Programming with Visual Output

Processing, Java, Elementary Tricks

05.10.2015

Crowd Simulation Course

treyer@arch.ethz.ch
zuend@arch.ethz.ch

ETH
Eidgenössische Technische Hochschule Zürich
Swiss Federal Institute of Technology Zurich

DARCH

iA Chair of Information Architecture

Programming
Visual Output

Java syntax: repetition / whole picture

Processing examples explained

Exercise

Java Syntax

(Key)words vs punctuation

punctuation characters cannot be changed

words can

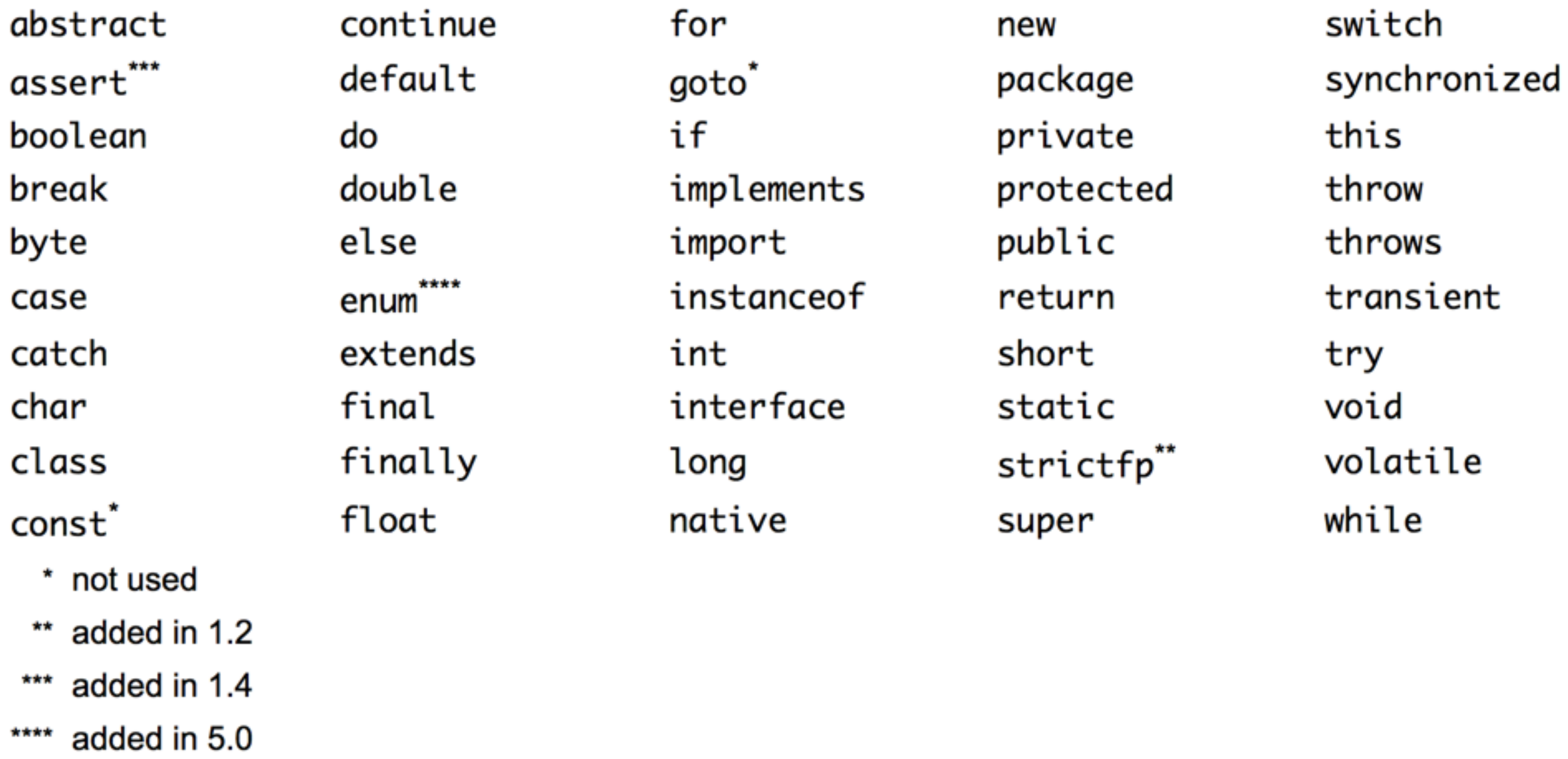
exception:

keywords (for, if, int, return, ...)

↑

and all other built-in data types:

int, double, ...



abstract
assert***
boolean
break
byte
case
catch
char
class
const*

continue
default
do
double
else
enum****
extends
final
finally
float

for
goto*
if
implements
import
instanceof
int
interface
long
native

new
package
private
protected
public
return
short
static
strictfp**
super

switch
synchronized
this
throw
throws
transient
try
void
volatile
while

* not used
** added in 1.2
*** added in 1.4
**** added in 5.0

http://docs.oracle.com/javase/tutorial/java/nutsandbolts/_keywords.html

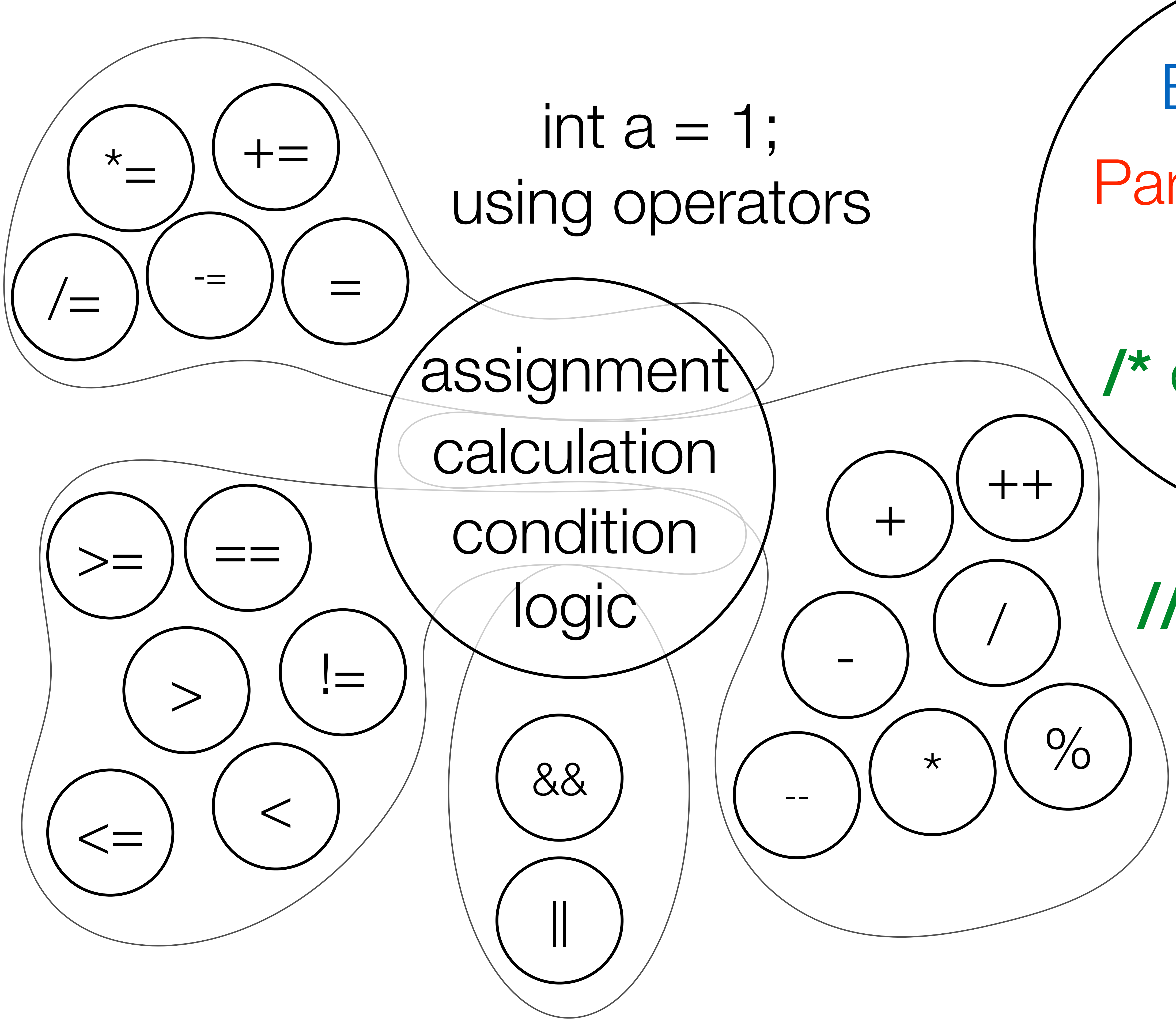
Java Syntax

Punctuation Characters

statements ;

bracket characters

int a = 1;
using operators



Block {...}

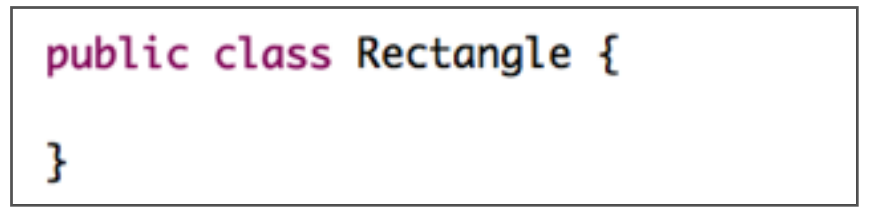
Parameters(...)

List[...]

