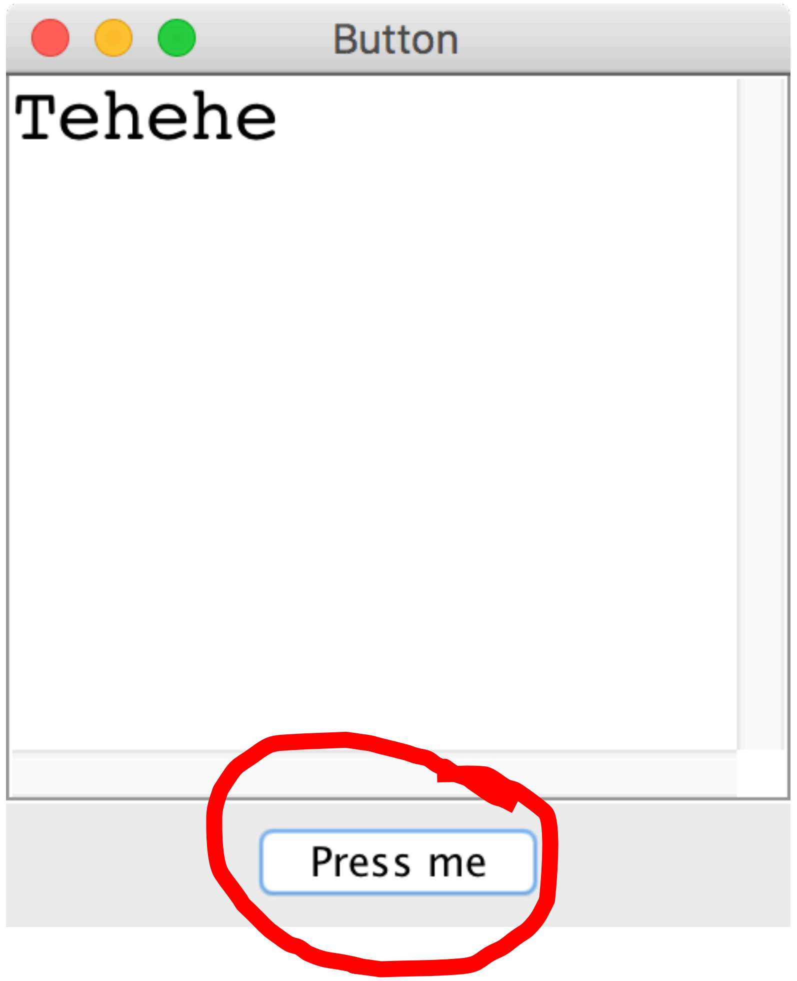
Data Structure Design I

Chris Piech
CS106A, Stanford University

Interactors

Button



Tehehe

`JButton`

JButton

```
JButton button = new JButton("Press me");
```

JButton

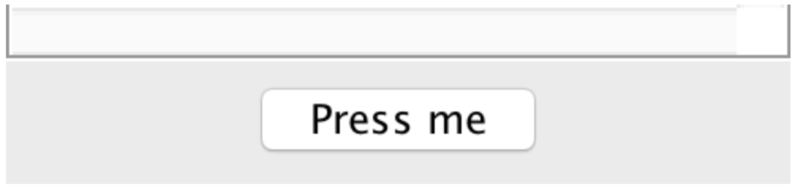
Button Text

$\downarrow$

```java
JButton button = new JButton("Press me");
```

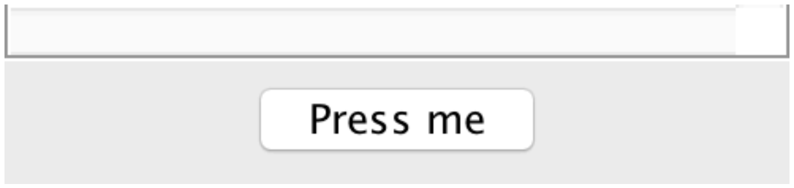
JButton

```
JButton button = new JButton("Press me");
add(button, SOUTH);
```



JButton

```
JButton button = new JButton("Press me");
add(button, SOUTH);
addActionListeners();
```



JButton

```
public void actionPerformed(ActionEvent e) {
   String actionCmd = e.getActionCommand();
   if(actionCmd.equals(“Press me”)) {
      println(“Tehehe”);
   }
}
```

JButton

```
public void actionPerformed(ActionEvent e) {
   String actionCmd = e.getActionCommand();
   if(actionCmd.equals(“Press me”)) {
      println(“Tehehe”);
   }
}
```

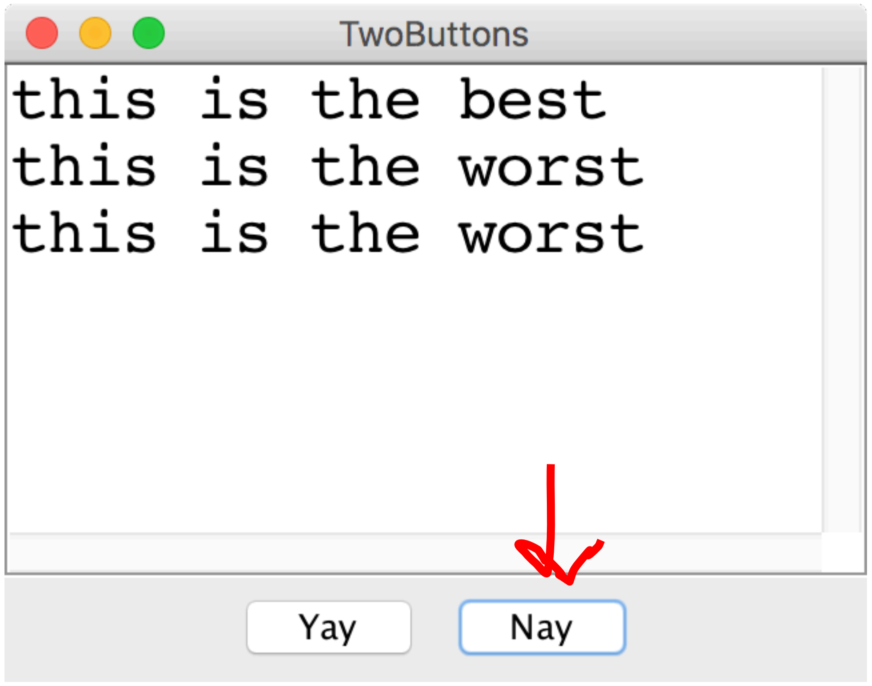
JButton

```
public void actionPerformed(ActionEvent e) {
   String actionCmd = e.getActionCommand();
   if(actionCmd.equals(“Press me”)) {
      println(“Tehehe”);
   }
}
```

JButton

```
public void actionPerformed(ActionEvent e) {
   String actionCmd = e.getActionCommand();
   if(actionCmd.equals(“Press me”)) {
      println(“Tehehe”);
   }
}
```

Two Buttons



```
this is the best
this is the worst
this is the worst
```

`JLabel`

JLabel

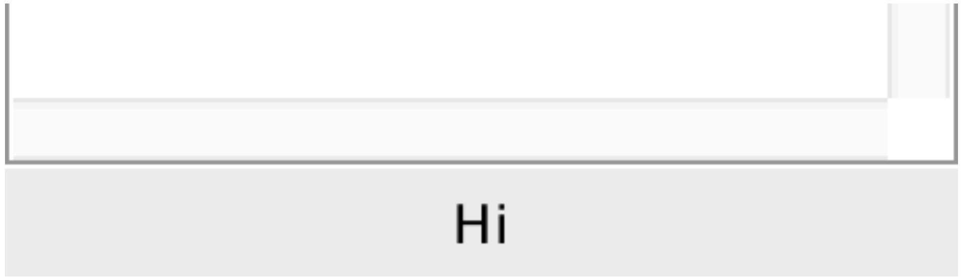
```
JLabel label = new JLabel("Hi");
```

JLabel

```java
JLabel label = new JLabel("Hi");
```

JLabel

```java
JLabel label = new JLabel("Hi");
add(label, SOUTH);
```



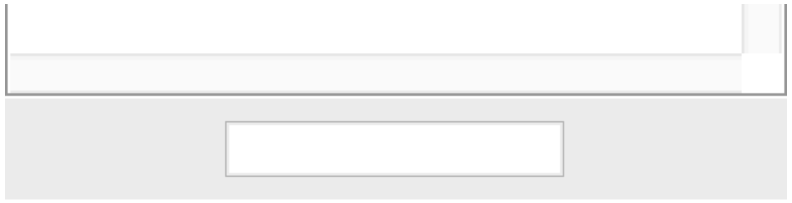
JTextField

JTextField

```
JTextField field = new JTextField(10);
add(field, SOUTH);
field.getText(); // returns string in field
field.setText("Marry me");
```

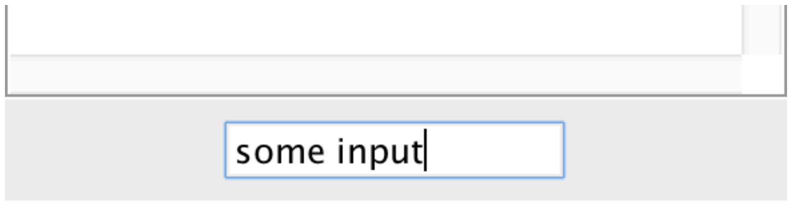
JTextField

```
JTextField field = new JTextField(10);
add(field, SOUTH);
field.getText(); // returns string in field
field.setText("Marry me");
```



