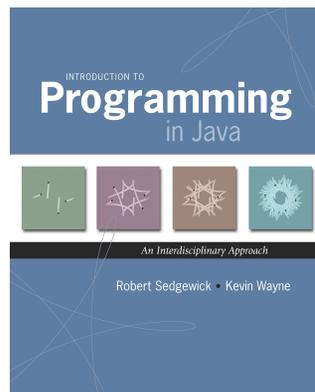


## 3.3 Designing Data Types



Introduction to Programming in Java: An Interdisciplinary Approach · Robert Sedgewick and Kevin Wayne · Copyright © 2008 · \* \*

Procedural programming. [verb-oriented]

- Tell the computer to do this.
- Tell the computer to do that.

Alan Kay's philosophy. Software is a **simulation** of the real world.

- We know (approximately) how the real world works.
- Design software to model the real world.

Objected oriented programming (OOP). [noun-oriented]

- Programming paradigm based on data types.
- Identify **things** that are part of the problem domain or solution.
- Things in the world **know** things: instance variables.
- Things in the world **do** things: methods.

### Alan Kay

Alan Kay. [Xerox PARC 1970s]

- Invented Smalltalk programming language.
- Conceived Dynabook portable computer.
- Ideas led to: laptop, modern GUI, OOP.



"The computer revolution hasn't started yet."

"The best way to predict the future is to invent it."

"If you don't fail at least 90 percent of the time, you're not aiming high enough."



Alan Kay  
2003 Turing Award

## Encapsulation



Bond. What's your escape route?

Saunders. Sorry old man. Section 26 paragraph 5, that information is on a need-to-know basis only. I'm sure you'll understand.

## Encapsulation

**Data type.** Set of values and operations on those values.

Ex. int, String, Complex, Vector, Document, GuitarString, Tour, ...

**Encapsulated data type.** Hide internal representation of data type.

**Separate implementation from design specification.**

- **Class** provides data representation and code for operations.
- **Client** uses data type as black box.
- **API** specifies contract between client and class.

## Intuition



Client



API

- volume
- change channel
- adjust picture
- decode NTSC signal



Implementation

- cathode ray tube
- electron gun
- Sony Wega 36XBR250
- 216 pounds

client needs to know  
how to use API

implementation needs to know  
what API to implement

Implementation and client need to  
agree on API ahead of time.

5

6

## Intuition



Client



API

- volume
- change channel
- adjust picture
- decode NTSC signal



Implementation

- gas plasma monitor
- Samsung FPT-6374
- wall mountable
- 4 inches deep

client needs to know  
how to use API

implementation needs to know  
what API to implement

Can substitute better implementation  
without changing the client.

## Counter Data Type

**Counter.** Data type to count electronic votes.

```
public class Counter {  
    public int count;  
    public final String name;  
    public Counter(String id) { name = id; }  
    public void increment() { count++; }  
    public int value() { return count; }  
}
```

**Legal Java client.**

```
Counter c = new Counter("Volusia County");  
c.count = -16022;
```

**Oops.** Al Gore receives -16,022 votes in Volusia County, Florida.

7

8

## Counter Data Type

Counter. Encapsulated data type to count electronic votes.

```
public class Counter {
    private int count;
    public final String name;

    public Counter(String id) { name = id; }
    public void increment() { count++; }
    public int value() { return count; }
}
```

Does not compile.

```
Counter c = new Counter("Volusia County");
c.count = -16022;
```

Benefit. Can guarantee that each data type value remains in a consistent state.

## Time Bombs

Internal representation changes.

- [Y2K] Two digit years: January 1, 2000.
- [Y2038] 32-bit seconds since 1970: January 19, 2038.
- [VIN numbers] We'll run out by 2010.



www.cartoonstock.com/directory/m/millennium\_time-bomb.asp

Lesson. By exposing data representation to client, need to sift through millions of lines of code in client to update.

## Changing Internal Representation

Encapsulation.

- Keep data representation hidden with `private` access modifier.
- Expose API to clients using `public` access modifier.

```
public class Complex {
    private final double re, im;

    public Complex(double re, double im) { ... }
    public double abs() { ... }
    public Complex plus(Complex b) { ... }
    public Complex times(Complex b) { ... }
    public String toString() { ... }
}
```

e.g., to polar coordinates

Advantage. Can switch internal representation without changing client.

