



Stichting NIOC en de NIOC kennisbank

Stichting NIOC (www.nioc.nl) stelt zich conform zijn statuten tot doel: het realiseren van congressen over informatica onderwijs en voorts al hetgeen met een en ander rechtstreeks of zijdelings verband houdt of daartoe bevorderlijk kan zijn, alles in de ruimste zin des woords.

De stichting NIOC neemt de archivering van de resultaten van de congressen voor zijn rekening. De website www.nioc.nl ontsluit onder "Eerdere congressen" de gearchiveerde websites van eerdere congressen. De vele afzonderlijke congresbijdragen zijn opgenomen in een kennisbank die via dezelfde website onder "NIOC kennisbank" ontsloten wordt.

Op dit moment bevat de NIOC kennisbank alle bijdragen, incl. die van het laatste congres (NIOC2023, gehouden op donderdag 30 maart 2023 jl. en georganiseerd door NHL Stenden Hogeschool). Bij elkaar bijna 1500 bijdragen!

We roepen je op, na het lezen van het document dat door jou is gedownload, de auteur(s) feedback te geven. Dit kan door je te registreren als gebruiker van de NIOC kennisbank. Na registratie krijg je bericht hoe in te loggen op de NIOC kennisbank.

Het eerstvolgende NIOC vindt plaats op donderdag 27 maart 2025 in Zwolle en wordt dan georganiseerd door Hogeschool Windesheim. Kijk op www.nioc2025.nl voor meer informatie.

Wil je op de hoogte blijven van de ontwikkeling rond Stichting NIOC en de NIOC kennisbank, schrijf je dan in op de nieuwsbrief via

www.nioc.nl/nioc-kennisbank/aanmelden_nieuwsbrief

Reacties over de NIOC kennisbank en de inhoud daarvan kun je richten aan de beheerder:

R. Smedinga kennisbank@nioc.nl.

Vermeld bij reacties jouw naam en telefoonnummer voor nader contact.

Programmeren in Java

Met JCreator en BlueJ



Door Wim de Lange,

SG Tabor, locatie Oscar Romero, Hoorn

18 april 2007

Mijn achtergrond

- Docent Informatiekunde en Informatica
- Opleiding via CODI, Java cursus met Visual Café
- Verzorg lessen Informatiekunde in de Brugklas en Informatica in de Tweede Fase HAVO en VWO

Lessen in programmeren op het Oscar

	Nu	Vanaf 2007
HAVO	JCreator-basis	Uitbreiden met BlueJ-basis
VWO	JCreator-basis, BlueJ-basis	Beide, BlueJ met verdieping

- Module gemaakt door Peter Harmsen en Ruud Koortens, vergelijkbaar met module van GertJan Laan van Instruct.
- Eerste kennismaking met kritische computertaal, confrontatie met de noodzaak van nauwkeurig werken.
- JCreator-omgeving is NIET visueel, bewuste keuze om gevoel voor code bij leerlingen aan te brengen. (“netjes coderen”)

- Er wordt alleen gewerkt binnen een Applet waardoor de “hele code ” is te overzien. (Eventueel op een website te plaatsen.)
- Nadeel van deze manier van leren programmeren is dat het gevoel voor objecten minder ontwikkeld wordt en er wordt niet aan zelfstandige applicaties gewerkt.
- Er was behoefte aan een leerling-vriendelijkere omgeving.

Vanaf dit schooljaar BlueJ



Wat is BlueJ?

Het is een geïntegreerde omgeving, speciaal ontworpen voor introductie van Objectgeoriënteerd programmeren. BlueJ is gebaseerd op Java en wordt regelmatig getest op systemen met Solaris, Linux, Macintosh en verschillende Windows-versies. Het zou moeten werken op alle platforms die een recente Java VM ondersteunen. Als freeware beschikbaar voor non-commerciële gebruikers.

Waar komt BlueJ vandaan?

Het wordt ontwikkeld en onderhouden door een groep onderzoekers van de Deakin University, Melbourne, Australië samen met mensen van de University of Kent in Canterbury, UK. Het wordt gesteund door Sun Microsystems.

Wat is het doel van de BlueJ-groep?

Men wilde voorzien in een makkelijk te gebruiken leeromgeving voor het onderwijs in Java aan beginnende studenten. Er is speciale aandacht besteed aan visualisatie - en interactietechnieken om zodoende een heel interactieve omgeving te creëren, die uitnodigt tot experimenteren en onderzoek.

Redenen voor mij om met BlueJ aan de slag te gaan

Allereerst zag ik kansen om objectoriëntatie beter uit te leggen, vooral de visuele manier van werken levert voordelen op didactisch gebied, de leerlingen van tegenwoordig zijn meer plaatjes gericht. Bovendien is het een complete omgeving, van ontwerpen tot en met testen kan allemaal binnen BlueJ.

Voor de toekomst zijn er mogelijkheden tot verdieping via de plug-ins.

BlueJ voor Havo/VWO



ISBN 90 460 0193 8

Inhoudsopgave

1. Objecten in BlueJ

- Objecten maken in BlueJ
- Een bericht naar een object sturen
- Nog een object maken
- De object-inspector van BlueJ

2. De schildpad

- De interface van de schildpad
- Methoden van Schildpad
- Een tweede schildpad maken
- Broncode van de klasse Schildpad
- Documentatie

3. Klassen

- Een klasse voor bankrekeningen
- De broncode van de klasse Bankrekening
- Lay-out van de broncode
- De klasse Leerling

4. Objecten met objecten

- Teams van leerlingen
- Verbeterde versie van de klasse Team
- Objecten bewaren in BlueJ
- Strings achter elkaar plakken

5. Collecties

- Herontwerp van de klasse Team
- Cijfers
- Een winkel
- Testen van de winkel
- Formatteren van de uitvoer

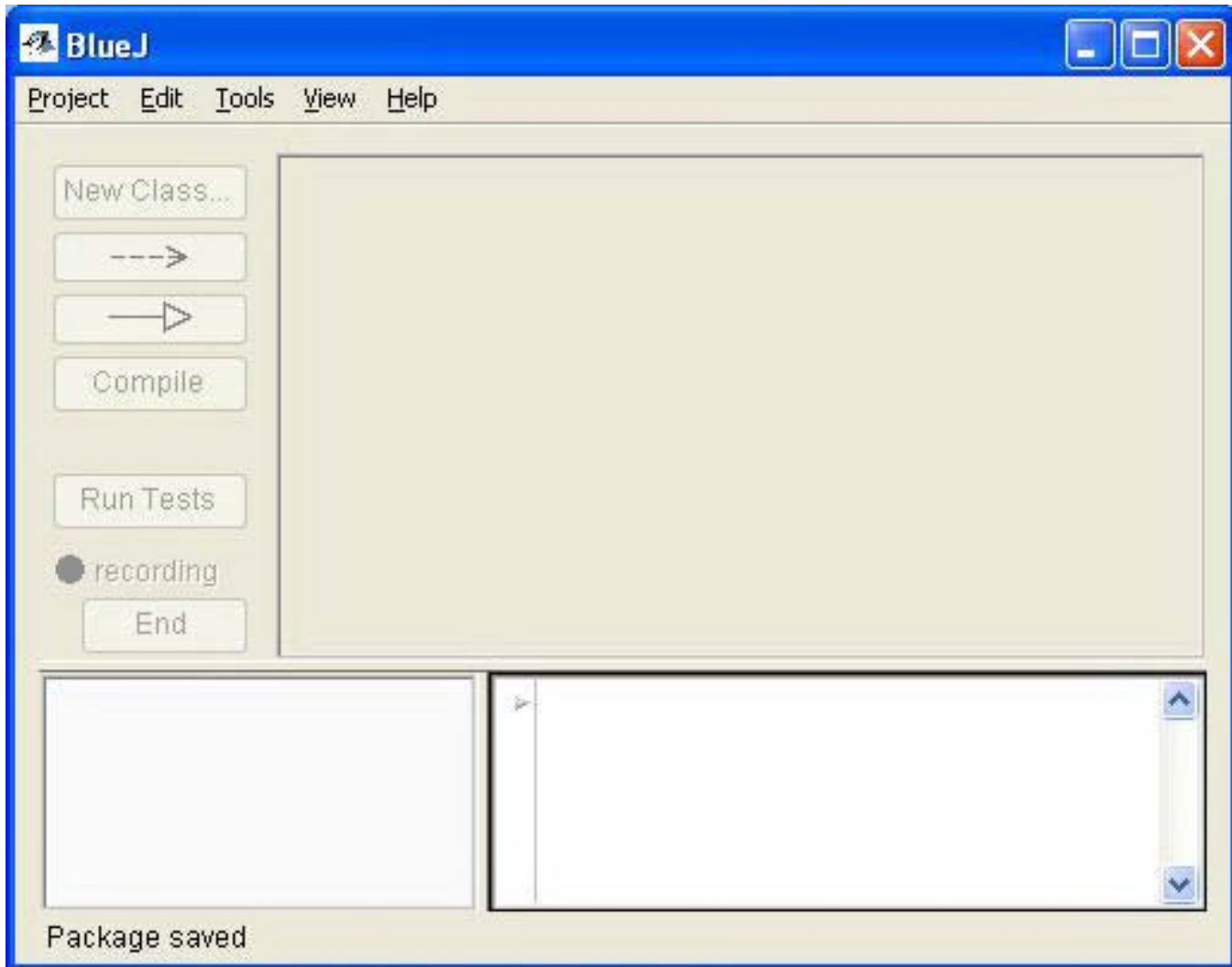
6. Keuzes en herhalingen

- Dobbelsteen
- Het for-statement
- Herhalingen met de schildpad
- Kop of munt
- Getal raden
- Yahtzee

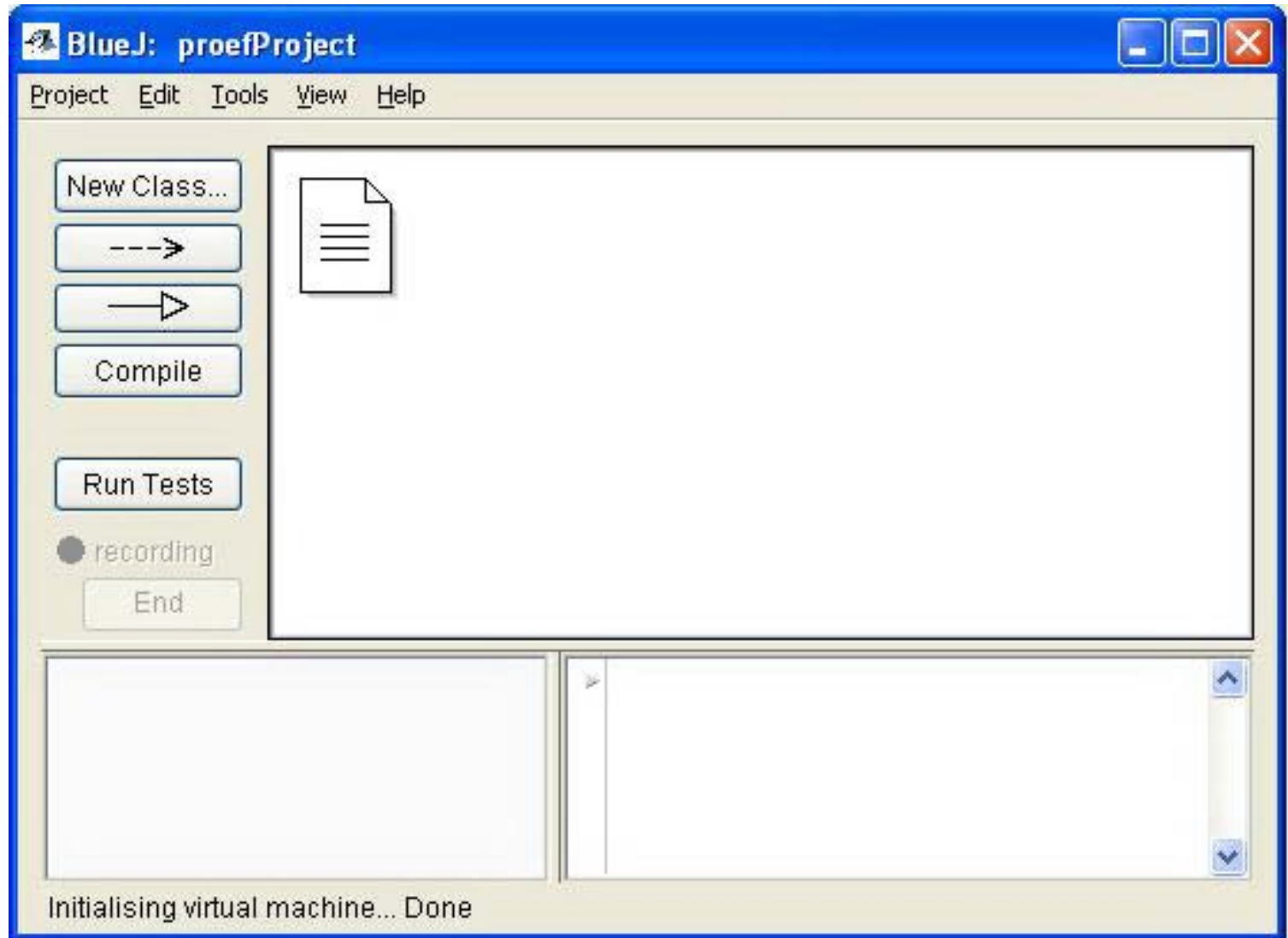
7. Gebruikersinterface

- Een Swing-applicatie
- JPanel, JButton en JTextField
- Gebruikersinterface voor het raadspel
- Afhandelen van twee verschillende events
- Een applet

