3.3 Designing Data Types
Object Oriented Programming

Procedural programming. [verb-oriented]
- Tell the computer to do this.
- Tell the computer to do that.

OOP 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 objects that are part of the problem domain or solution.
- Identity: objects are distinguished from other objects (references).
- State: objects in the world know things (instance variables).
- Behavior: objects 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 per cent of the time, you're not aiming high enough.”

— Alan Kay

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, ...

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.

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

Client

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

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

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

client needs to know how to use API

implementation needs to know what API to implement
Intuition

Client

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

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

Can substitute better implementation without changing the client.

Client needs to know how to use API

Implementation needs to know what API to implement
Counter Data Type

Counter. Data type to count electronic votes.

```java
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. Encapsulated data type to count electronic votes.

```java
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;
    }
}
```

Counter Data Type

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

Does not compile.

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

Encapsulation.

- Keep data representation hidden with \texttt{private} access modifier.
- Expose API to clients using \texttt{public} access modifier.

```java
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() { ... }
}
```

\textit{e.g., to polar coordinates}

\textbf{Advantage.} Can switch internal representation without changing client.

\textbf{Note.} All our data types are already encapsulated!
Time Bombs

Internal representation changes.

- [VIN numbers] We'll run out by 2010.

Lesson. By exposing data representation to client, might need to sift through millions of lines of code in client to update.
Encapsulated data types.
- Don't touch data and do whatever you want.
- Instead, ask object to manipulate its data.

"Ask, don't touch."

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

"principle of least privilege"
Immutability
Immutability

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

<table>
<thead>
<tr>
<th>mutable</th>
<th>immutable</th>
</tr>
</thead>
<tbody>
<tr>
<td>Picture</td>
<td>Charge</td>
</tr>
<tr>
<td>Histogram</td>
<td>Color</td>
</tr>
<tr>
<td>Turtle</td>
<td>Stopwatch</td>
</tr>
<tr>
<td>StockAccount</td>
<td>Complex</td>
</tr>
<tr>
<td>Counter</td>
<td>String</td>
</tr>
<tr>
<td>Java arrays</td>
<td>primitive types</td>
</tr>
</tbody>
</table>
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. Declaring an instance variable to be `final` means that you can assign it a value only once, in initializer or constructor.

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

Advantages.
- Helps enforce immutability.
- Prevents accidental changes.
- Makes program easier to debug.
- Documents that the value cannot change.
Spatial Vectors
**Vector Data Type**

**Set of values. Sequence of real numbers.** [Cartesian coordinates]

<table>
<thead>
<tr>
<th><strong>API.</strong></th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>public class Vector</strong></td>
</tr>
</tbody>
</table>
| Vector(double[] a) | create a vector with the given Cartesian coordinates 
| Vector plus(Vector b) | sum of this vector and b 
| Vector minus(Vector b) | difference of this vector and b 
| Vector times(double t) | scalar product of this vector and t 
| double dot(Vector b) | dot product of this vector and b 
| double magnitude() | magnitude of this vector 
| Vector direction() | unit vector with same direction as this vector |

\[
x = (0, 3, 4, 0), \quad y = (0, -3, 1, -4) \\
x + y = (0, 0, 5, -4) \\
3x = (0, 9, 12, 0) \\
x \cdot y = (0 \cdot 0) + (3 \cdot -3) + (14 \cdot 1) + (0 \cdot -4) = -5 \\
|x| = (0^2 + 3^2 + 4^2 + 0^2)^{1/2} = 5 \\
\vec{x} = x / |x| = (0, 0.6, 0.8, 0)
\]
Vector Data Type Applications

Relevance. A quintessential mathematical abstraction.

Applications.
- Statistics.
- Linear algebra.
- Clustering and similarity search.
- Force, velocity, acceleration, momentum, torque.
- ...

public class Vector {
    private int N;
    private double[] coords;

    public Vector(double[] a) {
        N = a.length;
        coords = new double[N];
        for (int i = 0; i < N; i++)
            coords[i] = a[i];
    }

    public double dot(Vector b) {
        double sum = 0.0;
        for (int i = 0; i < N; i++)
            sum += (coords[i] * b.coords[i]);
        return sum;
    }

    public Vector plus(Vector b) {
        double[] c = new double[N];
        for (int i = 0; i < N; i++)
            c[i] = coords[i] + b.coords[i];
        return new Vector(c);
    }
}
This. The keyword `this` is a reference to the invoking object.

Ex. When you invoke `a.magnitude()`, `this` is an alias for `a`. 
3.5 Case Study: Purple America
Challenge. Visualize election results.

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

2008 Presidential election
Data Visualization

Approach.
- Gather data from data sources on the web; save in local files.
- Build a modular program that reads files and draws maps.

McCain
Obama
2008 Presidential election

2008 Presidential election

McCain
Obama

Image of a map showing the 2008 Presidential election results.
Data Sources

If I'd known they wanted me to use all this info—"I would never have asked for it!"
Locate the Data Sources

**Geometric data.**
- www.census.gov/tiger/boundary
- Text files have boundaries of every state and county.  
  (format useful for programmers)

**Election returns.**
- www.uselectionatlas.org
  - Web site displays election results.  
  (need to screen scrape to extract raw data)

**Emerging standard.**
- Publish data in text format on the web (like geometric data).
- Write **mashup** program to produce visuals (like we’re doing)!
Geometric Data: States within the Continental US

USA data file. State names and boundary points.

% more USA.txt
-124.731216 24.544102  -66.980385 49.384365
104
Alabama
498
-88.200027   34.995548
-88.202919   35.007942
...
New Jersey
368
-74.695305   41.357330
-74.461754   41.250000
-74.366302   41.202801
...
-74.721313   41.347294
...

number of regions
bounding box
368 points
(longitude, latitude)
Geometric Data: Counties within a State

State data files. County names and boundary points.

% more NJ.txt
-75.560143 38.928589 -73.894402 41.357333
21
Atlantic
127
-74.877563 39.608414
-74.736694 39.729721
...

Mercer
88
-74.748825 40.424248
-74.722702 40.375301
-74.674507 40.384399
...
-74.808403 40.415401

88 points (longitude, latitude)

bounding box
number of regions
region
Screen Scraping the Election Returns

Screen scraping. Download html from web and parse.


NJ = FIPS 34
Screen Scraping the Election Returns (Java sketch)