Note. All our data types are already encapsulated!

## Ask, Don't Touch

Encapsulated data types.

- Don't touch data and do whatever you want.
- Instead, ask object to manipulate its data.

"Ask, don't touch."

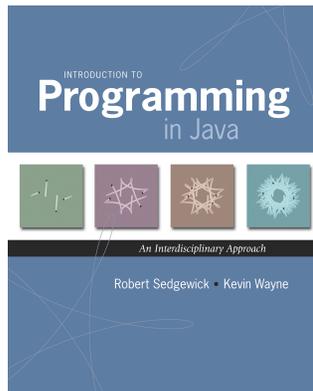


Adele Goldberg  
Former president of ACM  
Co-developed Smalltalk

Lesson. Limiting scope makes programs easier to maintain and understand.

"principle of least privilege"

## 3.5 Modular Programming



*Introduction to Programming in Java: An Interdisciplinary Approach · Robert Sedgewick and Kevin Wayne · Copyright © 2008 · \* \**

### Data Visualization

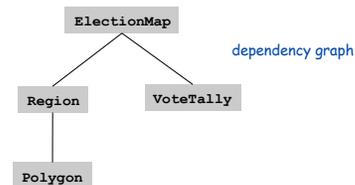
**Challenge.** Visualize election results.

*“If I can't picture it, I can't understand it.”*  
— *Albert Einstein*

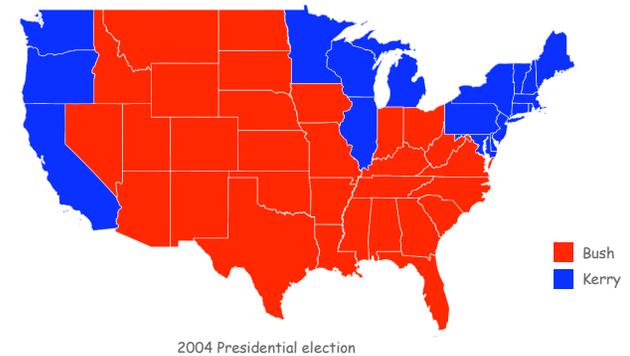


**Basic approach.**

- Gather data from web sources, save in local file.
- Build **modular program** that reads files, draws maps.



## Case Study: Red States, Blue States



### Modular Programming

**Modular programming.**

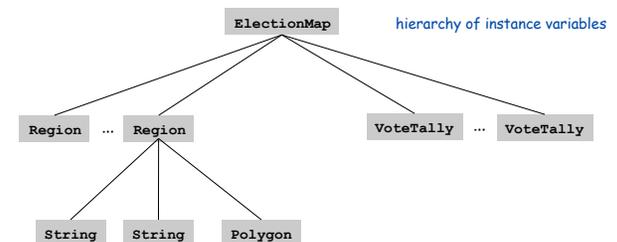
- Model problem by decomposing into components.
- Develop data type for each component.

**Polygon.** Geometric primitive.

**Region.** Name, postal abbreviation, polygonal boundary.

**Vote tally.** Number of votes for each candidate.

**Election map.** Regions and corresponding vote tallies for a given election.



# Data Sources

## Boundary Data: States within the Continental US

### Geometric data. [US census bureau]

- [www.census.gov/tiger/boundary](http://www.census.gov/tiger/boundary)
- NJ.txt has boundaries of every county in New Jersey.
- USA.txt that has boundary of every state.

format useful for programmers

### Election results. [David Leip]

- <http://uselectionatlas.org/RESULTS>
- Interactive and graphical.
- Need to screen-scrape to get data.

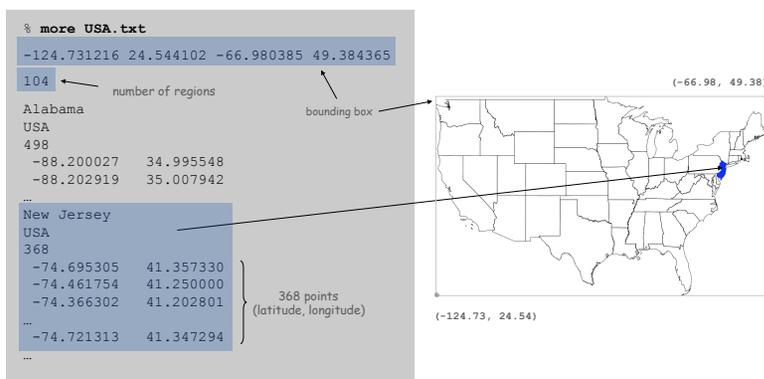
format useful for browsers and end-users  
(need to parse to extract raw data)

### Emerging standard.

- Publish data in text form on the web (like geometric data).
- Write programs to produce visuals (like we're doing!)
- Mashups.