JTextField

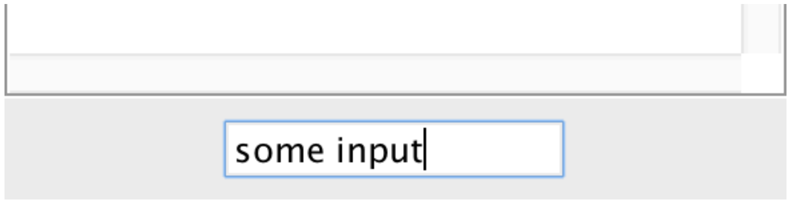
```
JTextField field = new JTextField(10);
add(field, SOUTH);
field.getText(); // returns string in field
field.setText("Marry me");
```



some input

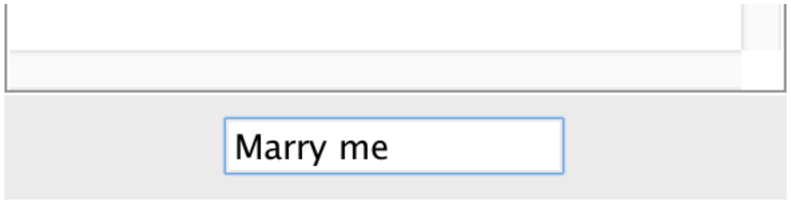
JTextField

```
JTextField field = new JTextField(10);
add(field, SOUTH);
field.getText(); // returns “some input”
field.setText("Marry me");
```


some input

JTextField

```
JTextField field = new JTextField(10);
add(field, SOUTH);
field.getText(); // returns “some input”
field.setText("Marry me");
```



Marry me

*in honor of Carlos, my freshman and sophmore roommate who just proposed

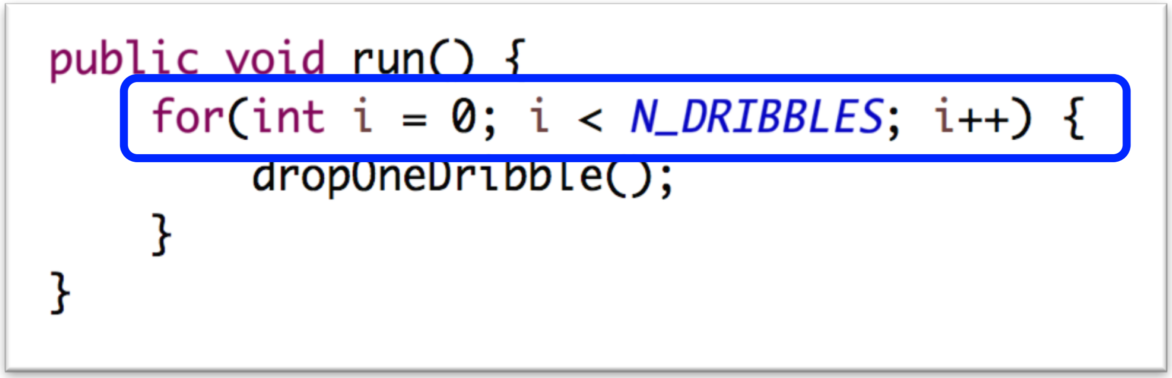
Recall the Dancing Children

Normal Program

Run Method

Normal Program

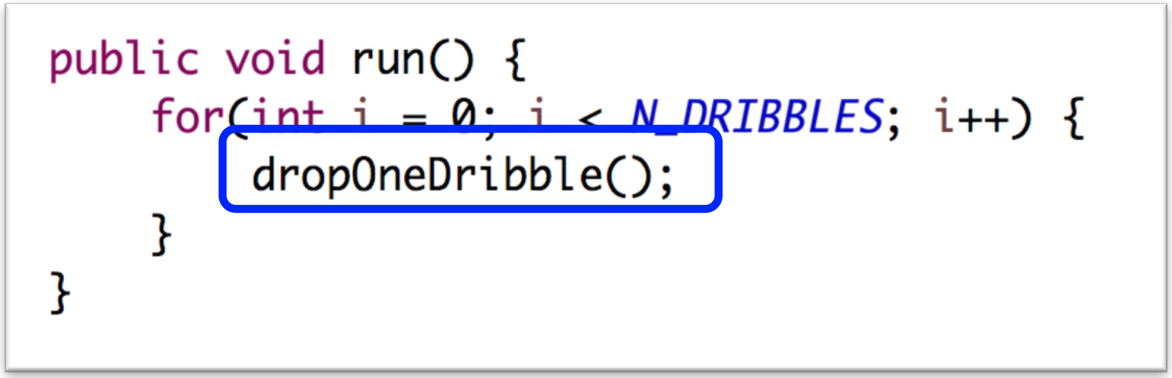
Run Method



```
public void run() {
    for(int i = 0; i < N_DRIBBLES; i++) {
        dropOneDribble();
    }
}
```

Normal Program

Run Method



```
public void run() {
    for(int i = 0; i < N_DRIBBLES; i++) {
        dropOneDribble();
    }
}
```

Normal Program

Run Method

New Listener Characters

Action Listener

Action Performed

Program Starts Running

Run Method

Action Performed

Add Action Listeners

Run Method

Action Performed

Action Listener

```
addActionListeners();
```

Program Runs as Usual

Run Method

Action Performed

Action Listener

Button Clicked!

Run Method

Action Performed

Action Listener

Calls Action Performed Method

Run Method

Action Performed

Action Listener

When done, Run continues.

Run Method

Action Performed

Action Listener

Keeps Doing Its Thing...

Run Method

Action Performed

Action Listener

Button Clicked!

Run Method

Action Performed

Action Listener

Calls Action Performed Method

Run Method

Action Performed

Action Listener

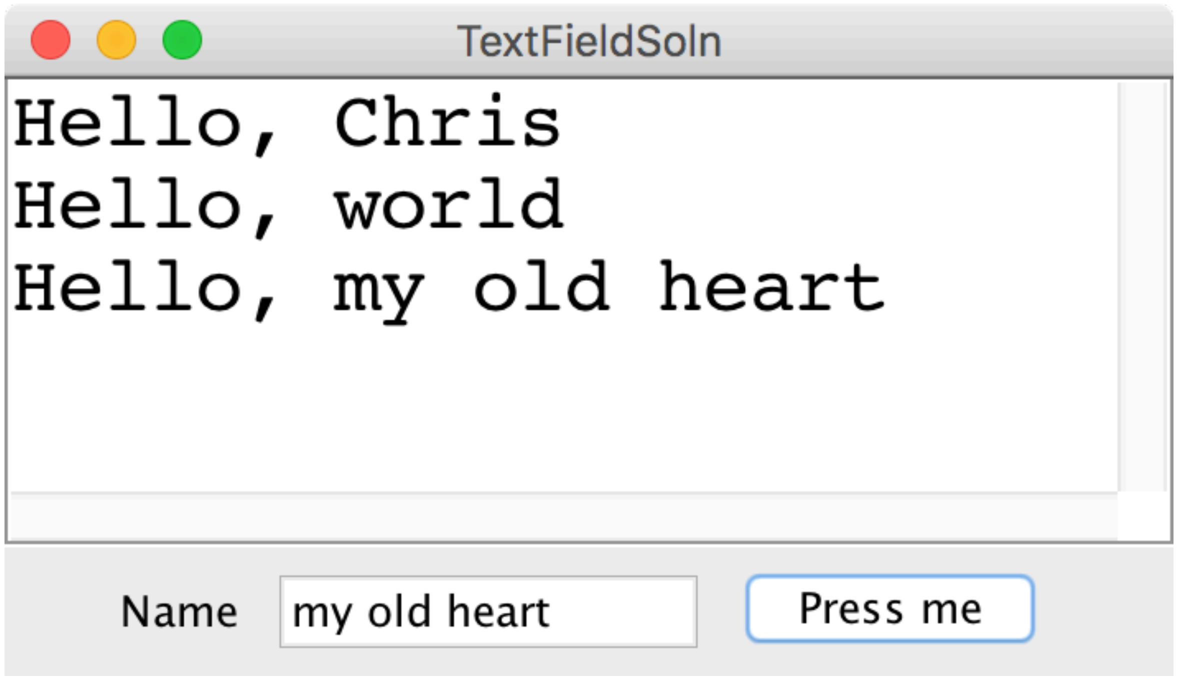
When done, Run continues.