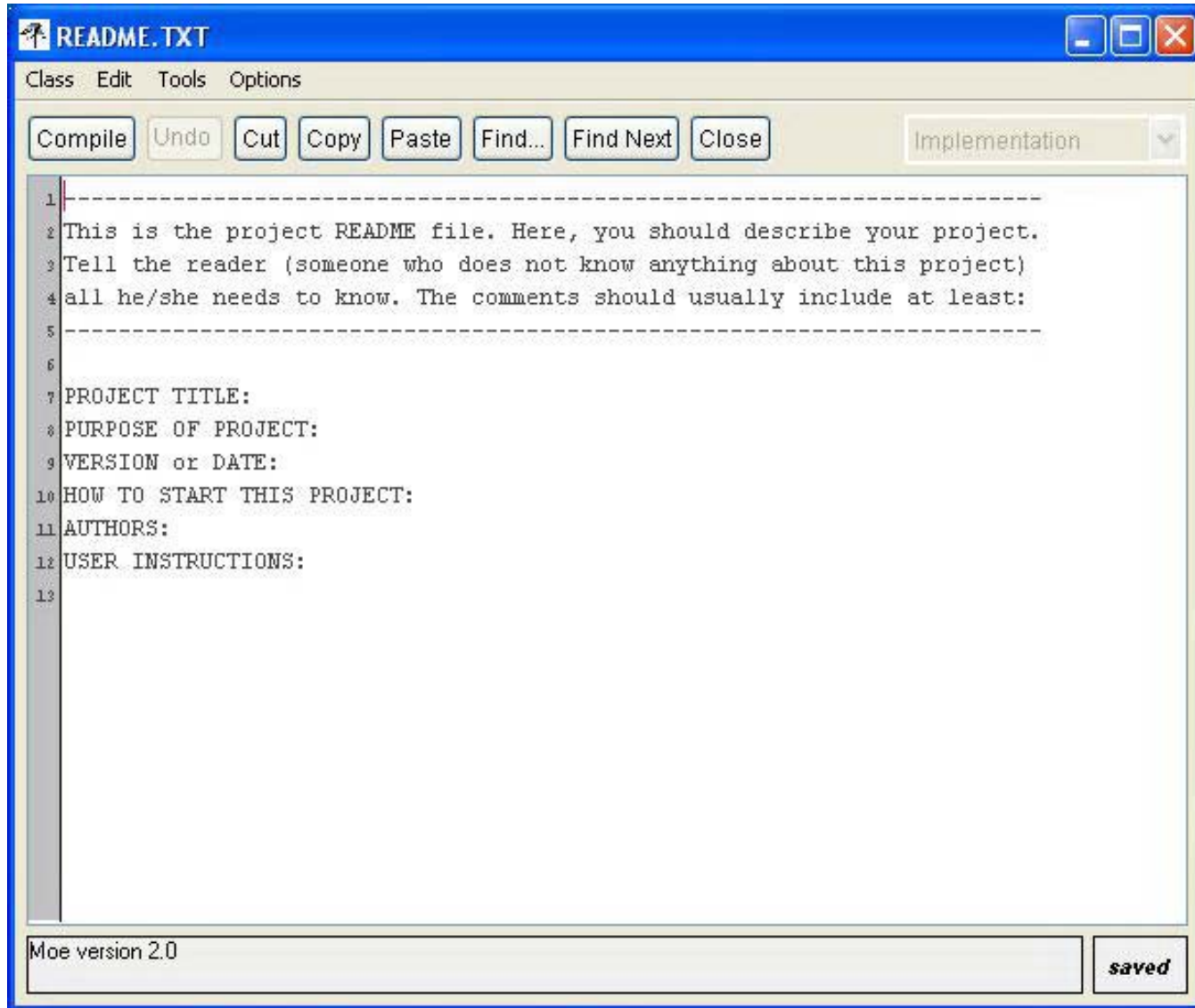
Screenshots



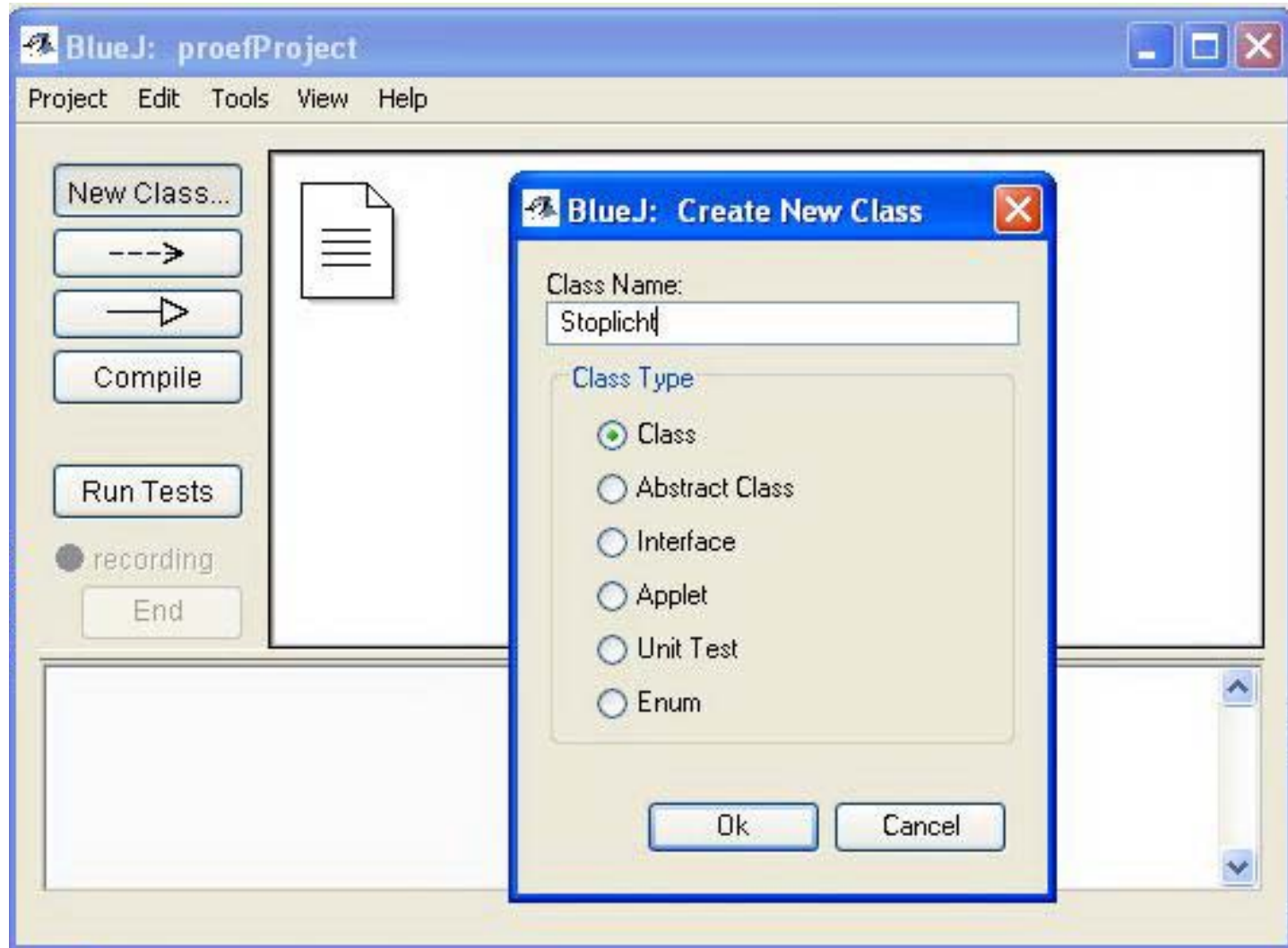
Screenshots



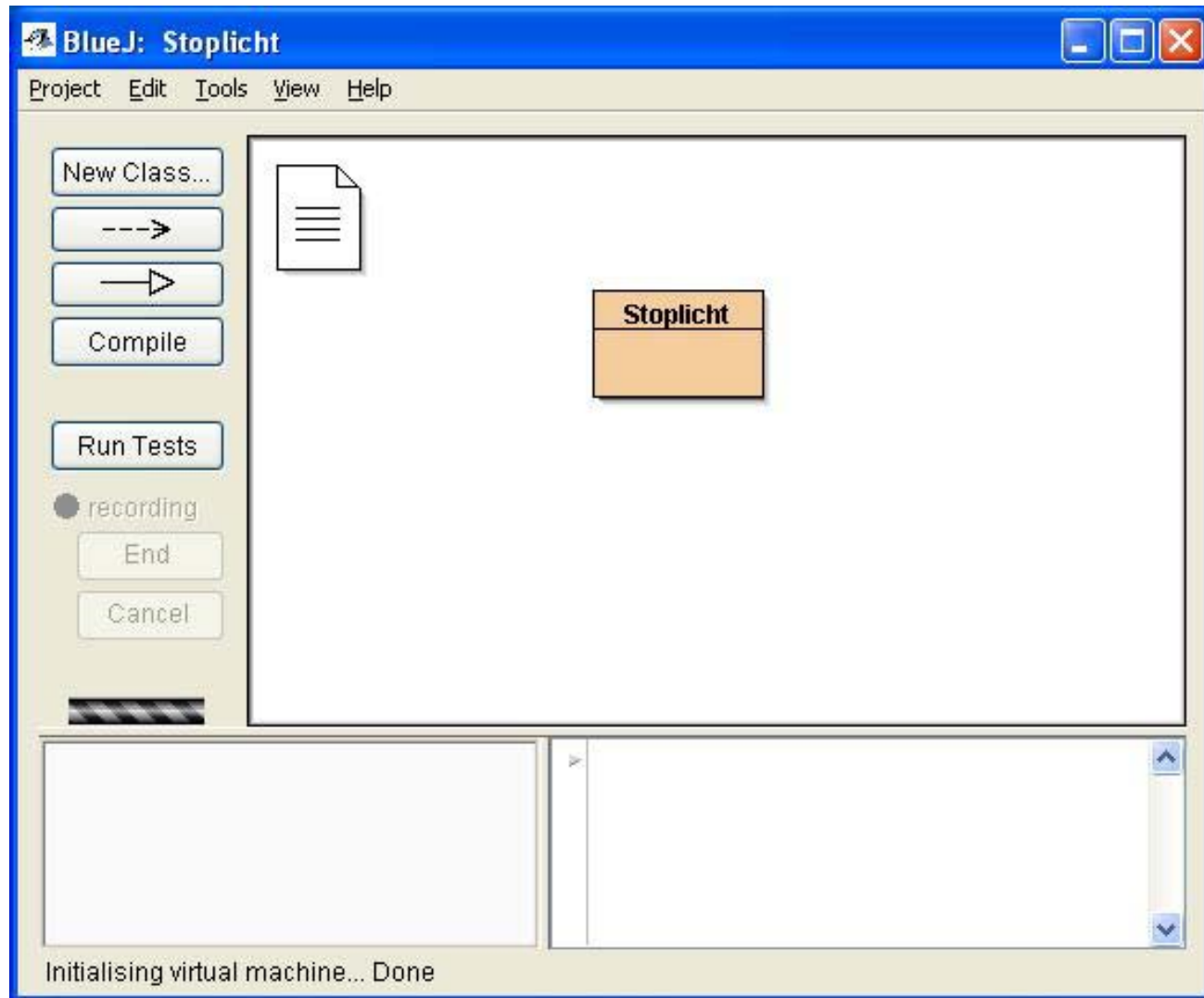
Screenshots



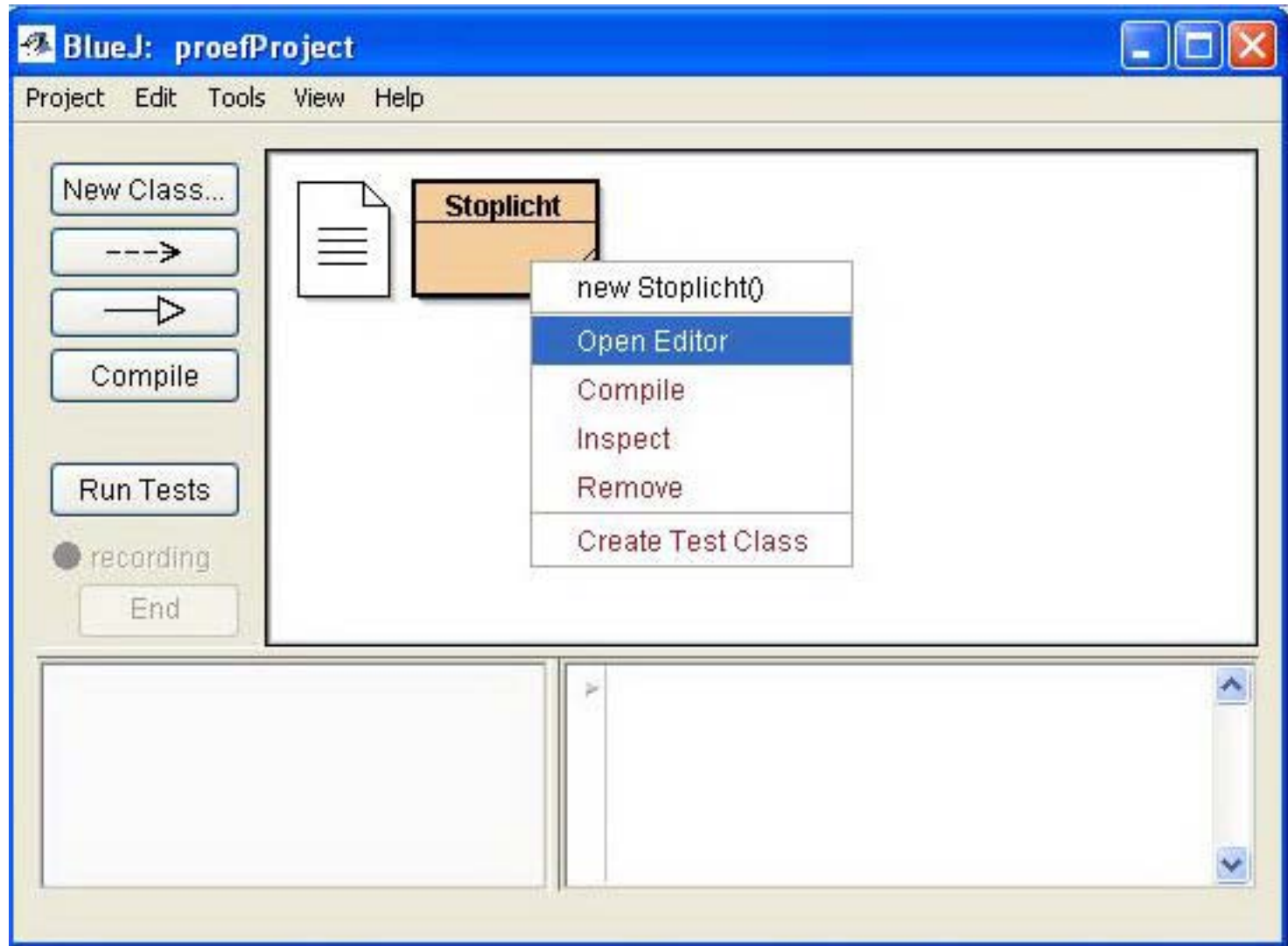
Screenshots



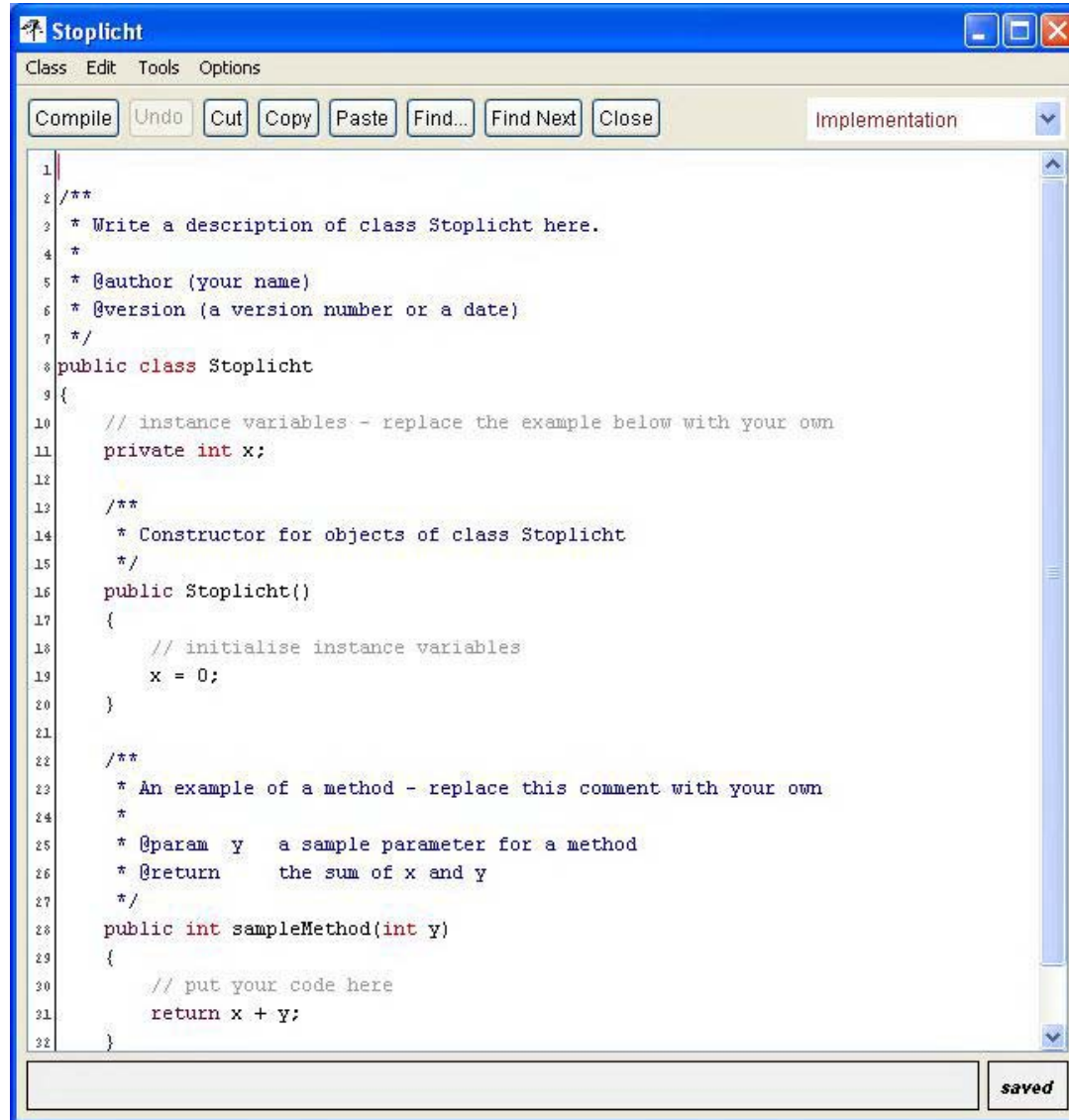
Screenshots



Screenshots



Screenshots

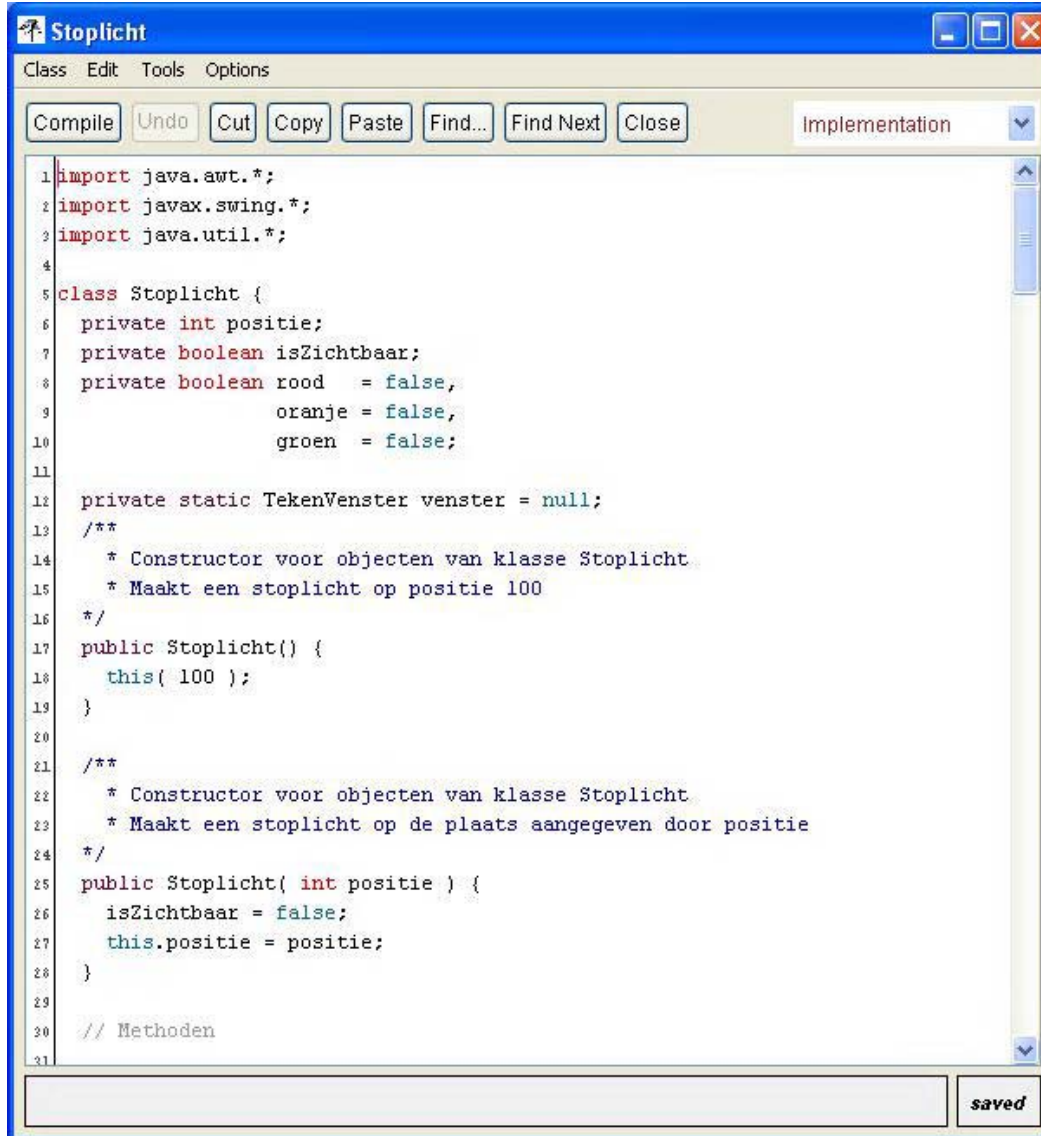


```
1 |
2 | /**
3 |  * Write a description of class Stoplicht here.
4 |  *
5 |  * @author (your name)
6 |  * @version (a version number or a date)
7 |  */
8 | public class Stoplicht
9 | {
10 |     // instance variables - replace the example below with your own
11 |     private int x;
12 |
13 |     /**
14 |      * Constructor for objects of class Stoplicht
15 |      */
16 |     public Stoplicht()
17 |     {
18 |         // initialise instance variables
19 |         x = 0;
20 |     }
21 |
22 |     /**
23 |      * An example of a method - replace this comment with your own
24 |      *
25 |      * @param y    a sample parameter for a method
26 |      * @return     the sum of x and y
27 |      */
28 |     public int sampleMethod(int y)
29 |     {
30 |         // put your code here
31 |         return x + y;
32 |     }

```

saved

Screenshots

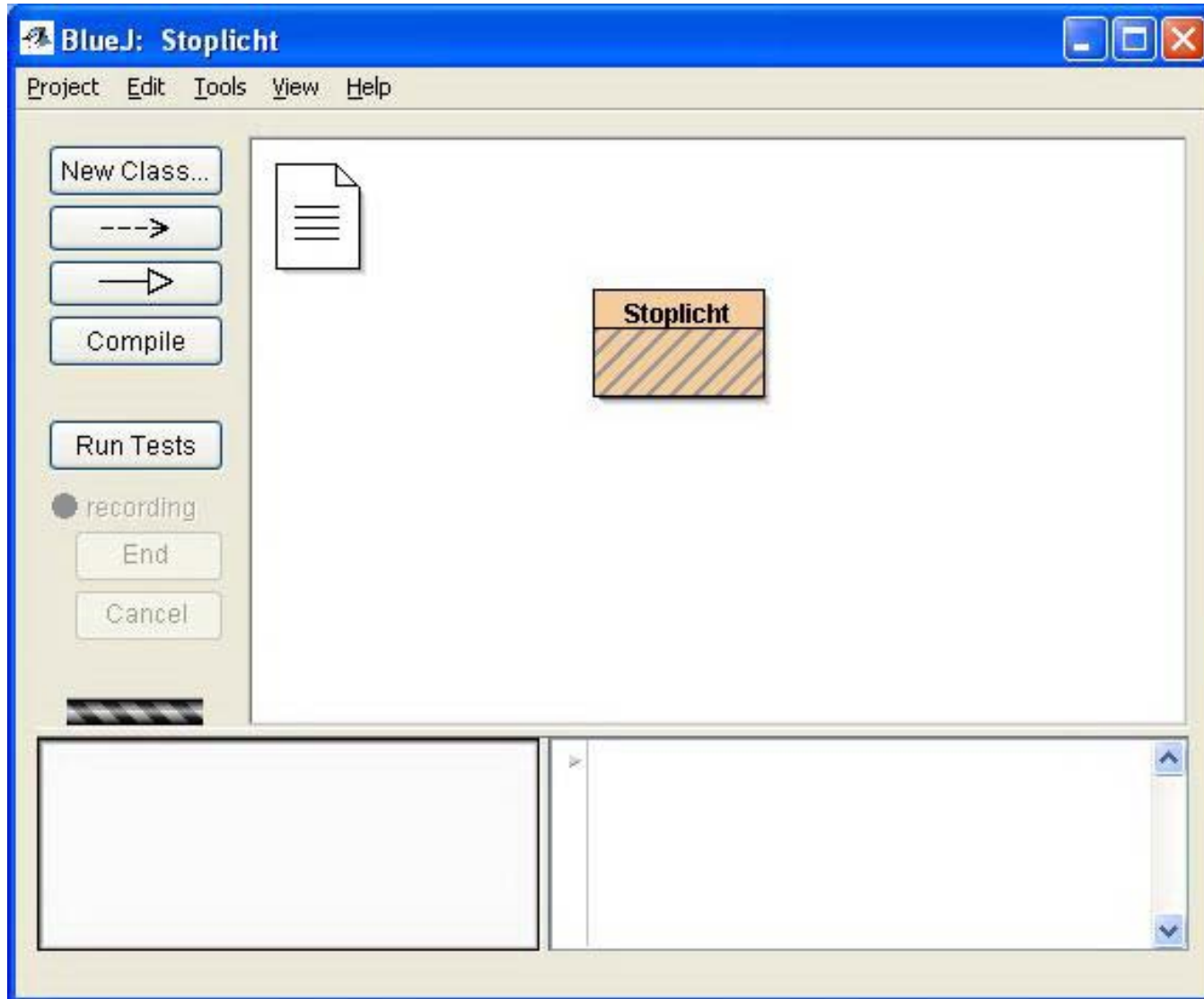


