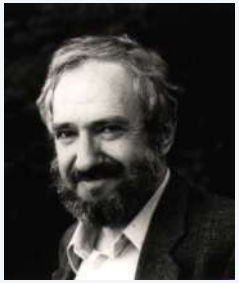
9. Creating Data Types

- Overview
- Point charges
- Turtle graphics
- Complex numbers

ADT for turtle graphics

A turtle is an idealized model of a plotting device.

An ADT allows us to write Java programs that manipulate turtles.



Seymour Papert
1928–

Values

position (x, y)	(.5, .5)	(.25, .75)	(.22, .12)
orientation	90°	135°	10°
			

API (operations)

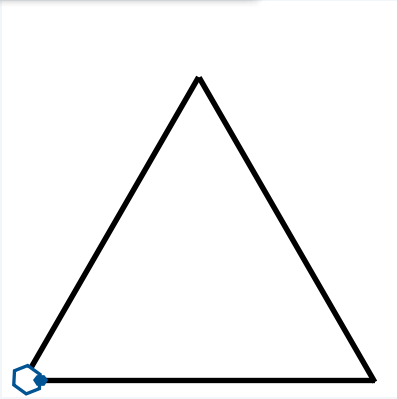
`public class Turtle`	
`Turtle(double x0, double y0, double q0)`	
`void turnLeft(double delta)`	*rotate* `delta` *degrees counterclockwise*
`void goForward(double step)`	*move distance* `step`*, drawing a line*

Turtle graphics implementation: Test client

Best practice. Begin by implementing a simple test client.


```
public static void main(String[] args)
{
   Turtle turtle = new Turtle(0.0, 0.0, 0.0);
   turtle.goForward(1.0);
   turtle.turnLeft(120.0);
   turtle.goForward(1.0);
   turtle.turnLeft(120.0);
   turtle.goForward(1.0);
   turtle.turnLeft(120.0);
}
```

```
% java Turtle
```



Note: Client drew triangle without computing $\sqrt{3}$

What we *expect*, once the implementation is done.

Turtle implementation: Instance variables and constructor

Instance variables define data-type values.

Constructors create and initialize new objects.

```
public class Turtle
{
    private double x, y;
    private double angle;

    public Turtle(double x0, double y0, double a0)
    {
        x = x0;
        y = y0;
        angle = a0;
    }
...
}
```

instance variables are *not* final

Values

position (*x*, *y*)	(.5, .5)	(.75, .75)	(.22, .12)
orientation	90°	135°	10°
			

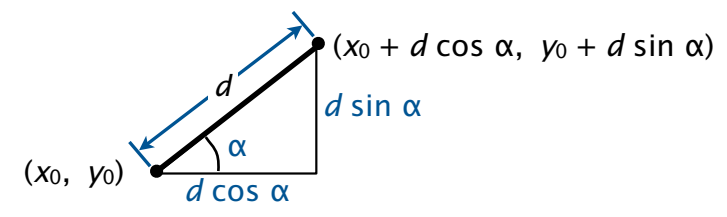
Turtle implementation: Methods

Methods define data-type operations (implement APIs).


```
public class Turtle
{
 ...
   public void turnLeft(double delta)
   { angle += delta; }
   public void goForward(double d)
   {
      double oldx = x;
      double oldy = y;
      x += d * Math.cos(Math.toRadians(angle));
      y += d * Math.sin(Math.toRadians(angle));
      StdDraw.line(oldx, oldy, x, y);
   }
 ...
}
```

API

public class Turtle	
Turtle(double x0, double y0, double q0)	
void turnLeft(double delta)	
void goForward(double step)	



Turtle implementation

text file named Turtle.java

```
public class Turtle
{
   private double x, y;
   private double angle;

   public Turtle(double x0, double y0, double a0)
   {
      x = x0;
      y = y0;
      angle = a0;
   }

   public void turnLeft(double delta)
   { angle += delta; }
   public void goForward(double d)
   {
      double oldx = x;
      double oldy = y;
      x += d * Math.cos(Math.toRadians(angle));
      y += d * Math.sin(Math.toRadians(angle));
      StdDraw.line(oldx, oldy, x, y);
   }

   public static void main(String[] args)
   {
     Turtle turtle = new Turtle(0.0, 0.0, 0.0);
     turtle.goForward(1.0); turtle.turnLeft(120.0);
     turtle.goForward(1.0); turtle.turnLeft(120.0);
     turtle.goForward(1.0); turtle.turnLeft(120.0);
   }
}
```