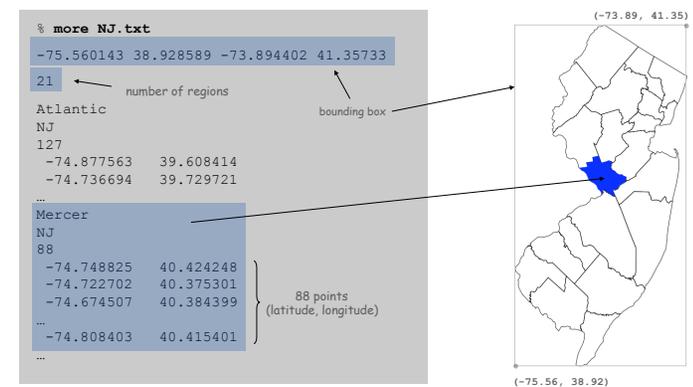
## Boundary Data: States within the Continental US

USA data file. State names and boundary points.



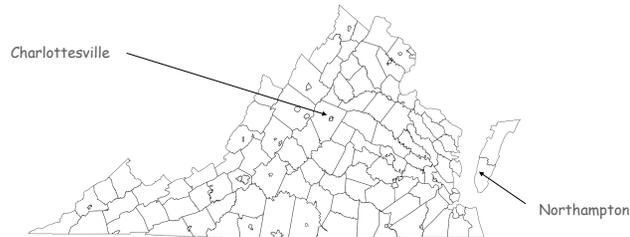
## Boundary Data: Counties within a State

State data files. County names and boundary points.

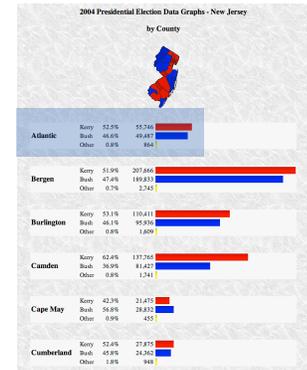


**Pieces.** A state can be comprised of several disjoint polygons.

**Holes.** A county can be entirely inside another county.



**Screen scrape.** Download html from web and parse.



county name is text between <b> and </b> tags that occurs after width:100px

```
<div>
<br /><b>2004 Presidential Election Data Graphs - New
Jersey<br /><br /></b>
</div>
<table border="1">
<tr>
<td colspan="3">Atlantic</td>
<td colspan="3">Kerry</td>
<td colspan="3">52.0%</td>
<td colspan="3">55,746</td>
<td colspan="3">Bush</td>
<td colspan="3">46.6%</td>
<td colspan="3">49,487</td>
<td colspan="3">Other</td>
<td colspan="3">0.8%</td>
<td colspan="3">864</td>
</tr>
<tr>
<td colspan="3">Bergen</td>
<td colspan="3">Kerry</td>
<td colspan="3">51.9%</td>
<td colspan="3">207,666</td>
<td colspan="3">Bush</td>
<td colspan="3">47.4%</td>
<td colspan="3">189,833</td>
<td colspan="3">Other</td>
<td colspan="3">0.7%</td>
<td colspan="3">3,741</td>
</tr>
<tr>
<td colspan="3">Burlington</td>
<td colspan="3">Kerry</td>
<td colspan="3">53.1%</td>
<td colspan="3">109,811</td>
<td colspan="3">Bush</td>
<td colspan="3">46.1%</td>
<td colspan="3">95,696</td>
<td colspan="3">Other</td>
<td colspan="3">0.8%</td>
<td colspan="3">1,600</td>
</tr>
<tr>
<td colspan="3">Camden</td>
<td colspan="3">Kerry</td>
<td colspan="3">62.4%</td>
<td colspan="3">137,765</td>
<td colspan="3">Bush</td>
<td colspan="3">36.9%</td>
<td colspan="3">81,427</td>
<td colspan="3">Other</td>
<td colspan="3">0.6%</td>
<td colspan="3">1,741</td>
</tr>
<tr>
<td colspan="3">Cape May</td>
<td colspan="3">Kerry</td>
<td colspan="3">43.3%</td>
<td colspan="3">21,675</td>
<td colspan="3">Bush</td>
<td colspan="3">56.8%</td>
<td colspan="3">28,832</td>
<td colspan="3">Other</td>
<td colspan="3">0.9%</td>
<td colspan="3">459</td>
</tr>
<tr>
<td colspan="3">Cumberland</td>
<td colspan="3">Kerry</td>
<td colspan="3">52.4%</td>
<td colspan="3">27,875</td>
<td colspan="3">Bush</td>
<td colspan="3">45.8%</td>
<td colspan="3">24,262</td>
<td colspan="3">Other</td>
<td colspan="3">1.8%</td>
<td colspan="3">948</td>
</tr>
</table>
```

