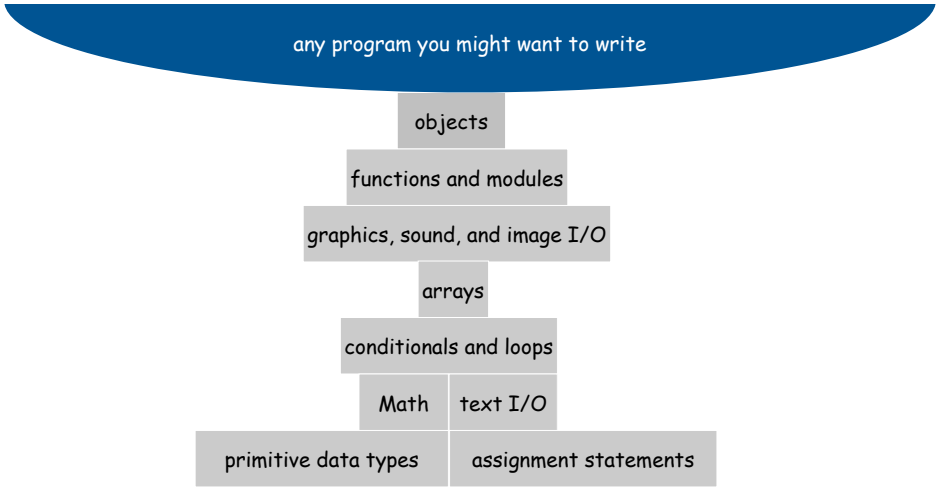
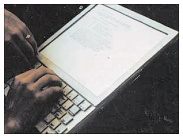
3.3 Designing Data Types



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 per cent of the time, you're not aiming high enough. "*
>
> *— Alan Kay*

Alan Kay
2003 Turing Award

Object Oriented Programming

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.

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.

Data type. Set of values and operations on those values.
Ex. `int`, `String`, `Complex`, `Vector`, `Document`, `GuitarString`, `Tour`, ...

Encapsulated (abstract) data type.
- Hide internal representation of values.
- Expose operations to client (in API).

Separates 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.

Bottom line.
You don't need to know how a data type is implemented in order to use it

Client

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

Implementation
- cathode ray tube
- electron gun
- Sony Wega 36XBR250
- 241 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.

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.

Counter Data Type

Counter. Encapsulated data type to count electronic votes.

```
public class Counter
{
   private int count;
   private 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.

Changing Internal Representation

Encapsulation.
- Keep data representation hidden with `private` access modifier.
- Expose API to client code 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!

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/millenium_time-bomb.asp

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

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

Thesis.
Limiting access to data makes programs easier to maintain and understand.

Immutability

Immutability

Immutable data type. Object's value cannot change once constructed.

mutable	*immutable*
`Picture`	`Charge`
`Histogram`	`Color`
`Turtle`	`Stopwatch`
`StockAccount`	`Complex`
`Counter`	`String`
Java arrays	primitive types

Immutability: Advantages and Disadvantages

Immutable data type. Object's value cannot change once constructed.

Advantages.
- Avoid aliasing bugs.
- Makes program easier to debug.
- Limits scope of code that can change values.
- Pass objects around without worrying about modification.

Disadvantage. New object must be created for every value.

Final Access Modifier

Final. Declaring an instance variable to be `final` means that you can assign it a value only once, in initializer or constructor.

this value doesn't change once the object is constructed

this value changes when instance method `increment()` is invoked

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

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

Advantages.

- Helps enforce immutability (immutable: all instance variables final).
- Prevents accidental changes.
- Makes program easier to debug.
- Documents that the value cannot not change.

TEQ on Data Type Design 1

[easy if you read pages 430-433]

Q. Is the following data type immutable?