```java
int year     = 2008;  // election year
String whole = "NJ";  // region name 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(whole + year + ".txt");
String input = in.readAll();

while (true) {
    // screen scrape region 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 region = input.substring(from + 3, to);

    // screen scrape vote totals for each candidate

    // save results to file
    file.println(region + "," + mccain + "," + obama + "," + other + ",");
}
```
Election Returns: By County

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

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

<table>
<thead>
<tr>
<th>State</th>
<th>McCain</th>
<th>Obama</th>
<th>Other</th>
</tr>
</thead>
<tbody>
<tr>
<td>Alabama</td>
<td>1,266,546</td>
<td>813,479</td>
<td>19,773</td>
</tr>
<tr>
<td>Alaska</td>
<td>193,841</td>
<td>123,594</td>
<td>8,762</td>
</tr>
<tr>
<td>Arizona</td>
<td>123,011</td>
<td>1,034,707</td>
<td>39,020</td>
</tr>
<tr>
<td>Arkansas</td>
<td>638,017</td>
<td>422,310</td>
<td>26,290</td>
</tr>
<tr>
<td>California</td>
<td>5,011,781</td>
<td>827,447</td>
<td>28,926</td>
</tr>
<tr>
<td>Colorado</td>
<td>1,073,584</td>
<td>1,288,568</td>
<td>39,197</td>
</tr>
<tr>
<td>Connecticut</td>
<td>629,428</td>
<td>997,772</td>
<td>19,592</td>
</tr>
<tr>
<td>Delaware</td>
<td>152,374</td>
<td>255,459</td>
<td>4,579</td>
</tr>
<tr>
<td>District of Columbia</td>
<td>1,736,7</td>
<td>2,458,002</td>
<td>2686</td>
</tr>
<tr>
<td>Florida</td>
<td>4,045,624</td>
<td>4,282,074</td>
<td>82,621</td>
</tr>
<tr>
<td>Georgia</td>
<td>2,048,744</td>
<td>1,844,137</td>
<td>39,222</td>
</tr>
<tr>
<td>Hawaii</td>
<td>1,205,66</td>
<td>3,258,71</td>
<td>7,131</td>
</tr>
<tr>
<td>Idaho</td>
<td>403,012</td>
<td>2,364</td>
<td>400,797</td>
</tr>
<tr>
<td>Illinois</td>
<td>2,031,527</td>
<td>3,419,673</td>
<td>7,1851</td>
</tr>
<tr>
<td>...</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Virginia</td>
<td>1,725,005</td>
<td>1,959,532</td>
<td>38,723</td>
</tr>
<tr>
<td>Washington</td>
<td>1,229,216</td>
<td>1,750,848</td>
<td>68,820</td>
</tr>
<tr>
<td>West Virginia</td>
<td>3,980,61</td>
<td>3,041,27</td>
<td>12,550</td>
</tr>
<tr>
<td>Wisconsin</td>
<td>1,262,393</td>
<td>1,677,211</td>
<td>4,3813</td>
</tr>
<tr>
<td>Wyoming</td>
<td>1,649,58</td>
<td>828,68</td>
<td>68,32</td>
</tr>
</tbody>
</table>
Real Data are Messy

Different data sources have different conventions.
- State names: NJ vs. New Jersey vs. FIPS 34.
- County names: LaSalle vs. La Salle, Kings County vs. Brooklyn.

Other annoyances.
- A state can be comprised of several disjoint polygons.
- A county can be entirely inside another county.
- County boundaries change over time.
- Write-in candidates.
- Unreported results.
- Alaska and Hawaii.

Bottom line. Must clean the data (but write a program to do most of it!)
Summary of Data Files

714 data files.  \[ (13 + 1) \times (50 + 1) \]

- Each file represents a "whole" divided into regions.
- One entry per region.

<table>
<thead>
<tr>
<th>whole</th>
<th>part</th>
<th>files</th>
<th>type of data</th>
</tr>
</thead>
<tbody>
<tr>
<td>USA</td>
<td>state</td>
<td>USA.txt</td>
<td>boundary</td>
</tr>
<tr>
<td></td>
<td></td>
<td>USA2008.txt</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>USA2004.txt</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>...</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>USA1960.txt</td>
<td></td>
</tr>
<tr>
<td>state</td>
<td>county</td>
<td>NJ.txt</td>
<td>boundary</td>
</tr>
<tr>
<td></td>
<td></td>
<td>NJ2008.txt</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>NJ2004.txt</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>...</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>NJ1960.txt</td>
<td></td>
</tr>
</tbody>
</table>

[similar files for all 50 states]
Modular Programming
Modular Programming

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

Region. State or county.
Vote tally. Number of votes for each candidate in a region.
Election map. Map of votes by region in a given election.

dependency graph
Region Data Type

Region. A state or county.

Set of values. Sequence of boundary points, name.

Operations. Create and draw.
public class Region {
    private final String name;    // name of region
    private final int N;          // number of boundary points
    private final double[] x, y;  // the points (x[i], y[i])

    public Region(String name, double[] x, double[] y) {
        this.name = name;
        this.N = x.length;
        this.x = new double[N];
        this.y = new double[N];
        for (int i = 0; i < N; i++) {
            this.x[i] = x[i];
            this.y[i] = y[i];
        }
    }

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

    public String name() {
        return name;
    }
}
Vote Tally Data Type

Vote tally. Election returns for one region.

Set of values. Number of votes for each candidate.

Operations.
- Create (whole, region, year).
- Number of votes for Republican, Democrat, and Independent candidates.

Mercer, NJ

McCain: 50,223
Obama: 107,926
Other: 2,229

needed to locate the data
public class VoteTally {
    private final int rep, dem, ind;

    public VoteTally(String region, String whole, int year) {
        In in = new In(whole + year + ".txt");
        String input = in.readAll();
        int i0 = input.indexOf(region);
        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 int rep() { return rep; }
    public int dem() { return dem; }
    public int ind() { return ind; }
}
Election Map Data Type

Election map. Map of votes by region in a given election.