/* comment */

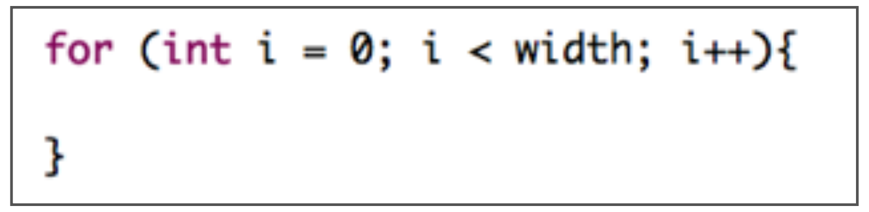
// comment

class



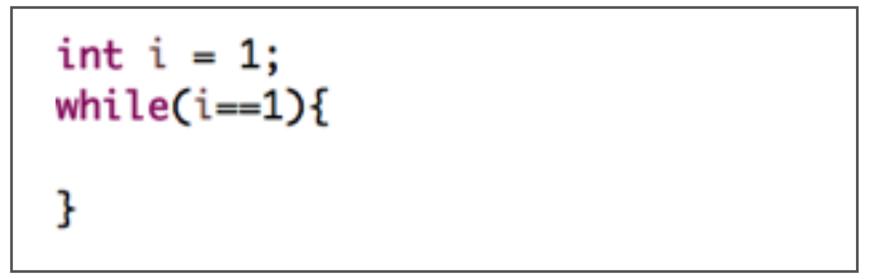
```
public class Rectangle {

}
```

for loop



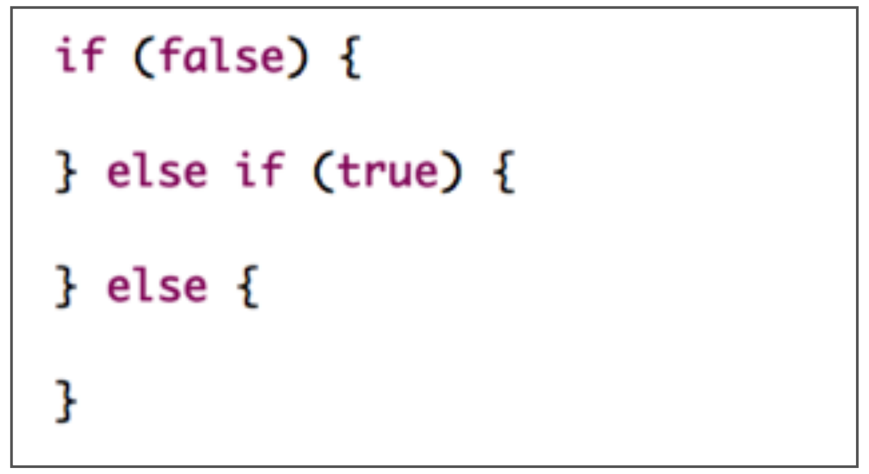
```
for (int i = 0; i < width; i++){

}
```

while loop



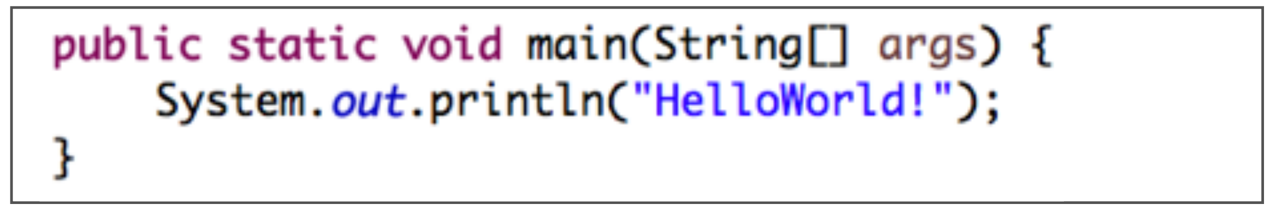
```
int i = 1;
while(i==1){

}
```

if / else
conditions



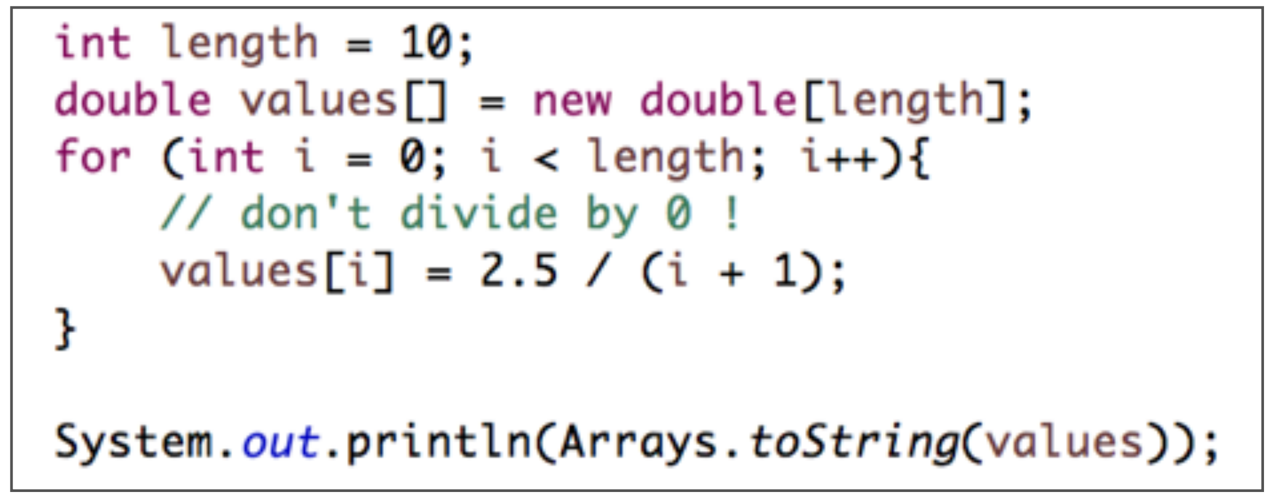
```
if (false) {

} else if (true) {

} else {

}
```

function
(also called "method")



```
public static void main(String[] args) {
    System.out.println("HelloWorld!");
}
```

initializing an array
using a for-loop to fill it with values
explain code with comments
print the array



```
int length = 10;
double values[] = new double[length];
for (int i = 0; i < length; i++){
    // don't divide by 0 !
    values[i] = 2.5 / (i + 1);
}

System.out.println(Arrays.toString(values));
```

```
[2.5, 1.25, 0.8333333333333334, 0.625, 0.5, 0.4166666666666667, 0.35714285714285715, 0.3125, 0.2777777777777778, 0.25]
```

Java Syntax

Punctuation Characters (Pro)

:

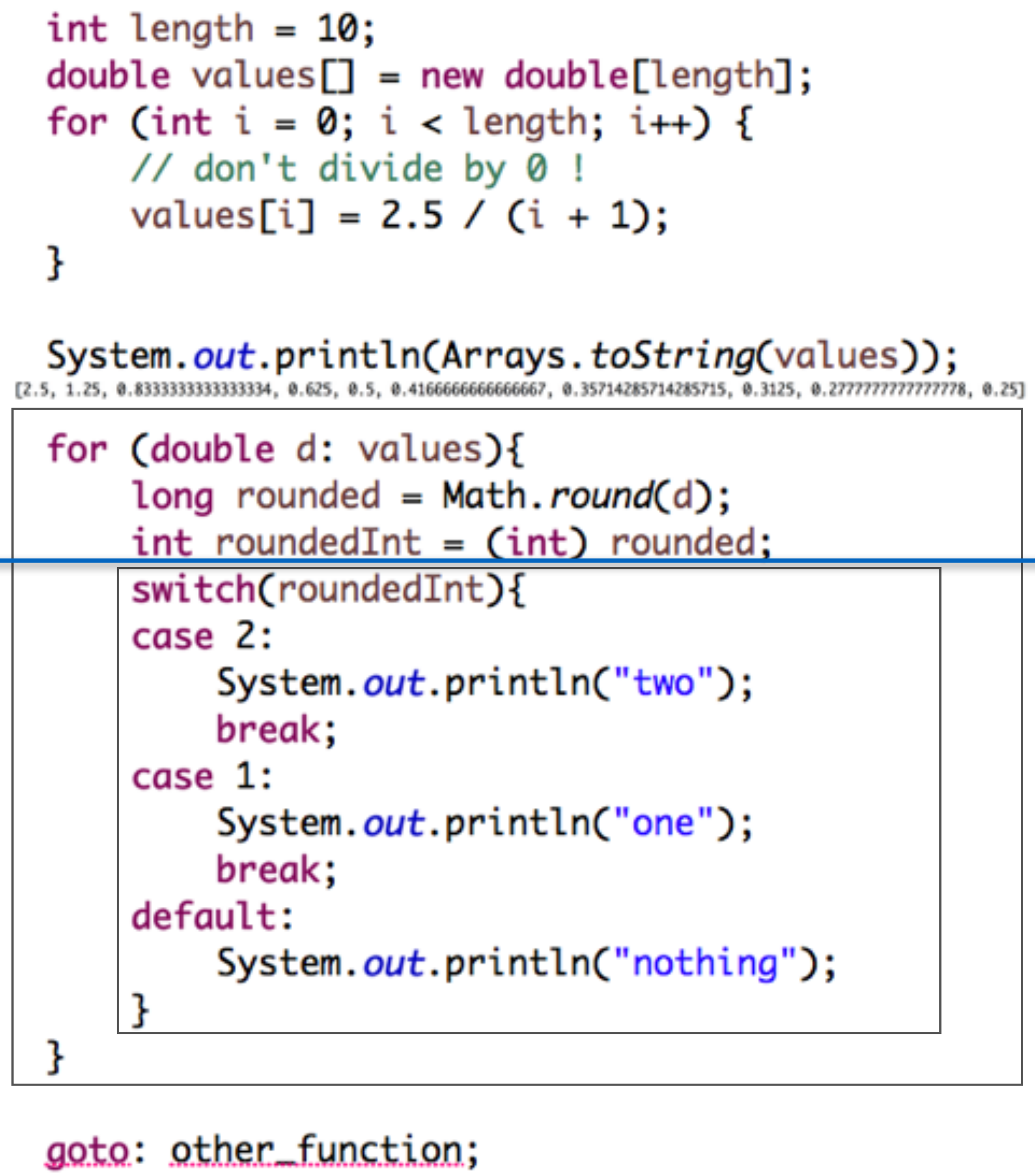
short cut in for loops using :

additional information

for special data types the if/else if /else block can be replaced with a *switch* statement, (old concept from C language)

goto statements = very, very old concept (not supported anymore in Java)
goto = jump to another position in code

Once software programs got a bit more complex in the 70/80ies,
code with "goto" statements turned out to be not understandable anymore.
Such code was called "spaghetti code" - too complex to follow.
Therefore computer scientists invented first functions (C)
and later the concept of object oriented programming (C++)



```
int length = 10;
double values[] = new double[length];
for (int i = 0; i < length; i++) {
    // don't divide by 0 !
    values[i] = 2.5 / (i + 1);
}

System.out.println(Arrays.toString(values));
[2.5, 1.25, 0.8333333333333334, 0.625, 0.5, 0.4166666666666667, 0.35714285714285715, 0.3125, 0.2777777777777778, 0.25]
for (double d: values){
    long rounded = Math.round(d);
    int roundedInt = (int) rounded;
    switch(roundedInt){
    case 2:
        System.out.println("two");
        break;
    case 1:
        System.out.println("one");
        break;
    default:
        System.out.println("nothing");
    }
}

goto: other_function;
```