Run Method

Action Performed

Action Listener

Text Field



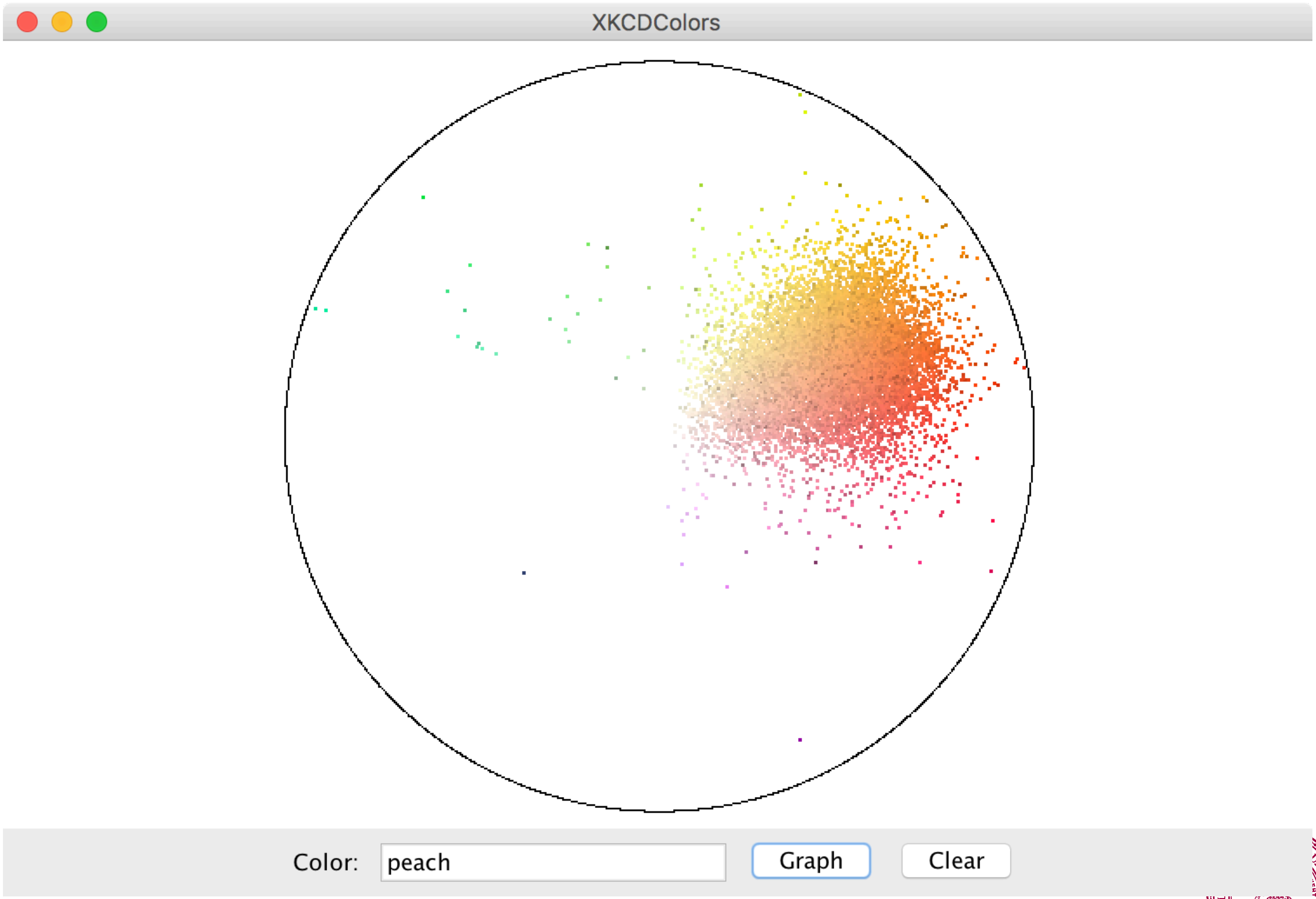
```
Hello, Chris
Hello, world
Hello, my old heart
```

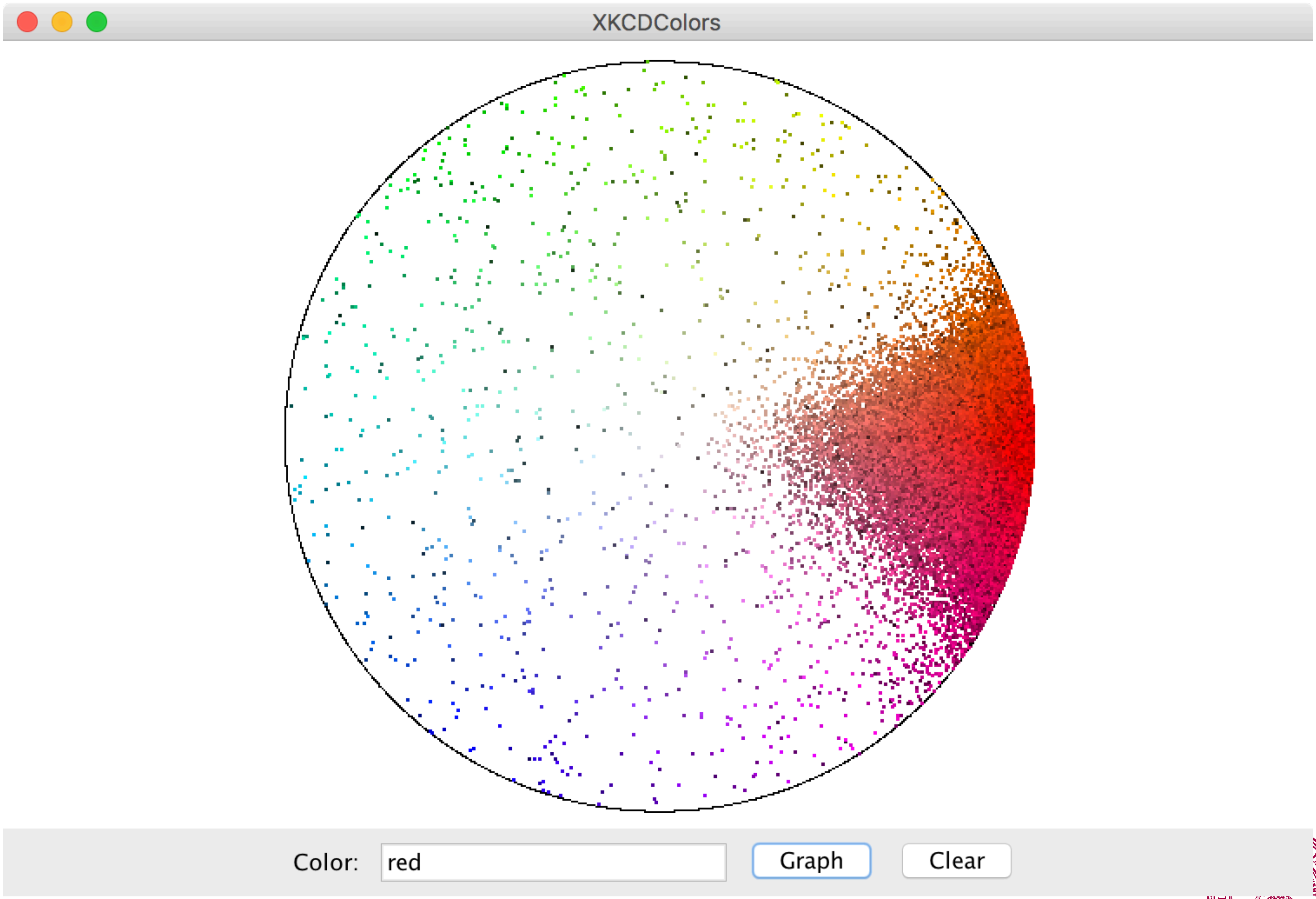
Something awesome

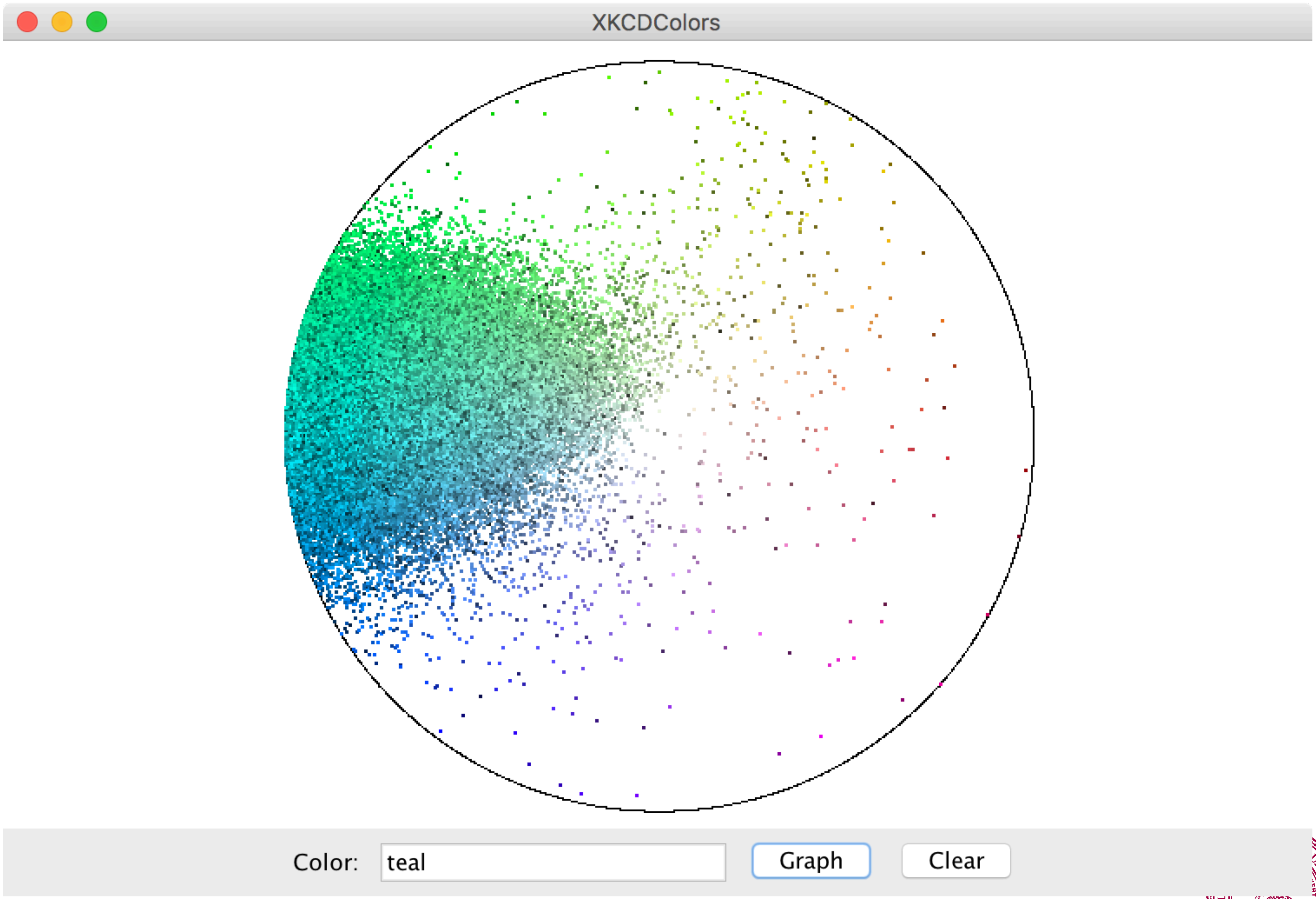
*thanks Keith for the idea

Putting two things together

The XKCD Color Survey


XKCDColors
Color:
peach
Graph
Clear


XKCDColors
Color:
red
Graph
Clear


XKCDColors
Color:
teal
Graph
Clear

The XKCD Color Survey

- Volunteers (online) were shown a randomly-chosen color and asked to name the color.
- The result is (after filtering) about 2.8 million RGB triplets and their names.