<http://uselectionatlas.org/RESULTS/datagraph.php?year=2004&fips=34>

Election Scrapper (sketch)

```
int year = 2004; // election year
String usps = "NJ"; // United States postal code for New Jersey
int fips = 34; // FIPS code for New Jersey

String url = "http://uselectionatlas.org/RESULTS/datagraph.php";
In in = new In(url + "?year=" + year + "&fips=" + fips);
Out file = new Out(usps + year + ".txt");
String input = in.readAll();

while (true) {

    // screen scrape county name
    int p = input.indexOf("width:100px", p);
    if (p == -1) break;
    int from = input.indexOf("<b>", p);
    int to = input.indexOf("</b>", from);
    String county = input.substring(from + 3, to);

    // screen scrape vote totals for each candidate

    // save results to file
    file.println(county + ", " + bush + ", " + kerry + ", " + nader + ",");
}
```

extract text between <b> and </b> tags, that occurs after width:100px

More Pitfalls

Data sources have different conventions.

- FIPS codes: NJ vs. 34.
- County names: LaSalle vs. La Salle, Kings County vs. Brooklyn.

Plenty of other minor annoyances.

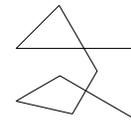
- Unreported results.
- Third-party candidates.
- Changes in county boundaries.

Bottom line. Need to clean up data (but write a program to do it!)

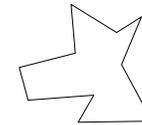
# Polygons and Regions

## Polygon Data Type

**Polygon.** Closed, planar path with straight line segments.  
**Simple polygon.** No crossing lines.



polygon  
(8 points)



simple polygon  
(10 points)



simple polygon  
(368 points)

26

## Polygon Data Type: Java Implementation

```
public class Polygon {
    private final int N;           // number of boundary points
    private final double[] x, y;  // the points (x[i], y[i])

    // read from input stream
    public Polygon(In in) {
        N = in.readInt();
        x = new double[N];
        y = new double[N];
        for (int i = 0; i < N; i++) {
            x[i] = in.readDouble();
            y[i] = in.readDouble();
        }
    }

    public void fill() { StdDraw.filledPolygon(x, y); }

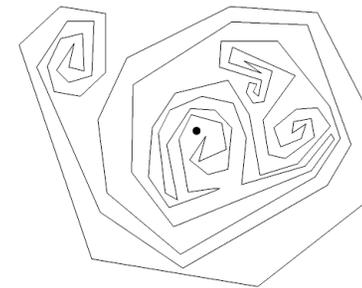
    public boolean contains(double x0, double y0) { ... } ← see COS 226
    public String toString() { ... } ← easy
}
```

27

## Polygon Diversion

Q. How to determine whether a point is **inside** a simple polygon?

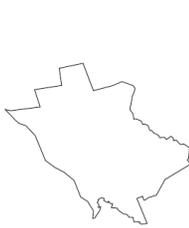
no two segments cross



28

## Region Data Type

**Region.** Represents a state or county.



Mercer, NJ  
88 point polygon



New Jersey, USA  
368 point polygon

30

## Region Data Type: Java Implementation

```
public class Region {
    private final String name; // name of region
    private final String usps; // postal abbreviation
    private final Polygon poly; // polygonal boundary

    public Region(String name, String usps, Polygon poly) {
        this.name = name;
        this.usps = usps;
        this.poly = poly;
    }

    public void draw() { poly.fill(); }

    public boolean contains(double x0, double y0) {
        return poly.contains(x0, y0);
    }

    public String toString() { ... }
}
```

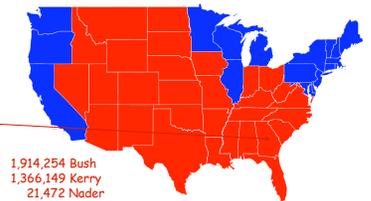
31

## Election Returns

### Election Returns: By State

Screen-scraping results. Number of votes for Bush, Kerry, Nader by region.