Crowd Simulation | 05.10.2015

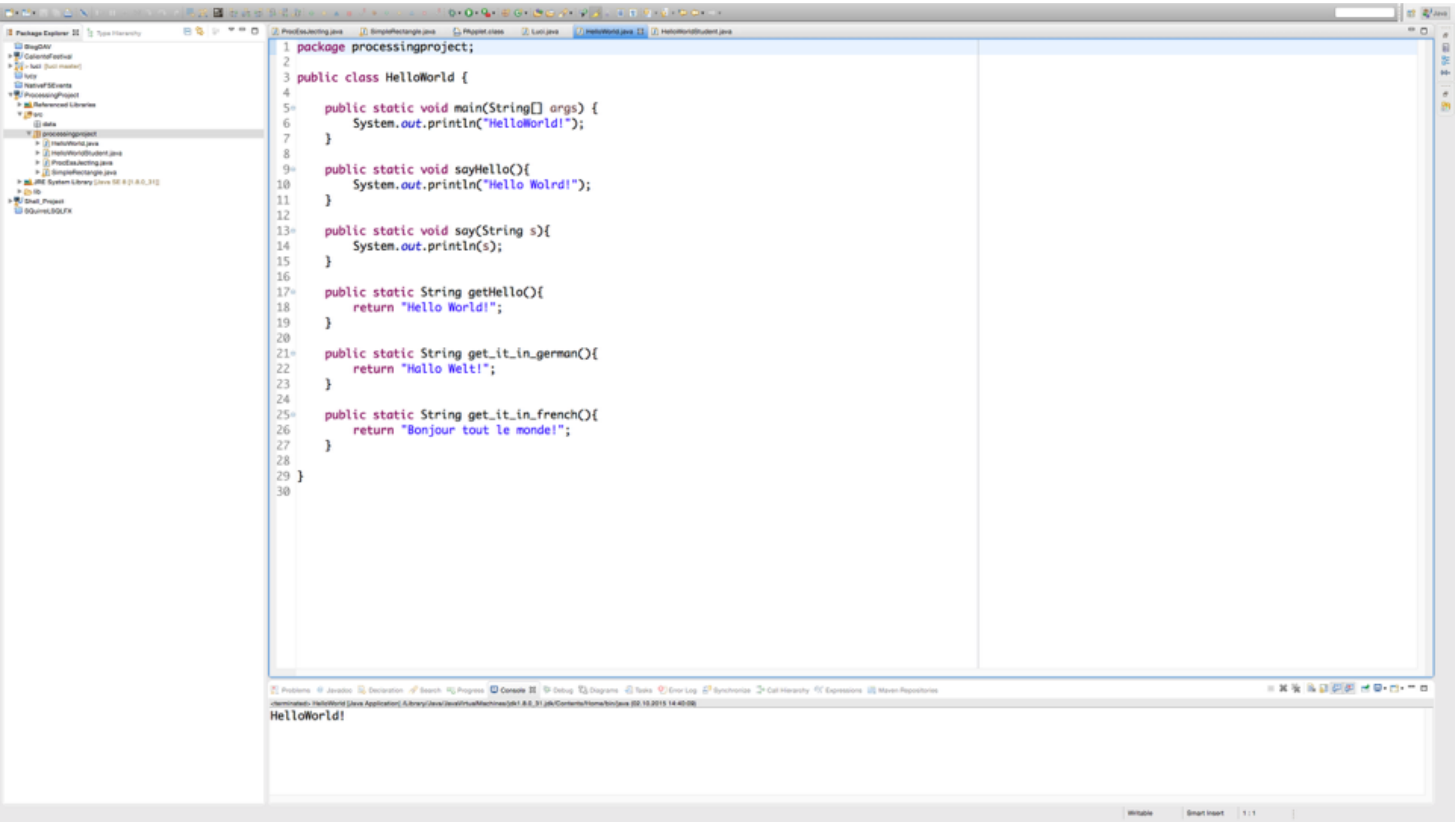
ETH
Eidgenössische Technische Hochschule Zürich
Swiss Federal Institute of Technology Zurich

DARCH

iA Chair of Information Architecture

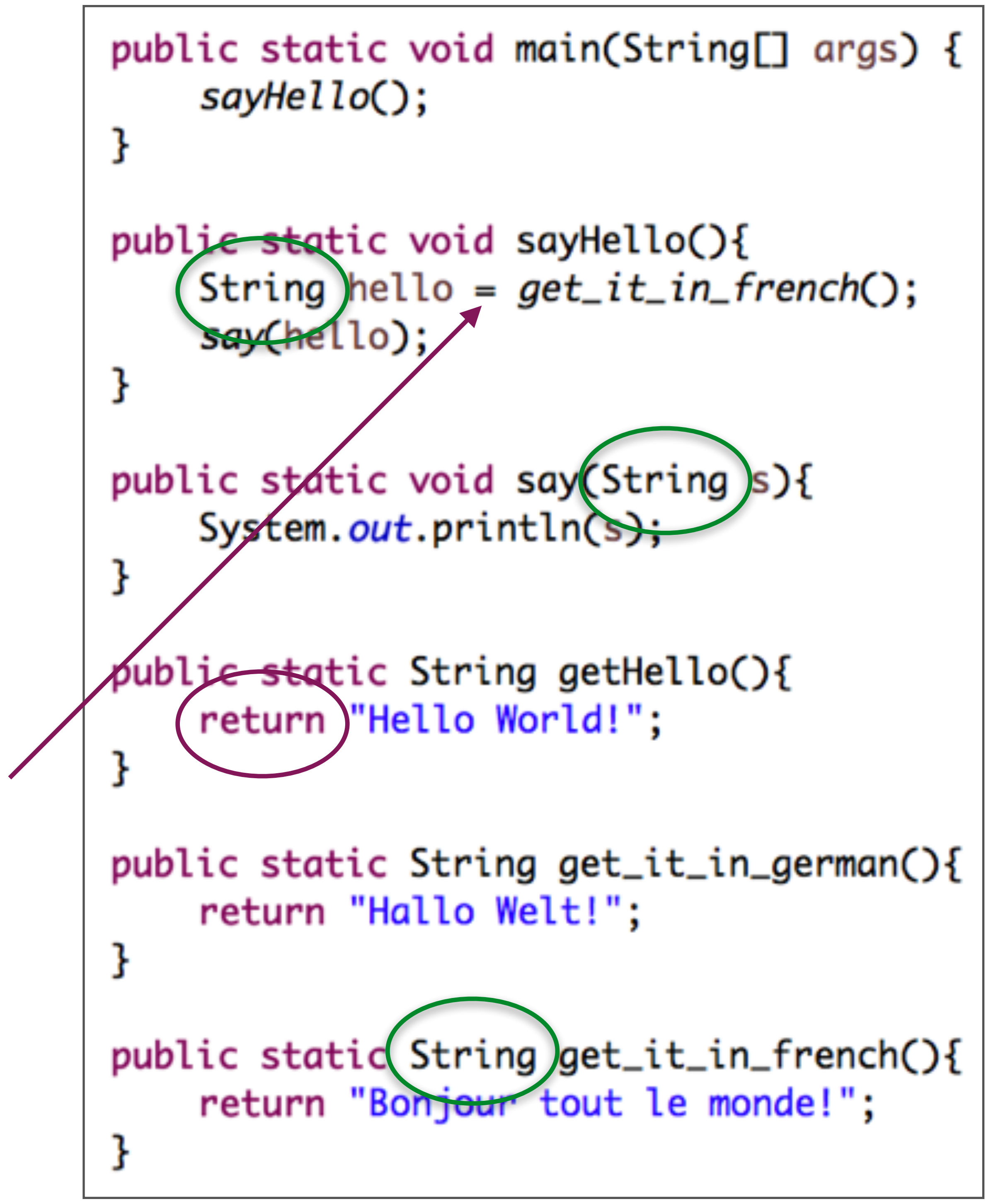
Functions

in Java called methods



```java
package processingproject;

public class HelloWorld {

    public static void main(String[] args) {
        System.out.println("HelloWorld!");
    }

    public static void sayHello(){
        System.out.println("Hello Wolrd!");
    }

    public static void say(String s){
        System.out.println(s);
    }

    public static String getHello(){
        return "Hello World!";
    }

    public static String get_it_in_german(){
        return "Hallo Welt!";
    }

    public static String get_it_in_french(){
        return "Bonjour tout le monde!";
    }

}
```

```
HelloWorld!
```



```java
public static void main(String[] args) {
    sayHello();
}

public static void sayHello(){
    String hello = get_it_in_french();
    say(hello);
}

public static void say(String s){
    System.out.println(s);
}

public static String getHello(){
    return "Hello World!";
}

public static String get_it_in_german(){
    return "Hallo Welt!";
}

public static String get_it_in_french(){
    return "Bonjour tout le monde!";
}
```

return

(result can be assigned to a variable)

return type

(type of the returned value)

Java Syntax

(Key)words / Syntax Coloring

punctuation characters cannot be changed

words can

exception:

`keywords (for, if, int, return, ...)`

↑

and all other built-in data types:

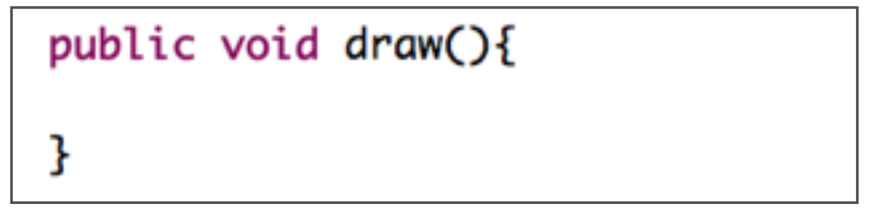
`int, double, ...`

method name

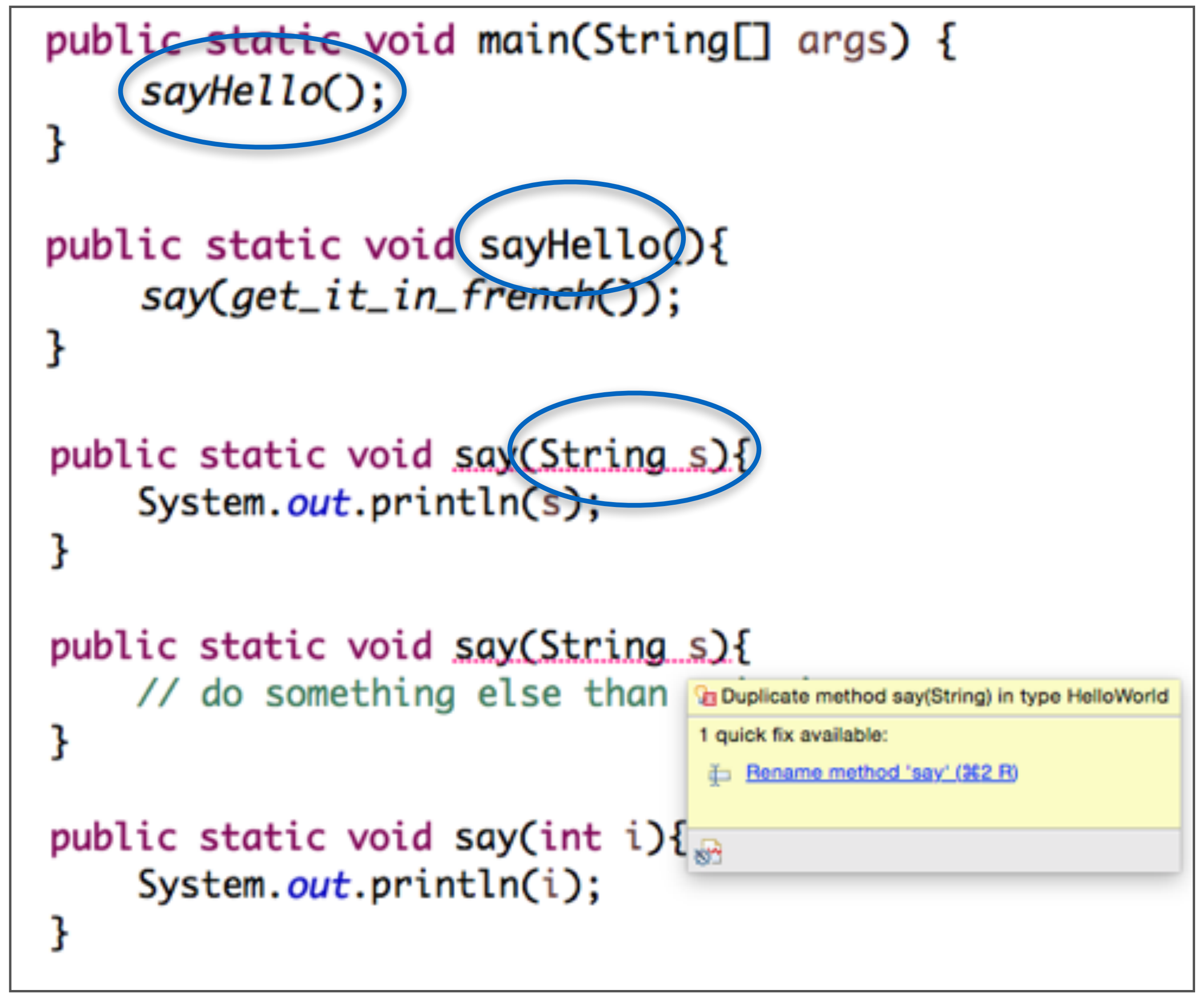
parameter declaration

same method name
same parameter declaration = error

same method name
different parameter declaration



```
public void draw(){

}
```

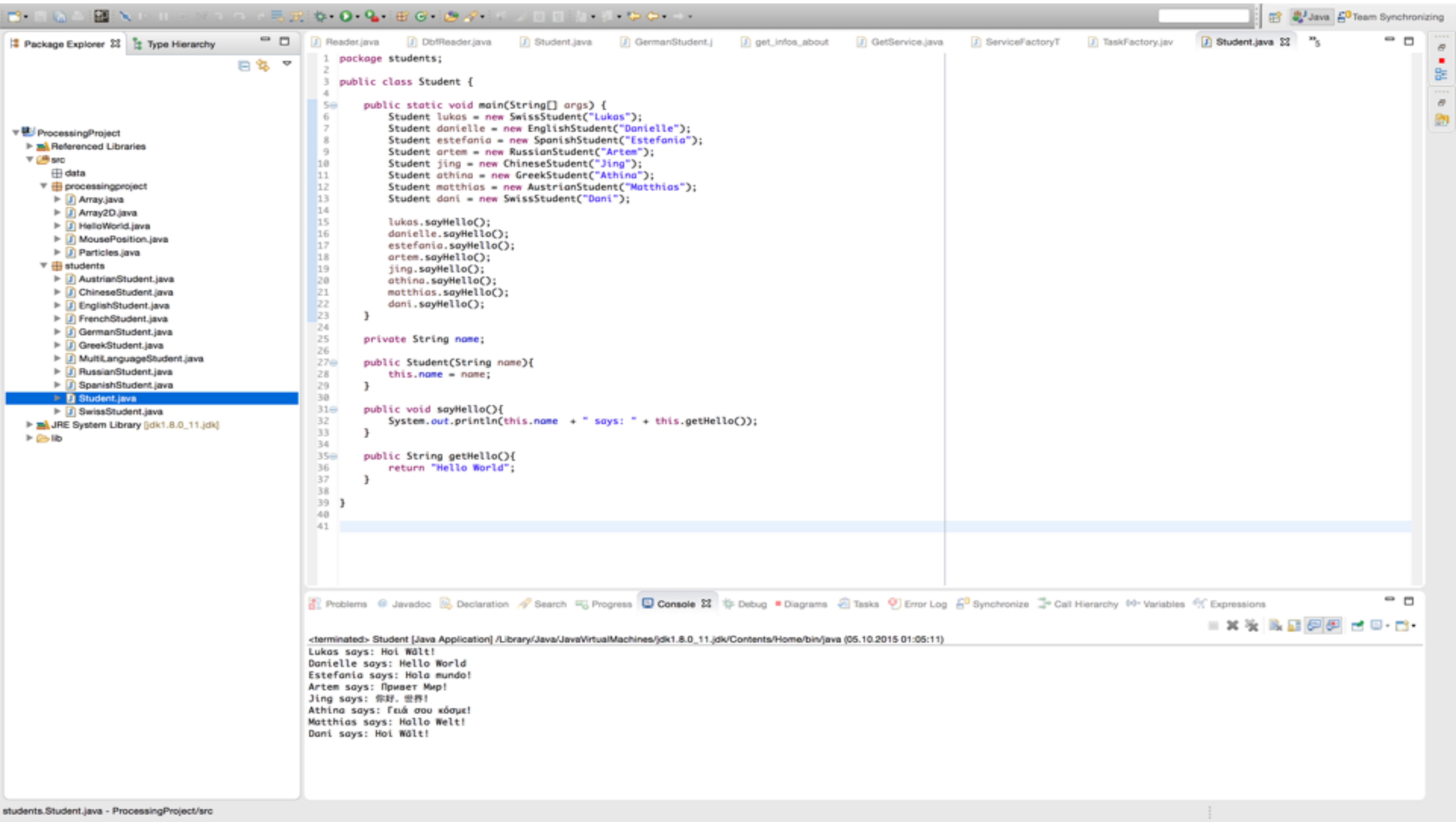


```
public static void main(String[] args) {
    sayHello();
}

public static void sayHello(){
    say(get_it_in_french());
}

public static void say(String s){
    System.out.println(s);
}

public static void say(String s){
    // do something else than
}

public static void say(int i){
    System.out.println(i);
}
```

Duplicate method say(String) in type HelloWorld
1 quick fix available:
Rename method 'say' (⌘2 R)

Objects / Classes

Define your own data types



```java
package students;

public class Student {

    public static void main(String[] args) {
        Student lukas = new SwissStudent("Lukas");
        Student danielle = new EnglishStudent("Danielle");
        Student estefania = new SpanishStudent("Estefania");
        Student artem = new RussianStudent("Artem");
        Student jing = new ChineseStudent("Jing");
        Student athina = new GreekStudent("Athina");
        Student matthias = new AustrianStudent("Matthias");
        Student dani = new SwissStudent("Dani");

        lukas.sayHello();
        danielle.sayHello();
        estefania.sayHello();
        artem.sayHello();
        jing.sayHello();
        athina.sayHello();
        matthias.sayHello();
        dani.sayHello();
    }

    private String name;

    public Student(String name){
        this.name = name;
    }

    public void sayHello(){
        System.out.println(this.name  + " says: " + this.getHello());
    }

    public String getHello(){
        return "Hello World";
    }

}
```

```
<terminated> Student [Java Application] /Library/Java/JavaVirtualMachines/jdk1.8.0_11.jdk/Contents/Home/bin/java (05.10.2015 01:05:11)
Lukas says: Hoi Wält!
Danielle says: Hello World
Estefania says: Hola mundo!
Artem says: Привет Мир!
Jing says: 你好, 世界!
Athina says: Γειά σου κόσμε!
Matthias says: Hallo Welt!
Dani says: Hoi Wält!
```

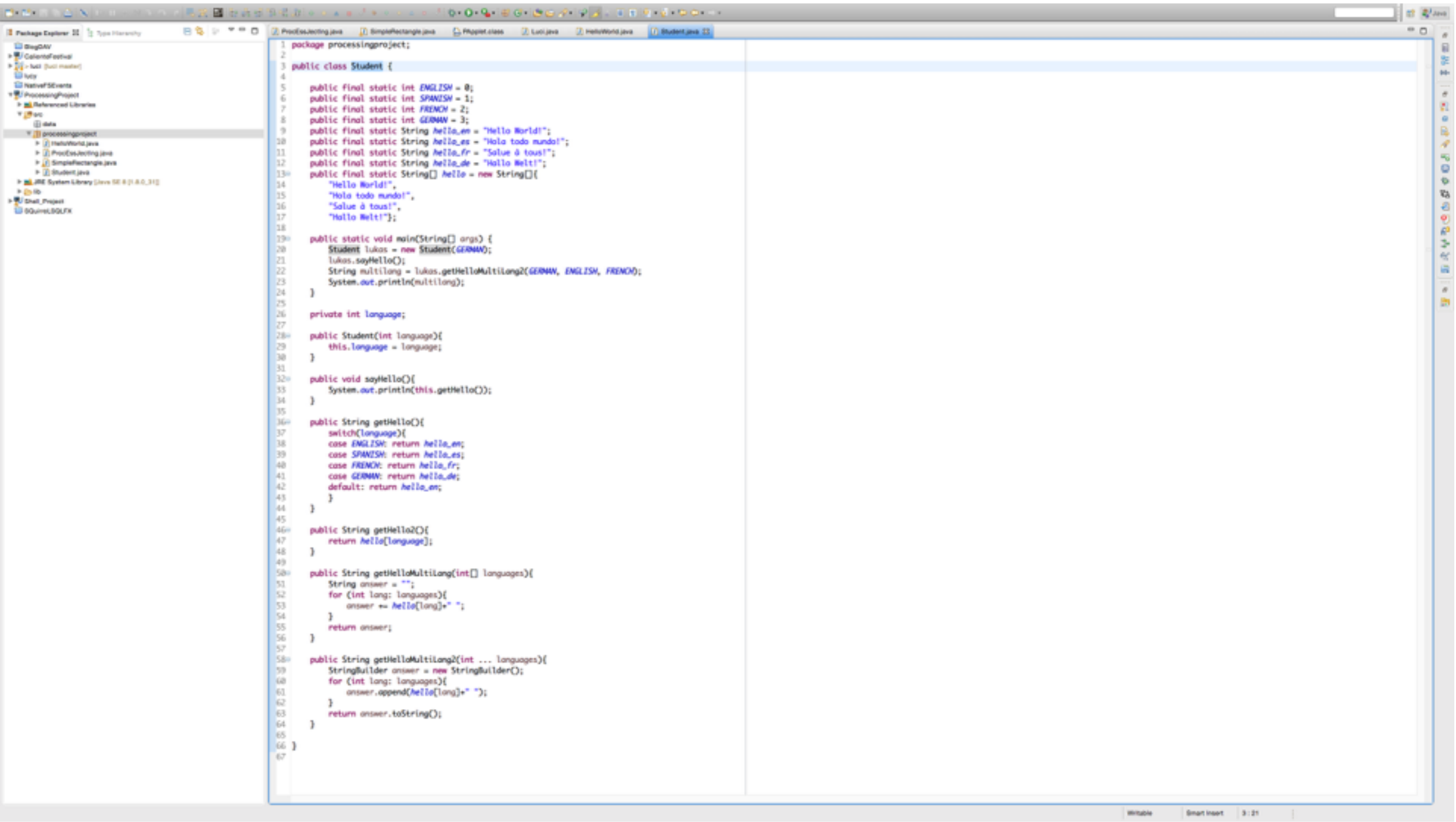
```java
public static void main(String[] args) {
    Student lukas = new Student("Lukas");
    lukas.sayHello();
}
```

