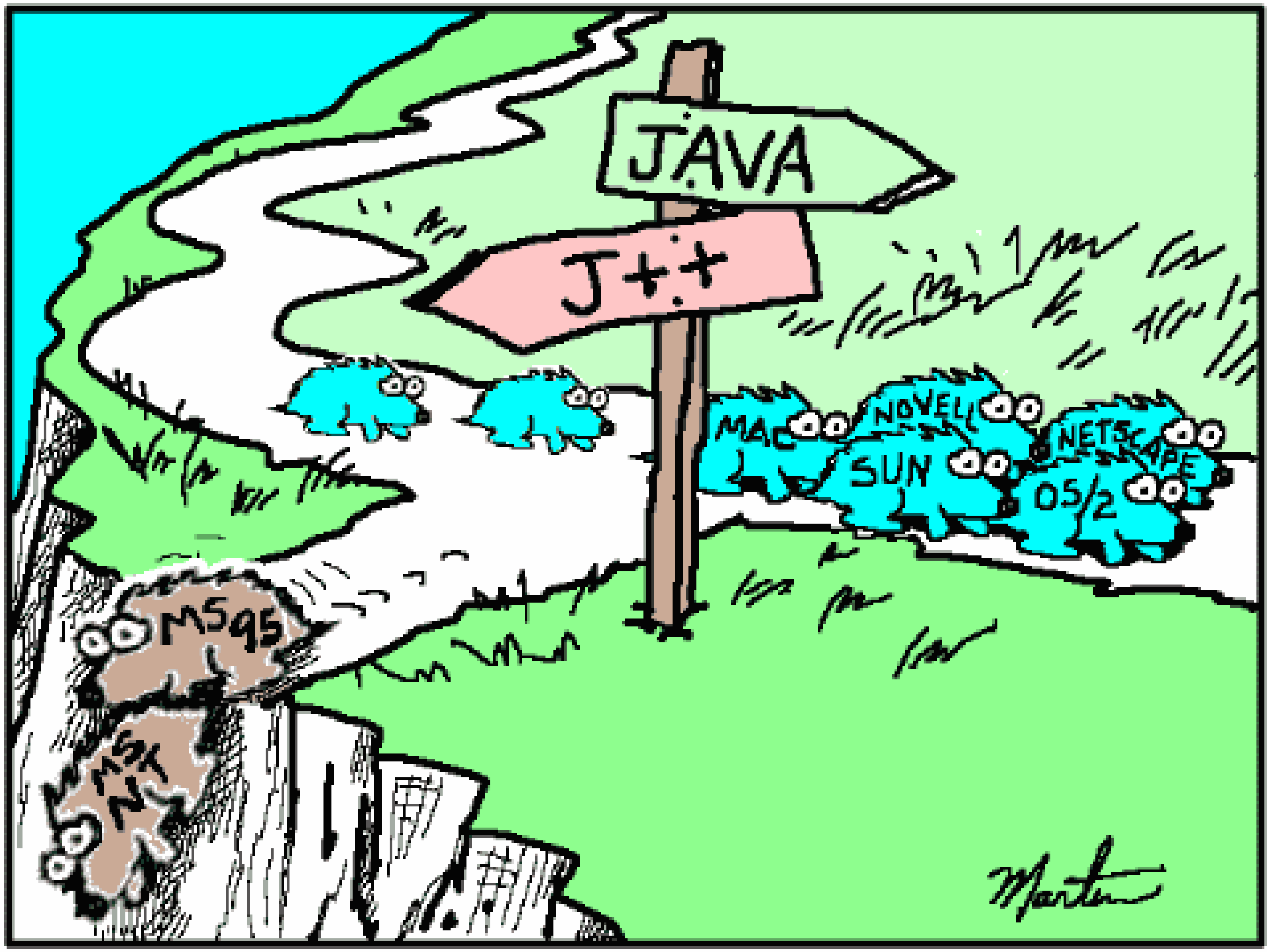
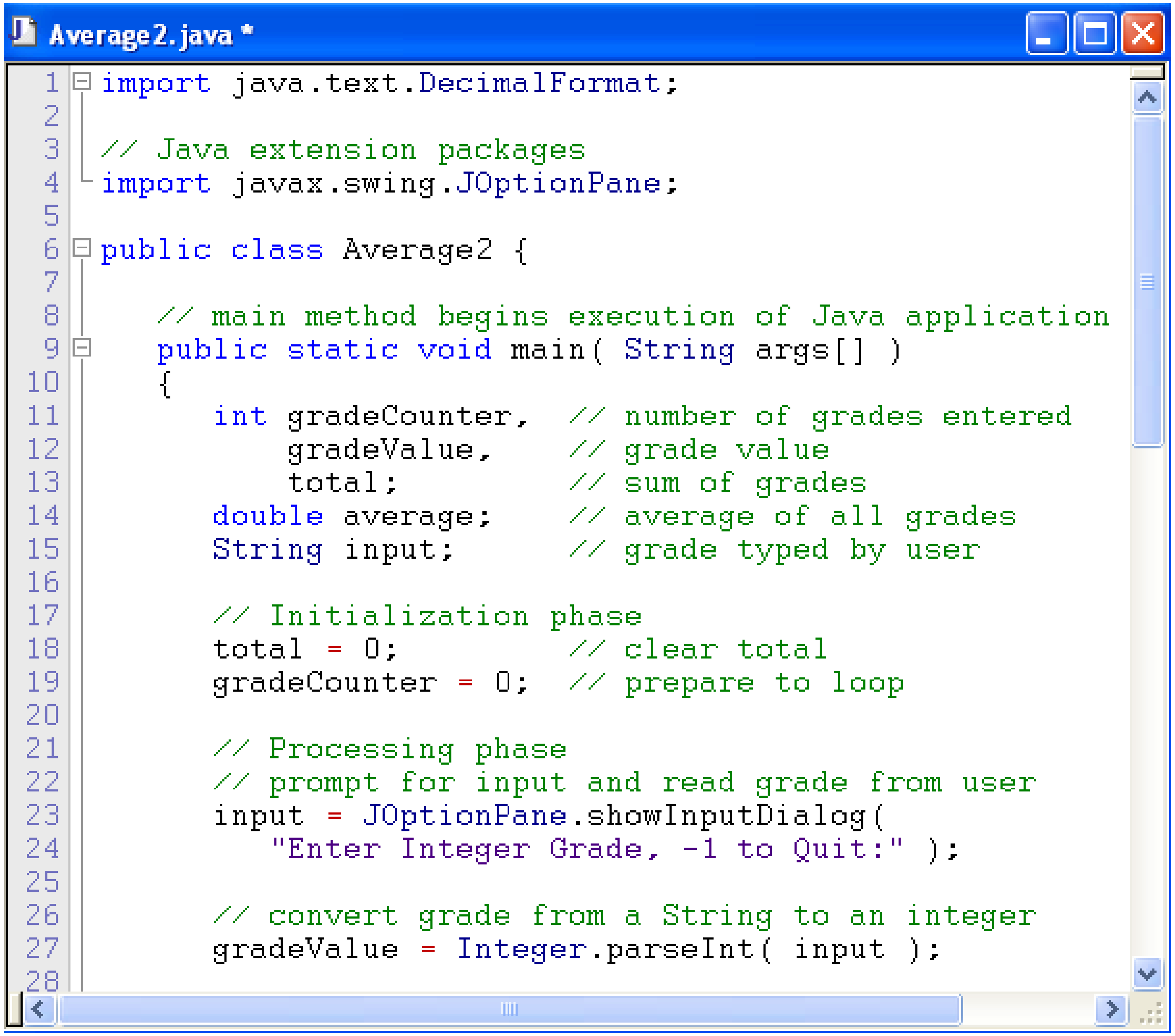
Loopy

