3.3 Designing Data Types

Alan Kay

• 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

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

Data type. Set of values and operations on those values. Ex: \texttt{int}, \texttt{String}, \texttt{Complex}, \texttt{Vector}, \texttt{Document}, \texttt{GuitarString}, \texttt{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.

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 needs to agree on API ahead of time.

Can substitute better implementation without changing the client.

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
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;
    }
}
```

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

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

Counter Data Type

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

Legal Java client.

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 `private` access modifier.
-Expose API to client code using `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() { ... }
}
```

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

www.cartoonstock.com/directory/e/millenium_time-bomb.asp

**Lesson.** By exposing data representation to client, 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."

Adele Goldberg
Former president of ACM
Co-developed Smalltalk

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

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 Access Modifier

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

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

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

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

Q. Is the following data type immutable?

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

---

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**

**Geometric data**
- [www.census.gov/tiger/boundary](http://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

**Election results**
- [http://uselectionatlas.org/RESULTS](http://uselectionatlas.org/RESULTS)
  - interactive and graphical
  - need to screen-scrape to get data

**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.

```html
div
  dr />do-2004 Presidential Election Data Graphs - New Jersey
  dr /dr
  div src="img.png?year=2004&amp;amp;state=NJ&amp;amp;year=2004"
  div
    dr
      tr td class="text" width="100px"
    div
  div


  Screen scraping the Election Returns

  county name is text between <b> and </b> tags, that occurs after width:100px
```
Election Scraper (sketch)

```java
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);
}
```

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.

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

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

Screen-scraping 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>1,234</td>
<td>1,345</td>
<td>234</td>
</tr>
<tr>
<td>Alaska</td>
<td>1,234</td>
<td>1,345</td>
<td>234</td>
</tr>
<tr>
<td>Arizona</td>
<td>1,234</td>
<td>1,345</td>
<td>234</td>
</tr>
<tr>
<td>California</td>
<td>1,234</td>
<td>1,345</td>
<td>234</td>
</tr>
<tr>
<td>Colorado</td>
<td>1,234</td>
<td>1,345</td>
<td>234</td>
</tr>
<tr>
<td>Connecticut</td>
<td>1,234</td>
<td>1,345</td>
<td>234</td>
</tr>
<tr>
<td>Delaware</td>
<td>1,234</td>
<td>1,345</td>
<td>234</td>
</tr>
<tr>
<td>District of Columbia</td>
<td>1,234</td>
<td>1,345</td>
<td>234</td>
</tr>
<tr>
<td>New Jersey</td>
<td>1,234</td>
<td>1,345</td>
<td>234</td>
</tr>
<tr>
<td>...</td>
<td>...</td>
<td>...</td>
<td>...</td>
</tr>
</tbody>
</table>

Pitfalls: Pieces and Holes

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

Holes. A county can be entirely inside another county.

Unreported results, write-ins, changes in county boundaries,...
### Election Return Data: By County