this makes our code more readable

objects are being created by creating an instance of a class

Casting

Convert from one type to another
if possible

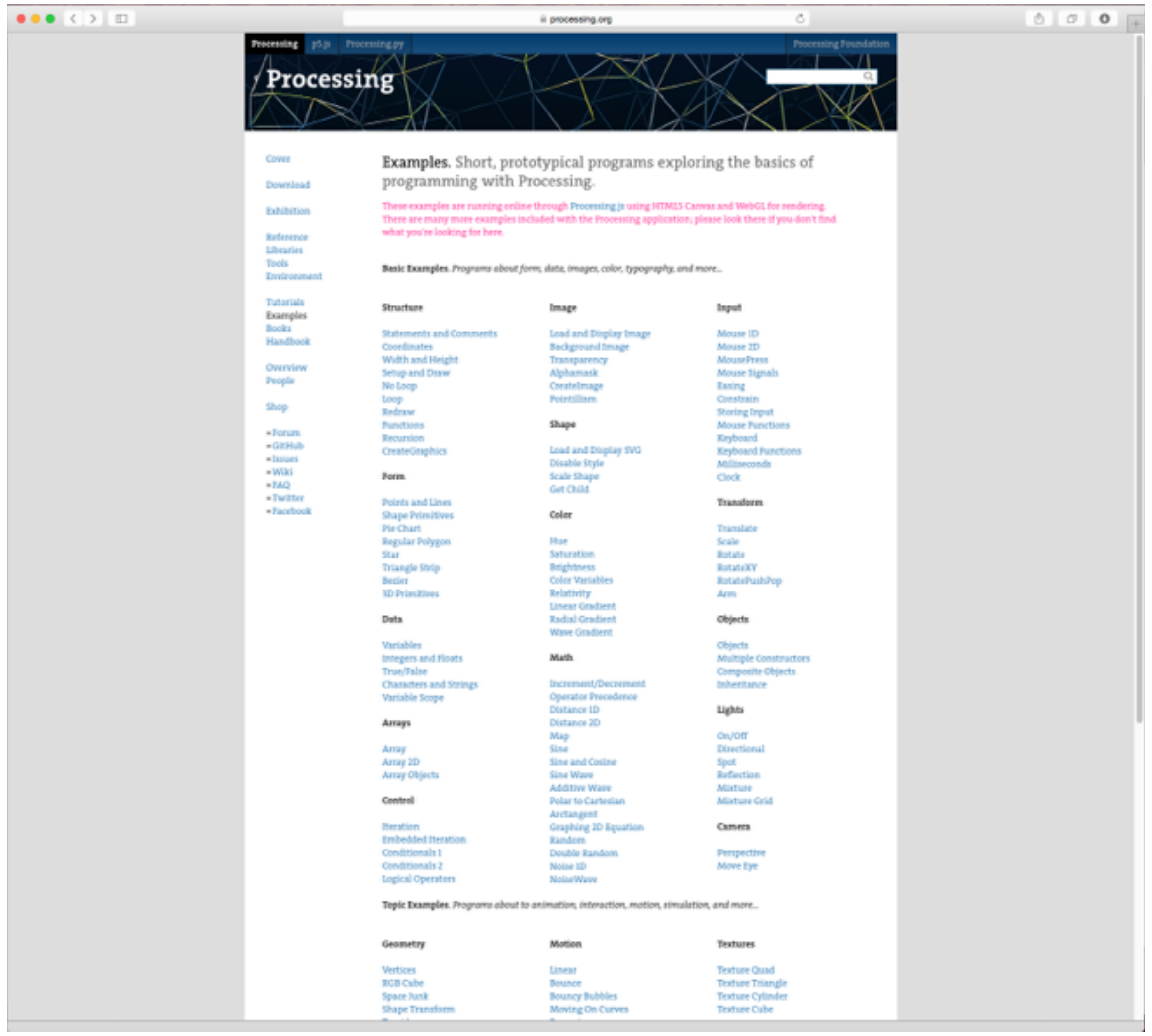


```java
package processingproject;

public class Student {

    public final static int ENGLISH = 0;
    public final static int SPANISH = 1;
    public final static int FRENCH = 2;
    public final static int GERMAN = 3;
    public final static String hello_en = "Hello World!";
    public final static String hello_es = "Hola todo mundo!";
    public final static String hello_fr = "Salue à tous!";
    public final static String hello_de = "Hallo Welt!";
    public final static String[] hello = new String[]{
        "Hello World!",
        "Hola todo mundo!",
        "Salue à tous!",
        "Hallo Welt!"};

    public static void main(String[] args) {
        Student lukas = new Student(GERMAN);
        lukas.sayHello();
        String multilang = lukas.getHelloMultiLang2(GERMAN, ENGLISH, FRENCH);
        System.out.println(multilang);
    }

    private int language;

    public Student(int language){
        this.language = language;
    }

    public void sayHello(){
        System.out.println(this.getHello());
    }

    public String getHello(){
        switch(language){
        case ENGLISH: return hello_en;
        case SPANISH: return hello_es;
        case FRENCH: return hello_fr;
        case GERMAN: return hello_de;
        default: return hello_en;
        }
    }

    public String getHello2(){
        return hello[language];
    }

    public String getHelloMultiLang(int[] languages){
        String answer = "";
        for (int lang: languages){
            answer += hello[lang]+" ";
        }
        return answer;
    }

    public String getHelloMultiLang2(int ... languages){
        StringBuilder answer = new StringBuilder();
        for (int lang: languages){
            answer.append(hello[lang]+" ");
        }
        return answer.toString();
    }

}
```

```java
double d = 4.987;
int i = (int) d;
System.out.println(i);

Student lukas = (Student) new GermanStudent();
lukas.sayHello();

((GermanStudent) lukas).saysSomethingGerman();
```

Processing

Examples & Sources (for Eclipse)



Examples. Short, prototypical programs exploring the basics of programming with Processing.

These examples are running online through Processing.js using HTML5 Canvas and WebGL for rendering. There are many more examples included with the Processing application; please look there if you don't find what you're looking for here.

Basic Examples. *Programs about form, data, images, color, typography, and more...*

Structure

Statements and Comments
Coordinates
Width and Height
Setup and Draw
No Loop
Loop
Redraw
Functions
Recursion
CreateGraphics

Form

Points and Lines
Shape Primitives
Pie Chart
Regular Polygon
Star
Triangle Strip
Bezier
3D Primitives

Data

Variables
Integers and Floats
True/False
Characters and Strings
Variable Scope

Arrays

Array
Array 2D
Array Objects

Control

Iteration
Embedded Iteration
Conditionals 1
Conditionals 2
Logical Operators

Image

Load and Display Image
Background Image
Transparency
Alphamask
CreateImage
Pointillism

Shape

Load and Display SVG
Disable Style
Scale Shape
Get Child

Color

Hue
Saturation
Brightness
Color Variables
Relativity
Linear Gradient
Radial Gradient
Wave Gradient

Math

Increment/Decrement
Operator Precedence
Distance 1D
Distance 2D
Map
Sine
Sine and Cosine
Sine Wave
Additive Wave
Polar to Cartesian
Arctangent
Graphing 2D Equation
Random
Double Random
Noise 1D
NoiseWave

Input

Mouse 1D
Mouse 2D
MousePress
Mouse Signals
Easing
Constrain
Storing Input
Mouse Functions
Keyboard
Keyboard Functions
Milliseconds
Clock

Transform

Translate
Scale
Rotate
RotateXY
RotatePushPop
Arm

Objects

Objects
Multiple Constructors
Composite Objects
Inheritance

Lights

On/Off
Directional
Spot
Reflection
Mixture
Mixture Grid

Camera

Perspective
Move Eye

Topic Examples. *Programs about to animation, interaction, motion, simulation, and more...*

