

Programming Environments

DrJava is not Java! It is an integrated environment for developing Java programs

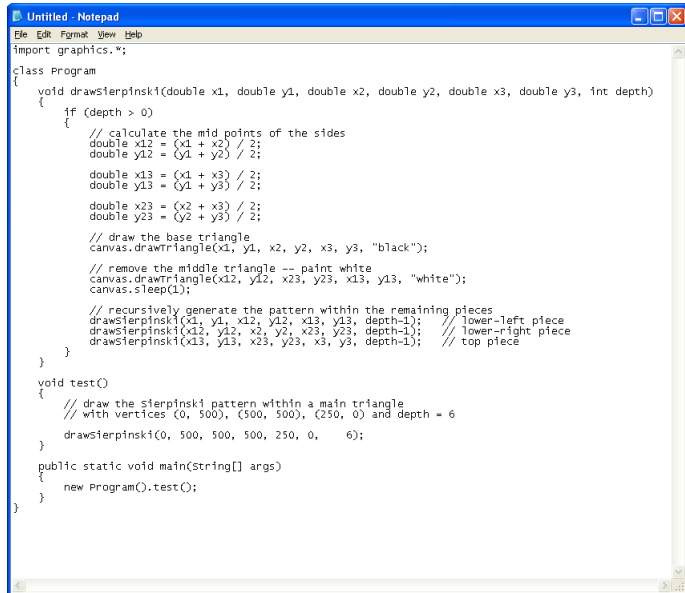
DrJava integrates the Edit => Compile => Run cycle

Other: jGrasp, BlueJ, Eclipse

Java Program Development Cycle

Edit => Compile => Run

Edit: create Program.java



```
Untitled - Notepad
File Edit Format View Help
import graphics.*;

class Program
{
    void drawSierpinski(double x1, double y1, double x2, double y2, double x3, double y3, int depth)
    {
        if (depth > 0)
        {
            // calculate the mid points of the sides
            double x12 = (x1 + x2) / 2;
            double y12 = (y1 + y2) / 2;
            double x13 = (x1 + x3) / 2;
            double y13 = (y1 + y3) / 2;
            double x23 = (x2 + x3) / 2;
            double y23 = (y2 + y3) / 2;

            // draw the base triangle
            canvas.drawTriangle(x1, y1, x2, y2, x3, y3, "black");

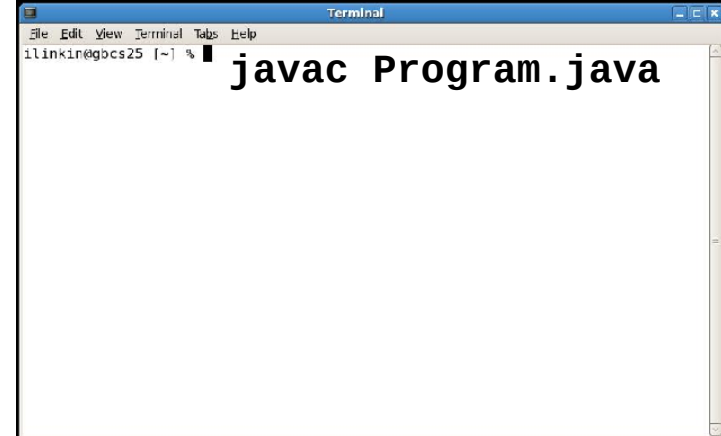
            // remove the middle triangle -- paint white
            canvas.drawTriangle(x12, y12, x23, y23, x13, y13, "white");
            canvas.sleep(1);

            // recursively generate the pattern within the remaining pieces
            drawSierpinski(x1, y1, x12, y12, x13, y13, depth-1); // lower-left piece
            drawSierpinski(x12, y12, x2, y2, x23, y23, depth-1); // lower-right piece
            drawSierpinski(x13, y13, x23, y23, x3, y3, depth-1); // top piece
        }
    }

    void test()
    {
        // draw the Sierpinski pattern within a main triangle
        // with vertices (0, 500), (500, 500), (250, 0) and depth = 6
        drawSierpinski(0, 500, 500, 500, 250, 0, 6);
    }

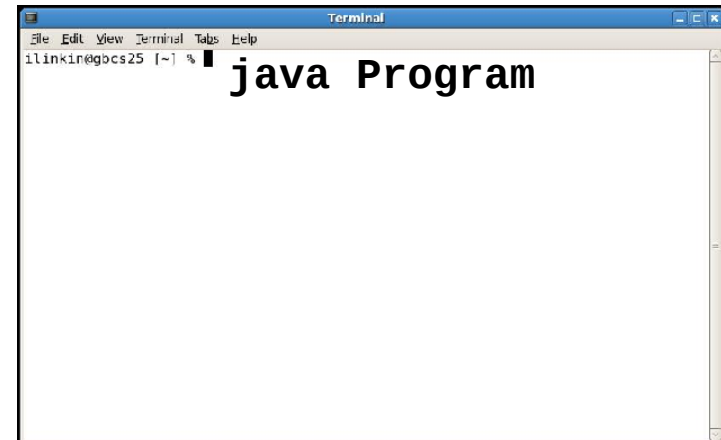
    public static void main(String[] args)
    {
        new Program().test();
    }
}
```

Compile: create Program.class



```
Terminal
ilinkin@gbcs25 [~] % javac Program.java
```

Run



```
Terminal
ilinkin@gbcs25 [~] % java Program
```

Program.java – human readable

Program.class – machine readable

Java Primitive Types

- **boolean** – for true/false values
- **char** – for single character values
- **short, int, long** – for integer values

short [-32,768 ... 32,767]

int [-2,147,483,648 ... 2,147,483,647]

long [-9,223,372,036,854,775,808 ... 9,223,372,036,854,775,807]

- **float, double** – for real values

float has smaller range than **double**

In how many ways can you say $x = x + 1$?