```
public class Vector
{
   private final double[] coords;

   public Vector(double[] a)
   { // Make a defensive copy to ensure immutability.
     coords = a;
   }

   public Vector plus(Vector b)   { ... }
   public Vector times(Vector b)  { ... }
   public double dot(Vector b)    { ... }
}
```

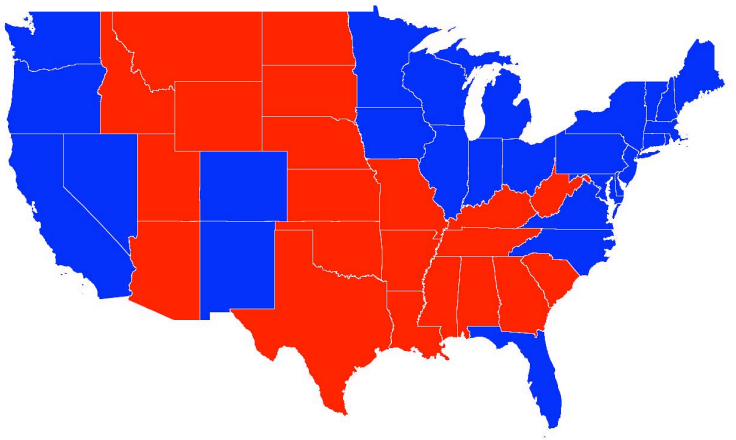

TEQ on Data Type Design 2

[easy if you read pages 430-433]

Q. Is the following data type immutable?

```
public class Vector
{
   private final double[] coords;

   public Vector(double[] a)
   { // Make a defensive copy to ensure immutability.
     coords = new double[a.length];
     for (int i = 0; i < a.length; i++)
     coords[i] = a[i];
   }

   public Vector plus(Vector b)   { ... }
   public Vector times(Vector b)  { ... }
   public double dot(Vector b)    { ... }
}
```


Modular Programming with Data Types: A Case Study



2008 Presidential election

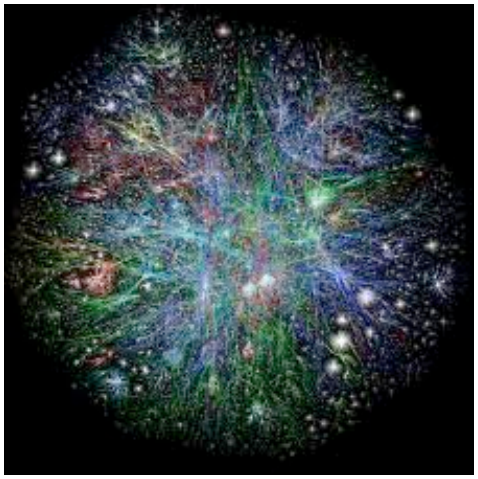
Modular Programming with Data Types

Challenge. Visualize election results.

Approach.

- Gather data from data sources on the web, save in local files.
- Build modular program that reads files, draws map.

Data Sources



Data Sources

Geometric data

TIGER: Topologically Integerated Geographic Encoding and Referencing

- `www.census.gov/tiger/boundary`
- text file `USA.txt` that has boundaries of every state
- text file `*.txt` for every state that has boundaries of every county

↖ useful for people who are writing programs

Election results

- `http://uselectionatlas.org/RESULTS`
- interactive and graphical
- need to screen-scrape to get data

↖ useful for people who want their programs written for them
(and who are therefore limited to the relatively few programs out there!)

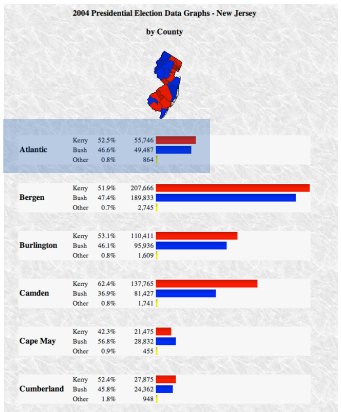
Emerging standard

- publish data in text form on the web (like geometric data)
- write programs to produce visuals (like we're doing!)

Screen Scraping the Election Returns

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

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



county name is text between and tags, that occurs after `width:100px`