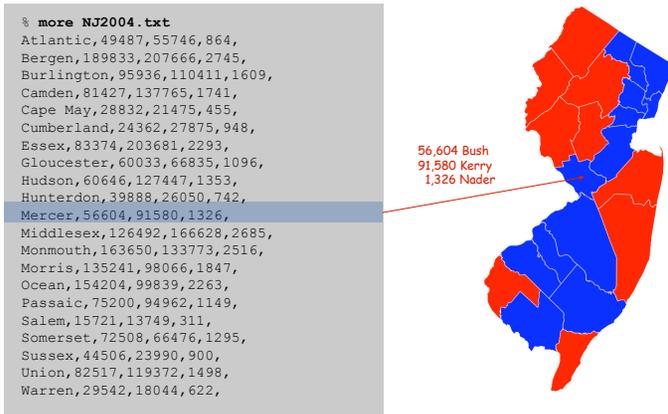
```
% more USA2004.txt
Alabama,1176394,693933,13122,
Alaska,190889,111025,10684,
Arizona,1104294,893524,14767,
Arkansas,572898,469953,12094,
California,5509826,6745485,164546,
Colorado,1101255,1001732,27343,
Connecticut,693826,857488,27455,
Delaware,171660,200152,3378,
District of Columbia,21256,202970,3360,
Florida,3964522,3583544,61744,
Georgia,1914254,1366149,21472,
Hawaii,194191,231708,3114,
Idaho,409235,181098,8114,
Kansas,736456,434993,16307,
Kentucky,1069439,712733,13688,
...
Virginia,1716959,1454742,26666,
Washington,1304894,1510201,43989,
West Virginia,423778,326541,5568,
Wisconsin,1478120,1489504,29383,
Wyoming,167629,70776,5023,
```



33

## Election Returns: By County

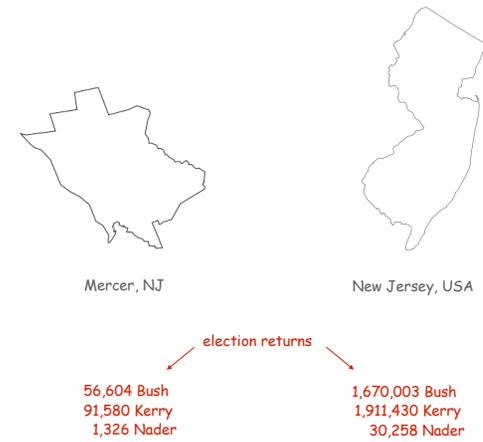
Screen-scraping results. Number of votes for Bush, Kerry, Nader by region.



34

## Vote Tally Data Type

VoteTally. Represents the election returns for one region.



35

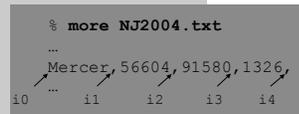
## Vote Tally Data Type: Java Implementation

```

public class VoteTally {
    private final int rep, dem, ind;

    public VoteTally(String name, String usps, int year) {
        In in = new In(usps + year + ".txt");
        String input = in.readAll();
        int i0 = input.indexOf(name);
        int i1 = input.indexOf(",", i0+1);
        int i2 = input.indexOf(",", i1+1);
        int i3 = input.indexOf(",", i2+1);
        int i4 = input.indexOf(",", i3+1);
        rep = Integer.parseInt(input.substring(i1+1, i2));
        dem = Integer.parseInt(input.substring(i2+1, i3));
        ind = Integer.parseInt(input.substring(i3+1, i4));
    }

    public Color getColor() {
        if (rep > dem) return StdDraw.RED;
        if (dem > rep) return StdDraw.BLUE;
        return StdDraw.BLACK;
    }
}
    
```



## Election Map

36

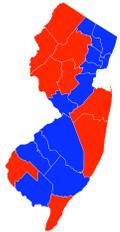
## Election Map Data Type

`ElectionMap`. Represents the election map for a given election.

