Class SimpleTurtle

java.lang.Object
  └ **SimpleTurtle**

Direct Known Subclasses:
Turtle

```
public class SimpleTurtle
extends Object
```

Class that represents a Logo-style turtle. The turtle starts off facing north. A turtle can have a name, has a starting x and y position, has a heading, has a width, has a height, has a visible flag, has a body color, can have a shell color, and has a pen. The turtle will not go beyond the model display or picture boundaries. You can display this turtle in either a picture or in a class that implements ModelDisplay. Copyright Georgia Institute of Technology 2004

Author:
Barb Ericson ericson@cc.gatech.edu

Constructor Summary

SimpleTurtle(int x, int y) Constructor that takes the x and y position for the turtle
SimpleTurtle(int x, int y, ModelDisplay display) Constructor that takes the x and y position and the model displayer
SimpleTurtle(int x, int y, Picture picture) Constructor that takes the x and y position and the picture to draw on
SimpleTurtle(ModelDisplay display) Constructor that takes a model display and adds a turtle in the middle of it
SimpleTurtle(Picture picture) Constructor that takes the picture to draw on and will appear in the middle

Method Summary

void	**backward**() Method to go backward by 100 pixels
void	**backward**(int pixels) Method to go backward a given number of pixels
void	**clearPath**() Method to clear the path (history of where the turtle has been)

void	**drawInfoString**(Graphics g) Method to draw the information string
void	**drop**(Picture dropPicture) Method to draw a passed picture at the current turtle location and rotation in a picture or model display
void	**forward**() Method to move the turtle foward 100 pixels
void	**forward**(int pixels) Method to move the turtle forward the given number of pixels
Color	**getBodyColor**() Method to get the body color
double	**getDistance**(int x, int y) Get the distance from the passed x and y location
double	**getHeading**() Method to get the current heading
int	**getHeight**() Method to return the height of this object
Color	**getInfoColor**() Method to get the information color
ModelDisplay	**getModelDisplay**() Method to get the model display for this simple turtle
String	**getName**() Method to get the name of the turtle
Pen	**getPen**() Method to get the pen
Color	**getPenColor**() Method to get the pen color
int	**getPenWidth**() Method to get the pen width
Picture	**getPicture**() Method to get the picture for this simple turtle
Color	**getShellColor**() Method to get the shell color
boolean	**getShowInfo**() Method to get value of show info
int	**getWidth**() Method to return the width of this object
int	**getXPos**() Method to get the current x position
int	**getYPos**() Method to get the current y position
void	**hide**() Method to hide the turtle (stop showing it) This doesn't affect the pen status

boolean	**isPenDown**() Method to check if the pen is down
boolean	**isVisible**() Method to get the value of the visible flag
void	**moveTo**(int x, int y) Method to move to turtle to the given x and y location
void	**paintComponent**(Graphics g) Method to paint the turtle
void	**penDown**() Method to set the pen down
void	**penUp**() Method to lift the pen up
void	**setBodyColor**(Color color) Method to set the body color which will also set the pen color
void	**setColor**(Color color) Method to set the color of the turtle.
void	**setHeading**(double heading) Method to set the heading
void	**setHeight**(int theHeight) Method to set the height of this object
void	**setInfoColor**(Color color) Method to set the information color
void	**setModelDisplay**(ModelDisplay theModelDisplay) Method to set the model display for this simple turtle
void	**setName**(String theName) Method to set the name of the turtle
void	**setPen**(Pen thePen) Method to set the pen
void	**setPenColor**(Color color) Method to set the pen color
void	**setPenDown**(boolean value) Method to set the pen down boolean variable
void	**setPenWidth**(int width) Method to set the pen width
void	**setPicture**(Picture pict) Method to set the picture for this simple turtle
void	**setShellColor**(Color color) Method to set the shell color
void	**setShowInfo**(boolean value) Method to show the turtle information string
void	**setVisible**(boolean value) Method to set the visible flag

void	**setWidth**(int theWidth) Method to set the width of this object
void	**show**() Method to show the turtle (doesn't affect the pen status
String	**toString**() Method to return a string with informaiton about this turtle
void	**turn**(int degrees) Method to turn the turtle the passed degrees use negative to turn left and pos to turn right
void	**turnLeft**() Method to turn left
void	**turnRight**() Method to turn right
void	**turnToFace**(int x, int y) Method to turn towards the given x and y
void	**turnToFace**(SimpleTurtle turtle) Method to turn to face another simple turtle
void	**updateDisplay**() Method to update the display of this turtle and also check that the turtle is in the bounds