```
<div>
<br /><b>2004 Presidential Election Data Graphs - New
Jersey<br /><br />by County</b><br /><br /><img
src="img.php?year=2004&amp;st=NJ&amp;type=map&amp;off=0"
alt="Map" /><br /><br /><div class="info"><table
cellpadding="2"><tr><td style="width:100px"
rowspan="3"><b>Atlantic</b></td><td class="cnd">Kerry</
td><td class="per">52.5&#37;</td><td class="dat">55,746</
td><td class="bar"><div class="bardem" style="width:
26.8%"> </div></td></tr><tr><td>Bush</td><td
class="per">46.6&#37;</td><td class="dat">49,487</
td><td><div class="barrep" style="width:23.8%"> </
div></td></tr><tr><td>Other</td><td
class="per">0.8&#37;</td><td class="dat">864</td><td><div
class="baroth" style="width:1.0%"> </div></td></
tr></table><br /><br /><table cellpadding="2"><tr><td
style="width:100px" rowspan="3"><b>Bergen</b></td><td
class="cnd">Kerry</td><td class="per">51.9&#37;</td><td
class="dat">207,666</td><td class="bar"><div
class="bardem" style="width:100.0%"> </div></td></
tr><tr><td>Bush</td><td class="per">47.4&#37;</td><td
class="dat">189,833</td><td><div class="barrep"
style="width:91.4%"> </div></td></tr><tr><td>Other</
td><td class="per">0.7&#37;</td><td class="dat">2,745</
td><td><div class="baroth" style="width:1.3%"> </
div></td></tr></table><br /><br /><table···
```

Election Scraper (sketch)

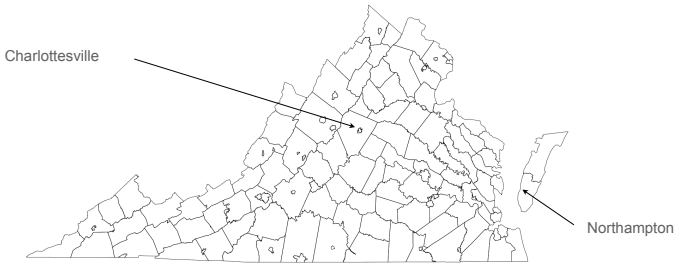
```
int year    = 2004;  // election year
String usps = "NJ";  // United States postal code for New Jersey
int fips    = 34;    // FIPS code for New Jersey

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)
{
  // 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);

  // scrape vote totals for each candidate
  int mccain = ...
  int obama = ...
  int other = ...


  // save results to file
  file.println(county + "," + mccain + "," + obama + "," + other + ",");
}
```

extract text between and tags, that occurs after width:100px

Pitfalls: Pieces and Holes

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

Holes. A county can be entirely inside another county.



Cleaning up the data

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, write-ins, changes in county boundaries,...

Design decisions.

- Write programs to clean up web data
- Keep results in local files (web data/format might change)

also USA2004.txt, NJ2004.txt ... for past elections

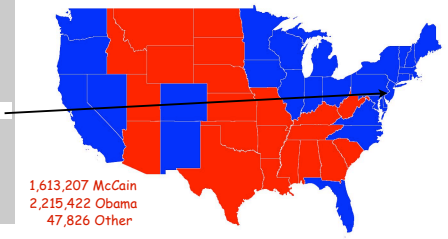
Starting point for case study

- **USA2008.txt**: election returns for US, one line per state
- **NJ2008.txt, ...** : election returns for each state, one line per county
- **USA.txt**: boundary data for US, one entry per state
- **NJ.txt, ...** : boundary data for each state, one entry per county

Election Return Data: By State

Screen-scraping results. Votes for McCain, Obama, Other by region.