The File Format

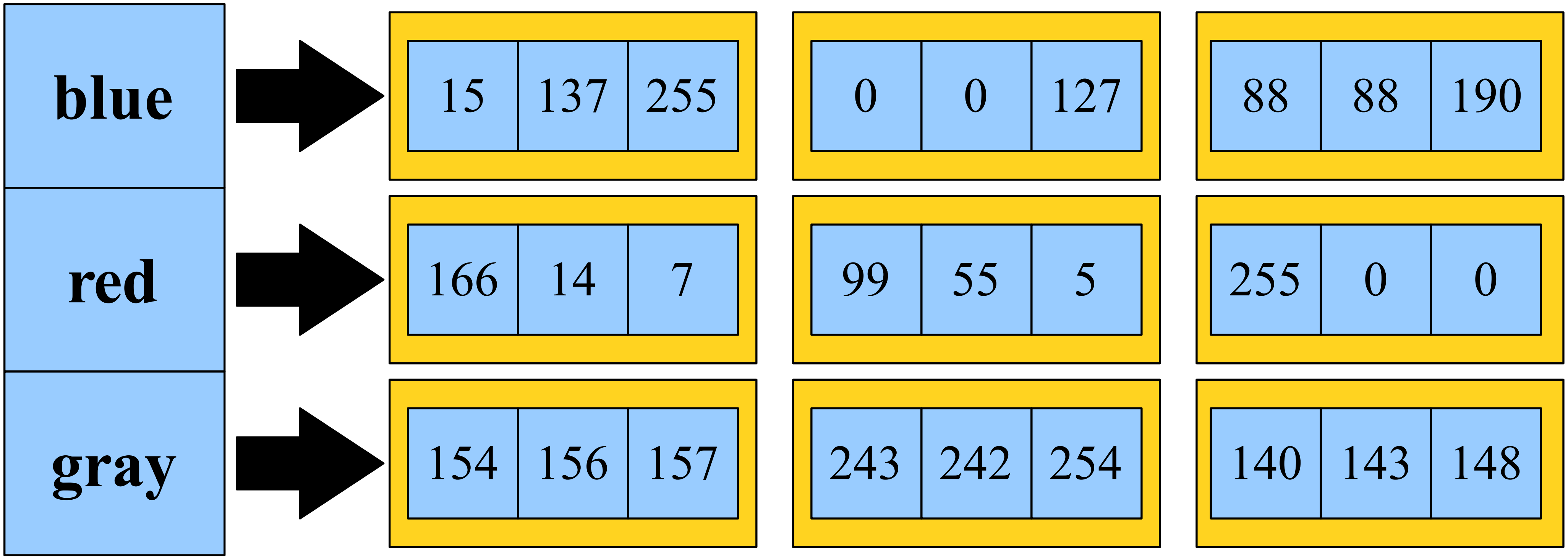
color-name,

red,

green,

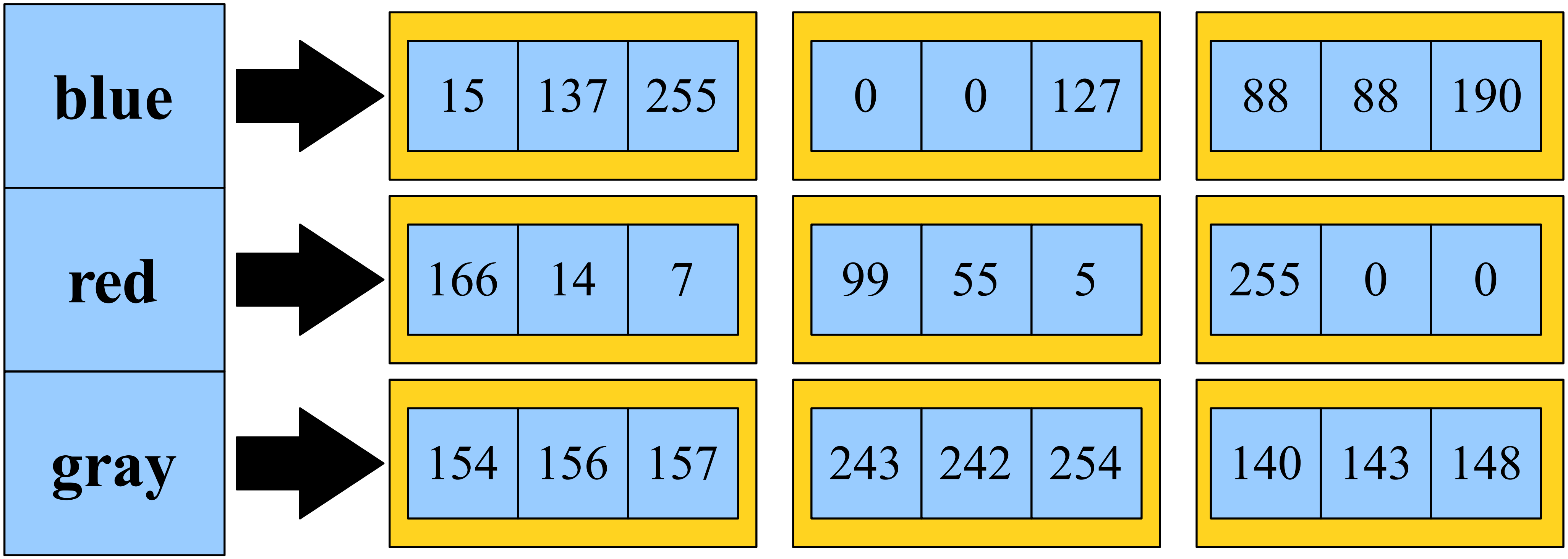
blue

How to Structure Data



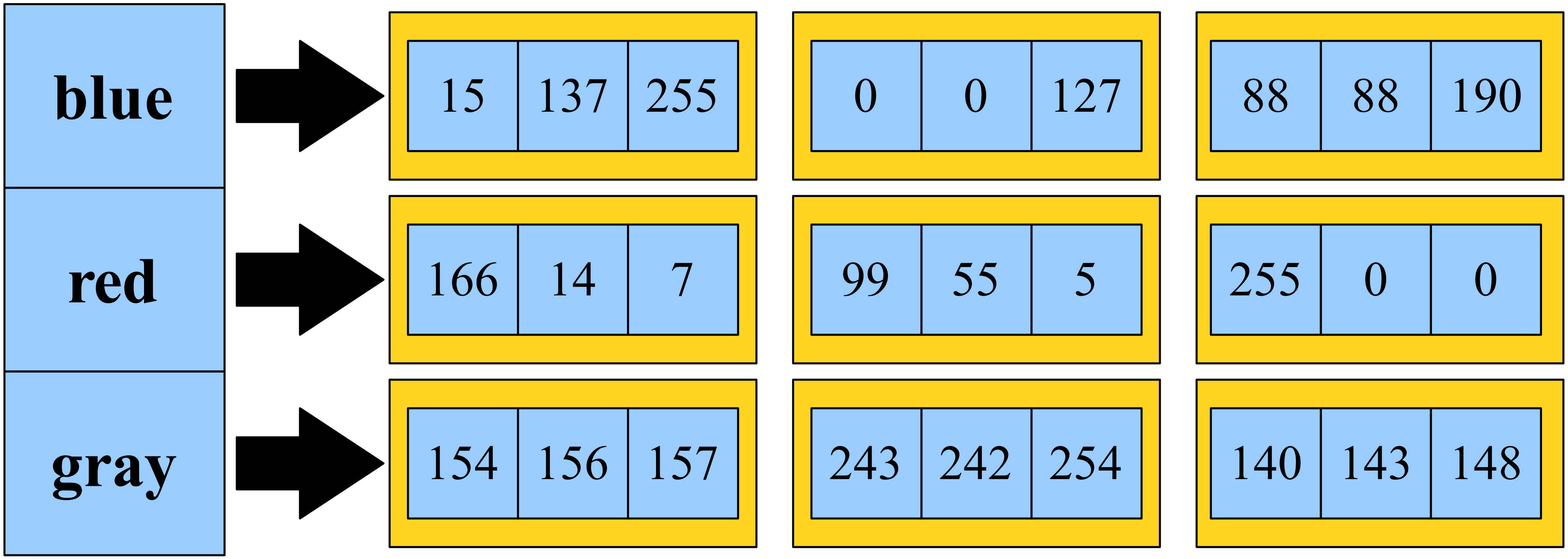
associate each color name with a list of colors

