Arrays, Comments, Graphics, and Management

Overview

Arrays
- Declaring, allocating, and initializing
- Indexing and length
- References
- Multidimensional arrays

Comments

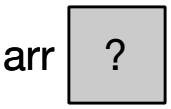
Graphics
- Still pictures
- Coordinate systems
- Animation
- Graphic User Interfaces (GUIs)

Management

Arrays

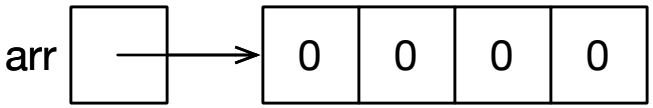
Declaring, allocating, and initializing

```
int[] arr;
```

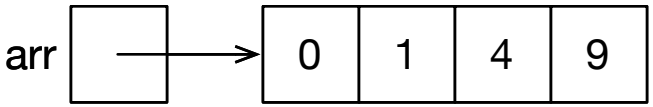



```
arr = new int[4];
```

Elements get default values.




```
for (int i = 0; i < arr.length; i++)
{
  arr[i] = i * i;
}
```



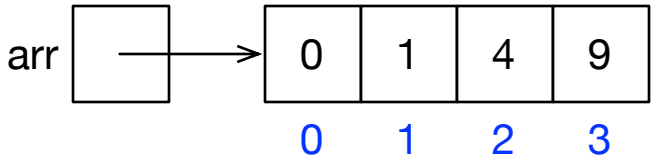
Equivalently:

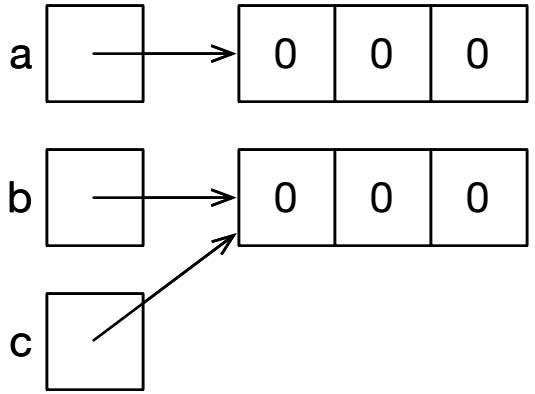
```
int[] arr = {0, 1, 4, 9};
```

If `arr` were already declared:

```
arr = new int[] {0, 1, 4, 9};
```

Indexing and length



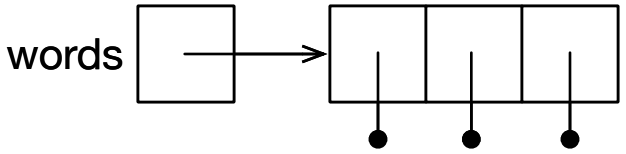


First element: `arr[0]`

Last element: `arr[3]`

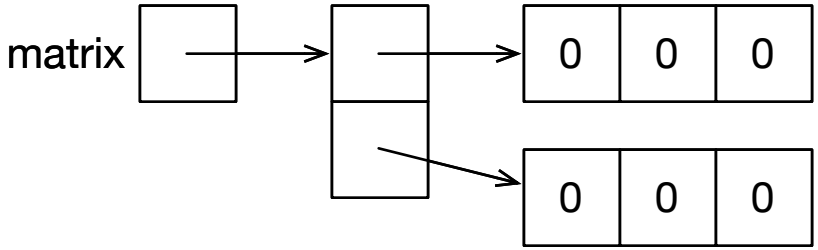
Length: `arr.length`

In general, the last element is `arr[arr.length - 1]`



References

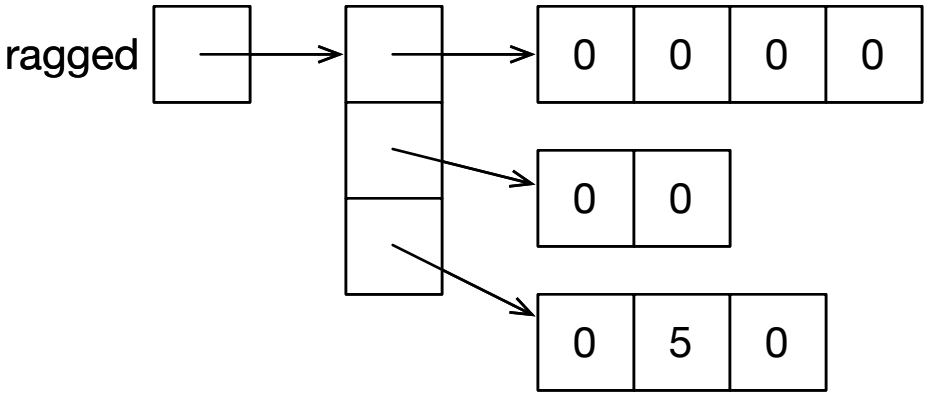
```
int[] a = new int[3];
int[] b = new int[3];
int[] c = b;
```



Now `b == c` (*aliasing*) but `b != a`.

A variable of an object type (e.g., an array) contains a *reference* (*pointer*) to some object. It refers to the entire object, not some part of it.

== is only true if both boxes being compared contain the same primitive value or references to the same object (not merely different objects with the same contents).



`null`, a reference that doesn't point at anything, is the default value for object types.

```
String[] words = new String[3];
```

Multidimensional arrays

```
int[][] matrix = new int[2][3];
```

Now:

`matrix.length` is 2.

`matrix[0].length` is 3.

`matrix[0][2]` is the upper right element.

`matrix[1]` is the bottom row.