The screenshot shows the JCreator IDE window titled "Stoplicht". The menu bar includes "Class", "Edit", "Tools", and "Options". The toolbar contains buttons for "Compile", "Undo", "Cut", "Copy", "Paste", "Find...", "Find Next", "Close", and a dropdown menu set to "Implementation". The main text area displays the following Java code:

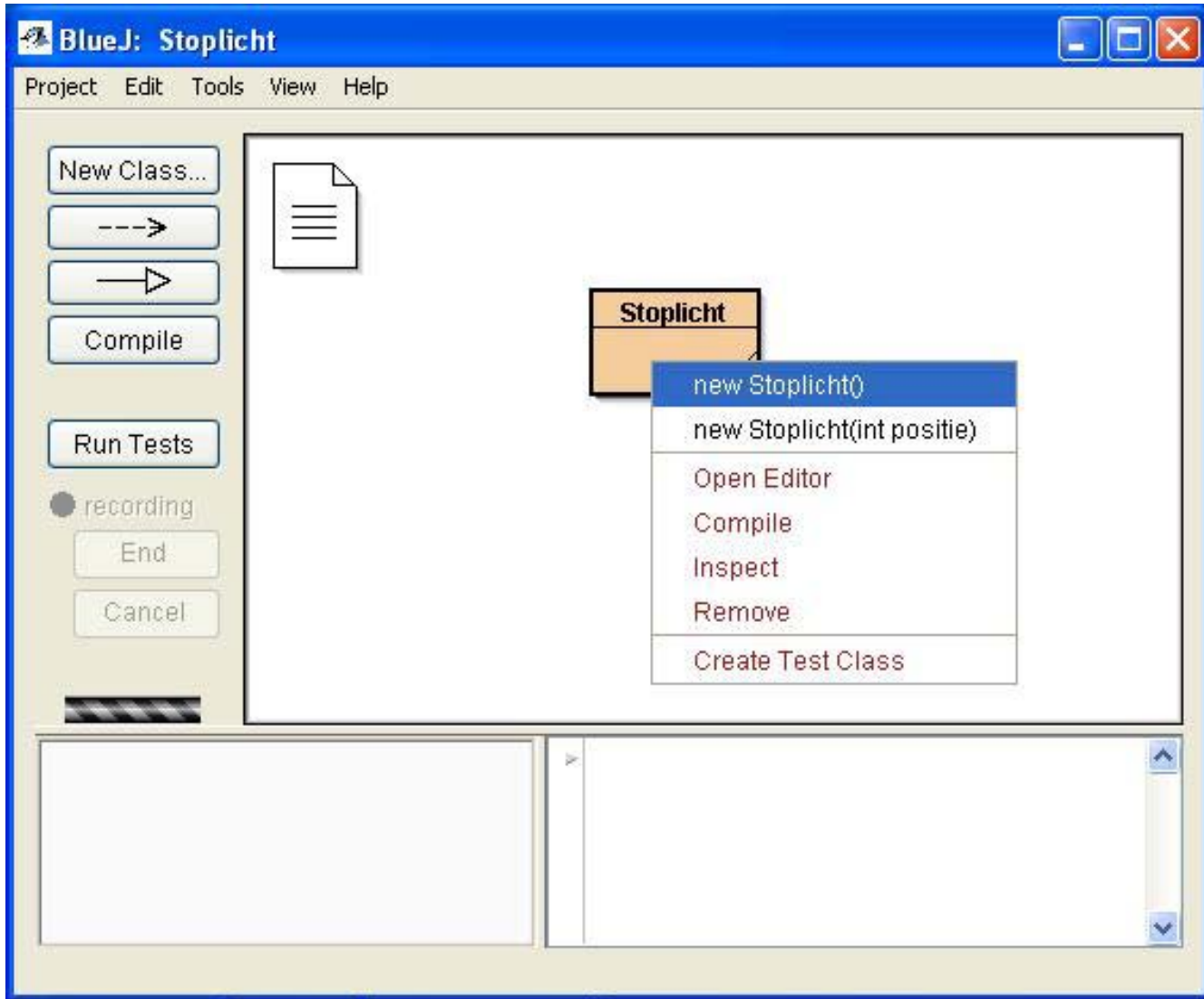
```
1 import java.awt.*;
2 import javax.swing.*;
3 import java.util.*;
4
5 class Stoplicht {
6     private int positie;
7     private boolean isZichtbaar;
8     private boolean rood = false,
9                 oranje = false,
10                groen = false;
11
12     private static TekenVenster venster = null;
13     /**
14      * Constructor voor objecten van klasse Stoplicht.
15      * Maakt een stoplicht op positie 100
16      */
17     public Stoplicht() {
18         this( 100 );
19     }
20
21     /**
22      * Constructor voor objecten van klasse Stoplicht.
23      * Maakt een stoplicht op de plaats aangegeven door positie
24      */
25     public Stoplicht( int positie ) {
26         isZichtbaar = false;
27         this.positie = positie;
28     }
29
30     // Methoden
31
```

The status bar at the bottom right of the window displays the word "saved".