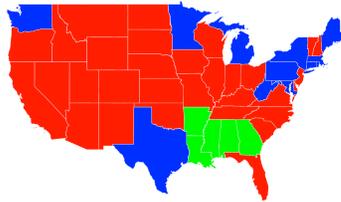
client

```
public static void main(String[] args) {
    String name = args[0];
    int year = Integer.parseInt(args[1]);
    ElectionMap election = new ElectionMap(name, year);
    election.show();
}
```

% java ElectionMap NJ 2004



% java ElectionMap USA 1968



38

## Election Map Data Type: Java Implementation

```
public class ElectionMap {
    private final int N;
    private final Region[] regions;
    private final VoteTally[] votes;

    public ElectionMap(String name, int year) {
        In in = new In(name + ".txt");
        // read in bounding box and rescale coordinates
        N = in.readInt();
        regions = new Region[N];
        votes = new VoteTally[N];
        for (int i = 0; i < N; i++) {
            String name = in.readLine();
            String usps = in.readLine();
            Polygon poly = new Polygon(in);
            regions[i] = new Region(name, usps, poly);
            votes[i] = new VoteTally(name, usps, year);
        }
    }

    public void show() {
        for (int i = 0; i < N; i++) {
            StdDraw.setPenColor(votes[i].getColor());
            regions[i].draw();
        }
    }
}
```

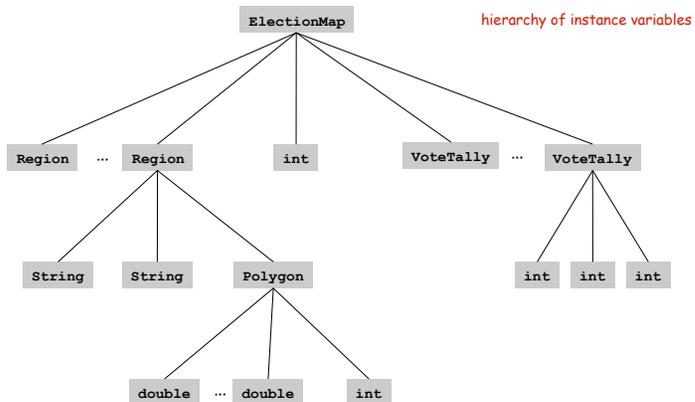
USE Polygon,  
Region, and  
VoteTally to  
build map

draw map

39

## Modular Programming

Modular program. Collection of data types.

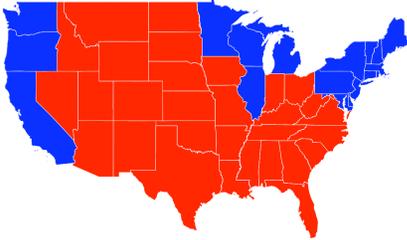


40

## Data Visualization

## Visual Display of Quantitative Information

Red states, blue states. Creates a misleading and polarizing picture.



Edward Tufte. Create charts with high data density that tell the truth.



42

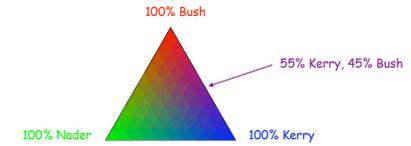
## Purple America

Idea. [Robert J. Vanderbei] Assign color based on number of votes.

<http://www.princeton.edu/~rvdb/JAVA/election2004>

- $a_1$  = Bush votes.
- $a_2$  = Nader votes.
- $a_3$  = Kerry votes.

$$(R, G, B) = \left( \frac{a_1}{a_1 + a_2 + a_3}, \frac{a_2}{a_1 + a_2 + a_3}, \frac{a_3}{a_1 + a_2 + a_3} \right)$$



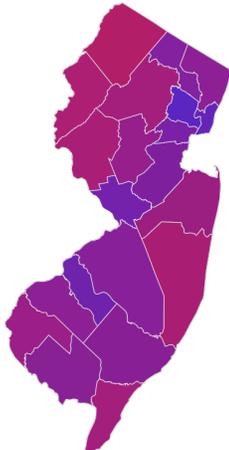
Implementation. Change one method!

```
public Color getColor() {
    int tot = dem + rep + ind;
    return new Color((float) rep/tot, (float) ind/tot, (float) dem/tot);
}
VoteTally.java
```