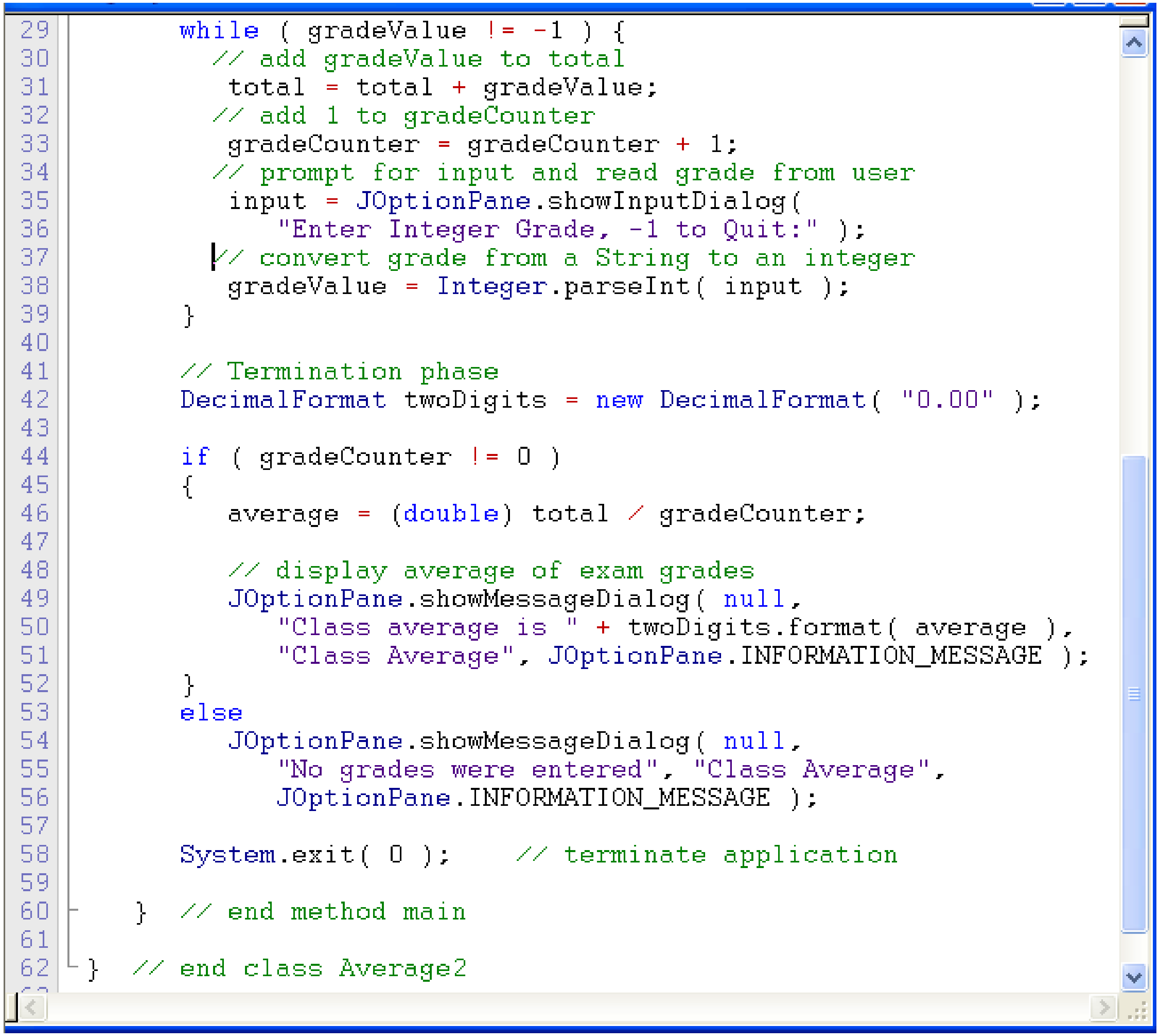


while loop with sentinel control



```java
import java.text.DecimalFormat;

// Java extension packages
import javax.swing.JOptionPane;

public class Average2 {

    // main method begins execution of Java application
    public static void main( String args[] )
    {
        int gradeCounter,  // number of grades entered
            gradeValue,    // grade value
            total;         // sum of grades
        double average;    // average of all grades
        String input;      // grade typed by user

        // Initialization phase
        total = 0;         // clear total
        gradeCounter = 0;  // prepare to loop

        // Processing phase
        // prompt for input and read grade from user
        input = JOptionPane.showInputDialog(
           "Enter Integer Grade, -1 to Quit:" );

        // convert grade from a String to an integer
        gradeValue = Integer.parseInt( input );

```

while loop with sentinel control continued…



```
      while ( gradeValue != -1 ) {
         // add gradeValue to total
         total = total + gradeValue;
         // add 1 to gradeCounter
         gradeCounter = gradeCounter + 1;
         // prompt for input and read grade from user
         input = JOptionPane.showInputDialog(
            "Enter Integer Grade, -1 to Quit:" );
         // convert grade from a String to an integer
         gradeValue = Integer.parseInt( input );
      }

      // Termination phase
      DecimalFormat twoDigits = new DecimalFormat( "0.00" );

      if ( gradeCounter != 0 )
      {
         average = (double) total / gradeCounter;

         // display average of exam grades
         JOptionPane.showMessageDialog( null,
            "Class average is " + twoDigits.format( average ),
            "Class Average", JOptionPane.INFORMATION_MESSAGE );
      }
      else
         JOptionPane.showMessageDialog( null,
            "No grades were entered", "Class Average",
            JOptionPane.INFORMATION_MESSAGE );

      System.exit( 0 );    // terminate application

   }  // end method main

}  // end class Average2
```