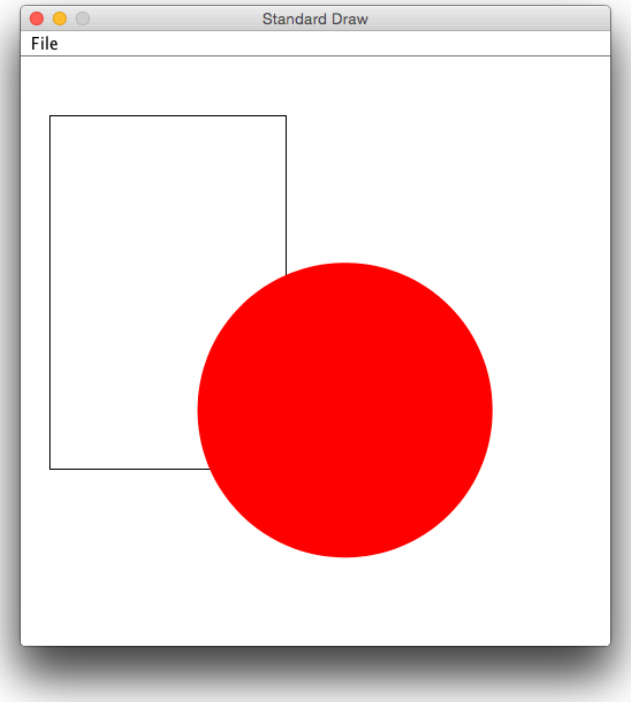
How do you make this?

Comments

```
// C++-style comment (to end of line)

/* C-style comment */

/** Javadoc comment */
```



Graphics

Still pictures

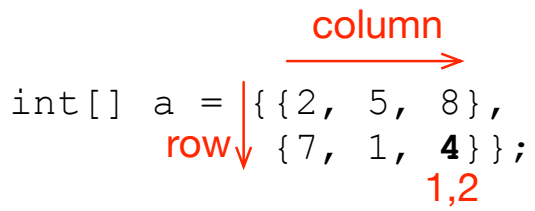
```
StdDraw.rectangle(0.25, 0.6, 0.2, 0.3);
StdDraw.setPenColor(StdDraw.RED);
StdDraw.filledCircle(0.55, 0.4, 0.25);
```

See the StdDraw API for much more!

Coordinate systems

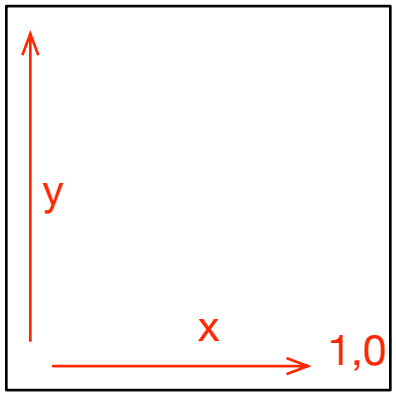
Arrays and mathematical matrices:

row, column



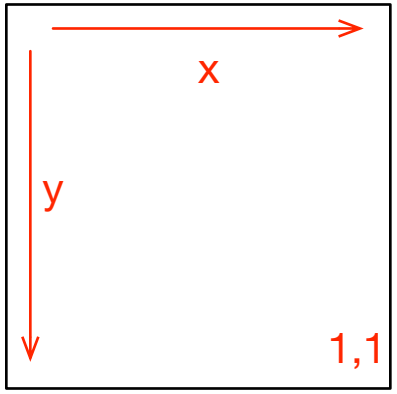
StdDraw and Cartesian plane:

x, y



Many computer graphics systems (including Java Swing):

x, y



For added fun, computer scientists start counting at 0 and everyone else starts at 1.

Animation

```
for (double i = 0.0; i < 1.0; i += 0.001) {
   StdDraw.clear();
   StdDraw.filledCircle(i, i, 0.1);
   StdDraw.show();
   StdDraw.pause(100); // msec (optional to slow animation)
}
```

Your life will be vastly easier if there is one method in charge of all of your drawing. (It may call others for subtasks.) Clear the window, redraw everything, and then show it. Don't try to edit the screen.

To avoid flickering, do this once at before starting your loop:

```
StdDraw.enableDoubleBuffering();
```

Graphic User Interfaces (GUIs)

In a loop:

1. Get input from the user
2. Modify your data structures
3. Redraw everything

Mouse events

```
while (!StdDraw.isMousePressed()) {
    // Wait for mouse press
}
```

Do something with `StdDraw.mouseX()`

Do something with `StdDraw.mouseY()`

```
while (StdDraw.isMousePressed()) {
    // Wait for mouse release
}
```

Keyboard events

```
if (StdDraw.hasNextKeyTyped()) {
    Do something with StdDraw.nextKeyTyped()
}
```

Queues up keys typed.
`nextKeyTyped` returns a char.
Cannot independently detect shift, ctrl, etc.

```
StdDraw.isKeyPressed(java.awt.event.KeyEvent.VK_SHIFT)
```

Does not queue up keys typed.
`isKeyPressed` returns a boolean indicating whether a specific key is down *right now*.
Keypresses are what you'd expect: `VK_A`, `VK_SPACE`, etc. Google KeyEvent for details.

Management

Management is about effectively using resources (time, computers, people, etc.).

Things to consider:

- Are workers (humans or machines) waiting for jobs to do?
- Are jobs waiting for workers to do them?
- Does the order in which jobs are done matter?
- How can jobs be broken down into smaller subtasks?
- Are resources being wasted due to miscommunication?
- How can workers be made more effective? (This includes better training and motivation for humans and better hardware and software for machines.)

The manager leads, but the whole team is responsible for these things.

Review

Arrays must be allocated.

Variables of object types (including arrays) are references.

Arrays can be multidimensional.

There are several forms of comments in Java.

StdDraw makes creating simple GUIs easy.

Management is about using resources effectively.