$x = x + 1;$

$x = x - 1;$

$x = x * 5;$

$x = x / 5;$

$x += 1;$

$x -= 1;$

$x *= 5;$

$x /= 5;$

$x++;$

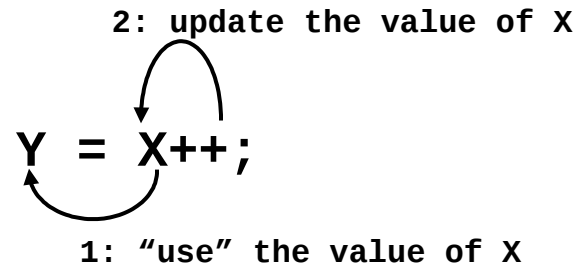
$x--;$

$++x;$

$--x;$

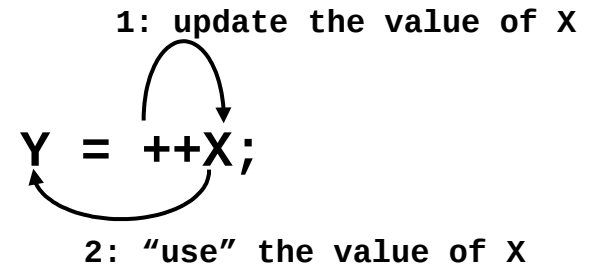
X++ vs ++X

X = 5;



Y is now 5
X is now 6

X = 5;



X is now 6
Y is also 6

X++ vs ++X

X = 5;

System.out.println(X++);

2: update X

1: "use" the value of X

displays 5
X is now 6

X = 5;

System.out.println(++X);

1: update X

2: "use" the value of X

X is now 6
displays 6

Integer Division

- Dividing integers always yields integer value

$$7 / 3 = 2$$

```
int x = 7;  
int y = 3;
```

```
int z1 = x / y;           // z1 is now 2
```

```
double z2 = x / y;       // z2 is now 2
```

- Remainder operator %

$$7 / 3 = 2 \quad \text{and 1 left over, so} \quad 7 \% 3 = 1$$

$$9 / 5 = 1 \quad \text{and 4 left over, so} \quad 9 \% 5 = 4$$

$$21 / 5 = 4 \quad \text{and 1 left over, so} \quad 21 \% 5 = 1$$

Remainder % Operator Use

```
if ( n % 2 == 0 )  
{  
    System.out.println( "n is even " );  
}
```

```
if ( n % 5 == 0 )  
{  
    System.out.println( "n is divisible by 5 " );  
}
```

```
player = ( player + 1 ) % n;
```

do-while loop

```
while ( something-is-true )  
{  
    action 1;  
    action 2;  
    ...  
    action n  
}
```

do-while guaranteed to
execute at least once

```
do  
{  
    action 1;  
    action 2;  
    ...  
    action n  
}  
while ( something-is-true )
```

Return in void Methods

```
void drawCirclesRow( double x, double y, double r, int n)
{
    if (n <= 0)
    {
        return;          // OK without value
    }

    ... rest of the code here ...
}
```

Enhanced FOR Loop

```
void printArray( double[] numbers )
{
    for ( int i = 0 ; i < numbers.length ; i++ )
    {
        double curValue = numbers[ i ];

        System.out.print( curValue + " ");
    }
    System.out.println();
}
```

```
void printArray( double[] numbers )
{
    for ( double curValue : numbers )
    {
        System.out.print( curValue + " ");
    }
    System.out.println();
}
```

Enhanced FOR Loop

```
void printArray( double[] numbers )  
{  
    for ( int i = 0 ; i < numbers.length ; i++ )  
    {  
        double curValue = numbers[ i ];  
  
        System.out.print( curValue + " ");  
    }  
    System.out.println();  
}
```

```
void printArray( double[] numbers )  
{  
    for ( double curValue : numbers )  
    {  
        System.out.print( curValue + " ");  
    }  
    System.out.println();  
}
```

Enhanced FOR Loop

```
double addNumbers( double[] numbers )
{
    double result = 0;

    for ( int i = 0 ; i < numbers.length ; i++ )
    {
        double curValue = numbers[ i ];

        result = result + curValue;
    }

    return result;
}
```

```
double addNumbers( double[] numbers )
{
    double result = 0;

    for ( double curValue : numbers )
    {
        result = result + curValue;
    }

    return result;
}
```

Enhanced FOR Loop

```
double addNumbers( double[] numbers )
{
    double result = 0;

    for ( int i = 0 ; i < numbers.length ; i++ )
    {
        double curValue = numbers[ i ];

        result = result + curValue;
    }

    return result;
}
```

```
double addNumbers( double[] numbers )
{
    double result = 0;

    for ( double curValue : numbers )
    {
        result = result + curValue;
    }

    return result;
}
```

Enhanced FOR Loop

```
void driftBalloons( )  
{  
    for ( int i = 0 ; i < balloons.size() ; i++ )  
    {  
        Balloon b = balloons.get( i );  
  
        b.drift( ... );  
    }  
}
```

```
void driftBalloons( )  
{  
    for ( Balloon b : balloons )  
    {  
        b.drift( ... );  
    }  
}
```

Enhanced FOR Loop

```
void driftBalloons( )  
{  
    for ( int i = 0 ; i < balloons.length ; i++ )  
    {  
        Balloon b = balloons.get( i );  
  
        b.drift( ... );  
    }  
}
```

```
void driftBalloons( )  
{  
    for ( Balloon b : balloons )  
    {  
        b.drift( ... );  
    }  
}
```