Stepwise Refinement

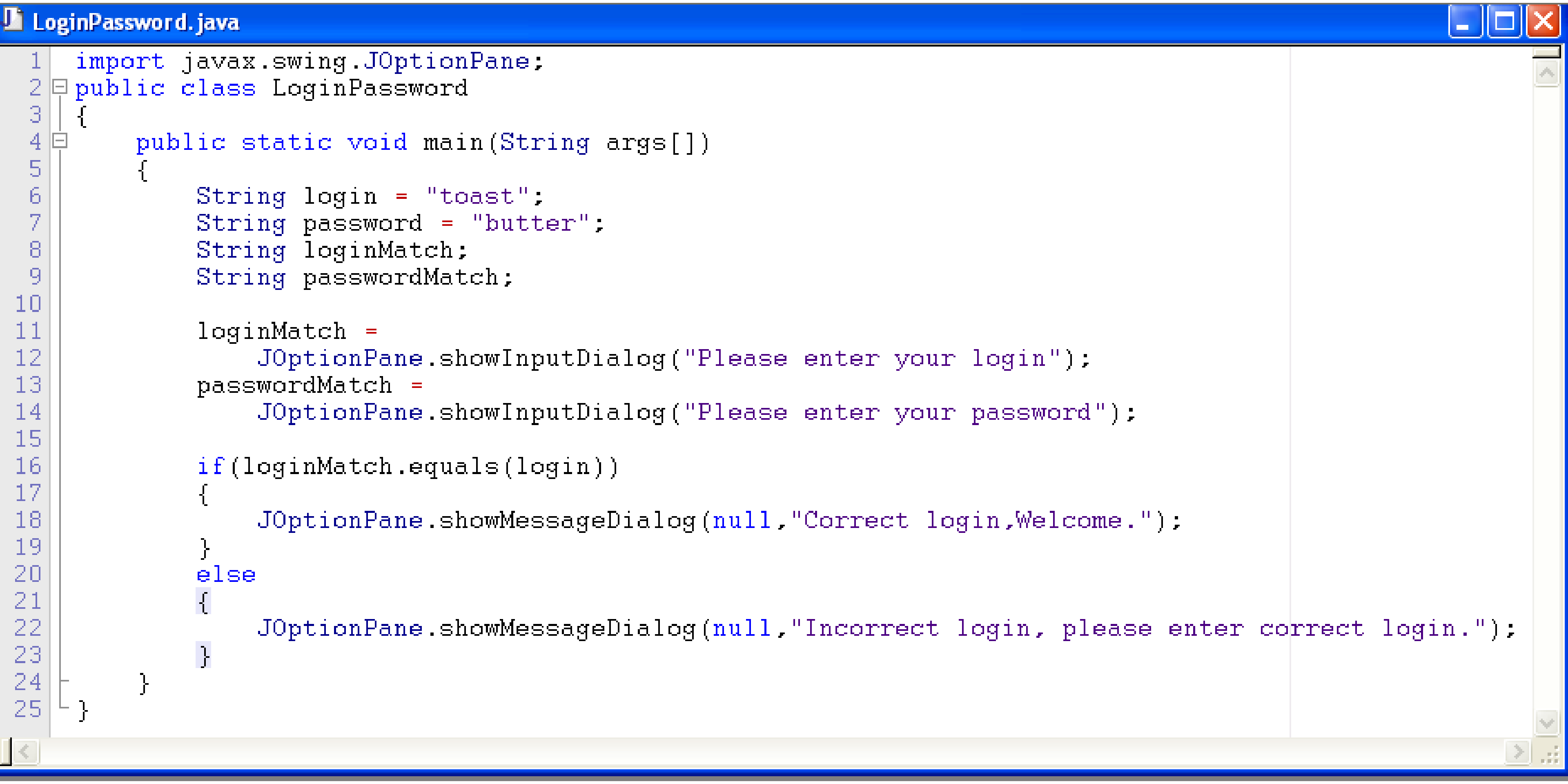
When attempting a complex problem such as that posed in the login/password program begin by breaking the program into smaller steps. Compile, test and debug these smaller sections before adding additional complexity to the program.

The following 3 steps are just a suggestion as to how you might go about breaking a larger program into smaller sections.

1. Have user enter a login and use if statement to determine if correct login was entered.
2. Use a nested loop to determine if password is correct after user has entered correct login.
3. Use a while loop to track the number of incorrect entries and close the program after a certain number of failed attempts.

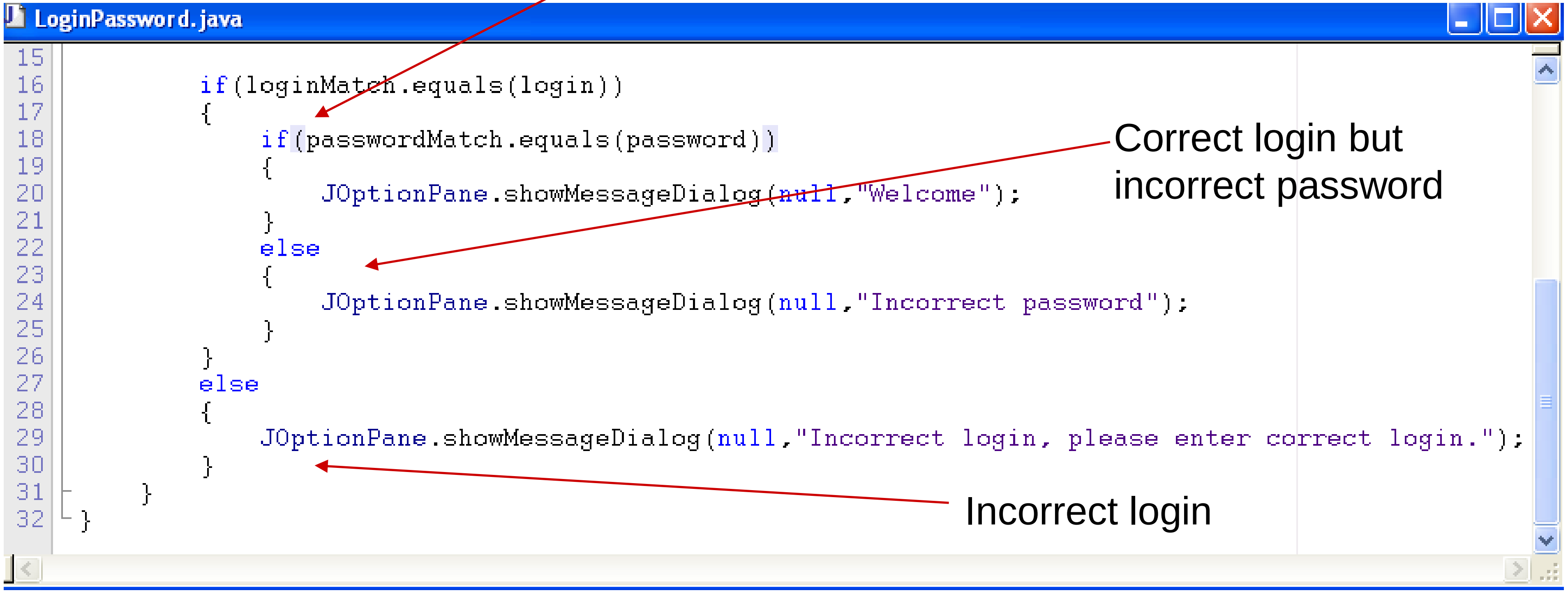
Login and Password

Start Small….code this amount and test with correct and incorrect logins


LoginPassword.java

```java
import javax.swing.JOptionPane;
public class LoginPassword
{
	public static void main(String args[])
	{
		String login = "toast";
		String password = "butter";
		String loginMatch;
		String passwordMatch;

		loginMatch =
			JOptionPane.showInputDialog("Please enter your login");
		passwordMatch =
			JOptionPane.showInputDialog("Please enter your password");

		if(loginMatch.equals(login))
		{
			JOptionPane.showMessageDialog(null,"Correct login,Welcome.");
		}
		else
		{
			JOptionPane.showMessageDialog(null,"Incorrect login, please enter correct login.");
		}
	}
}
```

Add a little more complexity



```
        if(loginMatch.equals(login))
        {
            if(passwordMatch.equals(password))
            {
                JOptionPane.showMessageDialog(null,"Welcome");
            }
            else
            {
                JOptionPane.showMessageDialog(null,"Incorrect password");
            }
        }
        else
        {
            JOptionPane.showMessageDialog(null,"Incorrect login, please enter correct login.");
        }
    }
}
```