How to Structure Data



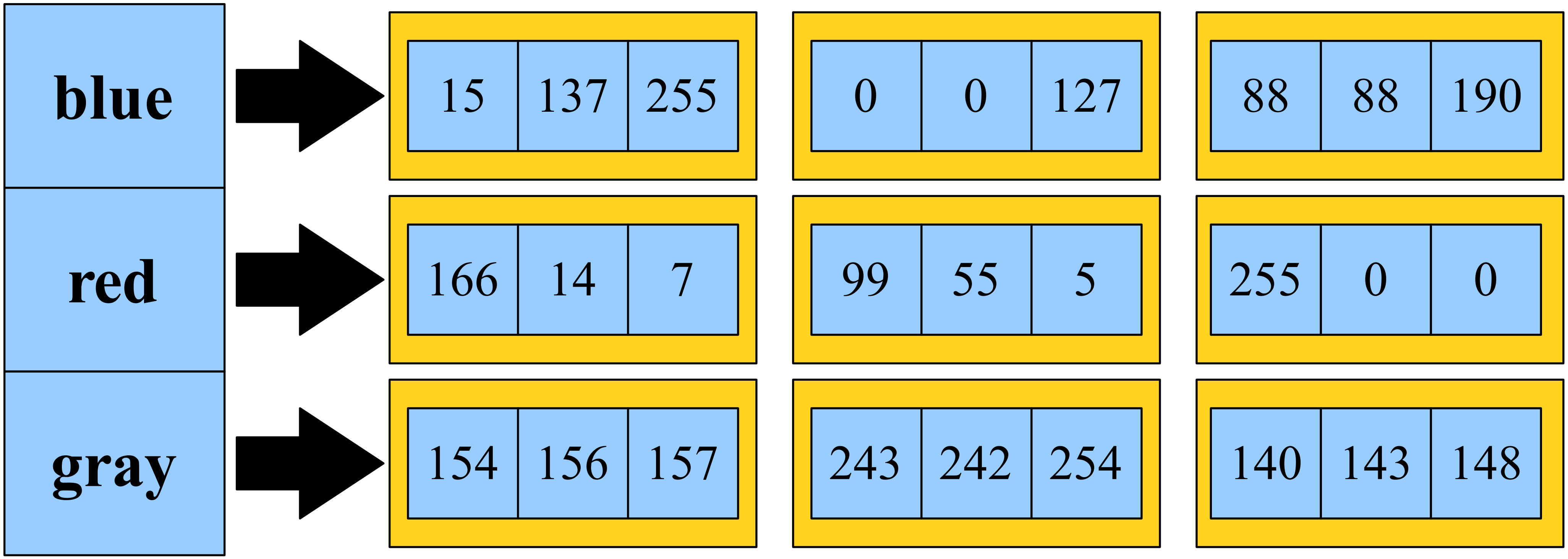
`HashMap`<*color name*, *list of colors*>

How to Structure Data



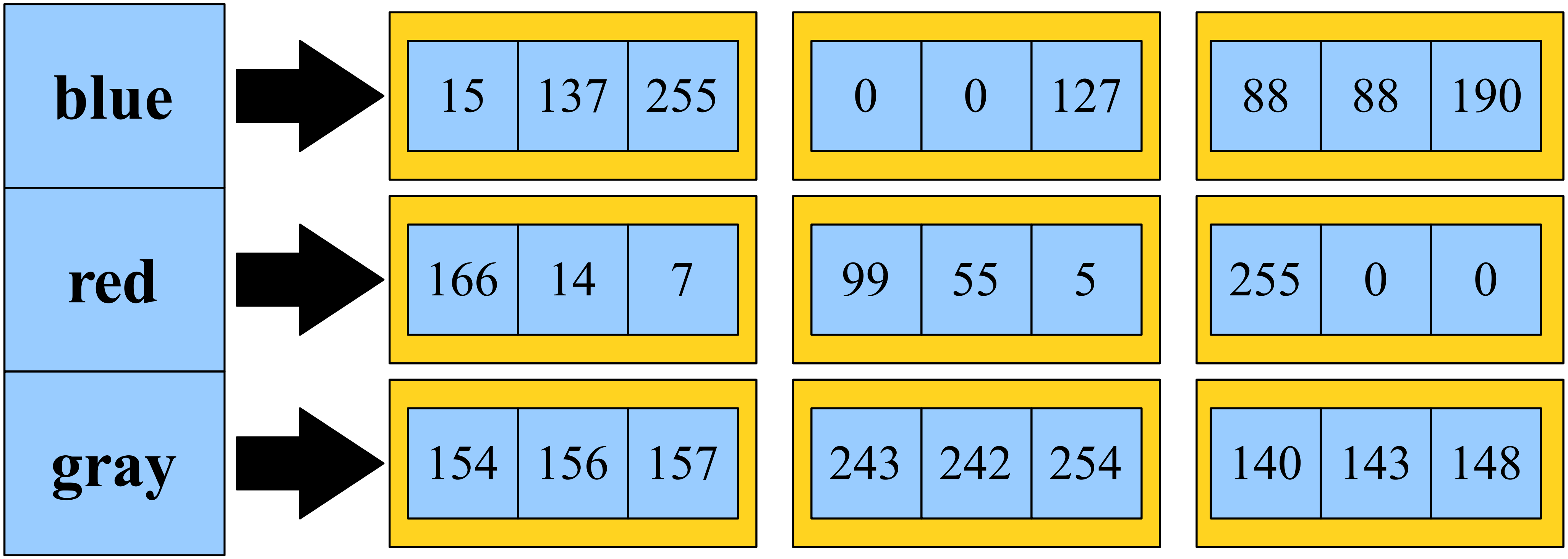
`HashMap<String,` ***list of colors*** `>`