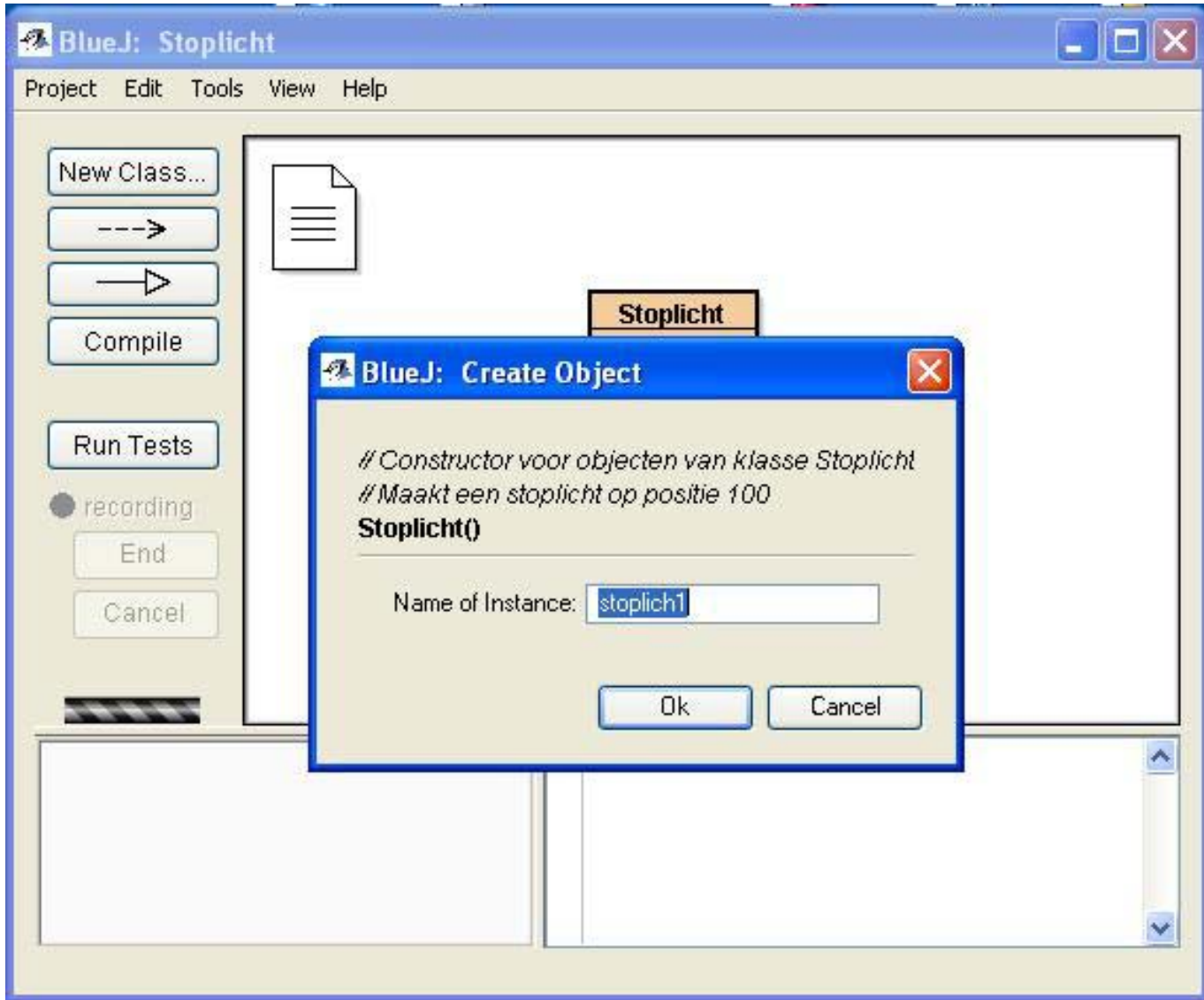
Screenshots



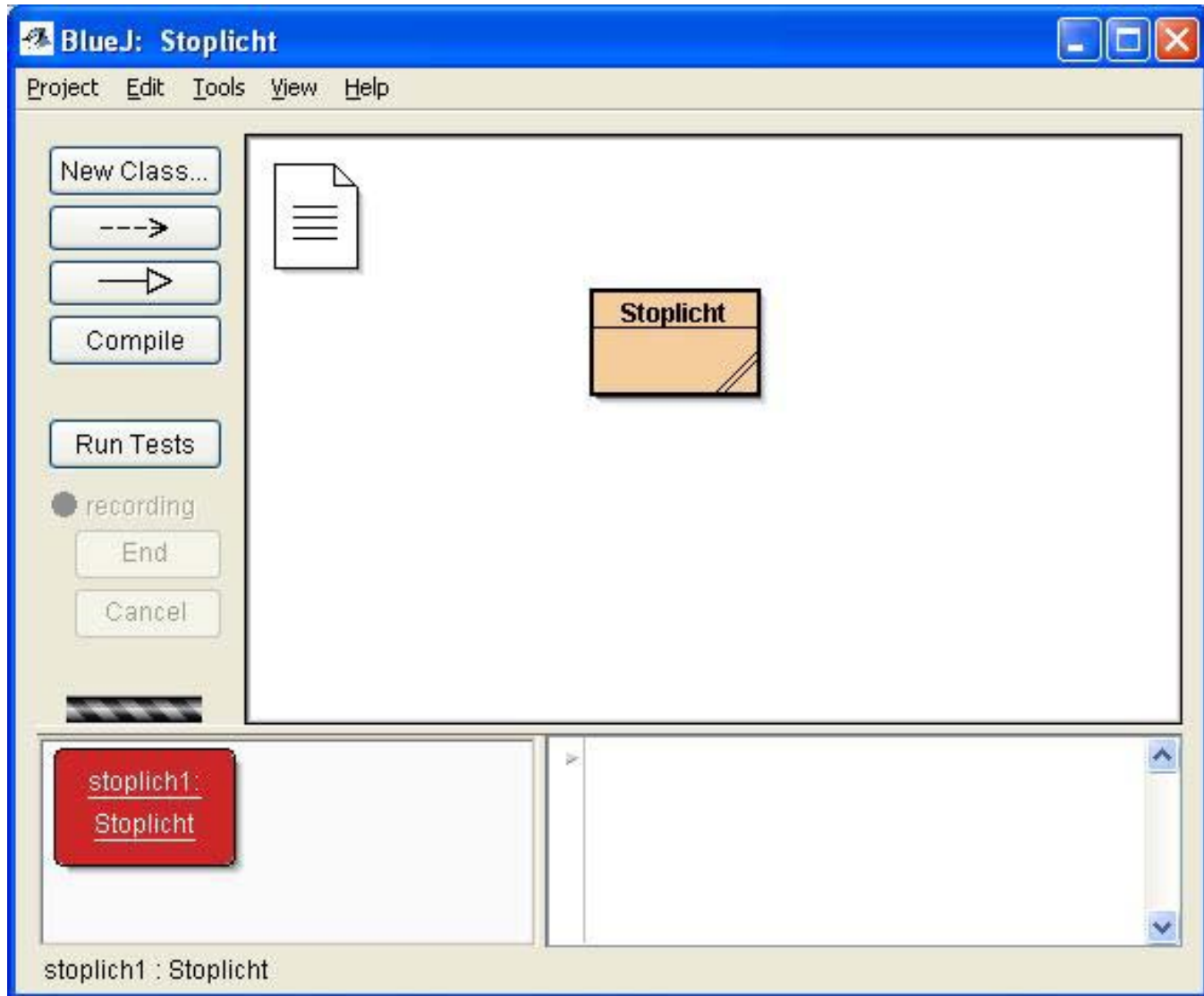
Screenshots



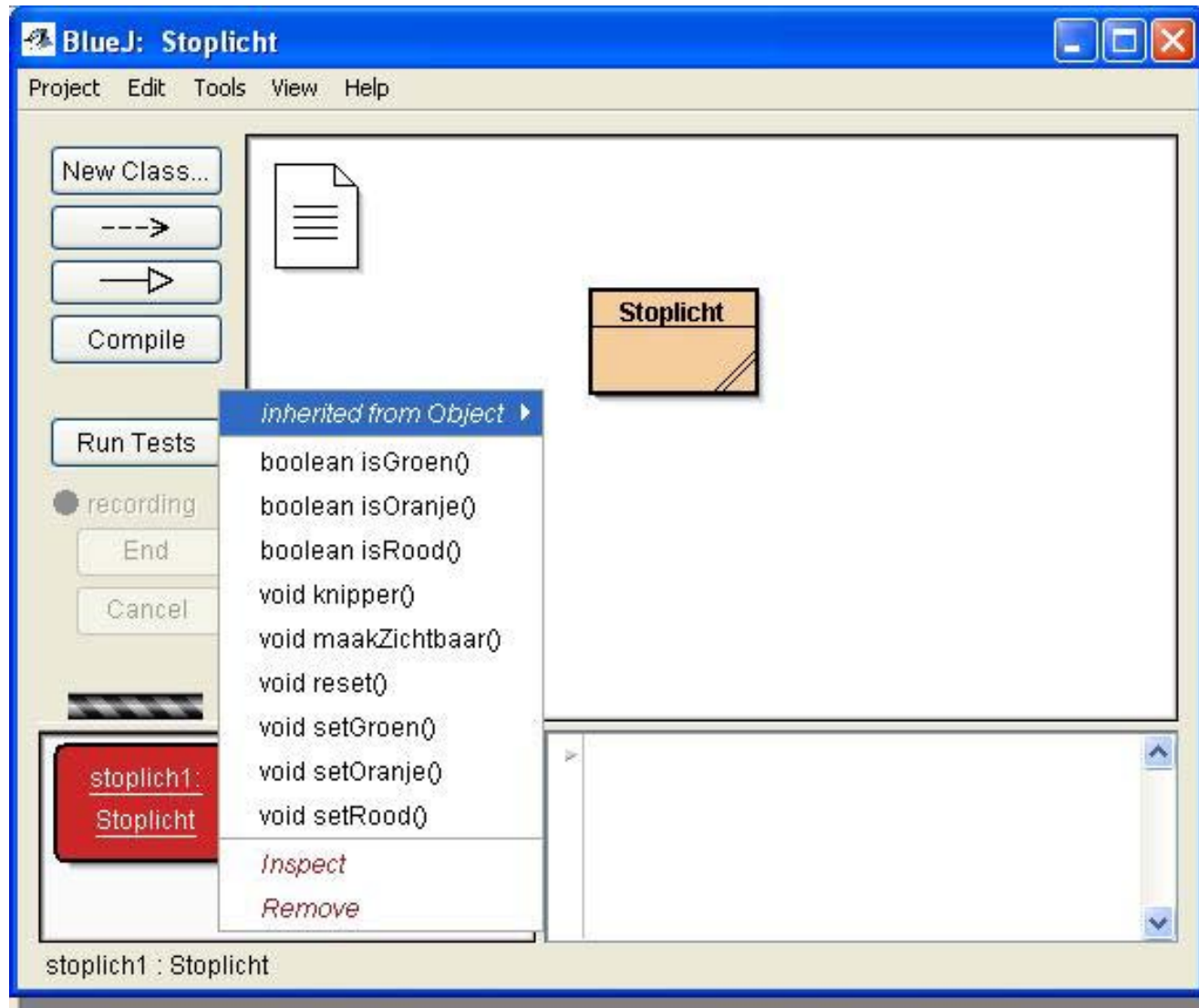
Screenshots



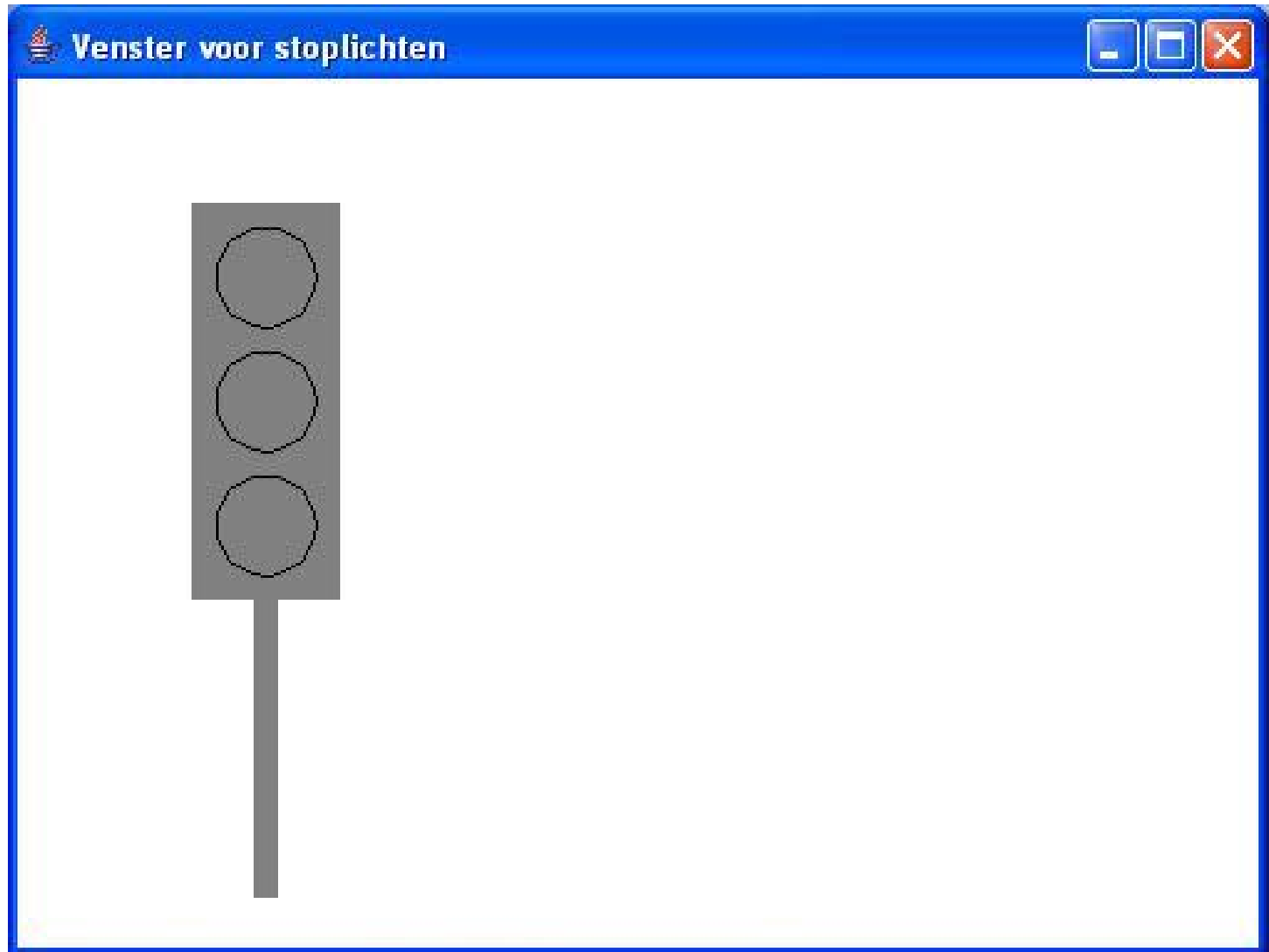
Screenshots



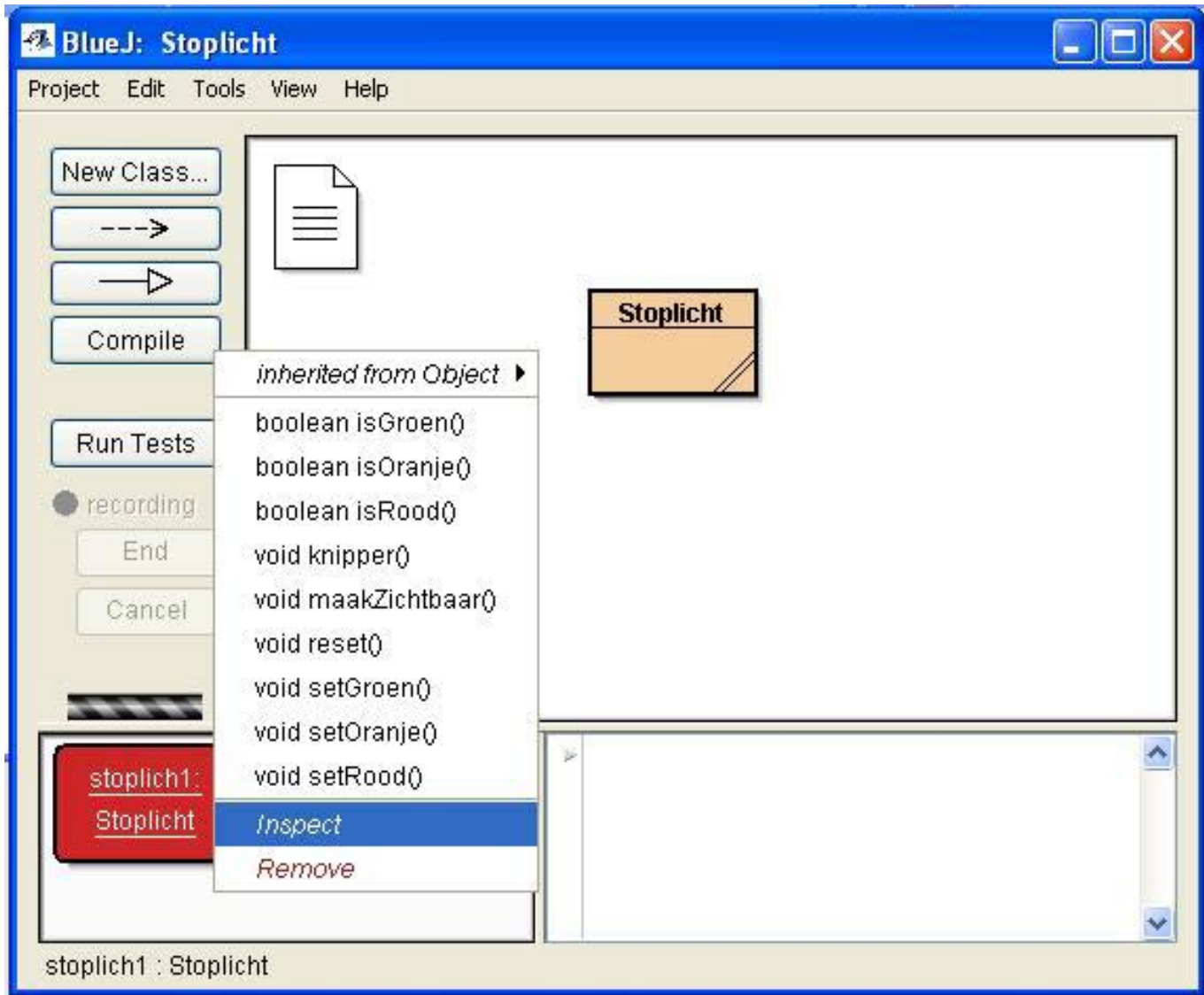
Screenshots



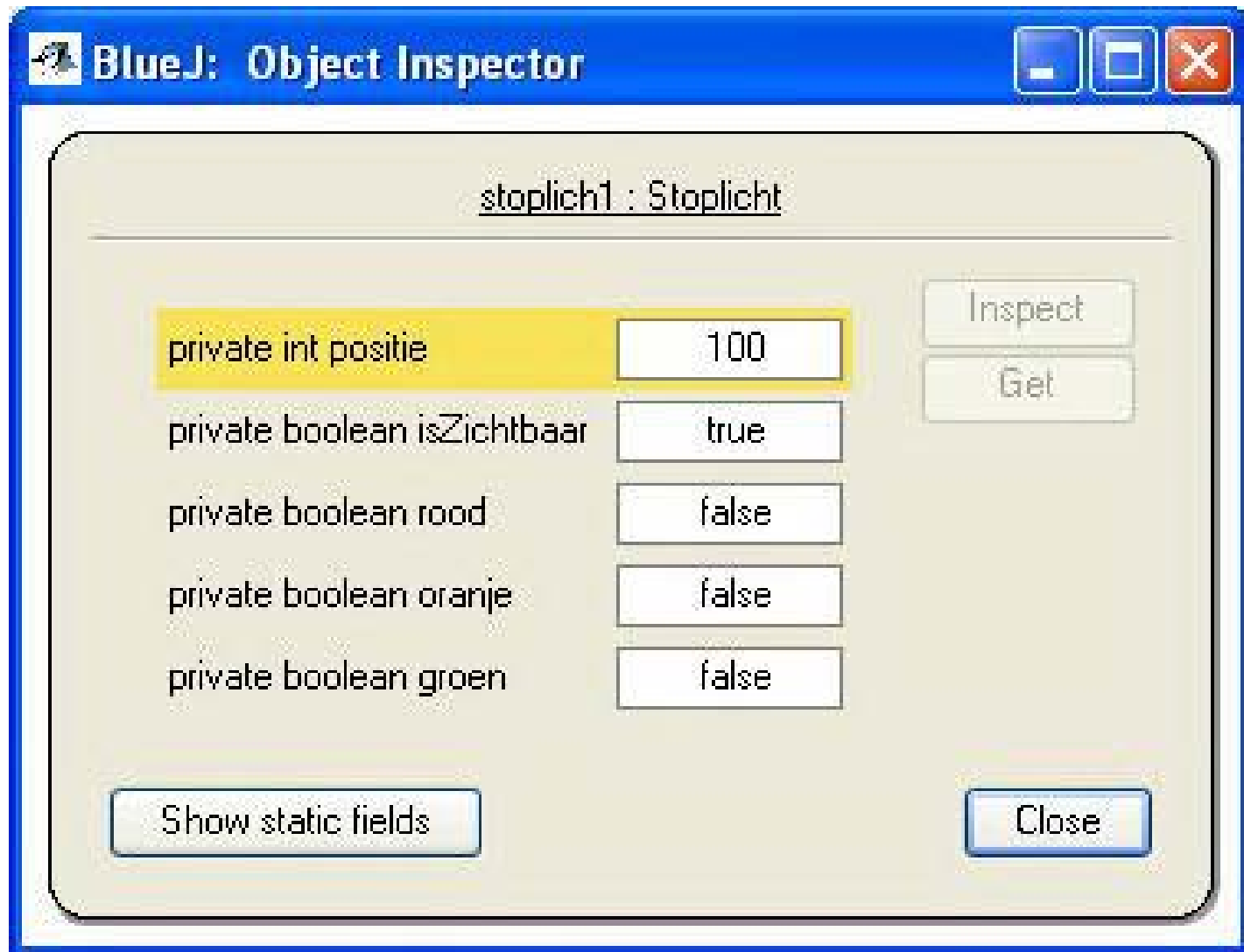
Screenshots



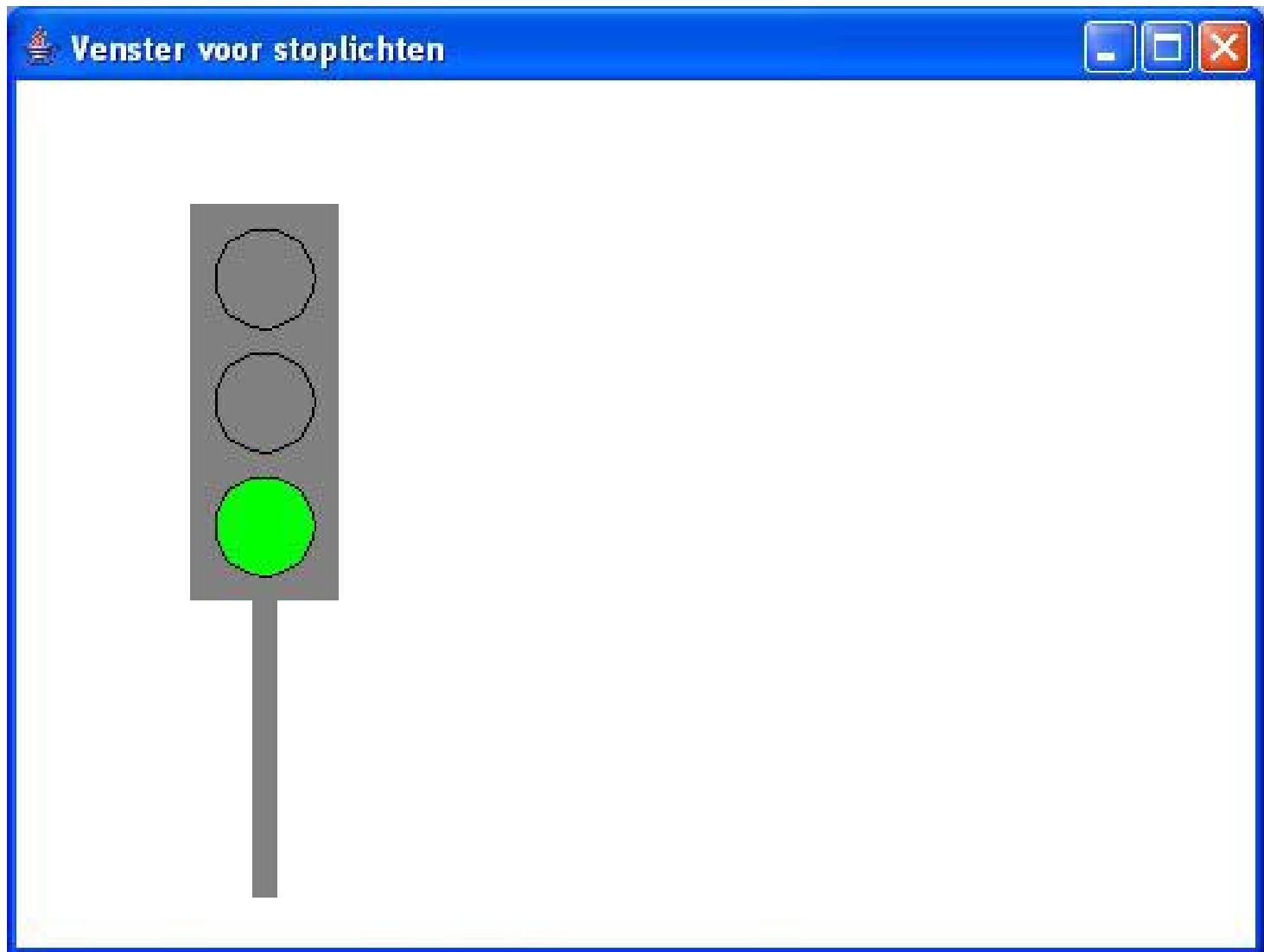
Screenshots



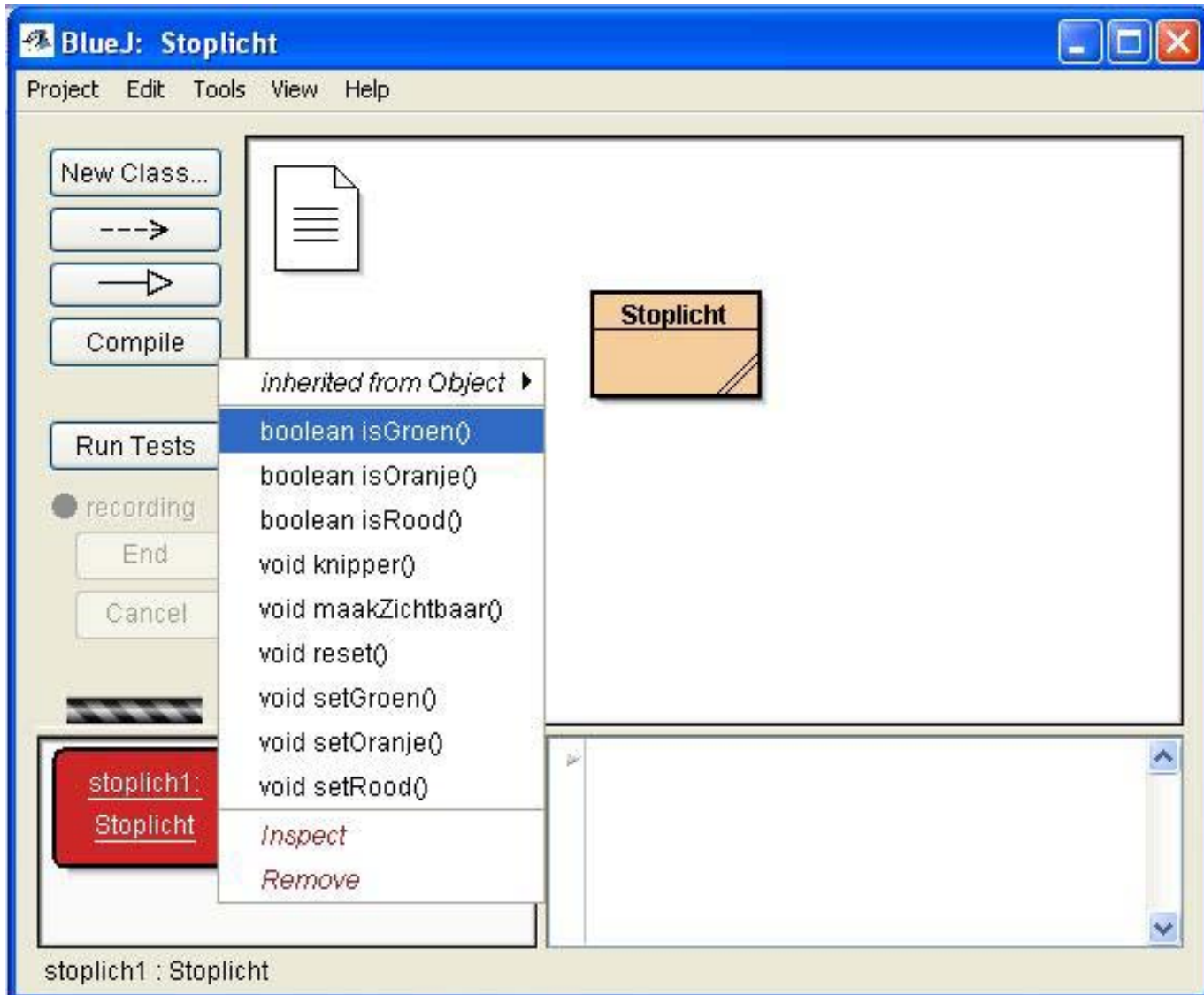
Screenshots



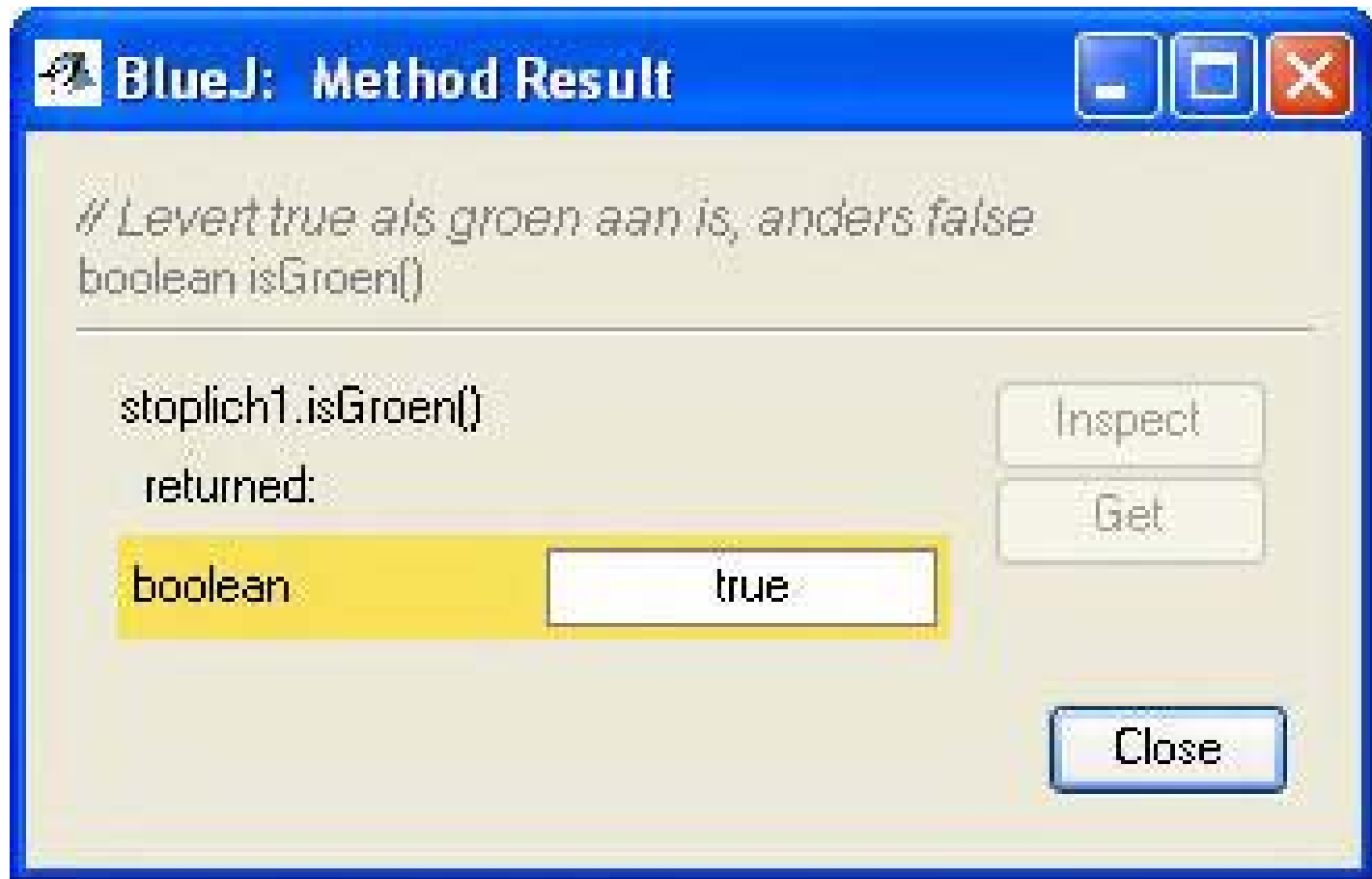
Screenshots



Screenshots