Nested if statement to reflect a correct login AND password

Correct login but incorrect password

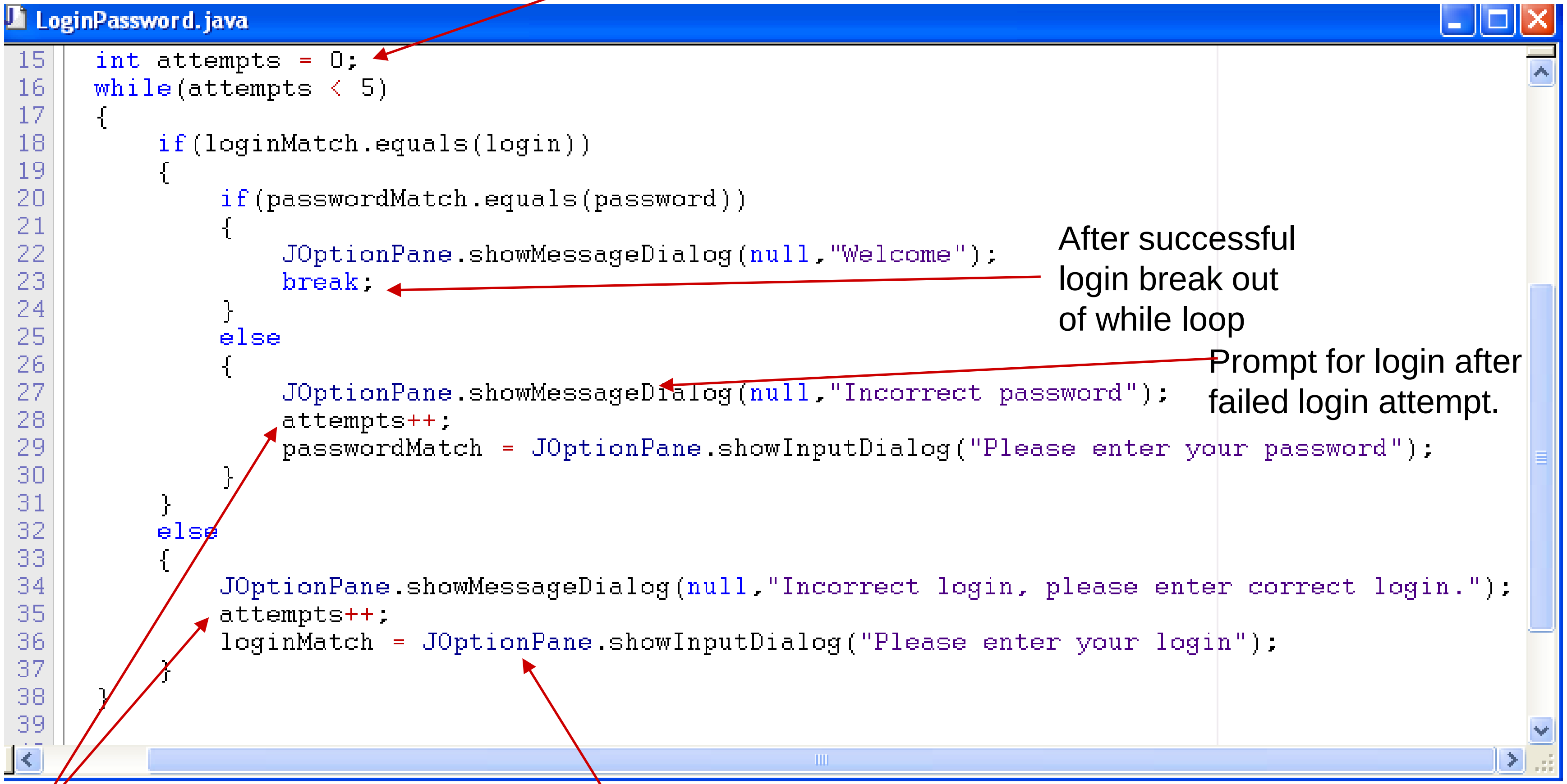
Incorrect login

Test with every combination of correct and incorrect password/login combinations to ensure you are getting the expected results.

Limit attempts using while loop

Encompass entire if statements block with a while loop that exits after 5 failed attempts.

Declare attempts variable to track failed attempts



```java
int attempts = 0;
while(attempts < 5)
{
    if(loginMatch.equals(login))
    {
        if(passwordMatch.equals(password))
        {
            JOptionPane.showMessageDialog(null,"Welcome");
            break;
        }
        else
        {
            JOptionPane.showMessageDialog(null,"Incorrect password");
            attempts++;
            passwordMatch = JOptionPane.showInputDialog("Please enter your password");
        }
    }
    else
    {
        JOptionPane.showMessageDialog(null,"Incorrect login, please enter correct login.");
        attempts++;
        loginMatch = JOptionPane.showInputDialog("Please enter your login");
    }
}
```

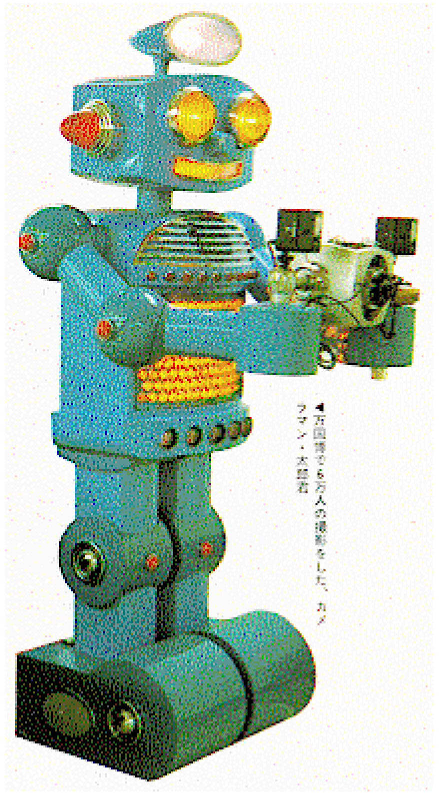
After successful login break out of while loop

Prompt for login after failed login attempt.

Failed login/password attempts result in attempts variable being incremented

Prompt for password after failed password attempt.

Robots

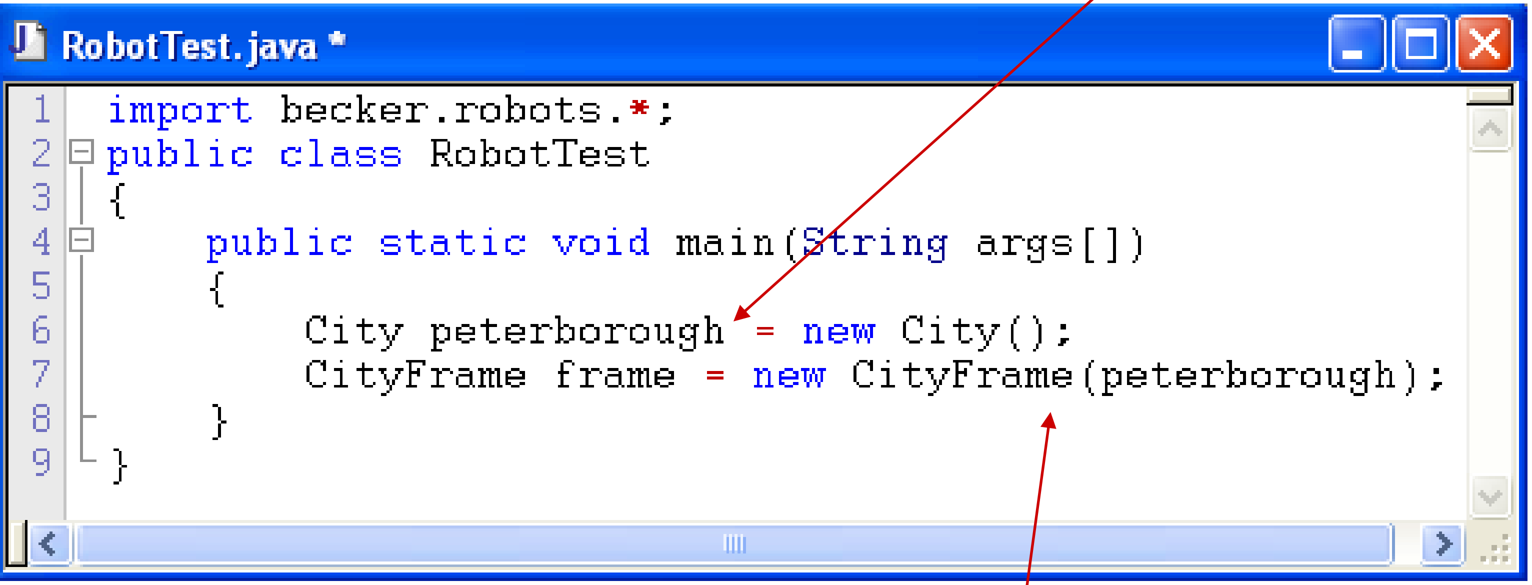


Make sure you have followed the directions in the previous lesson for downloading and installing the robot class packages (becker.jar)