Geometry

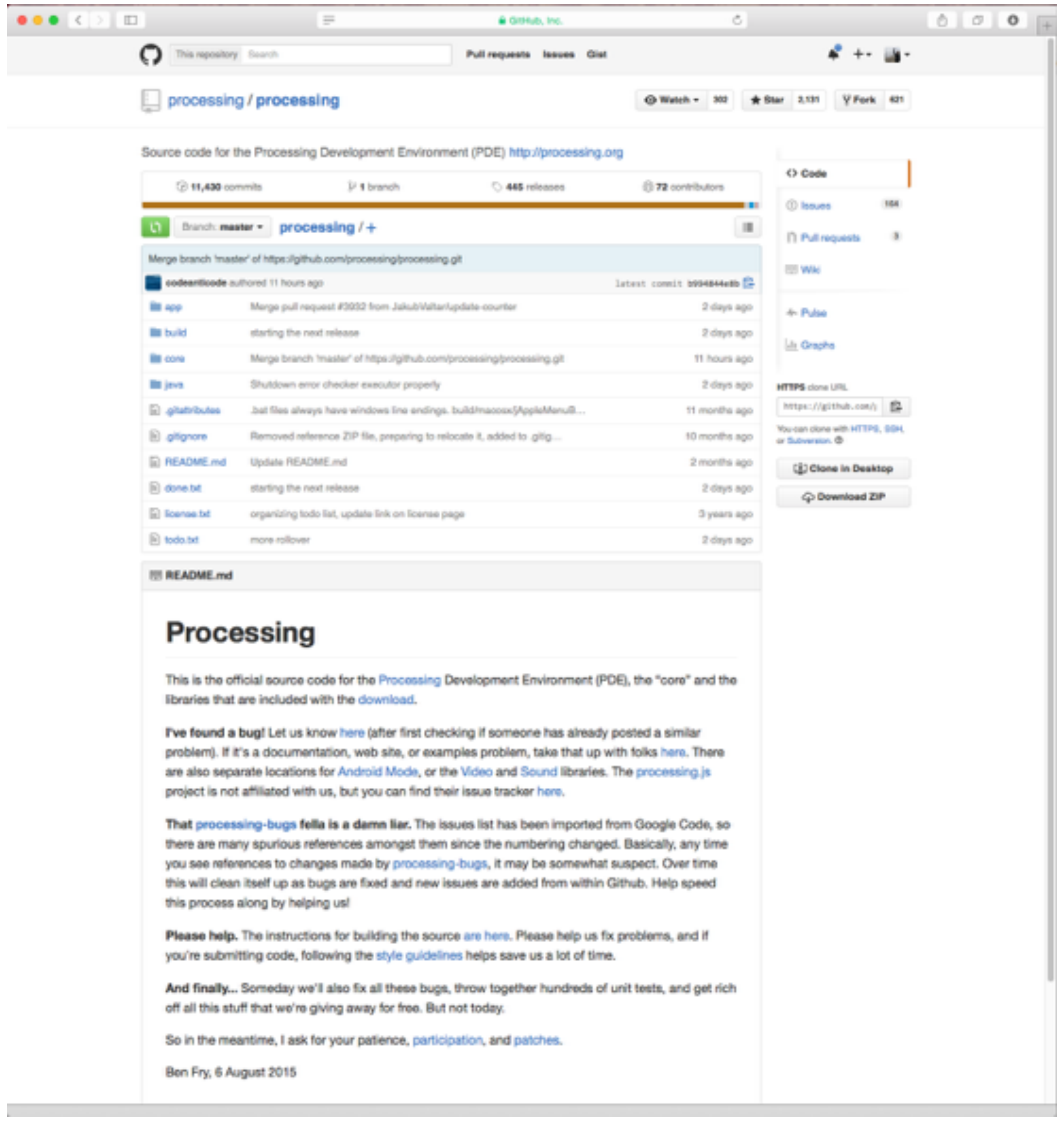
Vertices
RGB Cube
Space Junk
Shape Transform

Motion

Linear
Bounce
Bouncy Bubbles
Moving On Curves

Textures

Texture Quad
Texture Triangle
Texture Cylinder
Texture Cube



Source code for the Processing Development Environment (PDE) http://processing.org

11,430 commits · 1 branch · 445 releases · 72 contributors

Merge branch 'master' of https://github.com/processing/processing.git

codeanticode authored 11 hours ago — latest commit b994844e8b

app	Merge pull request #3932 from JakubValtar/update-counter	2 days ago
build	starting the next release	2 days ago
core	Merge branch 'master' of https://github.com/processing/processing.git	11 hours ago
java	Shutdown error checker executor properly	2 days ago
.gitattributes	.bat files always have windows line endings. build/macosx/AppleMenuB...	11 months ago
.gitignore	Removed reference ZIP file, preparing to relocate it, added to .gitig...	10 months ago
README.md	Update README.md	2 months ago
done.txt	starting the next release	2 days ago
license.txt	organizing todo list, update link on license page	3 years ago
todo.txt	more rollover	2 days ago

You can clone with HTTPS, SSH, or Subversion.

README.md

Processing

This is the official source code for the Processing Development Environment (PDE), the "core" and the libraries that are included with the download.

I've found a bug! Let us know here (after first checking if someone has already posted a similar problem). If it's a documentation, web site, or examples problem, take that up with folks here. There are also separate locations for Android Mode, or the Video and Sound libraries. The processing.js project is not affiliated with us, but you can find their issue tracker here.

That processing-bugs fella is a damn liar. The issues list has been imported from Google Code, so there are many spurious references amongst them since the numbering changed. Basically, any time you see references to changes made by processing-bugs, it may be somewhat suspect. Over time this will clean itself up as bugs are fixed and new issues are added from within Github. Help speed this process along by helping us!

Please help. The instructions for building the source are here. Please help us fix problems, and if you're submitting code, following the style guidelines helps save us a lot of time.

And finally... Someday we'll also fix all these bugs, throw together hundreds of unit tests, and get rich off all this stuff that we're giving away for free. But not today.

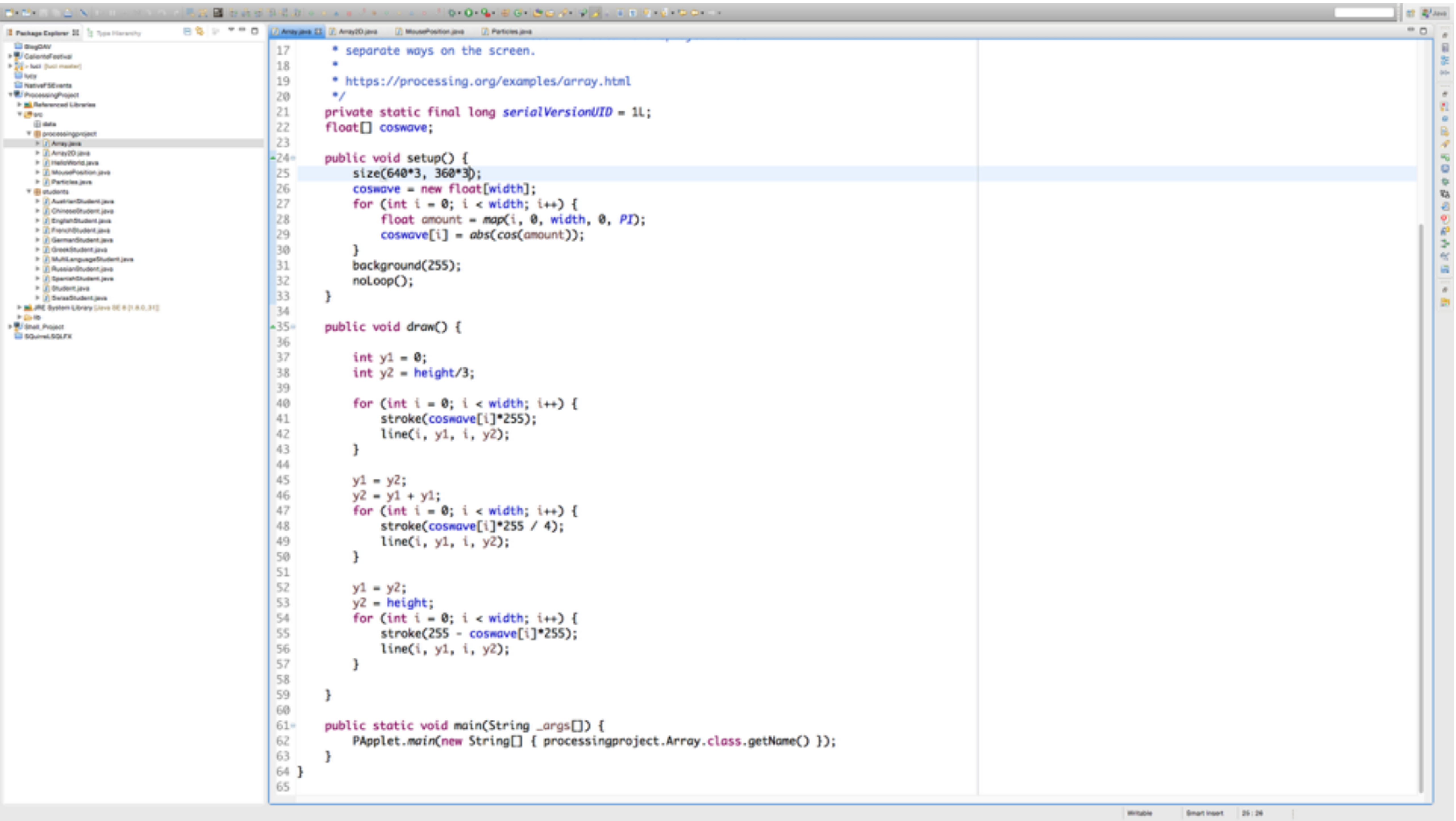
So in the meantime, I ask for your patience, participation, and patches.