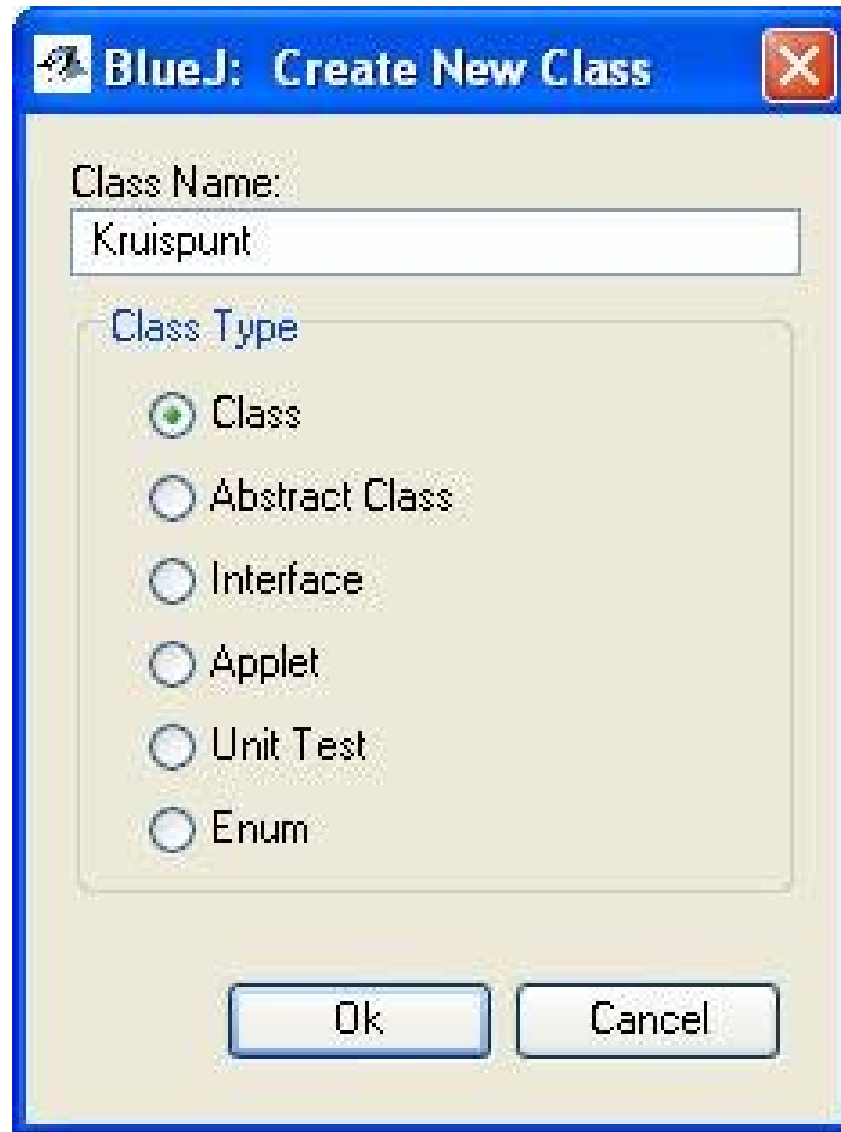
Screenshots



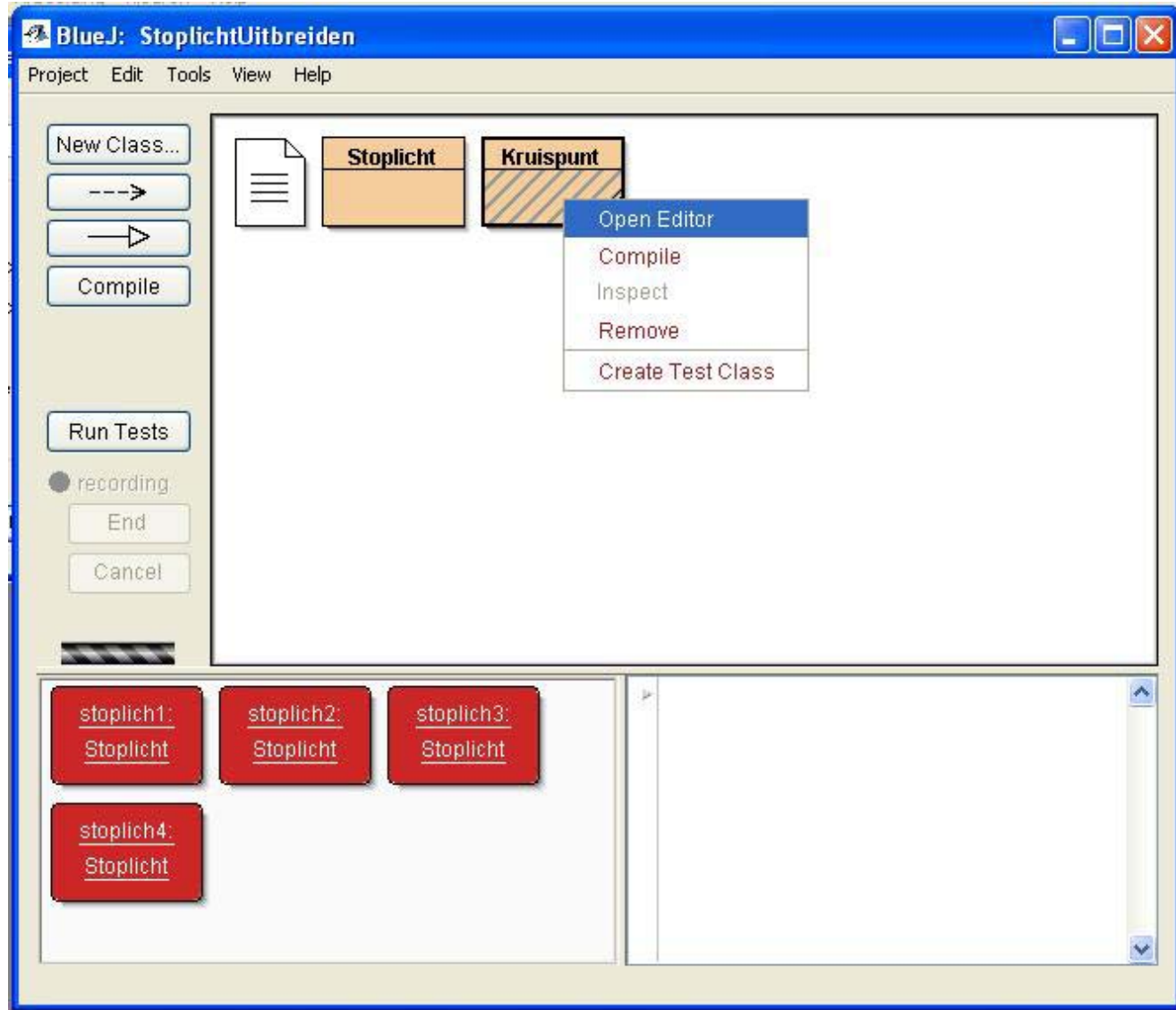
Screenshots



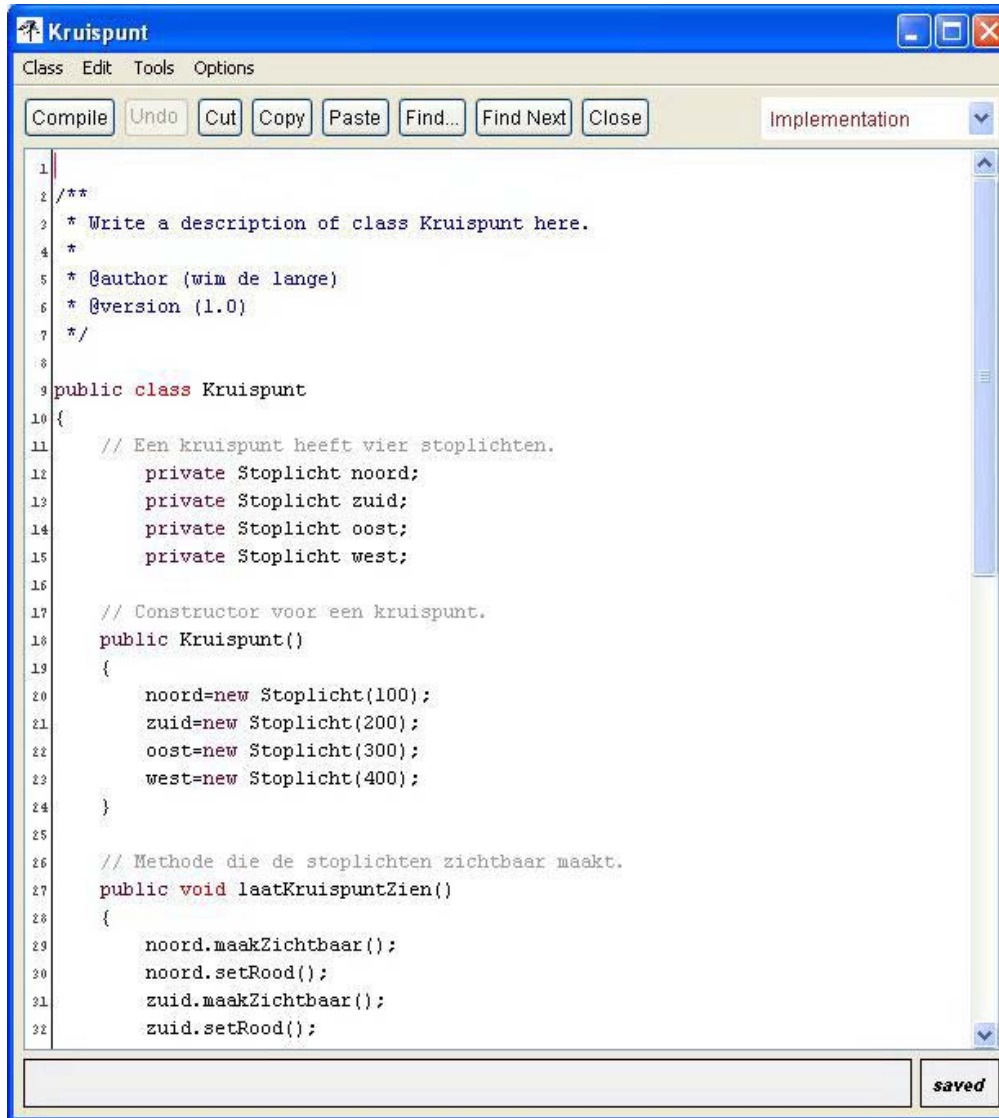
Screenshots



Screenshots



Screenshots

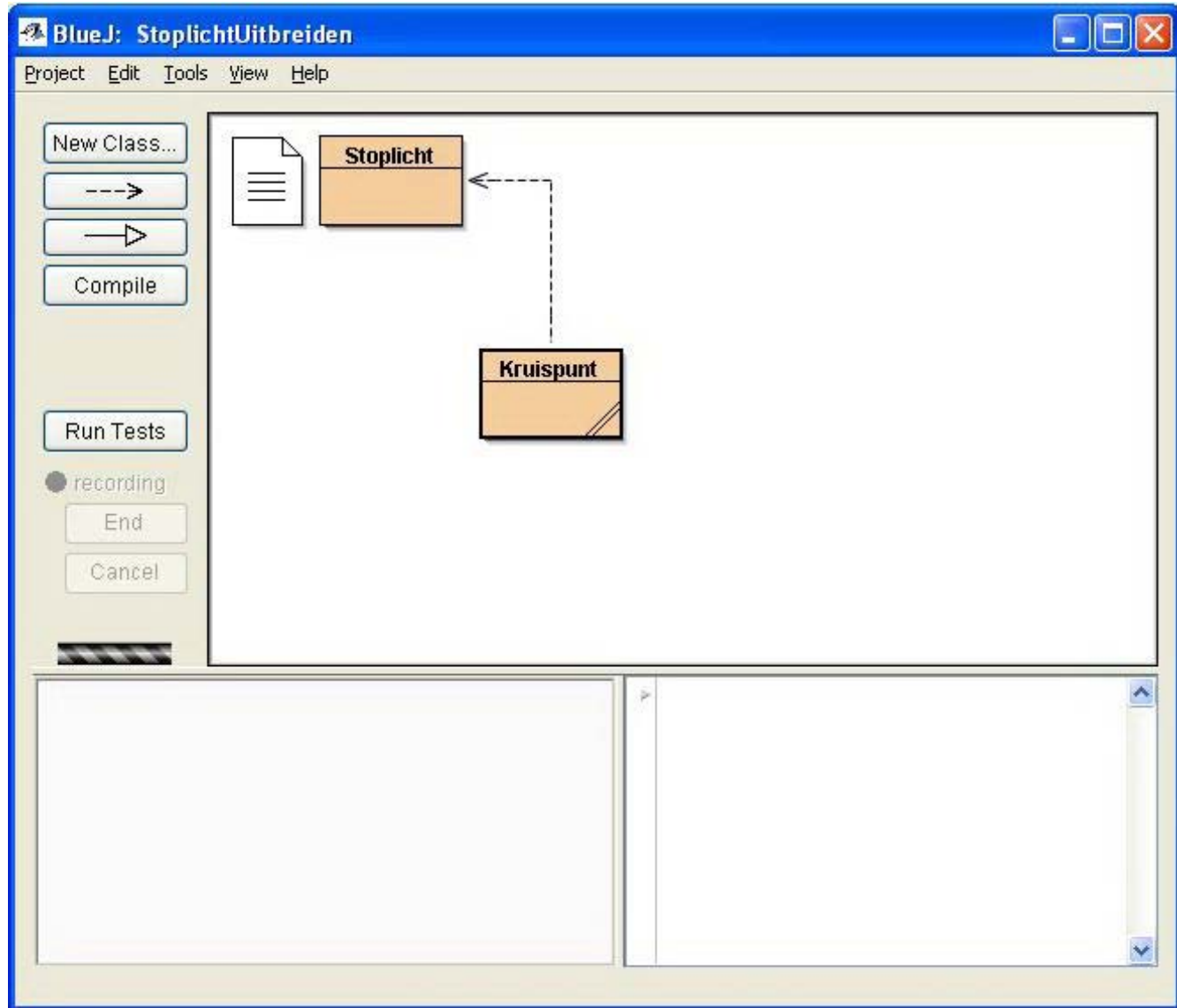


The screenshot shows the JCreator IDE window titled "Kruispunt". The menu bar includes "Class", "Edit", "Tools", and "Options". The toolbar contains buttons for "Compile", "Undo", "Cut", "Copy", "Paste", "Find...", "Find Next", and "Close". A dropdown menu is open, showing "Implementation". The main text area displays the following Java code:

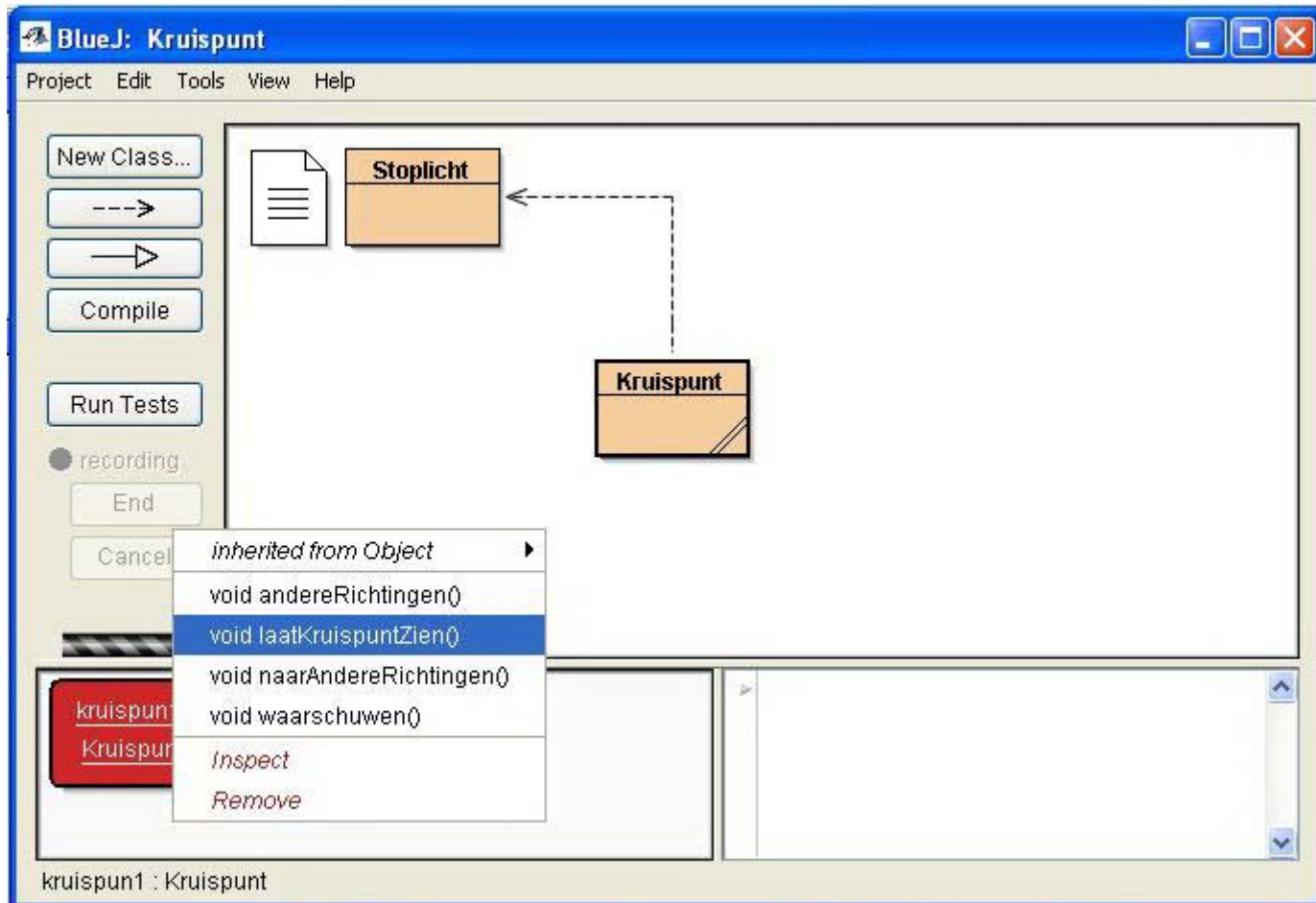
```
1 |
2 | /**
3 |  * Write a description of class Kruispunt here.
4 |  *
5 |  * @author (wim de lange)
6 |  * @version (1.0)
7 |  */
8 |
9 | public class Kruispunt
10 | {
11 |     // Een kruispunt heeft vier stoplichten.
12 |     private Stoplicht noord;
13 |     private Stoplicht zuid;
14 |     private Stoplicht oost;
15 |     private Stoplicht west;
16 |
17 |     // Constructor voor een kruispunt.
18 |     public Kruispunt()
19 |     {
20 |         noord=new Stoplicht(100);
21 |         zuid=new Stoplicht(200);
22 |         oost=new Stoplicht(300);
23 |         west=new Stoplicht(400);
24 |     }
25 |
26 |     // Methode die de stoplichten zichtbaar maakt.
27 |     public void laatKruispuntZien()
28 |     {
29 |         noord.maakZichtbaar();
30 |         noord.setRood();
31 |         zuid.maakZichtbaar();
32 |         zuid.setRood();
```

A "saved" button is visible in the bottom right corner of the IDE window.