```java
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 USA 1968
% java ElectionMap NJ 2008
public class ElectionMap {
    private final int REIGONS;    // number of regions
    private final Region[] regions;    // regions[j] = jth region
    private final VoteTally[] votes;    // votes[j] = jth vote tallies

    public ElectionMap(String whole, int year) {
        // see next slide
    }

    private Color getColor(VoteTally tally) {
        if (tally.rep() > tally.dem()) return StdDraw.RED;
        else if (tally.dem() > tally.rep()) return StdDraw.BLUE;
        else return StdDraw.BLACK;
    }

    public void show() {
        for (int j = 0; j < REIGONS; j++) {
            StdDraw.setPenColor(getColor(votes[j]));
            regions[j].draw();
        }
    }
}
public ElectionMap(String whole, int year) {
    In in = new In(whole + ".txt");
    // read in bounding box and rescale coordinates
    REGIONS = in.readInt();
    regions = new Region[REGIONS];
    votes = new VoteTally[REGIONS];
    for (int j = 0; j < REGIONS; j++) {
        String region = in.readLine();
        int N = in.readInt();
        double[] x = new double[N];
        double[] y = new double[N];
        for (int i = 0; i < N; i++) {
            x[i] = in.readDouble();
            y[i] = in.readDouble();
        }
        regions[j] = new Region(part, x, y);
        votes[j] = new VoteTally(region, whole, year);
    }
}
Modular Programming

**Modular program.** Collection of interacting data types.

```
Modular Programming

Modular program. Collection of interacting data types.

ElectionMap
  Region ... Region int
    String double[] double[] int
    ... VoteTally ... VoteTally
      int int int
```

hierarchy of instance variables
Data Visualization
Visual Display of Quantitative Information

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

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.

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

Implementation. Change only one method in ElectionMap.java.

```java
public Color getColor() {
    int dem = tally.dem(), rep = tally.rep(), ind = tally.ind();
    int tot = tally.dem + tally.rep + tally.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.
3D Visualization

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

Robert J. Vanderbei
www.princeton.edu/~rvdb/JAVA/election2004
Election 2004: The Final Tally

How Much Each County Counted

Popular Vote, By County

Popular Vote, By Population

This map, which uses the size of circles to indicate a winning margin, shows how Senator Kerry had huge margins in many counties with big cities, while President Bush had relatively similar but consistent margins in suburban and rural counties.

Largest vote margin for Kerry: Cook County, Ill. (+805,857 votes)

Largest vote margin for Bush: Orange County, Calif. (+155,010 votes)

Second largest vote margin for Bush: Tarrant County, Tex. (+113,163 votes)

Eugene Pyatigorsky (based on New York Times visualization)

www.ocf.berkeley.edu/~gene/media/docs/Election.pdf
Cartograms

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

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

Modular programming.
- Break a large program into smaller independent components.
- Develop a data type for each component.
- Ex: 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.

Data visualization. You can do it! [worthwhile to learn from Tufte]