Anatomy of a robot program

Here is the bare minimum code required to create a city and display it in a frame

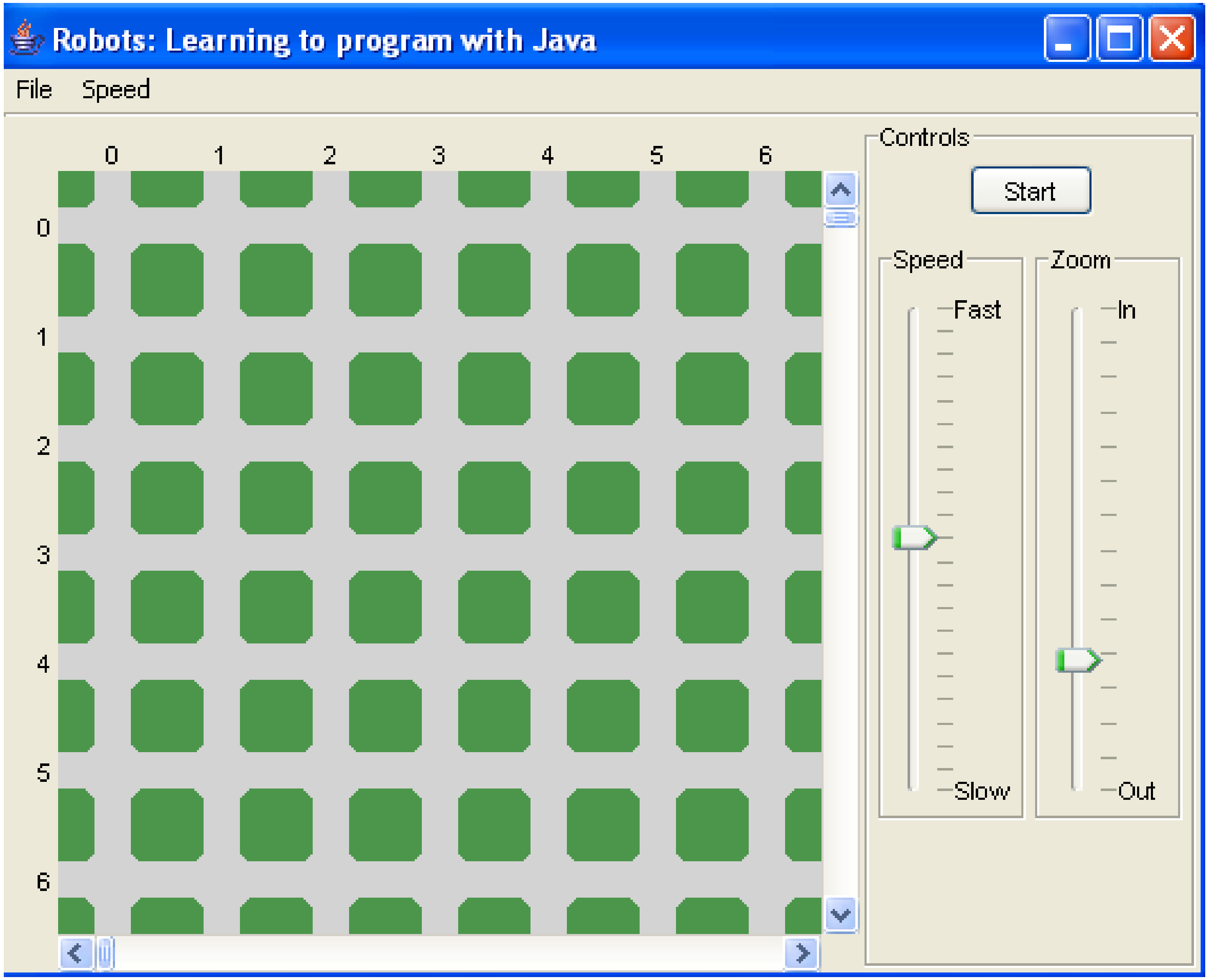
Declare city



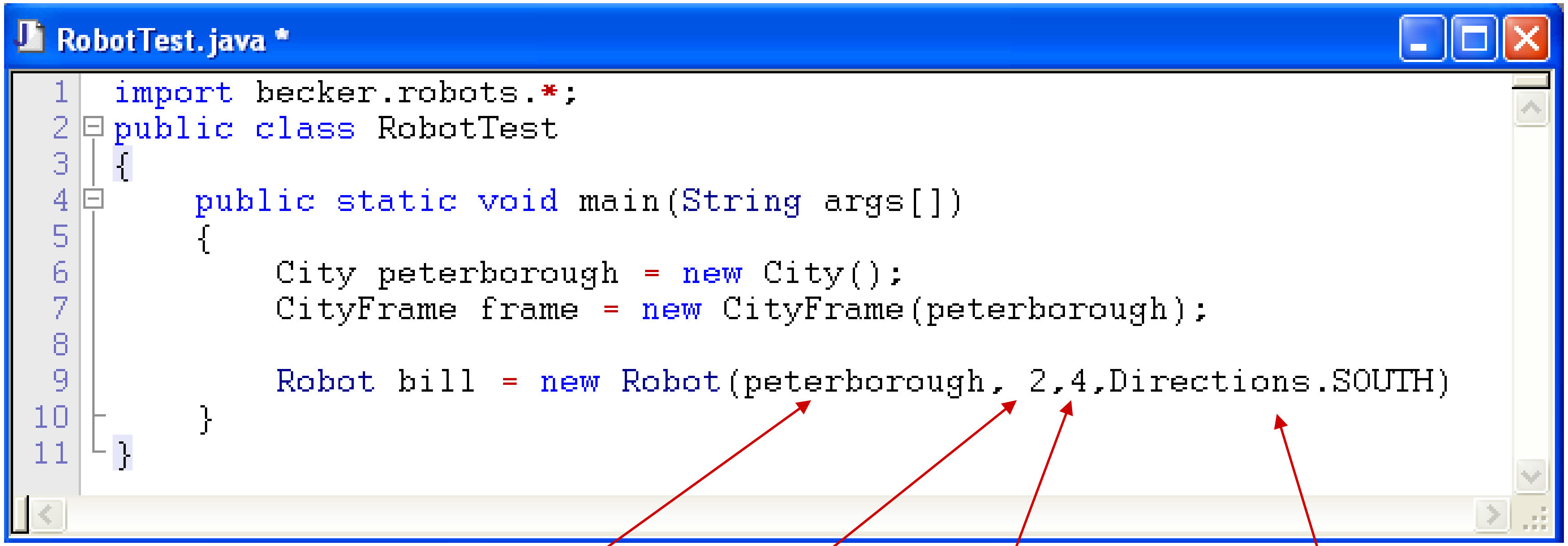
```java
import becker.robots.*;
public class RobotTest
{
    public static void main(String args[])
    {
        City peterborough = new City();
        CityFrame frame = new CityFrame(peterborough);
    }
}
```

Embed city in a frame

The result…an empty city.