How to Structure Data



```
HashMap<String, ArrayList<color>>
```

How to Structure Data



```
HashMap<String, ArrayList<Color>>
```

Further Reading

- http://blog.xkcd.com/2010/05/03/color-survey-results/

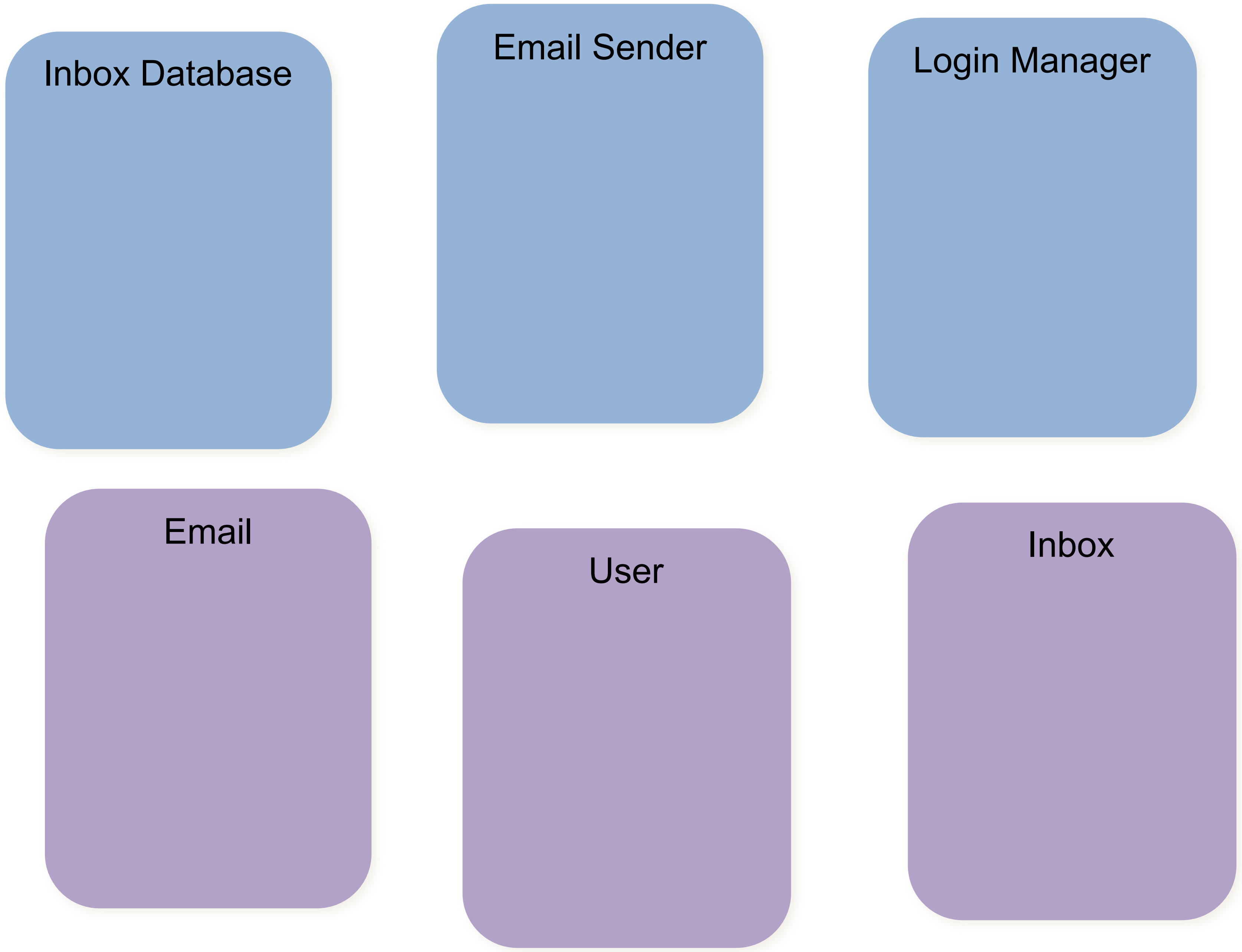
```
pause(1000);
```

Some *large* programs are in Java

Apache OpenOffice™

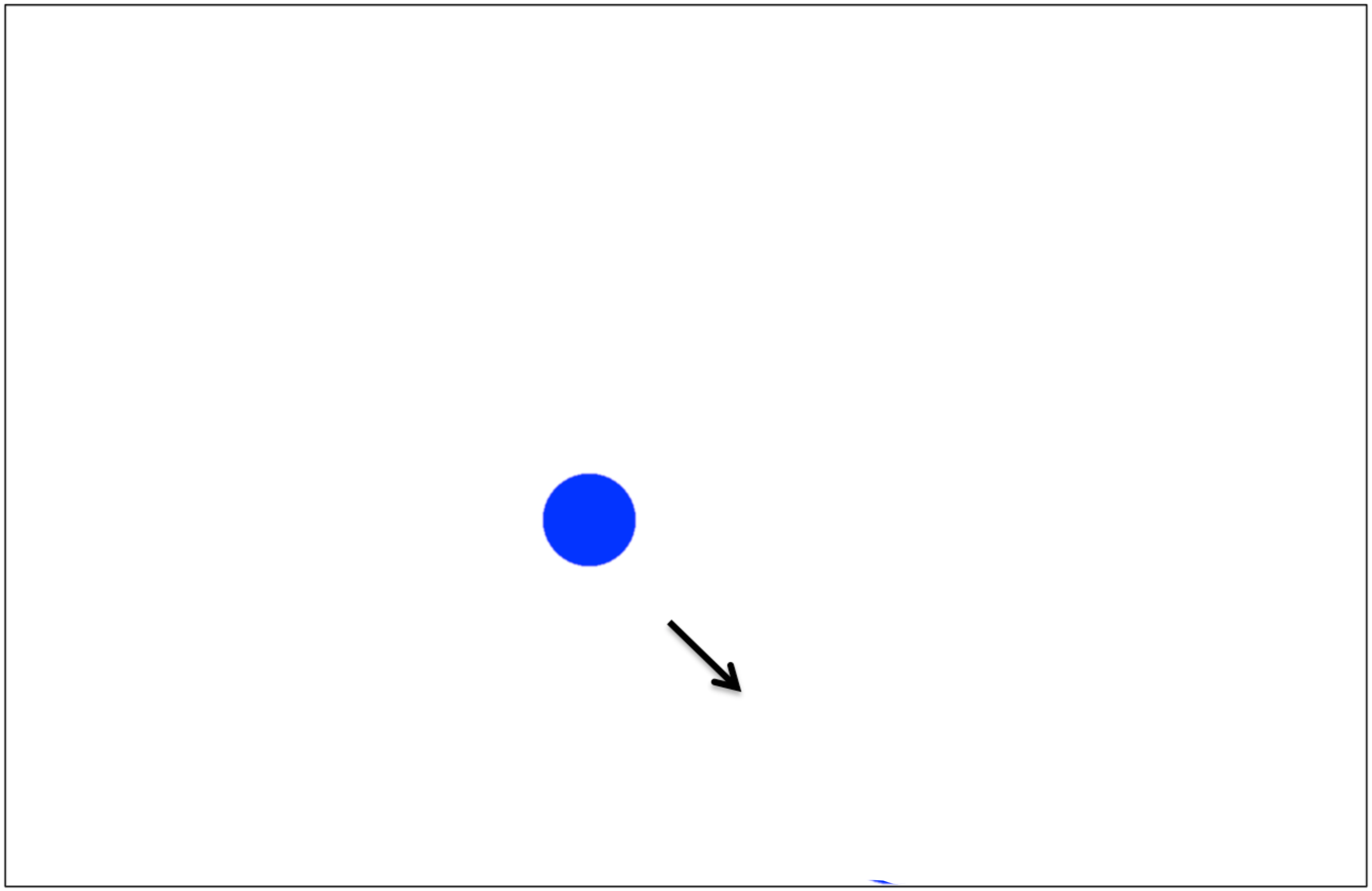
How?

Define New Variable Types

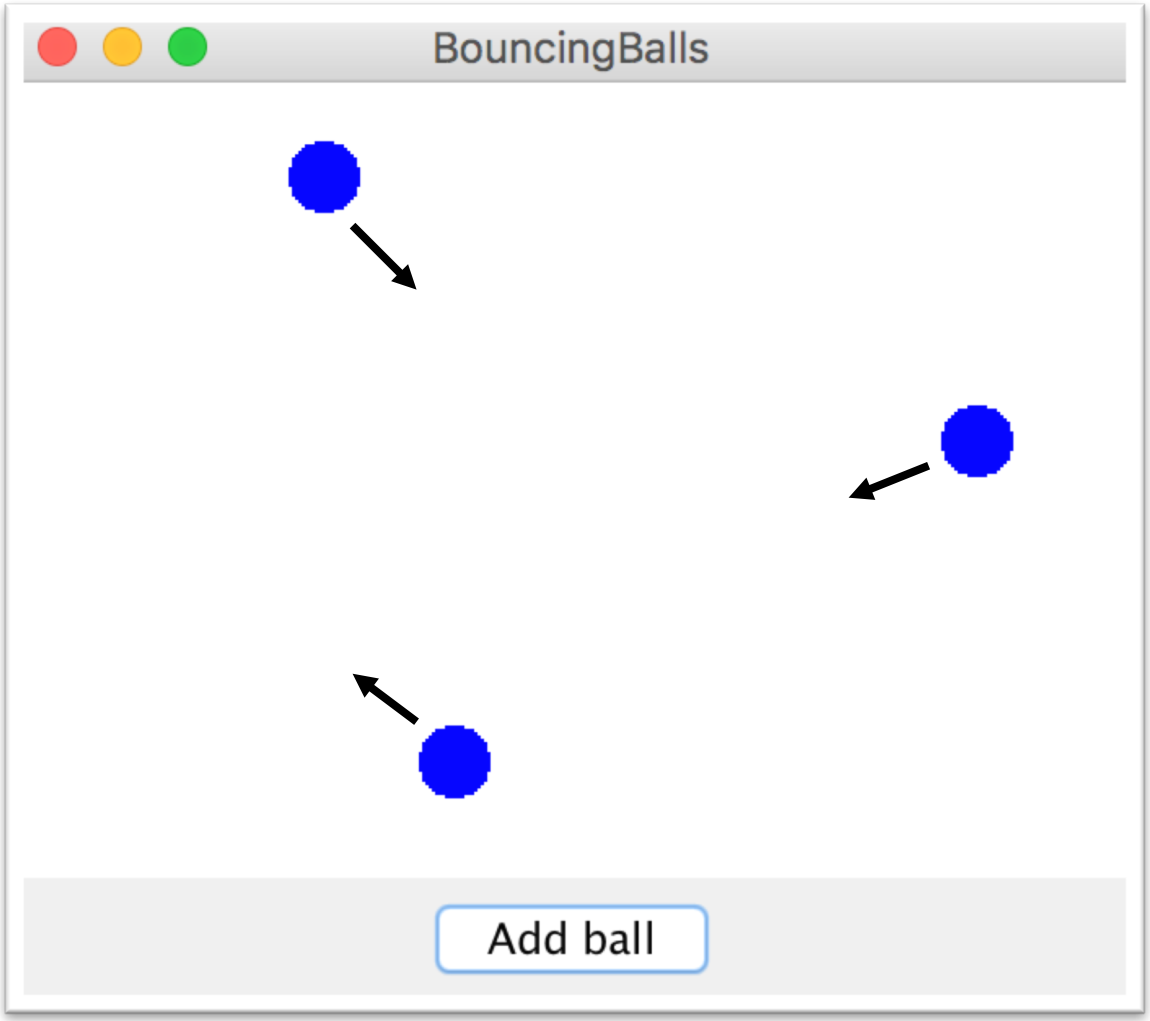


Even small programs define new variable types

Can you do this?



Bouncing Balls



Classes define new Variable Types

- A student registration system needs to store info about students, but Java has no **`Student`** type.