```
% more USA2008.txt
Alabama,1266546,813479,19773,
Alaska,193841,123594,8762,
Arizona,1230111,1034707,39020,
Arkansas,638017,422310,26290,
California,5011781,8274473,289260,
Colorado,1073584,1288568,39197,
Connecticut,629428,997772,19592,
Delaware,152374,255459,4579,
District of Columbia,17367,245800,2686,
...
New Jersey,1613207,2215422,47826,
...
Virginia,1725005,1959532,38723,
Washington,1229216,1750848,68820,
West Virginia,398061,304127,12550,
Wisconsin,1262393,1677211,43813,
Wyoming,164958,82868,6832,
```

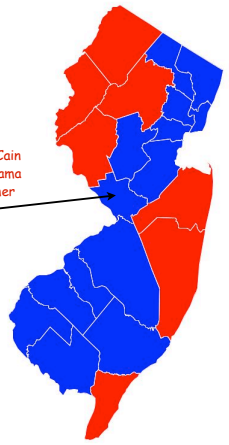


1,613,207 McCain
2,215,422 Obama
47,826 Other

Election Return Data: By County

Screen-scraping results. Votes for McCain, Obama, Other by region.

```
% more NJ2008.txt
Atlantic,49902,67830,1517,
Bergen,186118,225367,4424,
Burlington,89626,131219,2930,
Camden,73819,159259,3304,
Cape May,27288,22893,802,
Cumberland,22360,34919,915,
Essex,74063,240306,2181,
Gloucester,60315,77267,1848,
Hudson,55360,154140,2116,
Hunterdon,39092,29776,1147,
Mercer,50223,107926,2229,
Middlesex,123695,193812,4283,
Monmouth,160433,148737,4244,
Morris,132331,112275,2913,
Ocean,160677,110189,4111,
Passaic,72552,113257,1904,
Salem,14816,16044,672,
Somerset,70085,79321,1672,
Sussex,44184,28840,1393,
Union,78768,141417,2241,
Warren,27500,20628,980,
```



Boundary Data: States within the Continental US

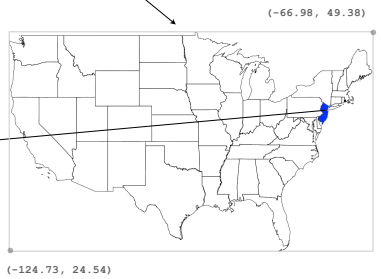
USA data file. State names and boundary points.

Data source: US census bureau, www.census.gov/tiger/boundary.

```
% more USA.txt

-124.731216 24.544102 -66.980385 49.384365

104

Alabama
USA
498
 -88.200027   34.995548
 -88.202919   35.007942
...
New Jersey
USA
368
 -74.695305   41.357330
 -74.461754   41.250000
 -74.366302   41.202801
...
 -74.721313   41.347294
...
```

bounding box

368 points (latitude, longitude)



Boundary Data: Counties within a State

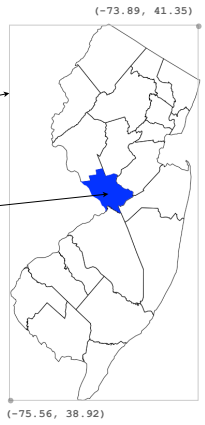
State data files. County names and boundary points.

Data source: US census bureau, www.census.gov/tiger/boundary.

