3.4 Encapsulation and ADTs

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.

Abstract Data Types

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

Abstract data type. Data type whose internal representation is hidden.

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
- 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.

Intuition

Client

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

Implementation
- gas plasma monitor
- Pioneer PDP-502MX
- 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 {
   int count;

   public Counter()  { count = 0;     }
   public void hit() { count++;       }
   public int get()  { return count; }

}
```

Legal Java client.

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

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

Counter ADT

Counter. Abstract data type to count electronic votes.

```
public class Counter {
   private int count;

   public Counter()  { count = 0;     }
   public void hit() { count++;       }
   public int get()  { return count; }

}
```

Does not compile.

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

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

Changing Internal Representation

Java ADTs.

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

```
public class Complex {
   private double re;
   private double 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 coordiantes

Advantage. Can switch internal representation without changing client.

Note. All our data types are already ADTs!

Time Bombs

Representation changes.

- [Y2K] Two digit years: January 1, 2000.
- [Y2038] 32-bit seconds since 1970: January 19, 2038.
- [ZIP codes] USPS changed from ZIP to ZIP + 4 code in 1983.
- [VIN numbers] Will run out by 2010 ⇒ representation change ahead!

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

Ask, Don't Touch

Encapsulation.
- Can'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"

3.3 Modular Programming

Review

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

A Java class allows us to define a data type by:
- Specifying a set of values (instance variables).
- Defining operations on those values (methods).

Modular programming. Break up a larger program into smaller, independent pieces.
- Class = program that defines a data type.
- Client = program that uses a data type.

Procedural vs. 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.

Alan Kay

Alan Kay. [Xerox PARC 1970s]

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

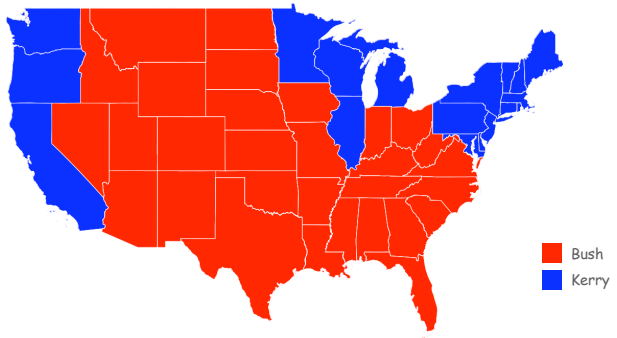
"The computer revolution hasn't started yet."

"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

Red States, Blue States



2004 Presidential election

Data Abstraction

Data abstraction. Model problem by decomposing into components.

Polygon. Geometric primitive.
Region. Name, postal abbreviation, polygonal boundary.
Vote tally. Number of votes for each candidate.
Election map. Set of regions and corresponding vote tallies for a given election.

Geographic Boundaries

Boundary Data: States within the USA

USA data file. State names and boundary points.

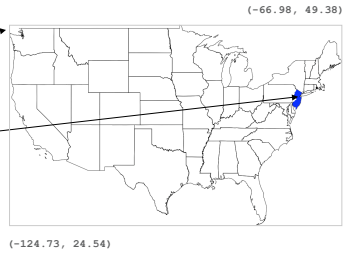
Data source: US census bureau, www.census.gov/tiger/boundary.

```
% 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
…
```

number of regions

bounding box

368 points (latitude, longitude)



Boundary Data: Counties within a State

State data files. County names and boundary points.

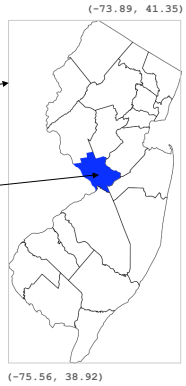
Data source: US census bureau, www.census.gov/tiger/boundary.

```
% more NJ.txt
-75.560143 38.928589 -73.894402 41.35733
21
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
…
```

number of subregions

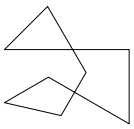
bounding box

88 points (latitude, longitude)

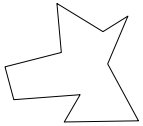


Polygon ADT

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



polygon (8 points)



simple polygon (10 points)

simple polygon (368 points)

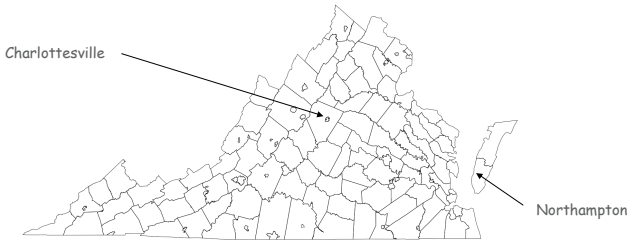
Polygon ADT

```
public class Polygon {
   private int N;           // number of boundary points
   private 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 double perimeter()                    { … }
   public boolean contains(double x0, double y0) { … }
   public String toString()                     { … }
}
```

Polygon: Pieces and Holes

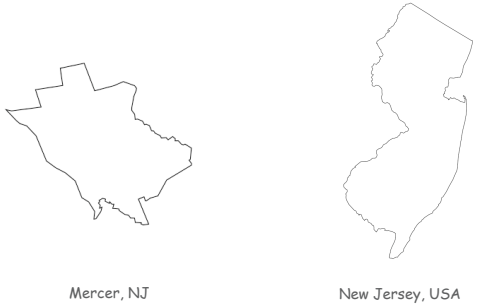
Pieces. A state can be comprised of several disjoint polygons.

