COSC 1P02
Method Summary

`Math`

Method/Value	meaning
`E`	*constant*: the mathematical constant `e`
`PI`	*constant*: the mathematical constant `π`
v = `abs(`*x*`)`	returns the absolute value of *x*
v = `acos(`*x*`)`	returns the arc cosine of *x*
v = `asin(`*x*`)`	returns the arc sine of *x*
v = `atan(`*x*`)`	returns the arc tangent of *x*
v = `cos(`*x*`)`	returns the cosine of *x*
v = `log(`*x*`)`	returns the natural logarithm of *x*
v = `pow(`*a*`,`*b*`)`	returns a^b
v = `random()`	returns a random value between `0.0` and `1.0`
v = `sin(`*x*`)`	returns the sine of *x*
v = `sqrt(`*x*`)`	returns the square root of *x*
v = `tan(`*x*`)`	returns the tangent of *x*

`PictureDisplayer`

method	meaning
d = `new PictureDisplayer()`	*constructor*: creates a new displayer with canvas 200x200
d = `new PictureDisplayer(`*pic*`)`	*constructor*: creates a new displayer with canvas to fit *pic* and with *pic* placed on displayer
d = `new PictureDisplayer(`*width*`,` *height*`)`	*constructor*: creates a new displayer with canvas of specified *height* and *width*
d`.close()`	wait until user presses `Close` button and close displayer
d`.placePicture(`*pic*`)`	place *pic* on the displayer
d`.waitForUser()`	wait until user presses `OK` before continuing

`Picture`

method	meaning
p = `new Picture()`	*constructor*: creates a picture object loading pixels from a file selected via a file open dialog
p = `new Picture(`*width*`,`*height*`)`	*constructor*: creates a picture object with specified *height* and *width* with all pixels `white`
i = *p*`.getHeight()`	returns height (in pixels) of picture
q = *p*`.getPixel(`*x*`,`*y*`)`	returns pixel in column *x* of row *y*
i = *p*`.getWidth()`	returns width (in pixels) of picture
b = *p*`.hasNext()`	returns `true` if another pixel is available
q = *p*`.next()`	returns the next available pixel
p`.save()`	present file save dialog to allow user to save picture as modified

Pixel

method	meaning
`i = q.getBlue()`	obtain blue color channel of pixel
`c = q.getColor()`	obtain color of pixel
`r = q.getDistance(color)`	returns the color distance between this pixel's color and *color*
`i = q.getGreen()`	obtain green color channel of pixel
`i = q.getRed()`	obtain red color channel of pixel
`q.setBlue(v);`	change blue channel value to *v*
`q.setColor(color)`	change color of pixel to *color*
`q.setGreen(v)`	change green channel value to *v*
`q.setRed(v)`	change red channel value to *v*

Color

method	meaning
`red, green, …, RED, GREEN, …`	*constant*: standard colors
`c = new Color(r,g,b)`	*constructor*: creates a new color object with specified *r*, *g* and *b* components
`c = new Color(value)`	*constructor*: creates a new color object with color `value (0-16,777,215)`

SoundPlayer

method	meaning
`p = new SoundPlayer()`	*constructor*: creates a new sound player
`p = new SoundPlayer(sound)`	*constructor*: creates a new player with *sound* placed on player
`p.close()`	wait until user presses `Close` button and close player
`p.placeSound(sound)`	place *sound* on the player
`p.waitForUser()`	wait until user presses `OK` before continuing

Sound

method	meaning
`s = new Sound()`	*constructor*: creates a sound object loading samples from a file selected via a file open dialog
`s = new Sound(length,like)`	*constructor*: creates a sound object with *length* samples and all other attributes the same as the sound `like`, with all amplitudes zero
`i = s.getNumSamples()`	returns the number of samples in sound
`i = s.getSampleRate()`	returns sampling rate of sound
`sa = s.getSample(pos)`	returns sample at position *pos*
`b = s.hasNext()`	returns `true` if another sample is available
`sa = s.next()`	returns the next available sample
`p.save()`	present file save dialog to allow user to save picture as modified

Sample

method	meaning
i = *sa*.getAmp()	obtain amplitude of the sample
sa.setAmp(*v*);	change amplitude of sample to *v*