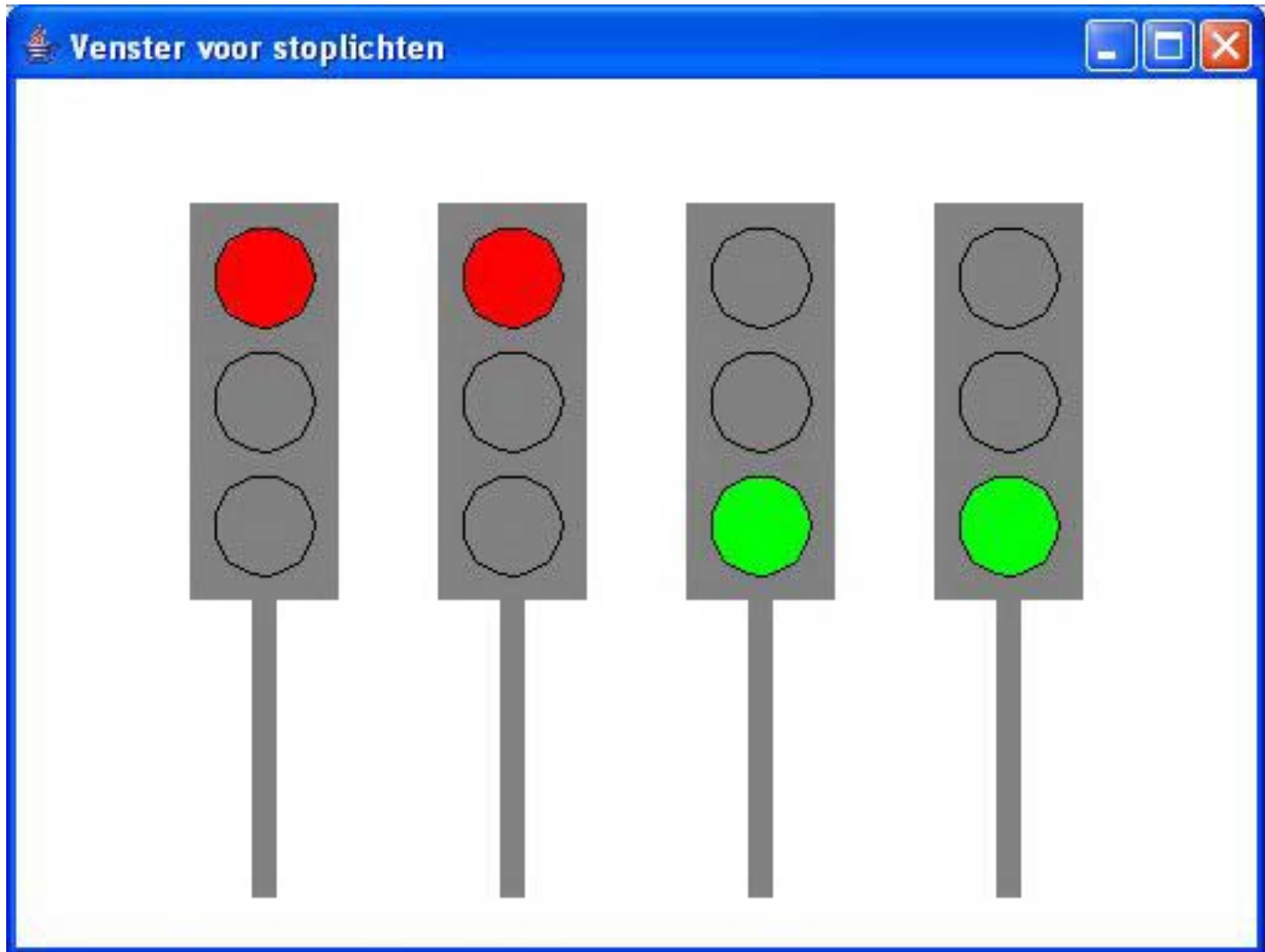
Screenshots



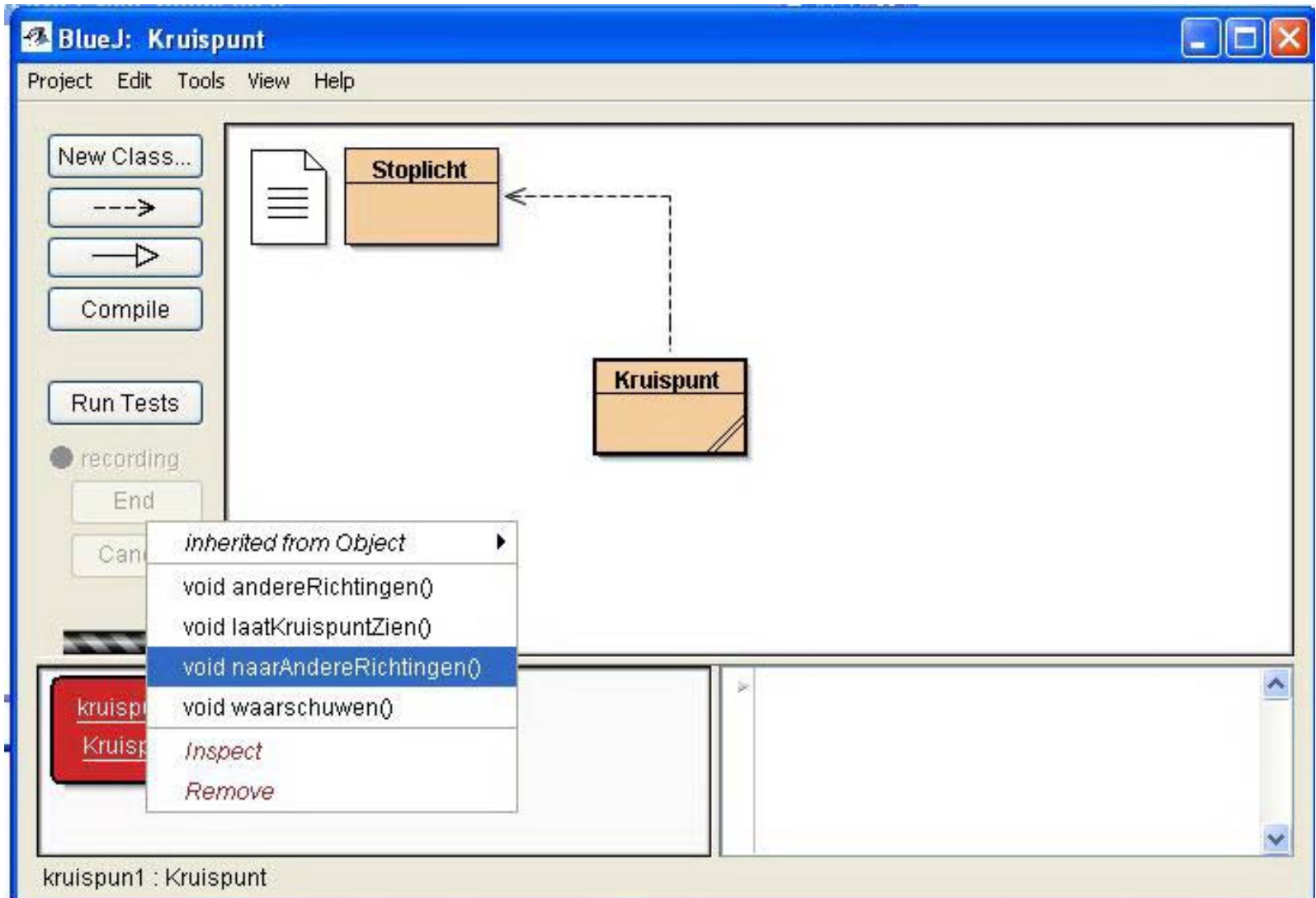
Screenshots



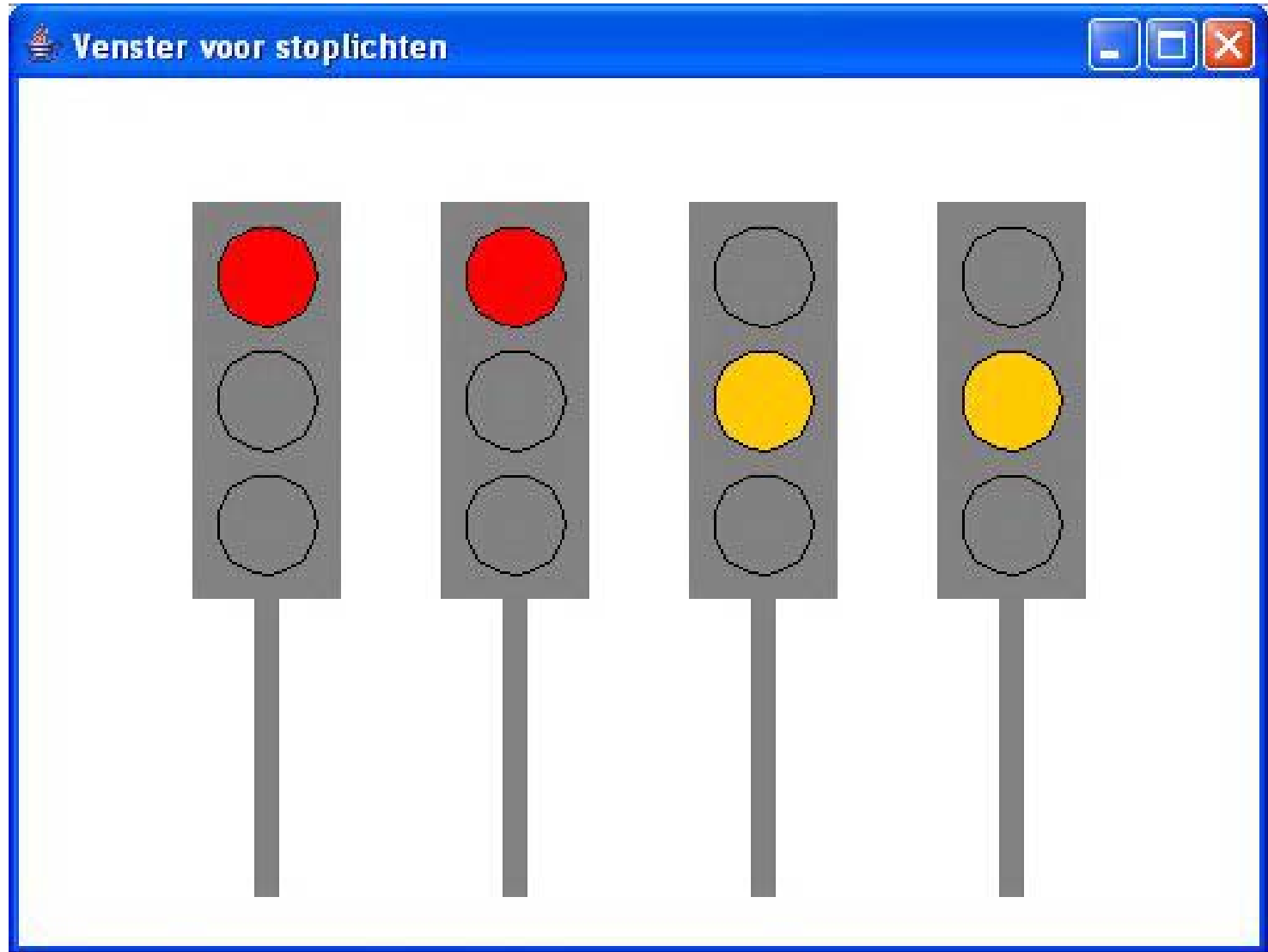
Screenshots



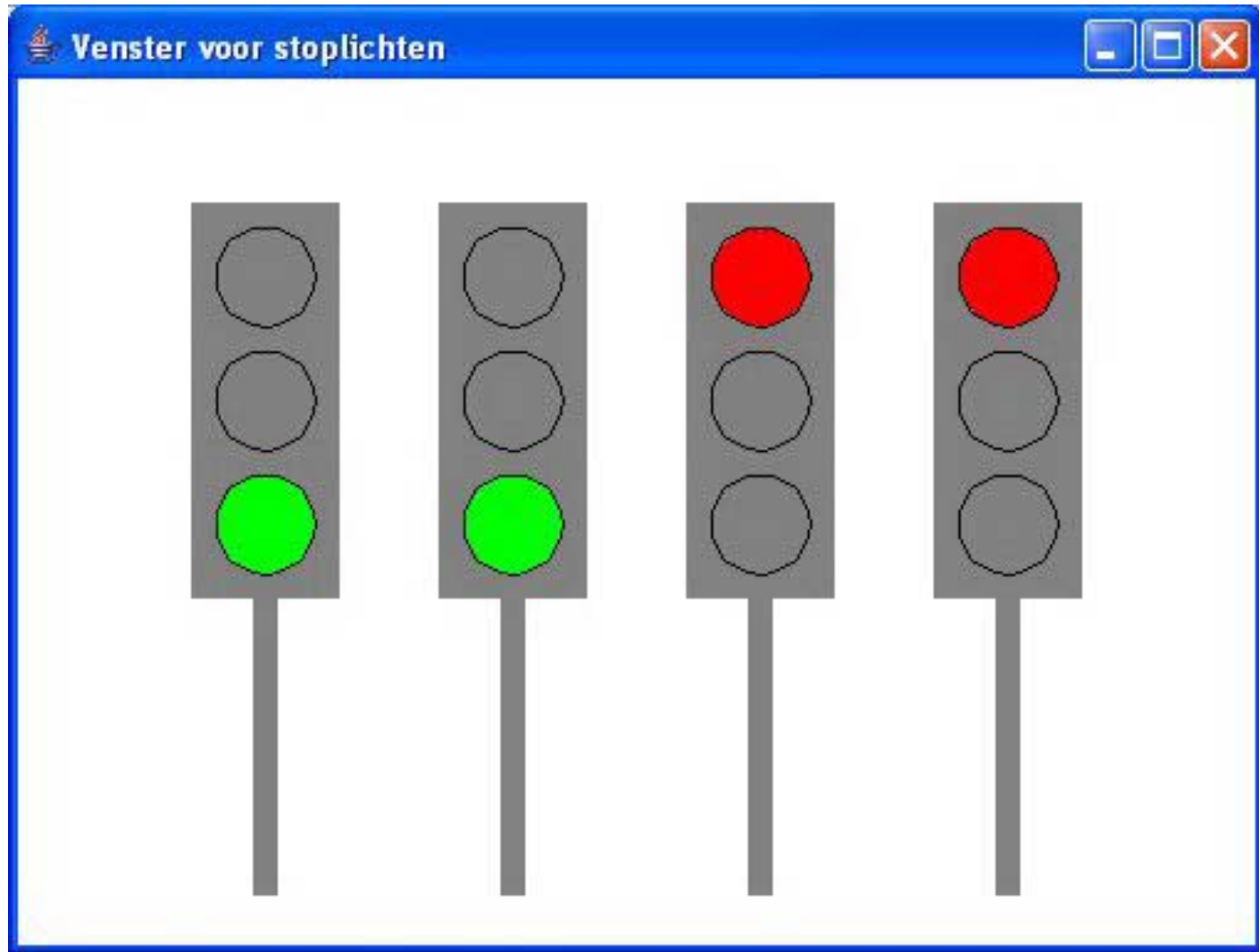
Screenshots



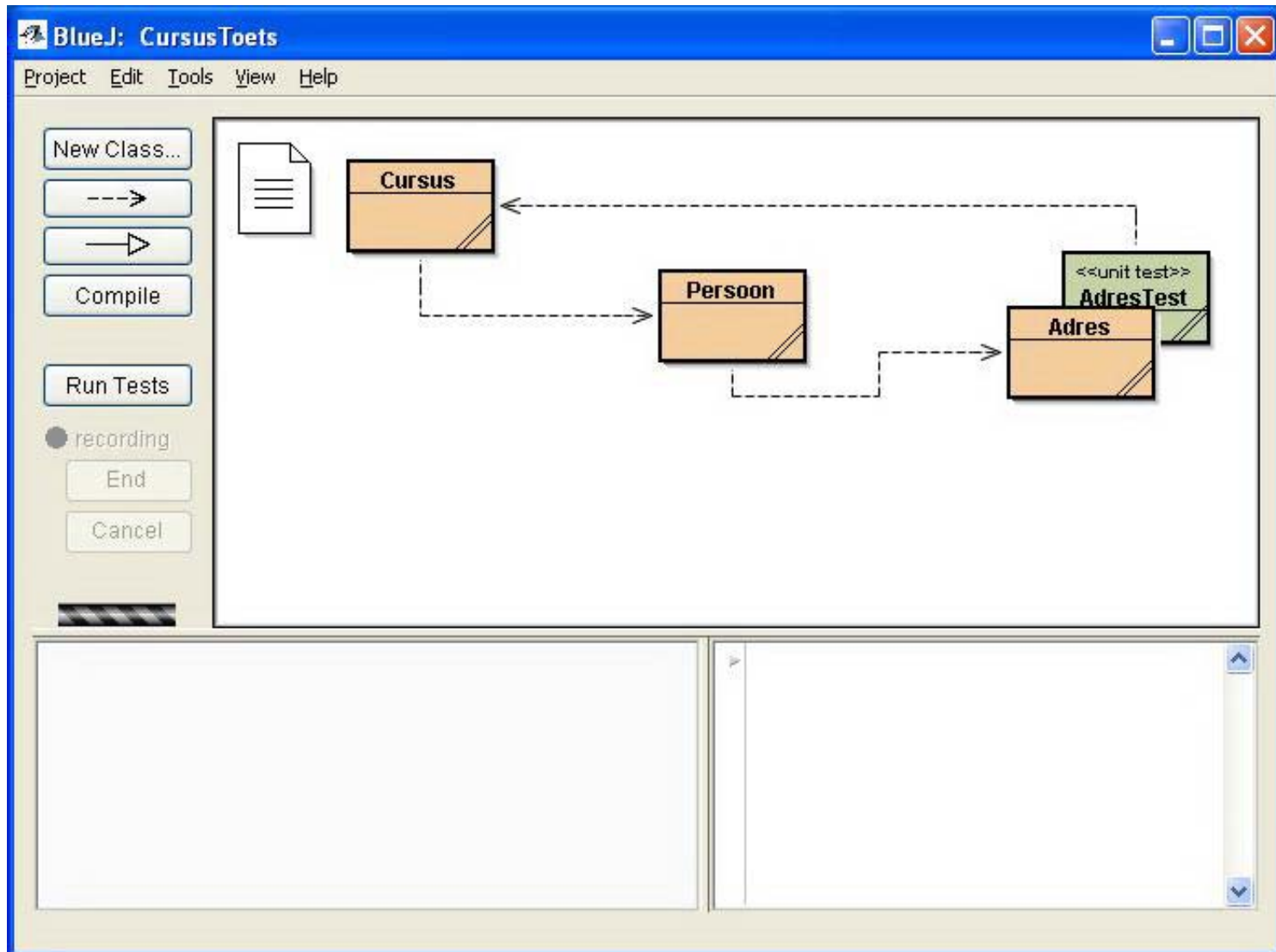
Screenshots



Screenshots



Screenshots



Screenshots

The screenshot shows the BlueJ IDE interface for a project named 'CursusToets'. The main workspace displays a class diagram with three classes: Cursus, Persoon, and Adres. Cursus is connected to Persoon, and Persoon is connected to Adres. A dashed arrow points from Adres back to Cursus. A context menu is open over the Adres class, listing various actions. The 'Test Fixture to Object Bench' option is highlighted in blue. The menu items are: Test All, No test methods, Create Test Method..., Object Bench to Test Fixture, Test Fixture to Object Bench, Open Editor, Compile, Inspect, and Remove. On the left side of the IDE, there are buttons for 'New Class...', 'Compile', 'Run Tests', and a 'recording' indicator with 'End' and 'Cancel' buttons.

Screenshots

The screenshot displays the BlueJ IDE window titled "BlueJ: CursusToets". The menu bar includes "Project", "Edit", "Tools", "View", and "Help". On the left side, there is a toolbar with buttons for "New Class...", "New Class" (dashed arrow), "New Class" (solid arrow), "Compile", "Run Tests", and a "recording" indicator with "End" and "Cancel" buttons. The main workspace shows a class diagram with three classes: "Cursus", "Persoon", and "Adres". "Cursus" is connected to "Persoon" and "Adres" by dashed arrows. "Persoon" is connected to "Adres" by a dashed arrow. A unit test class, "AdresTest", is shown as a green box with the stereotype "unit test" and is connected to "Adres" by a dashed arrow. Below the workspace, there is a variable monitor showing several variables: "cursus1: Cursus", "persoon2: Persoon", "lind: Adres", "persoon1: Persoon", "persoon3: Persoon", "voert: Adres", and "dorpsstr: Adres". The status bar at the bottom left shows "dorpsstr : Adres".

Helaas :

EINDE

Meer over BlueJ op www.bluej.org