```
% more NJ.txt

-75.560143 38.928589 -73.894402 41.35733

21   ← number of subregions

Atlantic
NJ
127
 -74.877563   39.608414
 -74.736694   39.729721
...
Mercer
NJ
88
 -74.748825   40.424248
 -74.722702   40.375301
 -74.674507   40.384399
...
 -74.808403   40.415401
...
```

bounding box

88 points (latitude, longitude)



Summary: Data Sources

(13 + 1)*(50 + 1) = 714 Data files

- each file represents a "whole" region divided into "parts"
- one entry per "part"

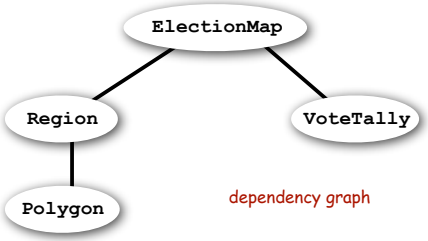
whole	part	files	type of data
USA	state	USA.txt	boundary
		USA2008.txt USA2004.txt ... USA1960.txt	election return
state	county	NJ.txt	boundary
		NJ2008.txt NJ2004.txt ... NJ1960.txt	election return
		[similar files for all 50 states]	

Modular Programming with Data Types

Challenge. Visualize election results.

Approach.

- Gather data from web sources, save in local files.
- Build modular program that reads files, draws map.
- Each module is an immutable data type.

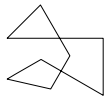


dependency graph

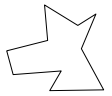
Polygon. Geometric primitve.
Region. State or county.
Vote Tally. Number of votes for each candidate.
Election Map. The map of "parts" for a given "whole" region in a given year.

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)

Set of values. Sequence of N boundary points
Operations.

- read from input stream
- draw (filled with the current pen color)
- [perimeter, area, many other useful operations might be included]

see COS 226

Design issue. Implement general data type or one just for this problem ?

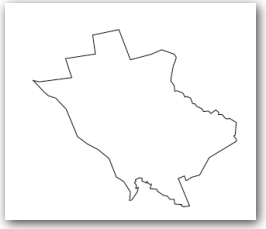
Polygon Data Type Implementation

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

   public Polygon(In in)
   {  // Read from input stream.
      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); }
}
```


Region Data Type

Region. State or county.
Set of values. Polygon

Ex.



Operations.

- create
- draw (filled with the current pen color)

Region Data Type Implementation

```
public class Region
{
   private final Polygon poly;  // polygonal boundary

   public Region(Polygon poly)
   {
      this.poly = poly;
   }

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

}
```


Vote Tally Data Type

Vote Tally. Election returns for one region

Set of values. # votes for republican, democrat, other

Ex.

```
50223   107926   2229
```

2008 returns for Mercer county
50,223 McCain
107,926 Obama
2,229 Other

Operations.

- create (whole, part, year)
- return a color representation of the vote

all needed to locate the data!

```
% more NJ2008.txt
...
Mercer,50223,107926,2229,
...
```

blue
when democrat beats republican

Vote Tally Data Type Implementation

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

   public VoteTally(String part, String whole, int year)
   {
      // Read and parse election return data file.
      In in = new In(whole + year + ".txt");
      String input = in.readAll();
      int i0 = input.indexOf(part);
      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.GREEN;
   }
}
```

```
% more NJ2008.txt
...
Mercer,50223,107926,2229,
...
i0   i1   i2   i3   i4
```

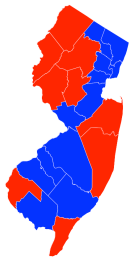

Election Map Data Type

`ElectionMap`. The map of "parts" for a given "whole" region in a given year.

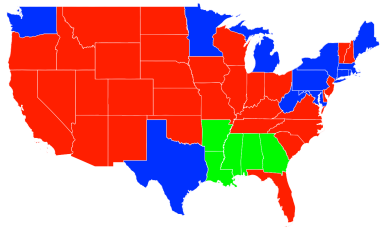
Client:

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