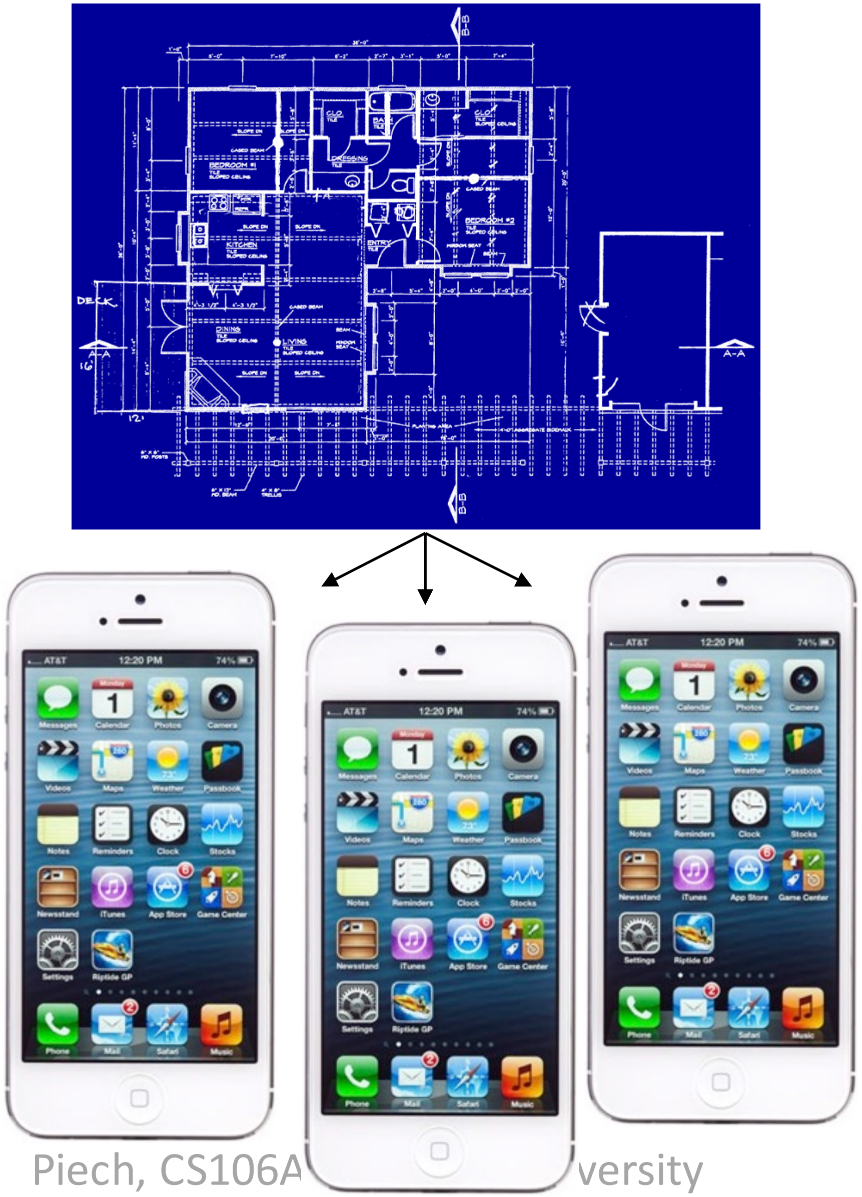
- A music synthesizer app might want to store information about users' accounts, but Java has no **`Instrument`** type.

- However, Java does provide a feature for us to add new data types to the language: **classes**.
  - Writing a class defines a new data type.

Classes are like blueprints

class: A template for a new type of variable.

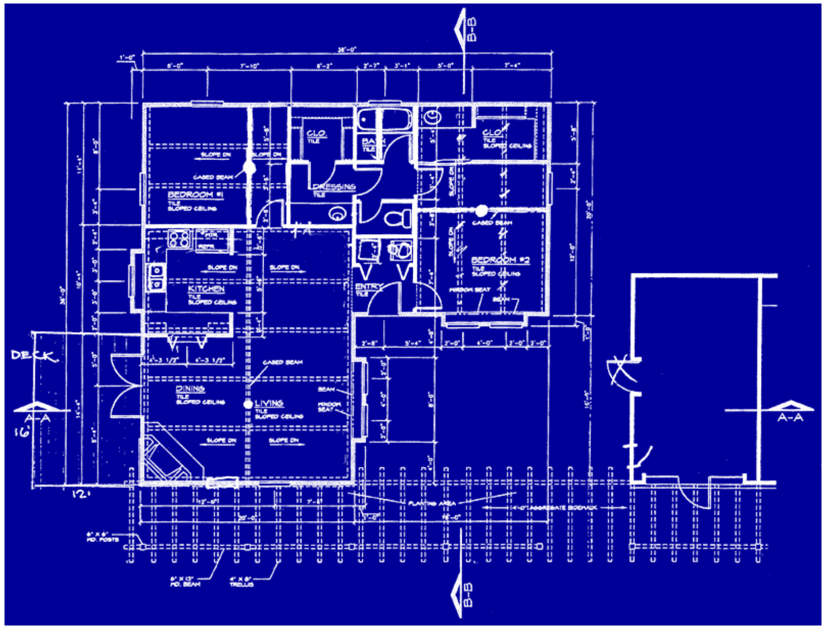
A blueprint is a helpful analogy



#key: Classes define new variable *types*

#key: Classes decompose your program across files

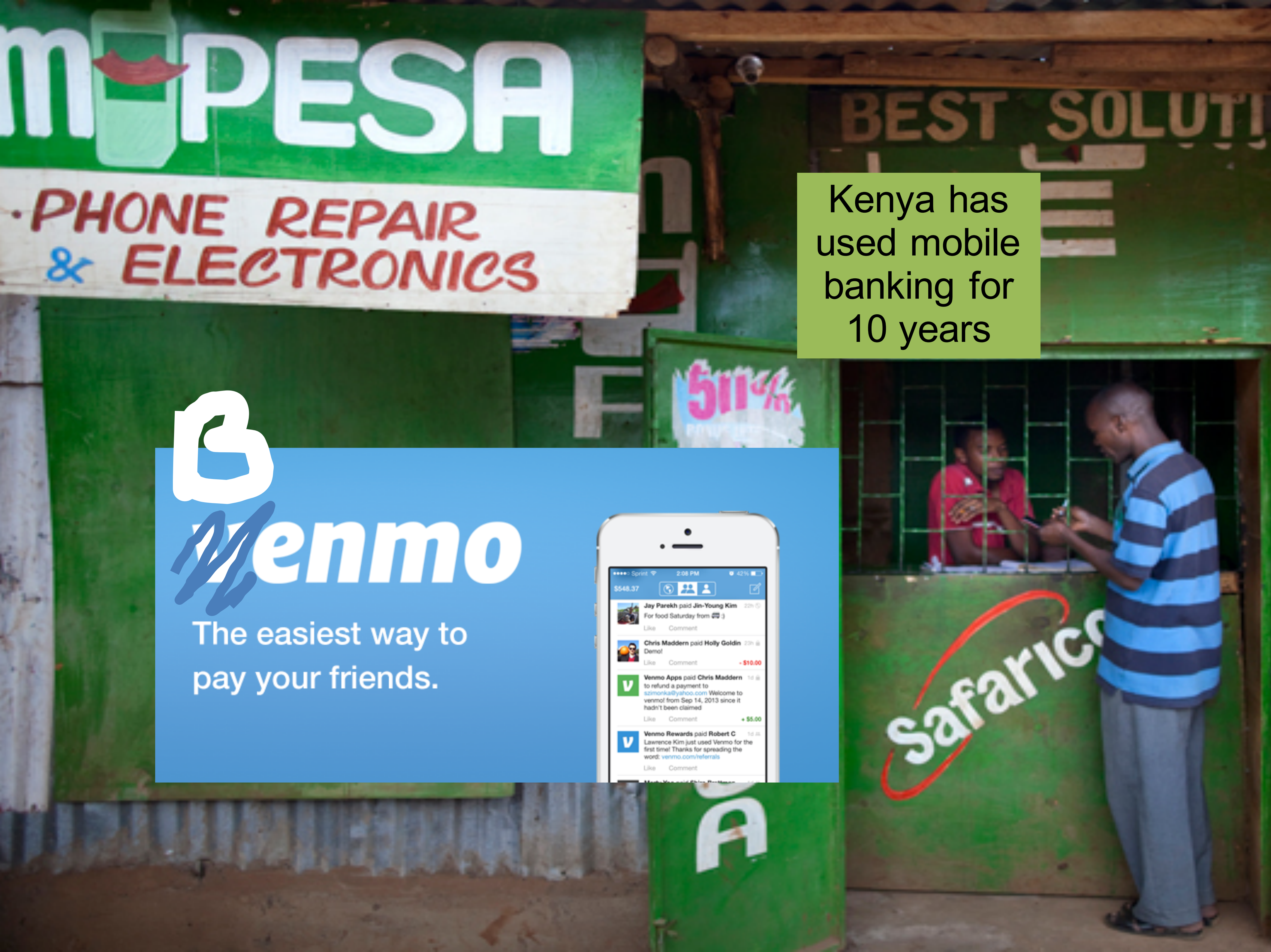
Classes are like blueprints



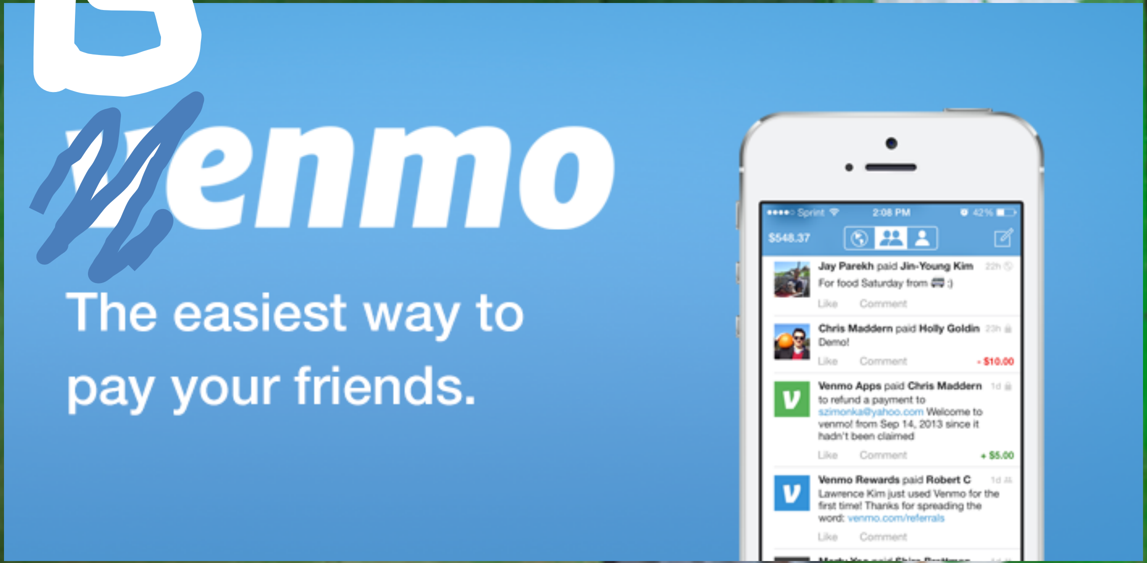
To design a new variable type you must specify three things:
1. What subvariables make up this new variable type?
2. What methods can you call on a variable of this type?
3. What happens when you make a new instance of this type?