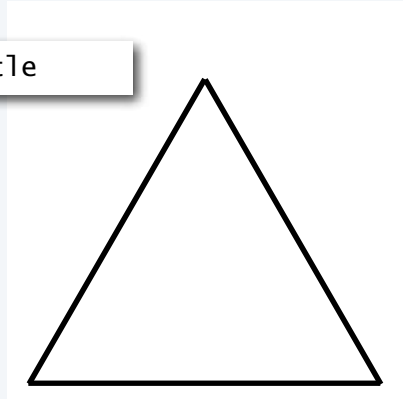
instance variables

constructor

methods

test client

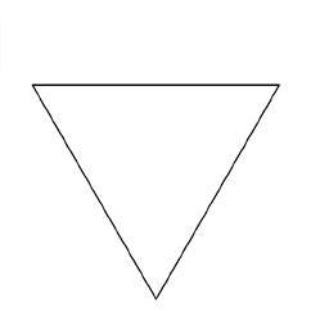
```
% java Turtle
```



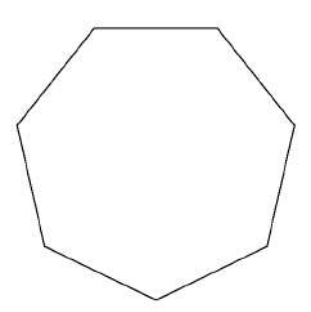
Turtle client: N-gon

```
public class Ngon
{
   public static void main(String[] args)
   {
      int N        = Integer.parseInt(args[0]);
      double angle = 360.0 / N;
      double step  = Math.sin(Math.toRadians(angle/2.0));
      Turtle turtle = new Turtle(0.5, 0, angle/2.0);
      for (int i = 0; i < N; i++)
      {
         turtle.goForward(step);
         turtle.turnLeft(angle);
      }
   }
}
```

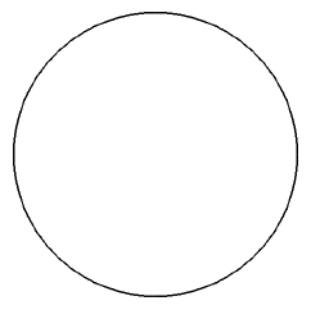
```
% java Ngon 3
```



```
% java Ngon 7
```



```
% java Ngon 1440
```

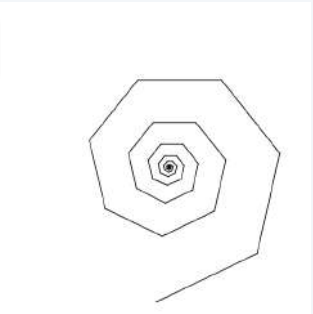


Turtle client: Spira Mirabilis

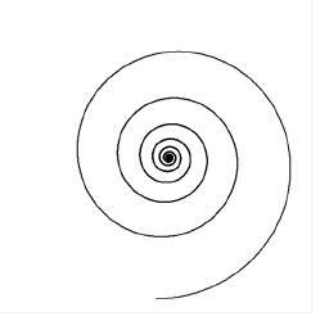
```
public class Spiral
{
   public static void main(String[] args)
   {
      int N        = Integer.parseInt(args[0]);
      double decay = Double.parseDouble(args[1]);
      double angle = 360.0 / N;
      double step  = Math.sin(Math.toRadians(angle/2.0));
      Turtle turtle = new Turtle(0.5, 0, angle/2.0);
      for (int i = 0; i < 10 * N; i++)
      {
         step /= decay;
         turtle.goForward(step);
         turtle.turnLeft(angle);
      }
   }
}
```

```
% java Spiral 3 1.2
```

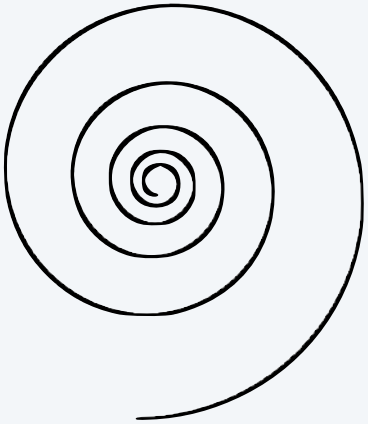
```
% java Spiral 7 1.2
```

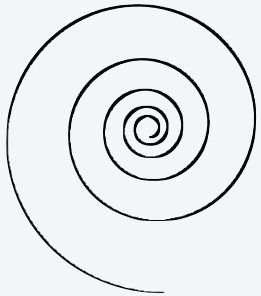


```
% java Spiral 1440 1.0004
```



Spira Mirabilis in the wild





Pop quiz 1 on OOP

Q. Fix the serious bug in this code:

```
public class Turtle
{
    private double x, y;
    private double angle;

    public Turtle(double x0, double y0, double a0)
    {
       double x = x0;
       double y = y0;
       double angle = a0;
    }
...
}
```

Pop quiz 1 on OOP

Q. Fix the serious bug in this code:

```
public class Turtle
{
   private double x, y;
   private double angle;

   public Turtle(double x0, double y0, double a0)
   {
      double x = x0;
      double y = y0;
      double angle = a0;
   }
...
}
```

A. Remove type declarations.

They create local variables, which are *different* from the instance variables!

Object-oriented programmers pledge. "I *will not* shadow instance variables"

Every programmer makes this mistake, and it is a difficult one to detect.