Lets add a robot…we'll call him bill



```
import becker.robots.*;
public class RobotTest
{
    public static void main(String args[])
    {
        City peterborough = new City();
        CityFrame frame = new CityFrame(peterborough);

        Robot bill = new Robot(peterborough, 2,4,Directions.SOUTH)
    }
}
```

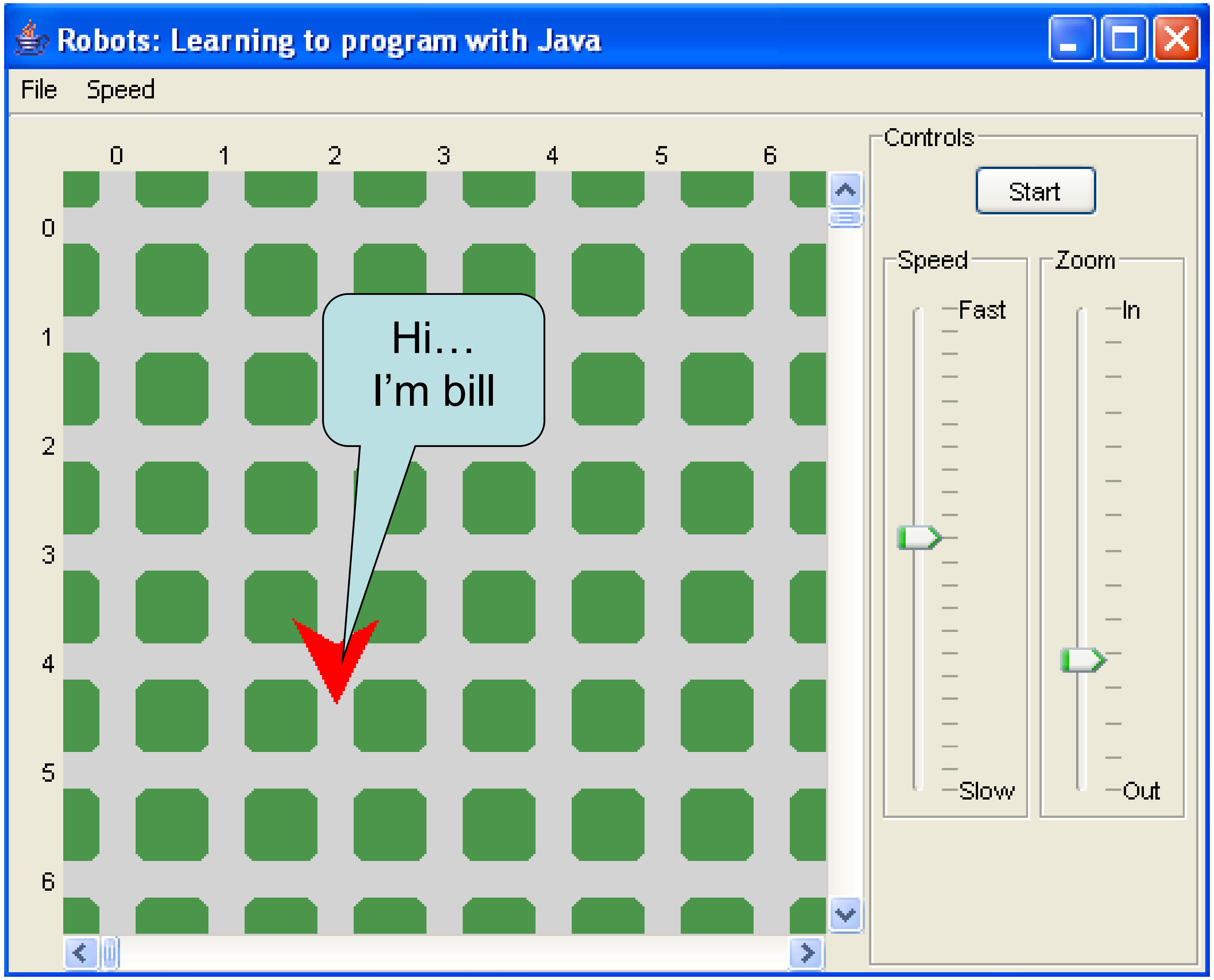
Parameter 1: city where robot exists

Parameter 2: x axis

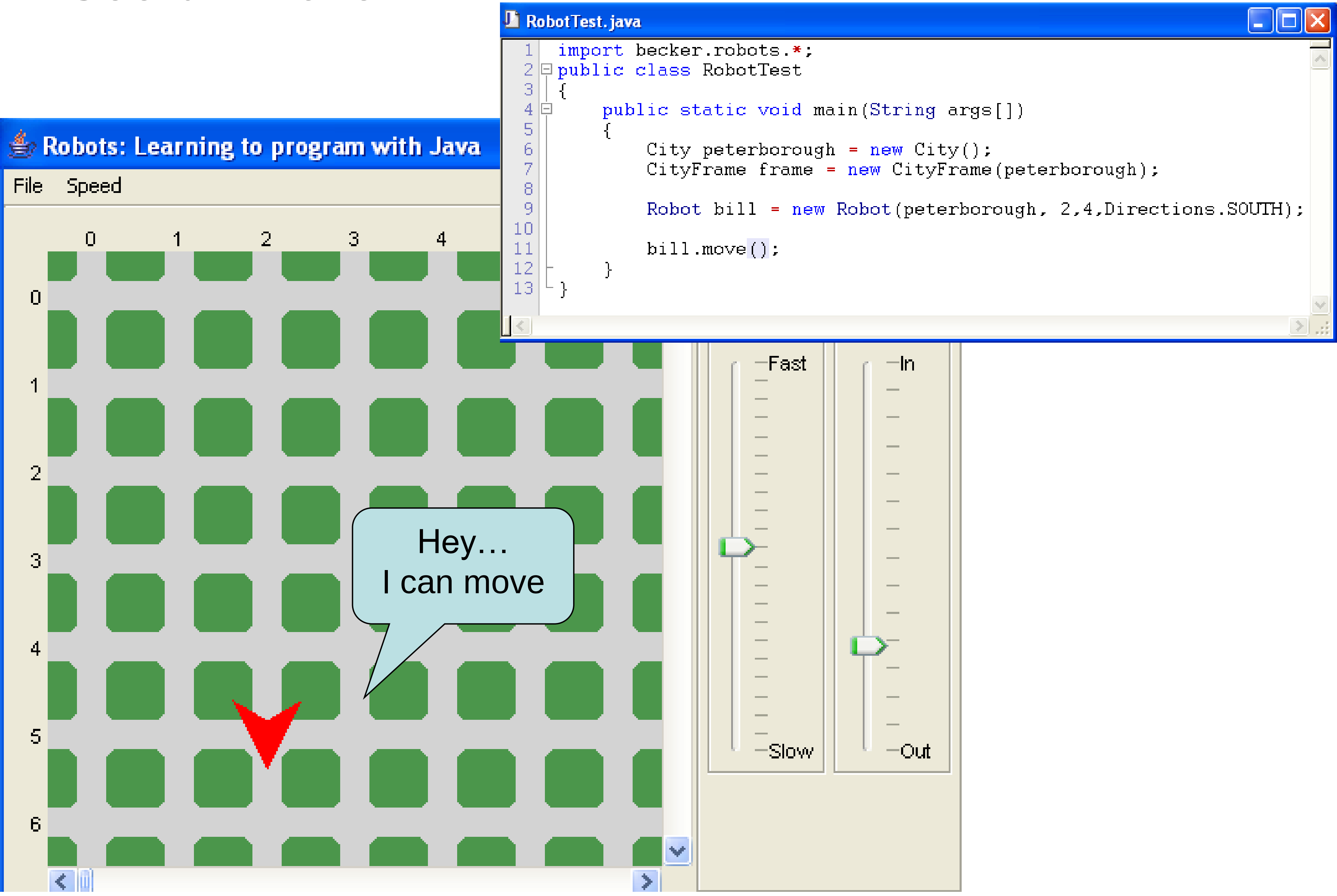
Parameter 3: y axis