Holes. A county can be entirely inside another county.



Region Data Type

Region. Represents a state or county.



Region Data Type: Java Implementation

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

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

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

   public boolean contains(double x, double y) {
      return poly.contains(x, y);
   }

   public String toString() { … }

}
```

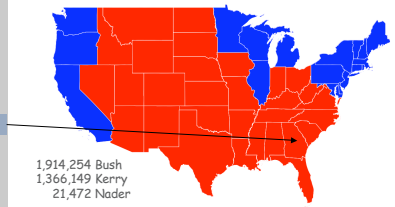

Election Returns

Election Returns: States within the USA

Election returns. Number of votes for Bush, Kerry, Nader by region.

Data source: David Leip, www.uselectionatlas.org.

```
% 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,
```

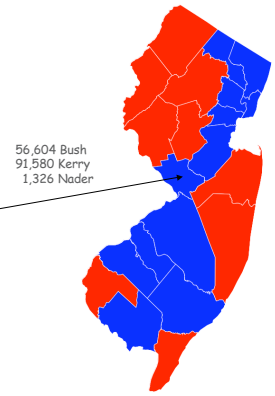


Election Returns: Counties within a State

Election returns. Number of votes for Bush, Kerry, Nader by region.

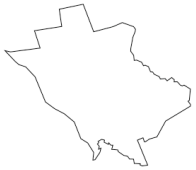
Data source: David Leip, www.uselectionatlas.org.

```
% more NJ2004.txt
Atlantic,49487,55746,864,
Bergen,189833,207666,2745,
Burlington,95936,110411,1609,
Camden,81427,137765,1741,
Cape May,28832,21475,455,
Cumberland,24362,27875,948,
Essex,83374,203681,2293,
Gloucester,60033,66835,1096,
Hudson,60646,127447,1353,
Hunterdon,39888,26050,742,
Mercer,56604,91580,1326,
Middlesex,126492,166628,2685,
Monmouth,163650,133773,2516,
Morris,135241,98066,1847,
Ocean,154204,99839,2263,
Passaic,75200,94962,1149,
Salem,15721,13749,311,
Somerset,72508,66476,1295,
Sussex,44506,23990,900,
Union,82517,119372,1498,
Warren,29542,18044,622,
```



Vote Tally Data Type

VoteTally. Represents the election returns for one region.



Mercer, NJ
56,604 Bush
91,580 Kerry
1,326 Nader

New Jersey, USA
1,670,003 Bush
1,911,430 Kerry
30,258 Nader

Vote Tally Data Type: Java Implementation

```
public class VoteTally {
   private 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;
   }
}
```

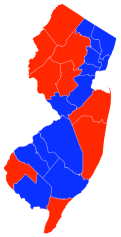
```
% more NJ2004.txt
…
Mercer,56604,91580,1326,
…
i0   i1   i2   i3   i4
```

ElectionMap Data Type

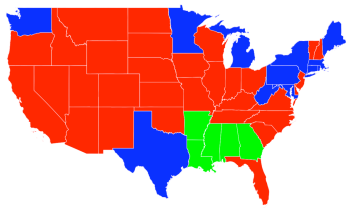
ElectionMap. Represents the election map for a given election.

```
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
```



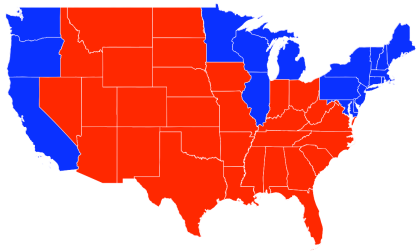
Election Map Data Type: Java Implementation

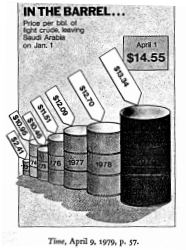
```
public class ElectionMap {
   private int N;
   private Region[] regions;
   private VoteTally[] votes;

   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].fill();
      }
   }
}
```


Data Visualization

Visual Display of Quantitative Information

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





Time, April 9, 1979, p. 57.

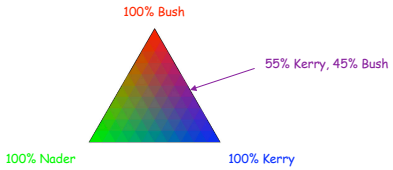
Edward Tufte. Create charts with high data density and that tell the truth.

Purple

Goal. Assign color based on number of votes for each candidate.

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

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

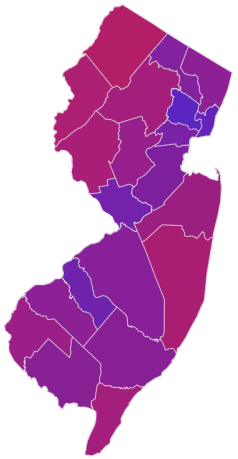


VoteTally.java