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

<table>
<thead>
<tr>
<th>County</th>
<th>McCain Votes</th>
<th>Obama Votes</th>
<th>Other Votes</th>
</tr>
</thead>
<tbody>
<tr>
<td>Atlantic</td>
<td>49902</td>
<td>67830</td>
<td>1517</td>
</tr>
<tr>
<td>Bergen</td>
<td>18618</td>
<td>25387</td>
<td>4424</td>
</tr>
<tr>
<td>Burlington</td>
<td>89626</td>
<td>131290</td>
<td>2390</td>
</tr>
<tr>
<td>Camden</td>
<td>73819</td>
<td>95259</td>
<td>3304</td>
</tr>
<tr>
<td>Cape May</td>
<td>27388</td>
<td>28935</td>
<td>802</td>
</tr>
<tr>
<td>Essex</td>
<td>74063</td>
<td>24056</td>
<td>2311</td>
</tr>
<tr>
<td>Gloucester</td>
<td>60115</td>
<td>77267</td>
<td>1848</td>
</tr>
<tr>
<td>Hudson</td>
<td>59360</td>
<td>194140</td>
<td>2116</td>
</tr>
<tr>
<td>Hunterdon</td>
<td>39692</td>
<td>29776</td>
<td>1147</td>
</tr>
<tr>
<td>Mercer</td>
<td>50223</td>
<td>107926</td>
<td>2239</td>
</tr>
<tr>
<td>Middlesex</td>
<td>-75.8912</td>
<td>-38.9875</td>
<td>-74.8418</td>
</tr>
<tr>
<td>Monmouth</td>
<td>140433</td>
<td>148737</td>
<td>4244</td>
</tr>
<tr>
<td>Morris</td>
<td>132333</td>
<td>112275</td>
<td>2913</td>
</tr>
<tr>
<td>Ocean</td>
<td>160677</td>
<td>110189</td>
<td>4111</td>
</tr>
<tr>
<td>Passaic</td>
<td>72552</td>
<td>112927</td>
<td>1904</td>
</tr>
<tr>
<td>Salem</td>
<td>18814</td>
<td>16044</td>
<td>672</td>
</tr>
<tr>
<td>Somerset</td>
<td>70085</td>
<td>79321</td>
<td>1672</td>
</tr>
<tr>
<td>Sussex</td>
<td>44184</td>
<td>28840</td>
<td>1393</td>
</tr>
<tr>
<td>Union</td>
<td>78768</td>
<td>141417</td>
<td>2241</td>
</tr>
<tr>
<td>Warren</td>
<td>27550</td>
<td>20428</td>
<td>980</td>
</tr>
</tbody>
</table>

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

### 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](http://www.census.gov/tiger/boundary).

<table>
<thead>
<tr>
<th>State</th>
<th>Latitude, Longitude</th>
</tr>
</thead>
<tbody>
<tr>
<td>Alabama</td>
<td>-124.73, 34.99</td>
</tr>
<tr>
<td>USA</td>
<td>-88.20, 34.99</td>
</tr>
</tbody>
</table>

### Boundary Data: Counties within a State

**State data files:** County names and boundary points.

Data source: US census bureau, [www.census.gov/tiger/boundary](http://www.census.gov/tiger/boundary).

<table>
<thead>
<tr>
<th>County</th>
<th>Latitude, Longitude</th>
</tr>
</thead>
<tbody>
<tr>
<td>Atlantic</td>
<td>-74.87, 39.61</td>
</tr>
<tr>
<td>Bergen</td>
<td>-74.74, 40.42</td>
</tr>
<tr>
<td>Burlington</td>
<td>-74.73, 40.79</td>
</tr>
</tbody>
</table>

### Summary: Data Sources

- **(13 + 1)*(50 + 1) = 714 Data files**
  - Each file represents a "whole" region divided into "parts"
  - One entry per "part"
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.

### Polygon Data Type

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

#### 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]

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

### Polygon Data Type Implementation

```java
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

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

### Election Map

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

#### Polygon
- Geometric primitive.

#### Vote Tally
- Number of votes for each candidate.

#### Region
- State or county.

**Ex.**
Region Data Type Implementation

```java
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 Implementation

```java
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" + part);
        String input = in.readString();
        int i0 = input.indexOf("", i0+1);
        int i1 = input.indexOf("", i1+1);
        int i2 = input.indexOf("", i2+1);
        int i3 = input.indexOf("", i3+1);
        int i4 = input.indexOf("", i4+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;
    }
}
```

Vote Tally Data Type

Vote Tally. Election returns for one region

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

Ex.

```plaintext
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

```plaintext
blue when democrat beats republican
```

Election Map Data Type

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

Client:```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();
}
```
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();
        }
    }
}

Modular Programming

Modular program: Collection of data types.

hierarchy of instance variables
Visual Display of Quantitative Information

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

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

Purple New Jersey

% java ElectionMap NJ 2004

Purple America

% java ElectionMap USA 2008

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

Purple New Jersey

% java ElectionMap NJ 2004

Purple America

% java ElectionMap USA 2008

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

- $a_1 = \text{McCain votes}.$
- $a_2 = \text{Other votes}.$
- $a_3 = \text{Obama votes}.$

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

Implementation: change one method!

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

VoteTally.java

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

Cartograms

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

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

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).