Methods inherited from class java.lang.Object
clone, equals, finalize, getClass, hashCode, notify, notifyAll, wait, wait, wait

Constructor Detail

SimpleTurtle

```
public SimpleTurtle(int x,
                    int y)
```

Constructor that takes the x and y position for the turtle

Parameters:

x - the x pos

y - the y pos

SimpleTurtle

```
public SimpleTurtle(int x,
                    int y,
                    ModelDisplay display)
```

Constructor that takes the x and y position and the model displayer

Parameters:

`x` - the x pos
`y` - the y pos
`display` - the model display

SimpleTurtle

```
public SimpleTurtle(ModelDisplay display)
```

Constructor that takes a model display and adds a turtle in the middle of it

Parameters:

`display` - the model display

SimpleTurtle

```
public SimpleTurtle(int x,
                    int y,
                    Picture picture)
```

Constructor that takes the x and y position and the picture to draw on

Parameters:

`x` - the x pos
`y` - the y pos
`picture` - the picture to draw on

SimpleTurtle

```
public SimpleTurtle(Picture picture)
```

Constructor that takes the picture to draw on and will appear in the middle

Parameters:

`picture` - the picture to draw on

Method Detail

getDistance

```
public double getDistance(int x,
                          int y)
```

Get the distance from the passed x and y location

Parameters:

`x` - the x location
`y` - the y location

---

turnToFace

```
public void turnToFace(SimpleTurtle turtle)
```

Method to turn to face another simple turtle

---

turnToFace

```
public void turnToFace(int x,
                       int y)
```

Method to turn towards the given x and y

Parameters:

`x` - the x to turn towards
`y` - the y to turn towards

---

getPicture

```
public Picture getPicture()
```

Method to get the picture for this simple turtle

Returns:

the picture for this turtle (may be null)

---

setPicture

```
public void setPicture(Picture pict)
```

Method to set the picture for this simple turtle

Parameters:

`pict` - the picture to use

---

getModelDisplay

```
public ModelDisplay getModelDisplay()
```

Method to get the model display for this simple turtle

Returns:

the model display if there is one else null

---

setModelDisplay

```
public void setModelDisplay(ModelDisplay theModelDisplay)
```

Method to set the model display for this simple turtle

Parameters:

`theModelDisplay` - the model display to use

---

getShowInfo

```
public boolean getShowInfo()
```

Method to get value of show info

Returns:

true if should show info, else false

---

setShowInfo

```
public void setShowInfo(boolean value)
```

Method to show the turtle information string

Parameters:

`value` - the value to set showInfo to

---

getShellColor

```
public Color getShellColor()
```

Method to get the shell color

Returns:

the shell color

---

setShellColor

```
public void setShellColor(Color color)
```

Method to set the shell color

Parameters:

`color` - the color to use

---

getBodyColor

```
public Color getBodyColor()
```

Method to get the body color

Returns:
the body color

---

setBodyColor

```
public void setBodyColor(Color color)
```

Method to set the body color which will also set the pen color

Parameters:
`color` - the color to use

---

setColor

```
public void setColor(Color color)
```

Method to set the color of the turtle. This will set the body color

Parameters:
`color` - the color to use

---

getInfoColor

```
public Color getInfoColor()
```

Method to get the information color

Returns:
the color of the information string

---

setInfoColor

```
public void setInfoColor(Color color)
```

Method to set the information color

Parameters:
`color` - the new color to use

---

getWidth

```
public int getWidth()
```

Method to return the width of this object

Returns:
the width in pixels

---

getHeight

```
public int getHeight()
```

Method to return the height of this object

Returns:
the height in pixels

---

setWidth

```
public void setWidth(int theWidth)
```

Method to set the width of this object

Parameters:
`theWidth` - in width in pixels

---

setHeight

```
public void setHeight(int theHeight)
```

Method to set the height of this object

Parameters:
`theHeight` - the height in pixels

---

getXPos

```
public int getXPos()
```

Method to get the current x position

Returns:
the x position (in pixels)