```
% java ElectionMap NJ 2004
```



```
% java ElectionMap USA 1968
```



Election Map Data Type 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");  // boundary data file
      // Read in bounding box and rescale coordinates (omitted).
      N       = in.readInt();
      regions = new Region[N];
      votes   = new VoteTally[N];
      for (int i = 0; i < N; i++)
      {
         String  part = in.readLine();
         String  whole = in.readLine(); // redundant data
         Polygon poly = new Polygon(in);
         regions[i]   = new Region(poly);
         votes[i]     = new VoteTally(part, whole, year);
      }
   }

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

instance variables

use Polygon, Region, and VoteTally to build map

draw map

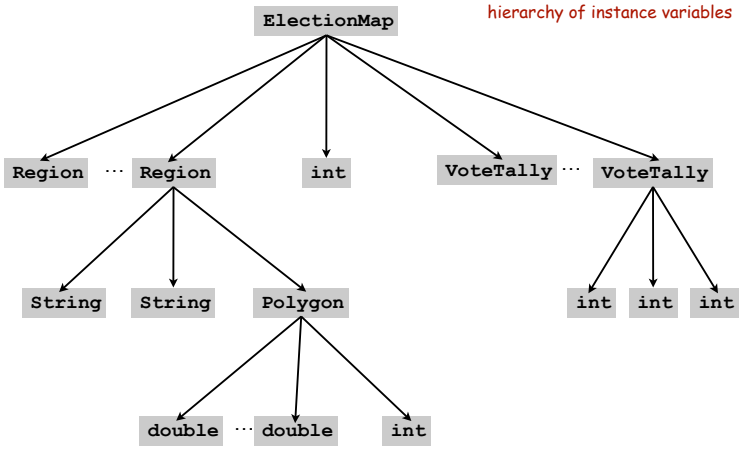
```
% more NJ.txt
...
Mercer
NJ
88
 -74.748825   40.424248
 -74.722702   40.375301
 -74.674507   40.384399
...
 -74.808403   40.415401
...
```


TEQ on Data Type Design 3

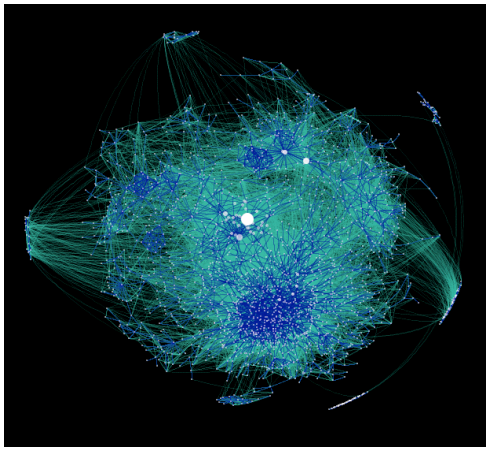
Q. Is `ElectionMap` immutable?

Modular Programming

Modular program: Collection of data types.

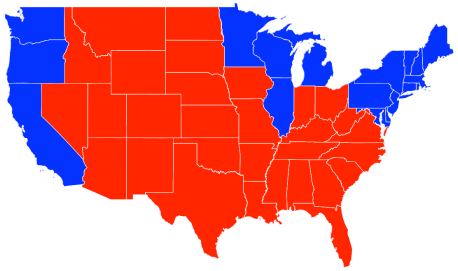


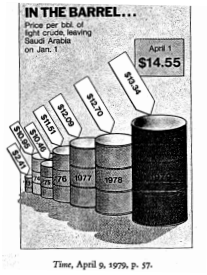
Data Visualization



Visual Display of Quantitative Information

Red states, blue states. Nice example, but a misleading and polarizing picture.





Time, April 9, 1979, p. 57.

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

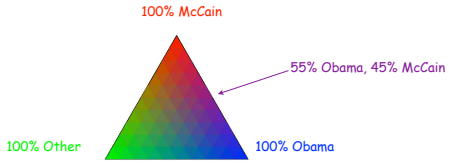
Purple America

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

- a_1 = McCain votes.
- a_2 = Other votes.
- a_3 = Obama votes.

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

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



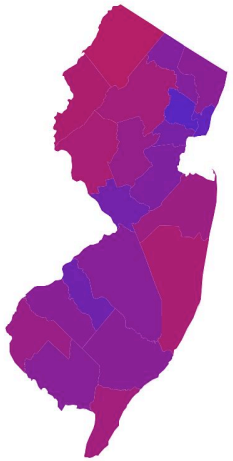
Implementation: change one method!

VoteTally.java

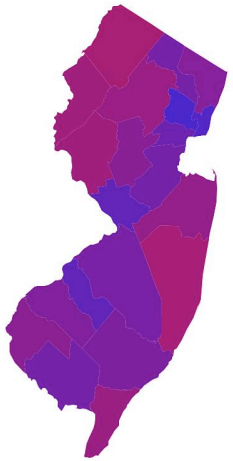
```
public Color getColor( )
{
   int tot = dem + rep + ind;
   return new Color((float) rep/tot, (float) ind/tot, (float) dem/tot);
}
```


Purple New Jersey