Ben Fry, 6 August 2015

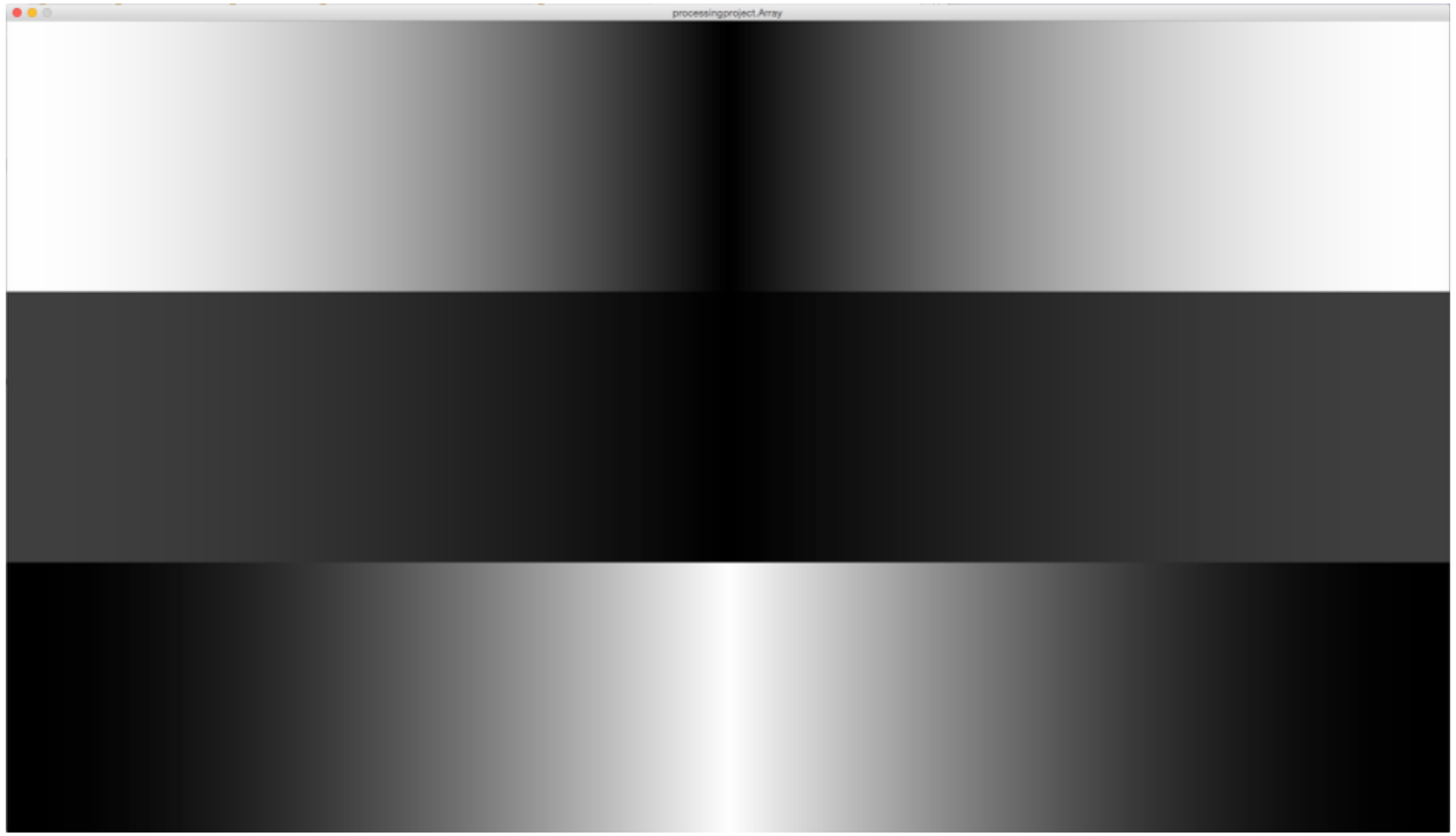
Processing

Examples 1: Array

https://processing.org/examples/array.html



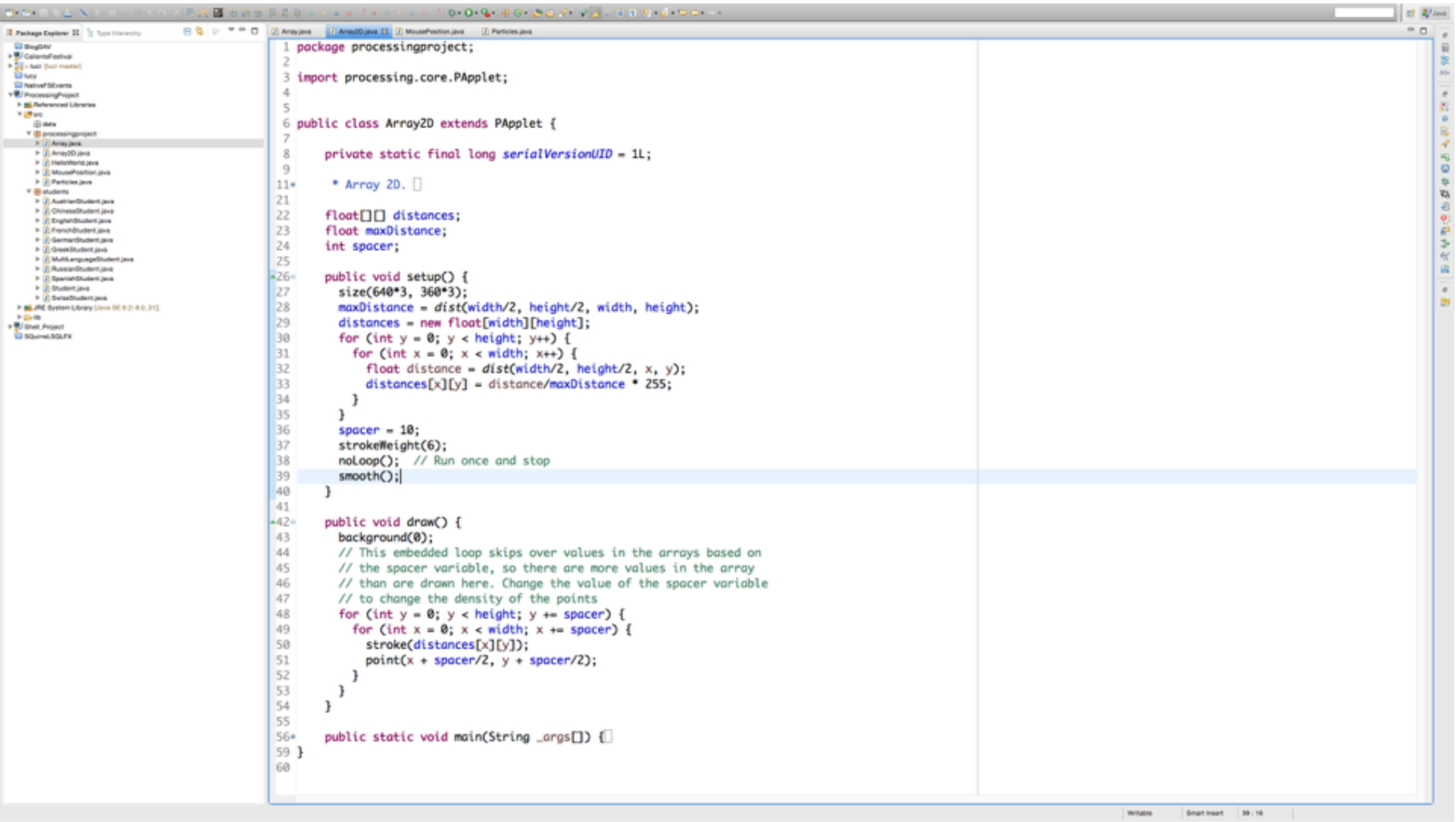
```
     * separate ways on the screen.
     *
     * https://processing.org/examples/array.html
     */
    private static final long serialVersionUID = 1L;
    float[] coswave;

    public void setup() {
        size(640*3, 360*3);
        coswave = new float[width];
        for (int i = 0; i < width; i++) {
            float amount = map(i, 0, width, 0, PI);
            coswave[i] = abs(cos(amount));
        }
        background(255);
        noLoop();
    }

    public void draw() {

        int y1 = 0;
        int y2 = height/3;

        for (int i = 0; i < width; i++) {
            stroke(coswave[i]*255);
            line(i, y1, i, y2);
        }

        y1 = y2;
        y2 = y1 + y1;
        for (int i = 0; i < width; i++) {
            stroke(coswave[i]*255 / 4);
            line(i, y1, i, y2);
        }

        y1 = y2;
        y2 = height;
        for (int i = 0; i < width; i++) {
            stroke(255 - coswave[i]*255);
            line(i, y1, i, y2);
        }

    }

    public static void main(String _args[]) {
        PApplet.main(new String[] { processingproject.Array.class.getName() });
    }
}
```



