lijn is aan de linkerkant de ingestelde afstand lager en aan de rechterkant de ingestelde afstand hoger.



## Vraag 2

### 1)

Voor deze vraag gaan we een animatie maken met verschillende botsende ballen.

Maak eerst de klasse **Bal**. Een Bal heeft een **x** en **y** coördinaat, een straal **r** en snelheidscomponenten van de snelheidsvector  $\vec{v} = (v_x, v_y)$ : **vx** en **vy**. Een Bal heeft ook een kleur (van type **Color**). Initialiseer de waarden van de bal met random waarden:

- Geef de bal een willekeurige x en y positie in het veld. Zorg dat de bal niet buiten de randen van het veld terechtkomt!
- Kies voor **r** een willekeurige waarde tussen 5 en 30
- Geef de snelheidsvector een willekeurige lengte tussen 0 en 5, en een willekeurige hoek tussen 0 en  $2\pi$  (in radialen)
- Geef de RGB kleurcomponenten (van Color) elk een random waarde tussen 0 en 255.

Maak vervolgens de methode “void drawme(Graphics g)”. Teken een Bal met de juiste dimensies, kleur en positie met het meegegeven Graphics-object **g**.

Maak een nieuwe klasse **Balpaneel**, die een extensie is van **JPanel**. Kies een schermgrootte van 750x750 pixels. Voorzie een lijst van **Bal** met 100 elementen, en teken deze.

### 2)

Voor het tweede deel van de vraag gaan we een animatie toevoegen. Hiervoor zal periodisch een methode worden opgeroepen van **Bal** die zijn staat elke  $\Delta t$  zal aanpassen. Noem deze methode “void updateme()”. Deze moet het volgende doen:

1. Incrementeer de positie met de snelheidsvector.
2. Wanneer de bal een verticale wand raakt, zorg dat de bal binnen de randen van het veld blijft.
3. Om een botsing te simuleren verander je het teken van **vy** bij het raken van de verticale wand.
4. Doe hetzelfde analoog voor de horizontale wanden van het veld.

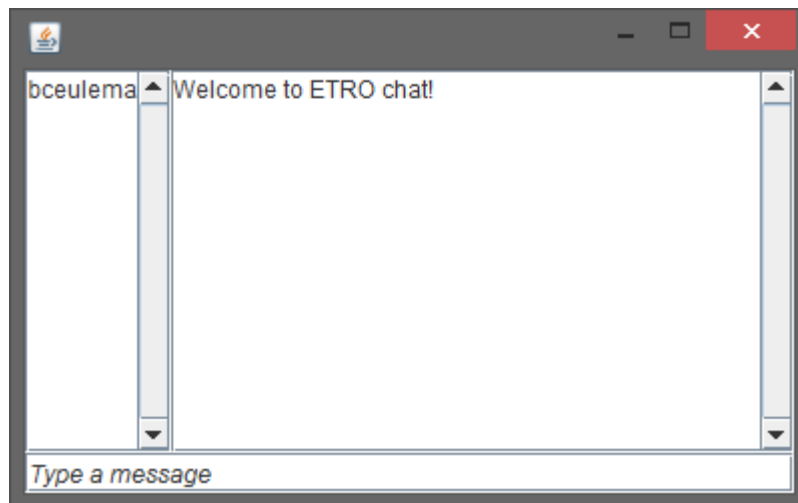
Voor de animatie hebben we een **Timer** en een **TimerTask** nodig. Maak een interne klasse **MyTimerTask** die een extensie is van **TimerTask**. Overschrijf de **run()** methode, en zorg dat deze de methode “void updateme()” van alle **Bal**-objecten in de **Bal**-lijst van **Balpaneel**.

Maak tenslotte een nieuwe **Timer** aan in de constructor. Zorg dat deze **MyTimerTask** periodisch elke 20ms oproept.

### Vraag 3

In deze vraag gaan we een eenvoudige grafische user interface (GUI) bouwen voor een chat-client. Op de laptop van de assistent zal een chat-server staan. Het programma dat jullie moeten schrijven zal hiernaar verbinden. Berichten die jullie naar de server verzenden worden doorgestuurd naar alle clients die op dat moment verbonden zijn.

Download alvast de SimpleConnection.jar en ChatMain.java bestanden van de cursus pagina.



*Figuur 1: Voorbeeld van een chat client GUI*

### GUI

De gedownloadde klasse **ChatMain** extend een **JFrame**. Aan dat JFrame wordt dan een **ChatPanel** object toegevoegd waarin de eigenlijke code komt te staan. Dit **ChatPanel** extends **JPanel** en heeft dan een methode `run()` waarin in een oneindige loop wordt geluisterd naar nieuwe berichten van de server. Het is jouw taak om de **ChatPanel** klasse te implementeren.

In de constructor van deze klasse maak je de GUI elementen aan die je wil gebruiken en voeg je ze toe aan het JPanel. In de meegegeven voorbeeld worden de volgende elementen gebruikt:

- JTextArea om de ontvangen tekst in weer te geven
- JTextArea om de lijst van online gebruikers in weer te geven
- JTextField waar de gebruiker in kan typen
- JScrollPane om scrollbars toe te voegen aan de gebruikte JTextAreas

Aan de JTextField moet je een ActionListener toevoegen zodat wanneer de gebruiker op ENTER drukt, de getypte tekst verzonden wordt. Dit gebeurt met de volgende lijn:  
`txtSend.addKeyListener(this);`

Om dit te laten werken moet de ChatPaneel klasse de KeyListener (**public class ChatPaneel extends JPanel implements KeyListener**)

Schrijf dan de nodige code in de “void KeyPressed(KeyEvent e)” methode. Er werd op ENTER gedrukt als e.getKeyCode() gelijk is aan 10.

Dus samengevat:

- 1) De constructor van ChatPaneel maakt een verbinding naar de server.
- 2) ChatPaneel bevat een methode run() die in een oneindige loop luistert naar nieuwe berichten van de server en deze verwerkt.
- 3) Wanneer de gebruiker een bericht typt in het JTextField en op ENTER drukt, wordt dit bericht naar de server verzonden.

### **Opdracht: Communicatie met een server**

Om met de server te communiceren gebruik je de **SimpleConnection** klasse die in de gedownloade **jar** file zit. (Voeg het jar-bestand toe in een package van je project. Rechterklik op de jar-file en klik vervolgens op “Build Path → Add to Build Path”)

Je kan een nieuw object van deze klasse maken:

```
SimpleConnection connection = new SimpleConnection();
```

Gebruik dan de volgende methodes:

```
public void connect(String host, int port)
// verbindt naar een server
public void sendText(String msg);
// stuur een bericht naar de server
public String readline();
// kijkt of de server een bericht stuurt en retournt dat als String
```

De readline() methode zal je telkens berichten teruggeven waarbij het eerste woord in hoofdletters staat en de betekenis van het bericht verduidelijkt.

“SAY <Jan> Hallo wereld” betekent dat er iets gezegd werd,

“ENTER Corneel” betekent dat er een nieuwe gebruiker in de chatroom is binnengekomen,

“LEAVE Corneel” betekent dat er een gebruiker de chatroom heeft verlaten,

“USERS Jan,Piet,Joris” geeft een lijst van de gebruikers die online zijn.