```
% java ElectionMap NJ 2004
```

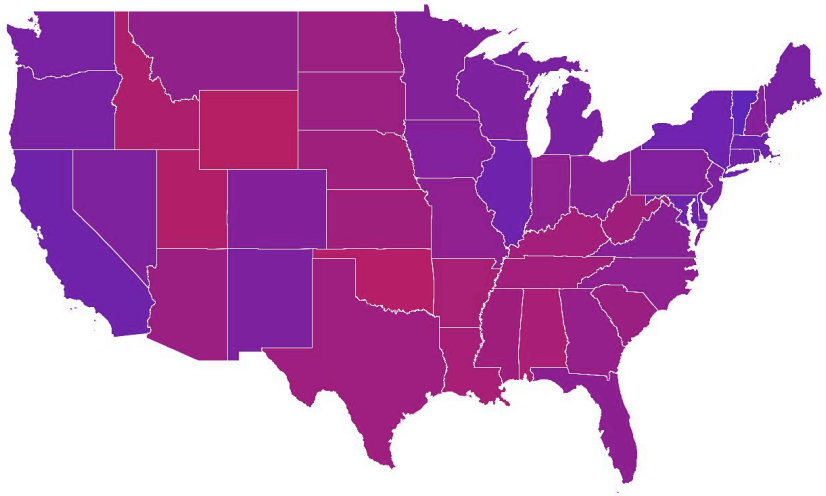


```
% java ElectionMap NJ 2008
```



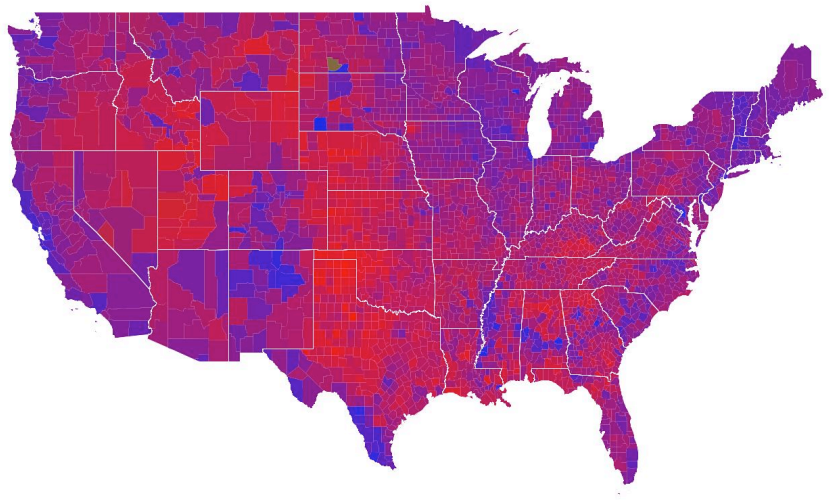
Purple America