43

## Purple New Jersey

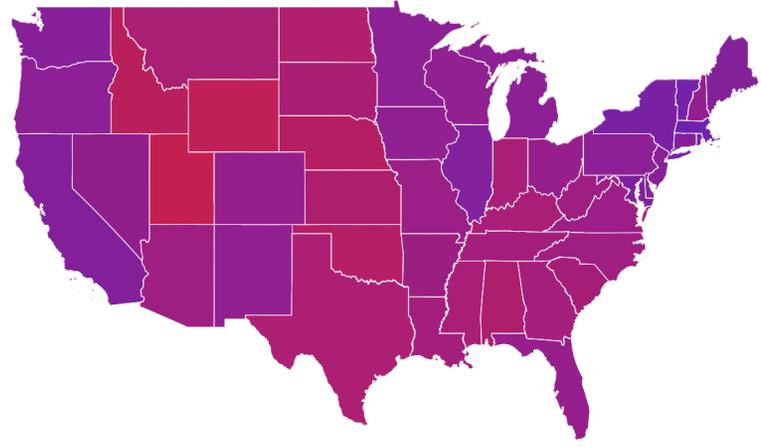
% java ElectionMap NJ 2004



44

## Purple America

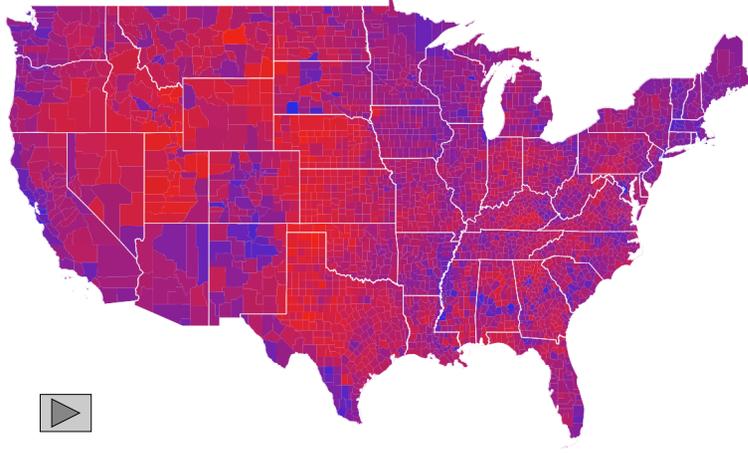
% java ElectionMap USA 2004



45

## Purple America

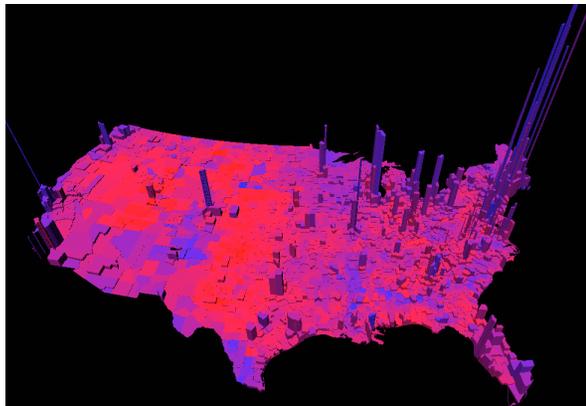
% java ElectionMap USA-county 2004



46

## 3D Visualization

**3D visualization.** Volume proportional to votes; azimuthal projection.



Robert J. Vanderbei  
www.princeton.edu/~rvdb/JAVA/election2004

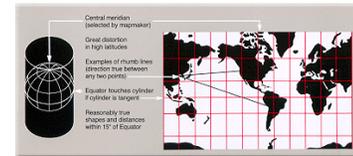
48

## Data Visualization: Design Issues

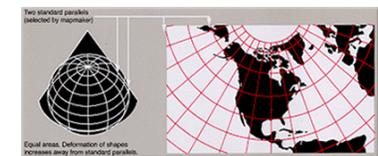
**Remark.** Humans perceive red more strongly than blue.