```
public Color getColor() {
   int total = dem + rep + ind;
   return new Color(1.0f*rep/total, 1.0f*ind/total, 1.0f*dem/total);
}
```

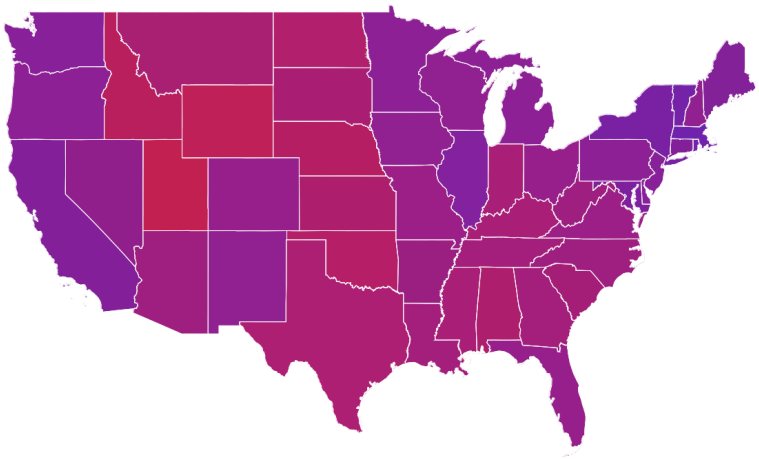

Purple New Jersey

```
% java ElectionMap NJ 2004
```



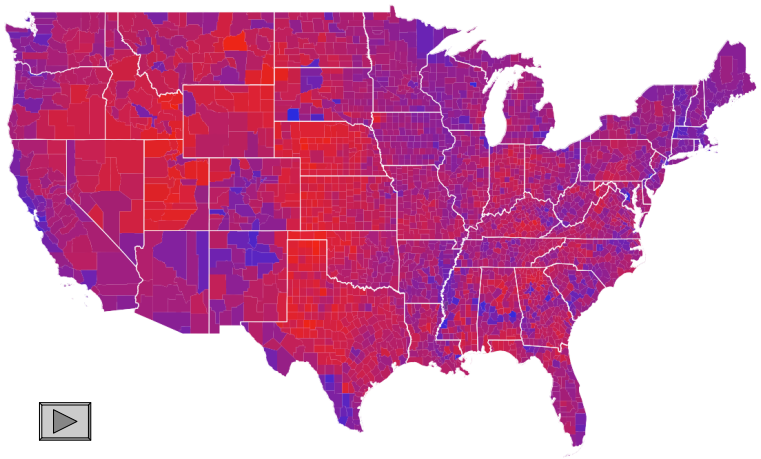
Purple America

```
% java ElectionMap USA 2004
```



Purple America

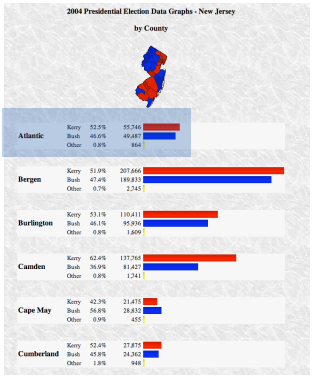
```
% java ElectionMap USA-county 2004
```



Screen Scraping the Election Returns

Screen scrape. Data available on Web; download html and parse.

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




```
<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 …
```

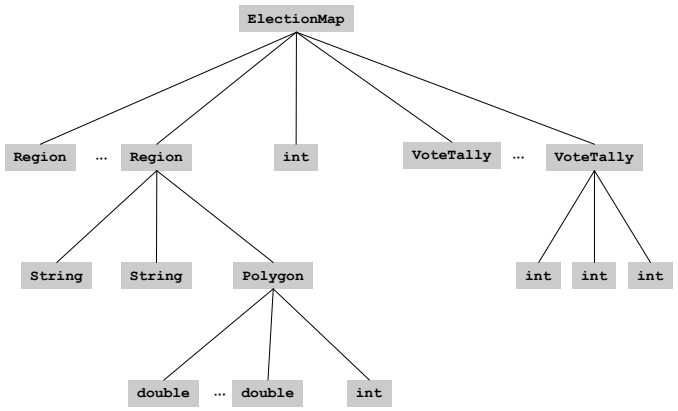

Election Scraper (sketch)

```
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) {
  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);

  // extract vote total for each candidate

  file.println(county + "," + bush + "," + kerry + "," + nader + ",");
}
```

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

Layers of Abstraction

Relationships among data types.



Summary

Modular programming.

- Break a large program into smaller independent modules.
- Ex: `Polygon`, `Region`, `VoteTally`, `ElectionMap`, `In`, `Out`, `Draw`.

Debug and test each piece independently. [unit testing]

- Each class can have its own `main()`.
- Spend less overall time debugging.

Ex: building large software project.

- Software architect specifies API.
- Each programmer implements one module.

Ex: build reusable libraries.

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