```
% java ElectionMap USA 2008
```



Purple America

```
% java ElectionMap USA-county 2008
```

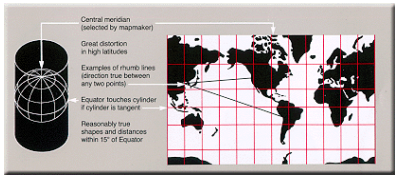


Data Visualization: Design Issues

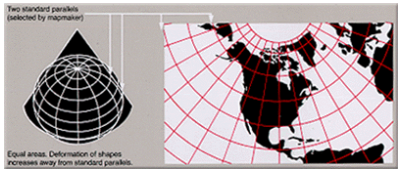
Remark. Humans perceive red more strongly than blue.

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

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



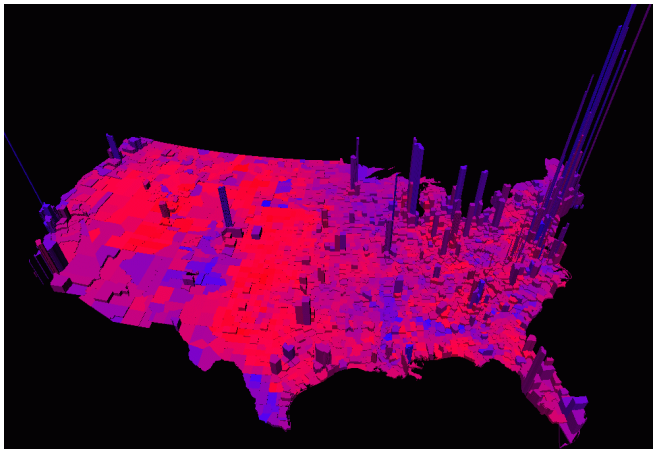
Mercator projection



Albers projection

3D Visualization

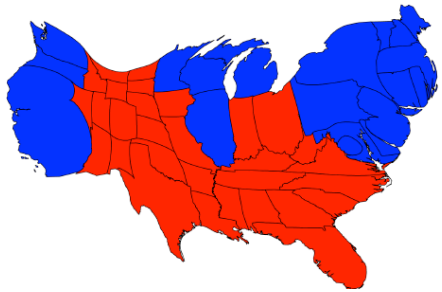
3D visualization. Volume proportional to votes; azimuthal projection.



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

Cartograms

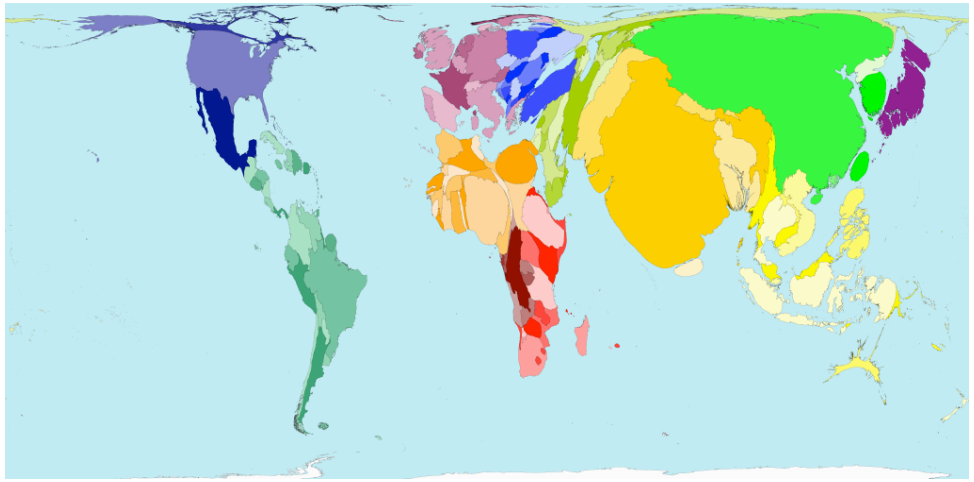
Cartogram. Area of state proportional to number of electoral votes.



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

Cartograms

Cartogram. Area of country proportional to population.



Summary

Modular programming.

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

Ex 1. Build large software project.

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

Ex 2. Build reusable libraries.

- Language designer extends language with ADTs.
- Programmers share extensive libraries.
- Ex: `In`, `Out`, `Draw`, `Polygon`, …

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