Parameter 4: Direction robot is facing

Meet bill….resident of Peterborough

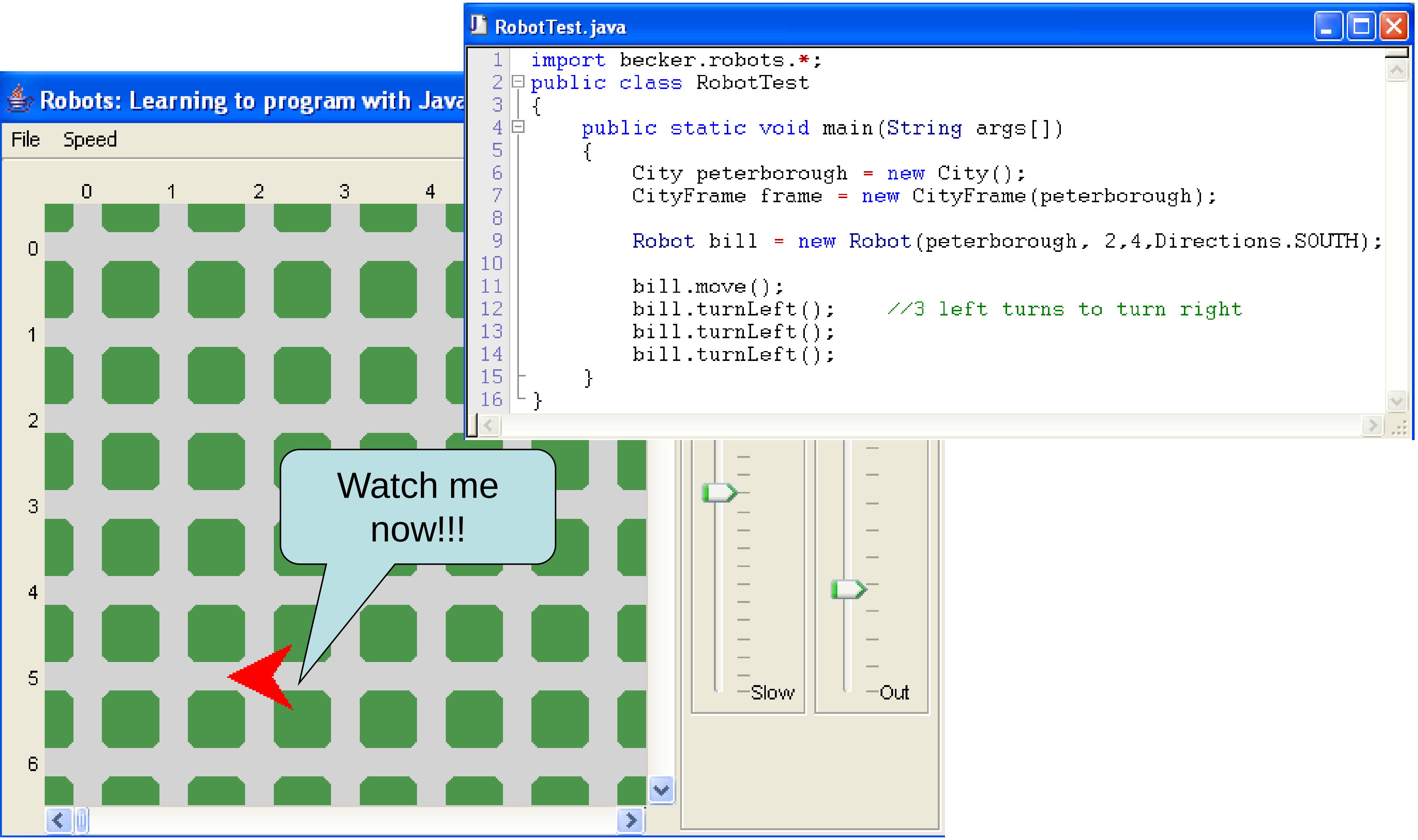


See bill move…



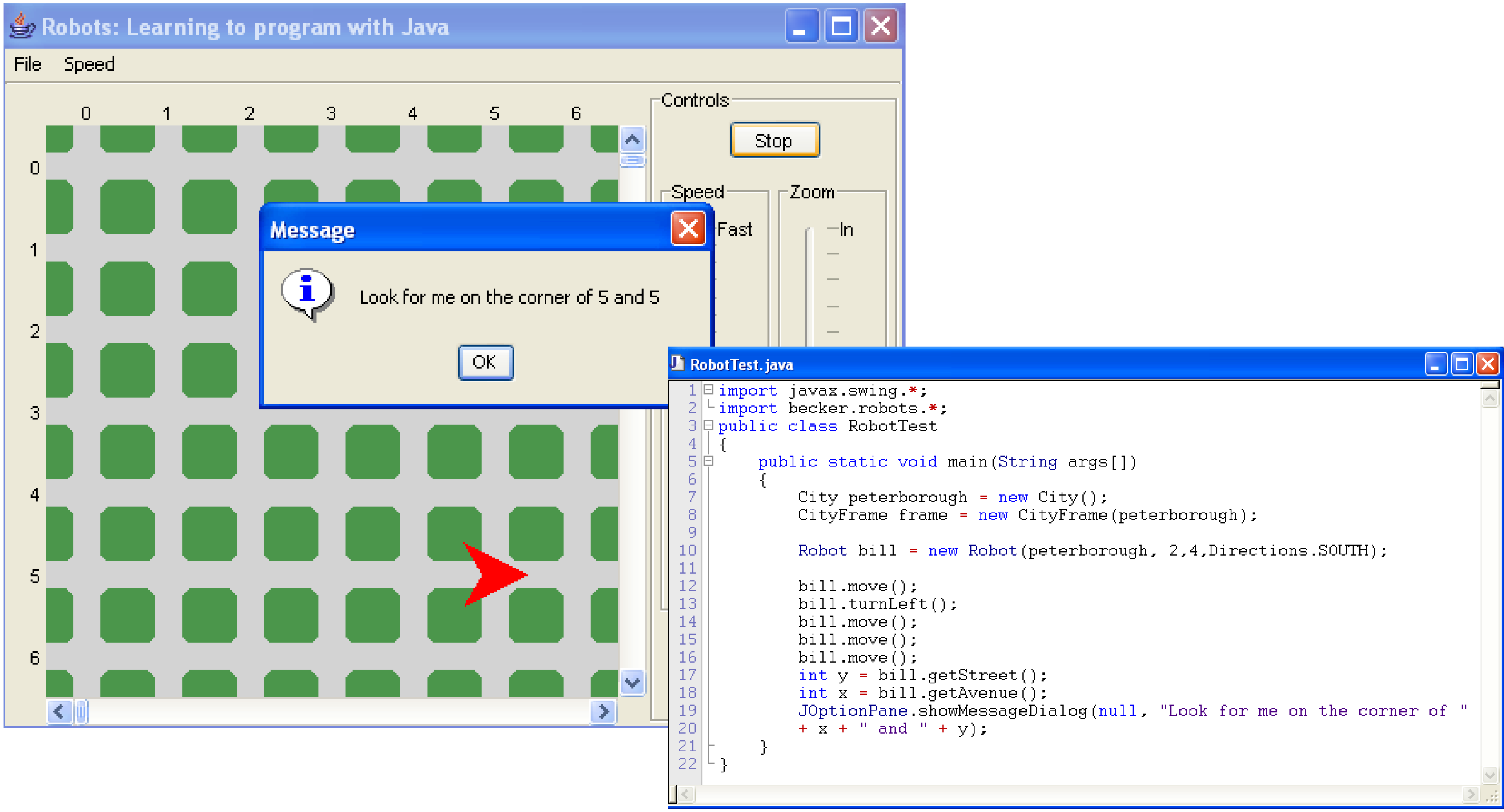
```java
import becker.robots.*;
public class RobotTest
{
    public static void main(String args[])
    {
        City peterborough = new City();
        CityFrame frame = new CityFrame(peterborough);

        Robot bill = new Robot(peterborough, 2,4,Directions.SOUTH);

        bill.move();
    }
}
```

See bill turn….