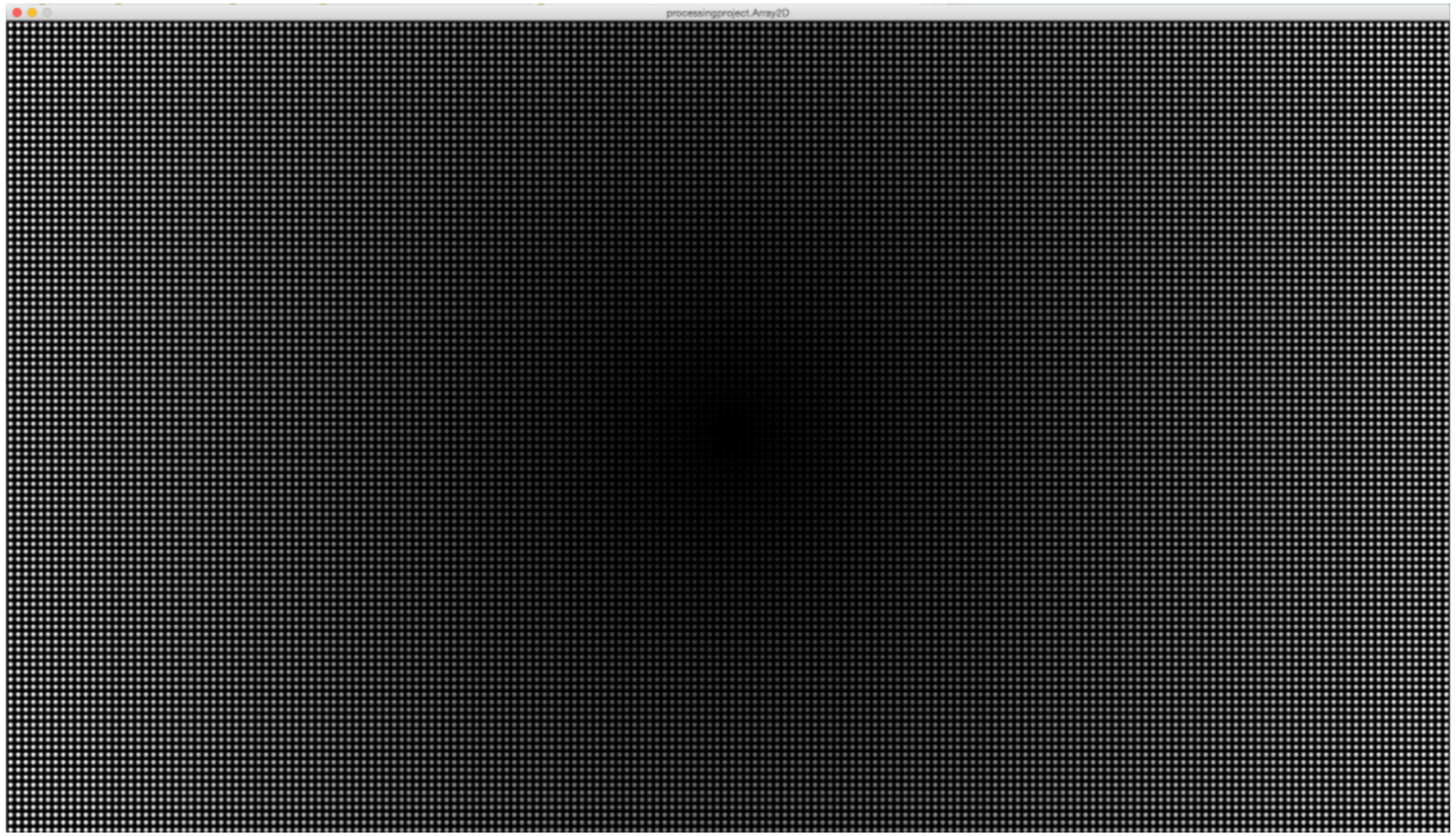
Processing

Examples 2: Array2D

https://processing.org/examples/array2d.html

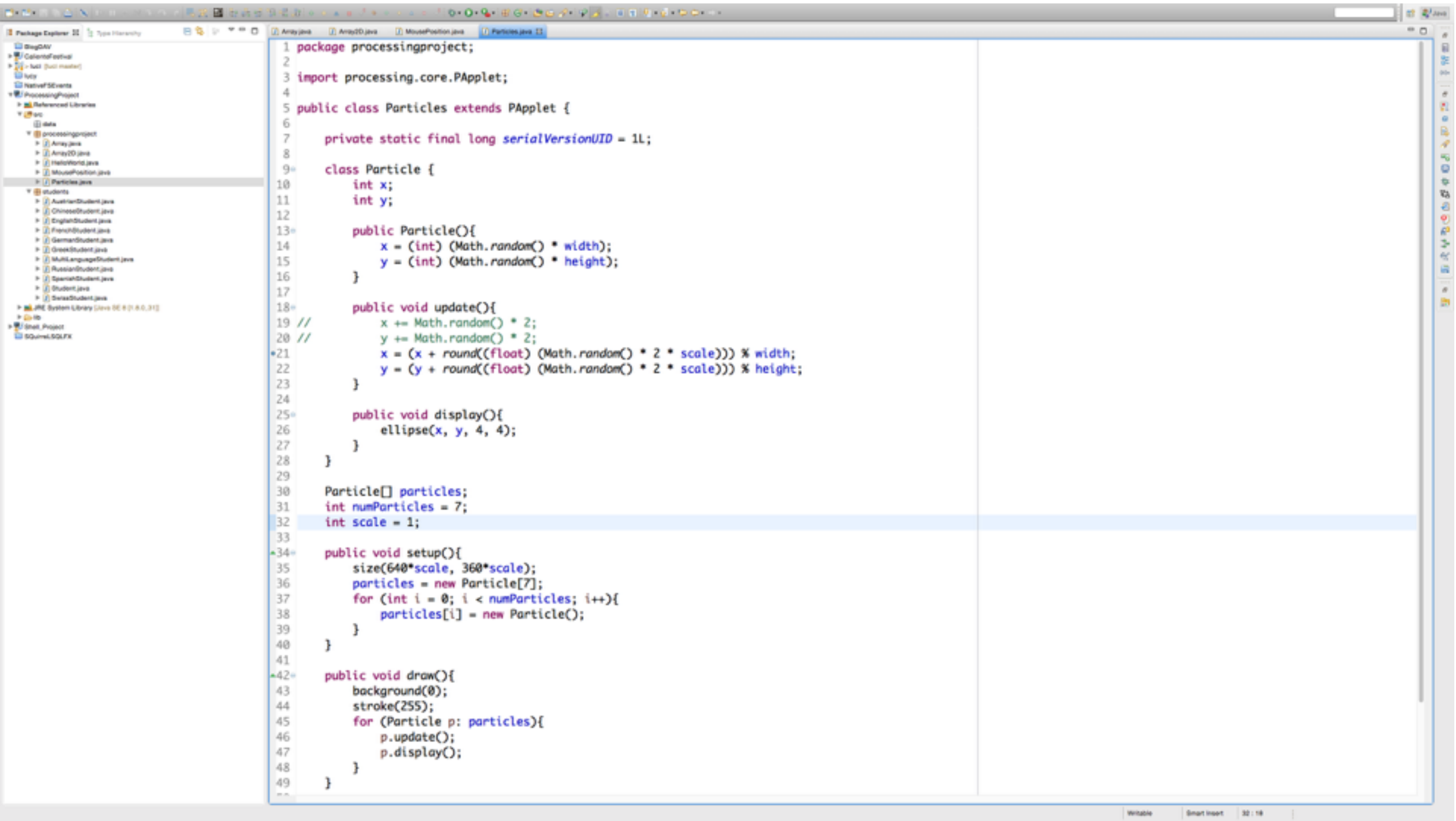


```
package processingproject;

import processing.core.PApplet;


public class Array2D extends PApplet {

    private static final long serialVersionUID = 1L;

     * Array 2D. 

    float[][] distances;
    float maxDistance;
    int spacer;

    public void setup() {
      size(640*3, 360*3);
      maxDistance = dist(width/2, height/2, width, height);
      distances = new float[width][height];
      for (int y = 0; y < height; y++) {
        for (int x = 0; x < width; x++) {
          float distance = dist(width/2, height/2, x, y);
          distances[x][y] = distance/maxDistance * 255;
        }
      }
      spacer = 10;
      strokeWeight(6);
      noLoop();  // Run once and stop
      smooth();
    }

    public void draw() {
      background(0);
      // This embedded loop skips over values in the arrays based on
      // the spacer variable, so there are more values in the array
      // than are drawn here. Change the value of the spacer variable
      // to change the density of the points
      for (int y = 0; y < height; y += spacer) {
        for (int x = 0; x < width; x += spacer) {
          stroke(distances[x][y]);
          point(x + spacer/2, y + spacer/2);
        }
      }
    }

    public static void main(String _args[]) {
}

```

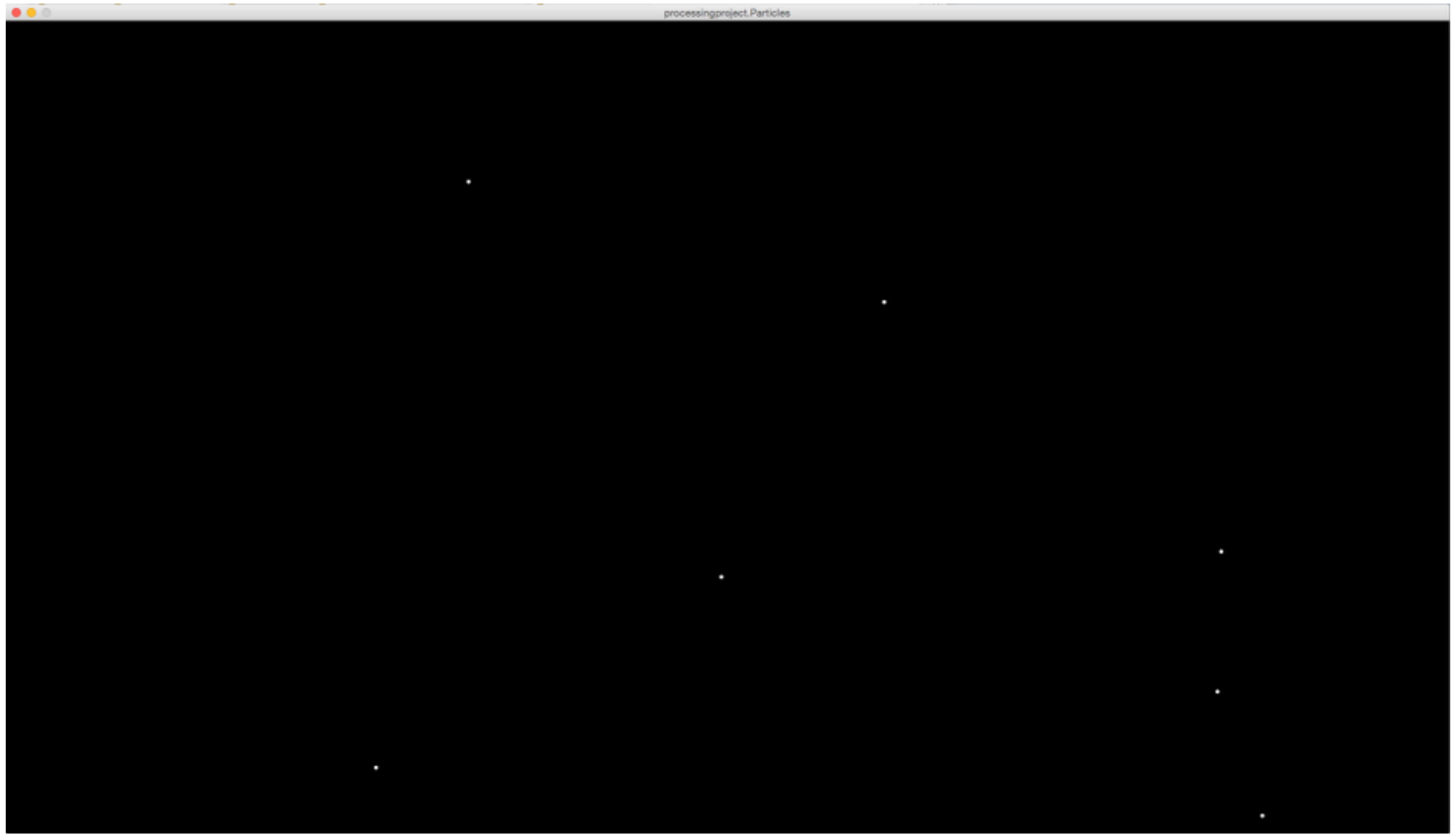


Processing

Examples 3: Particles



```java
package processingproject;

import processing.core.PApplet;

public class Particles extends PApplet {

    private static final long serialVersionUID = 1L;

    class Particle {
        int x;
        int y;

        public Particle(){
            x = (int) (Math.random() * width);
            y = (int) (Math.random() * height);
        }

        public void update(){
//          x += Math.random() * 2;
//          y += Math.random() * 2;
            x = (x + round((float) (Math.random() * 2 * scale))) % width;
            y = (y + round((float) (Math.random() * 2 * scale))) % height;
        }

        public void display(){
            ellipse(x, y, 4, 4);
        }
    }

    Particle[] particles;
    int numParticles = 7;
    int scale = 1;

    public void setup(){
        size(640*scale, 360*scale);
        particles = new Particle[7];
        for (int i = 0; i < numParticles; i++){
            particles[i] = new Particle();
        }
    }

    public void draw(){
        background(0);
        stroke(255);
        for (Particle p: particles){
            p.update();
            p.display();
        }
    }
```



Home Task

a) Modify the Array2D example. Make the black hole follow the mouse.

b) Create a Processing Applet in which at least 3 different geometric forms are moving.
(If we find out the code was copied from the internet ⇢ 0 pts)

Due 25.10.2015,
send to treyer@arch.ethz.ch