---

getYPos

```
public int getYPos()
```

Method to get the current y position

Returns:

the y position (in pixels)

---

getPen

```
public Pen getPen()
```

Method to get the pen

Returns:
the pen

---

setPen

```
public void setPen(Pen thePen)
```

Method to set the pen

Parameters:
`thePen` - the new pen to use

---

isPenDown

```
public boolean isPenDown()
```

Method to check if the pen is down

Returns:
true if down else false

---

setPenDown

```
public void setPenDown(boolean value)
```

Method to set the pen down boolean variable

Parameters:
`value` - the value to set it to

---

penUp

```
public void penUp()
```

Method to lift the pen up

---

penDown

```
public void penDown()
```

Method to set the pen down

---

getPenColor

```
public Color getPenColor()
```

Method to get the pen color

Returns:
the pen color

---

setPenColor

```
public void setPenColor(Color color)
```

Method to set the pen color

Parameters:
`color` - the color for the pen ink

---

setPenWidth

```
public void setPenWidth(int width)
```

Method to set the pen width

Parameters:
`width` - the width to use in pixels

---

getPenWidth

```
public int getPenWidth()
```

Method to get the pen width

Returns:
the width of the pen in pixels

---

clearPath

```
public void clearPath()
```

Method to clear the path (history of where the turtle has been)

---

getHeading

```
public double getHeading()
```

Method to get the current heading

Returns:
the heading in degrees

---

setHeading

```
public void setHeading(double heading)
```

Method to set the heading

Parameters:
`heading` - the new heading to use

---

getName

```
public String getName()
```

Method to get the name of the turtle

Returns:
the name of this turtle

---

setName

```
public void setName(String theName)
```

Method to set the name of the turtle

Parameters:
`theName` - the new name to use

---

isVisible

```
public boolean isVisible()
```

Method to get the value of the visible flag

Returns:
true if visible else false

---

hide

```
public void hide()
```

Method to hide the turtle (stop showing it) This doesn't affect the pen status

---

show

```
public void show()
```

Method to show the turtle (doesn't affect the pen status

---

setVisible

```
public void setVisible(boolean value)
```

Method to set the visible flag

Parameters:

`value` - the value to set it to

---

updateDisplay

```
public void updateDisplay()
```

Method to update the display of this turtle and also check that the turtle is in the bounds

---

forward

```
public void forward()
```

Method to move the turtle foward 100 pixels

---

forward

```
public void forward(int pixels)
```

Method to move the turtle forward the given number of pixels

Parameters:

`pixels` - the number of pixels to walk forward in the heading direction

---

backward

```
public void backward()
```

Method to go backward by 100 pixels

---

backward

```
public void backward(int pixels)
```

Method to go backward a given number of pixels

Parameters:

`pixels` - the number of pixels to walk backward

---

moveTo

```
public void moveTo(int x,
                   int y)
```

Method to move to turtle to the given x and y location

Parameters:

`x` - the x value to move to
`y` - the y value to move to

---

turnLeft

```
public void turnLeft()
```

Method to turn left

---

turnRight

```
public void turnRight()
```

Method to turn right

---

turn

```
public void turn(int degrees)
```

Method to turn the turtle the passed degrees use negative to turn left and pos to turn right

Parameters:

`degrees` - the amount to turn in degrees

---

drop

```
public void drop(Picture dropPicture)
```

Method to draw a passed picture at the current turtle location and rotation in a picture or model display

Parameters:

`dropPicture` - the picture to drop

---

paintComponent

`public void` **`paintComponent`**`(Graphics g)`

Method to paint the turtle

Parameters:

`g` - the graphics context to paint on

---

drawInfoString

`public void` **`drawInfoString`**`(Graphics g)`

Method to draw the information string

Parameters:

`g` - the graphics context

---

toString

`public String` **`toString`**`()`

Method to return a string with informaiton about this turtle

Overrides:

`toString` in class `Object`

Returns:

a string with information about this object