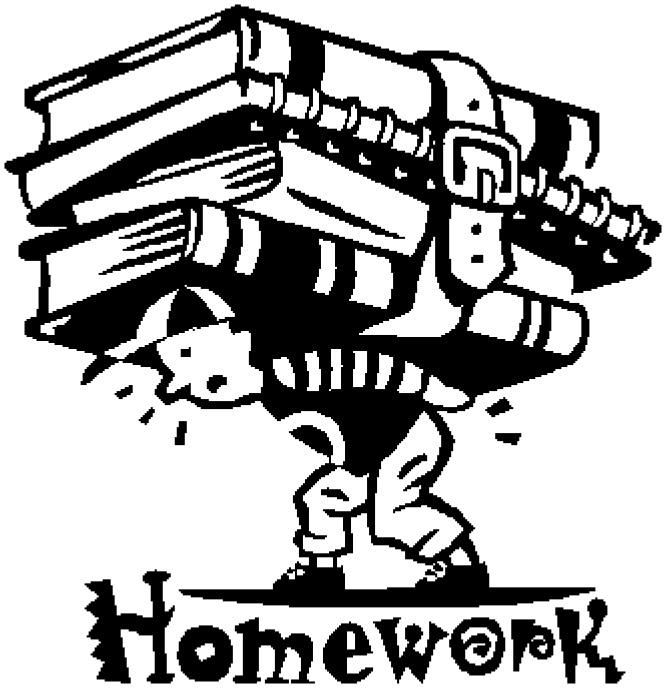
```java
import becker.robots.*;
public class RobotTest
{
      public static void main(String args[])
      {
            City peterborough = new City();
            CityFrame frame = new CityFrame(peterborough);

            Robot bill = new Robot(peterborough, 2,4,Directions.SOUTH);

            bill.move();
            bill.turnLeft();     //3 left turns to turn right
            bill.turnLeft();
            bill.turnLeft();
      }
}
```

Where is bill?



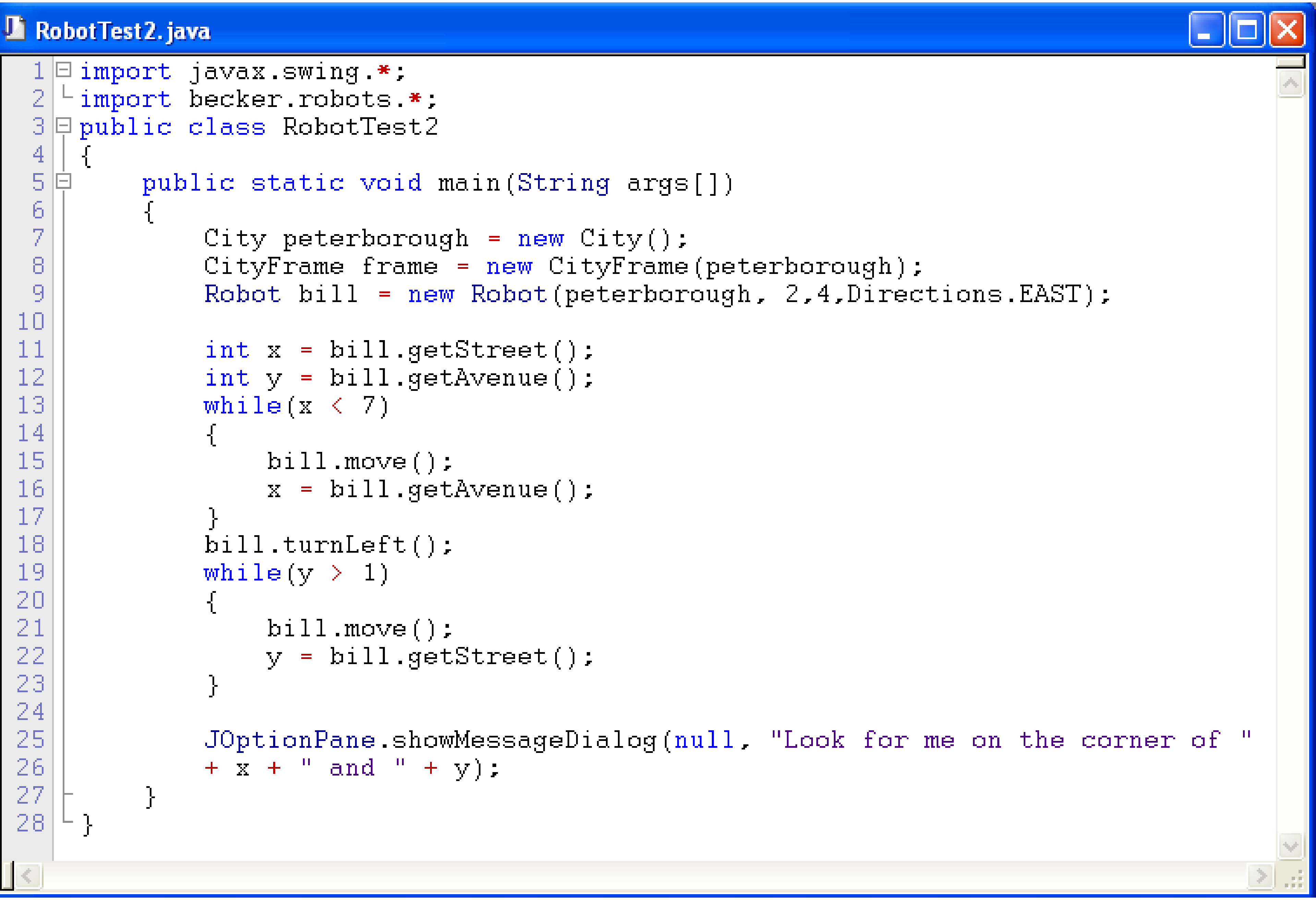
Look for me on the corner of 5 and 5

```java
import javax.swing.*;
import becker.robots.*;
public class RobotTest
{
    public static void main(String args[])
    {
        City peterborough = new City();
        CityFrame frame = new CityFrame(peterborough);

        Robot bill = new Robot(peterborough, 2,4,Directions.SOUTH);

        bill.move();
        bill.turnLeft();
        bill.move();
        bill.move();
        bill.move();
        int y = bill.getStreet();
        int x = bill.getAvenue();
        JOptionPane.showMessageDialog(null, "Look for me on the corner of "
        + x + " and " + y);
    }
}
```



Copy the code in the next slide. Now experiment with moving bill around the city using loops and if statements.

Home Work Code



RobotTest2.java

```java
import javax.swing.*;
import becker.robots.*;
public class RobotTest2
{
	public static void main(String args[])
	{
		City peterborough = new City();
		CityFrame frame = new CityFrame(peterborough);
		Robot bill = new Robot(peterborough, 2,4,Directions.EAST);

		int x = bill.getStreet();
		int y = bill.getAvenue();
		while(x < 7)
		{
			bill.move();
			x = bill.getAvenue();
		}
		bill.turnLeft();
		while(y > 1)
		{
			bill.move();
			y = bill.getStreet();
		}

		JOptionPane.showMessageDialog(null, "Look for me on the corner of "
		+ x + " and " + y);
	}
}
```