**Remark.** Amount of color should be proportional to number of votes, not geographic boundary.

**Remark.** Project latitude + longitude coordinates to 2d plane.



Mercator projection

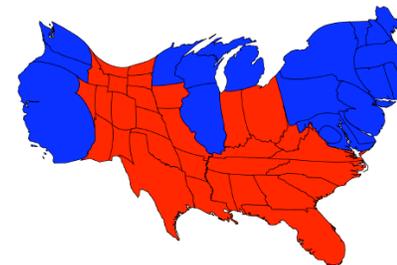


Albers projection

47

## Cartograms

**Cartogram.** Area of state proportional to number of electoral votes.

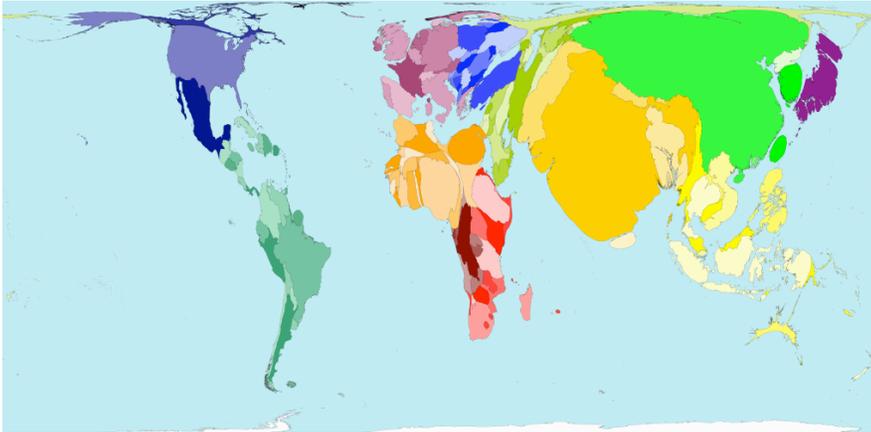


Michael Gastner, Cosma Shalizi, and Mark Newman  
www-personal.umich.edu/~mejn/election

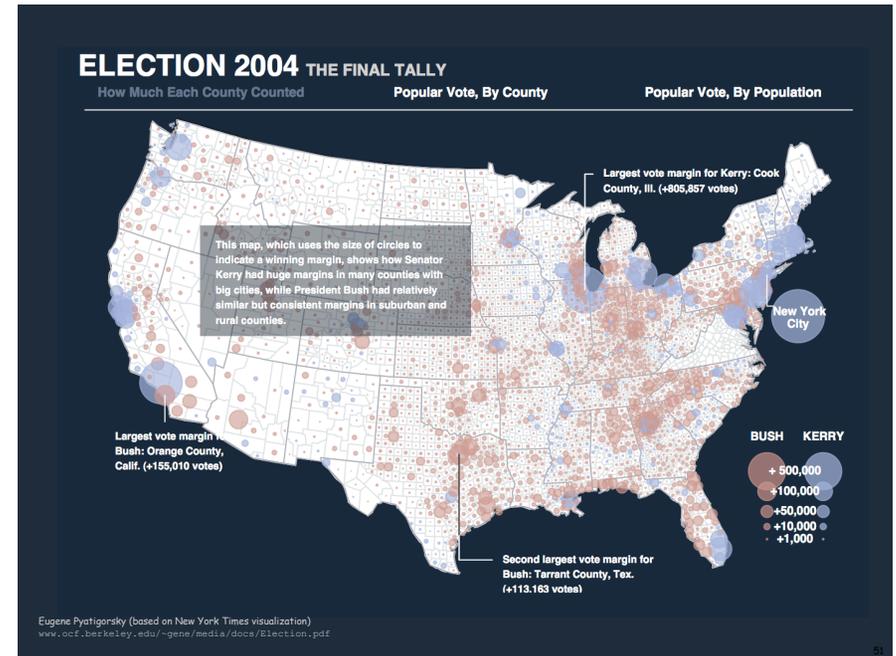
49

## Cartograms

**Cartogram.** Area of country proportional to population.



50



51

## Summary

### Modular programming.

- Break a large program into smaller independent components.
- Develop a **data type** for each component.
- Ex: Polygon, Region, VoteTally, ElectionMap, In, Out.

### Ex 1. Build large software project.

- Software architect specifies API.
- Each programmer implements one module.
- Debug and test each piece independently. [unit testing]

### Ex 2. Build reusable libraries.

- Language designer extends language with new data types.
- Programmers share extensive libraries.
- Ex: In, Out, Draw, Polygon, ...

**Data visualization.** You can do it! (worthwhile to learn from Tufte)

52