What is a class?

A class defines a new variable type



Kenya has used mobile banking for 10 years


B
~~V~~enmo

The easiest way to pay your friends.

Sprint 2:08 PM 42%
$548.37

Jay Parekh paid **Jin-Young Kim** 22h
For food Saturday from 🚌 :)
Like Comment

Chris Maddern paid **Holly Goldin** 23h
Demo!
Like Comment - $10.00

Venmo Apps paid **Chris Maddern** 1d
to refund a payment to szimonka@yahoo.com Welcome to venmo! from Sep 14, 2013 since it hadn't been claimed
Like Comment + $5.00

Venmo Rewards paid **Robert C** 1d
Lawrence Kim just used Venmo for the first time! Thanks for spreading the word: venmo.com/referrals
Like Comment

Classes: Take 1

This goes in BankAccount.java!

```
public class BankAccount {
   // the instance variable define what makes up the class
   public String name;
   public double money;
}
```

Instance variables have a special meaning

Classes: Take 1

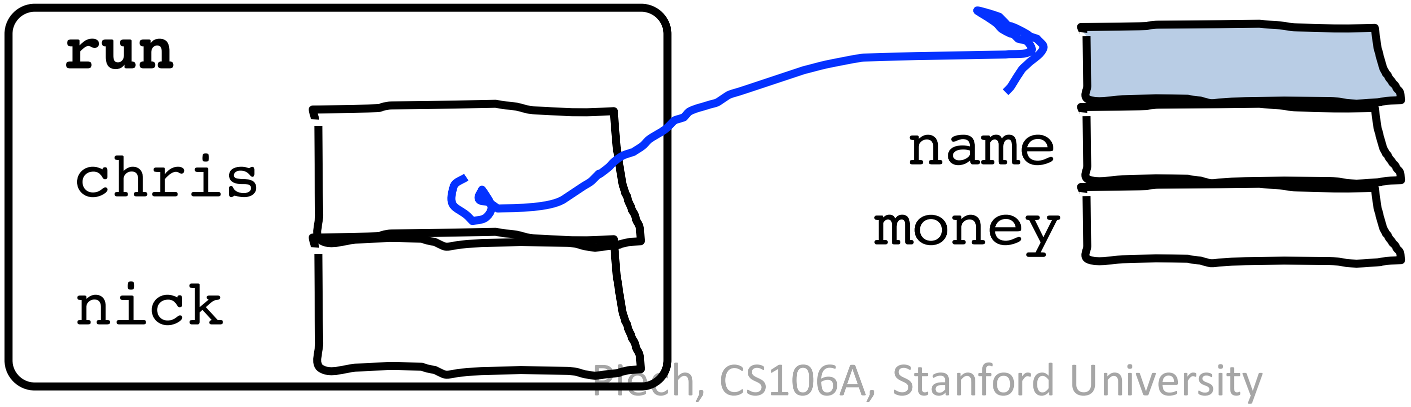
```
public class BankAccount {
   // the instance variable define what makes up the class
   public String name;
   public double money;
}
```

```
public class Benmo{
   public void run() {
      BankAccount chris = new BankAccount();
      BankAccount nick = new BankAccount();
      chris.name = "Chris";
      chris.money = 100;
      nick.name = "Nick";
      nick.money = 50;
   }
}
```

Classes: Take 1

```
public class BankAccount {
   // the instance variable define what makes up the class
   public String name;
   public double money;
}
```

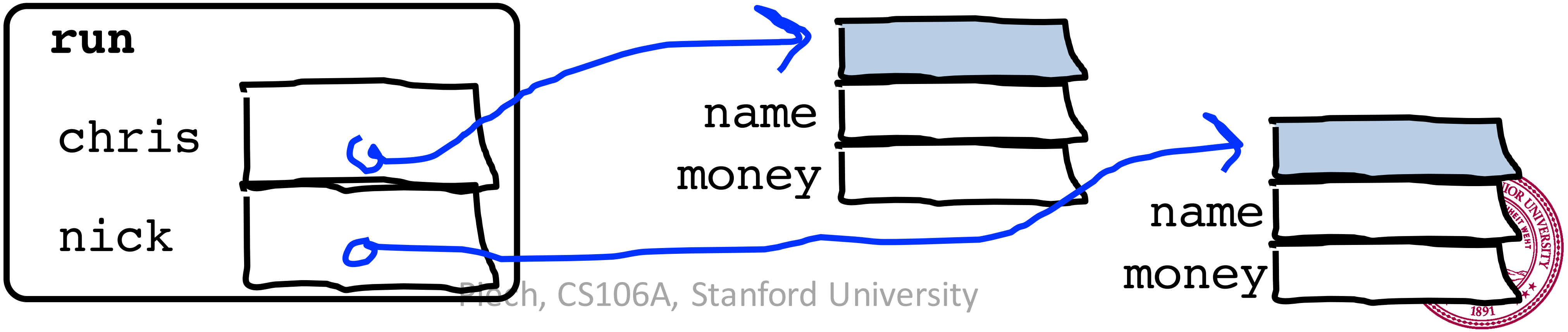
```
public class Benmo{
   public void run() {
      BankAccount chris = new BankAccount();
      BankAccount nick = new BankAccount();
      chris.name = “Chris”;
      chris.money = 100;
      nick.name = “Nick”;
      nick.money = 50;
   }
}
```



Classes: Take 1

```
public class BankAccount {
   // the instance variable define what makes up the class
   public String name;
   public double money;
}
```

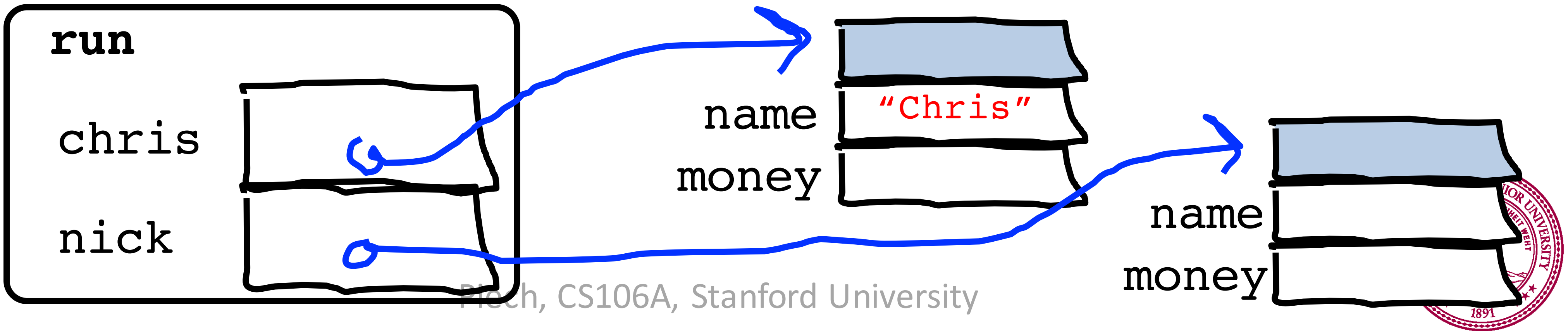
```
public class Benmo{
   public void run() {
      BankAccount chris = new BankAccount();
      BankAccount nick = new BankAccount();
      chris.name = “Chris”;
      chris.money = 100;
      nick.name = “Nick”;
      nick.money = 50;
   }
}
```



Classes: Take 1

```
public class BankAccount {
   // the instance variable define what makes up the class
   public String name;
   public double money;
}
```

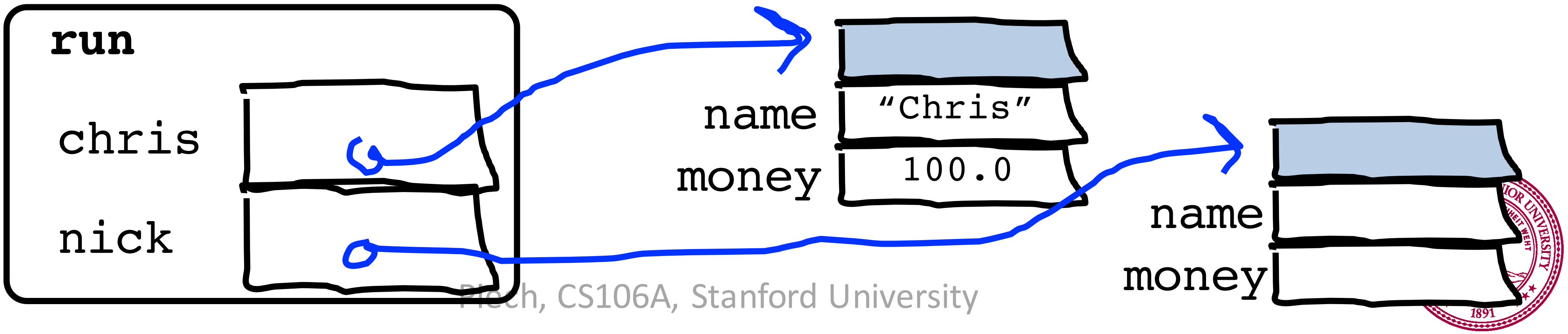
```
public class Benmo{
   public void run() {
      BankAccount chris = new BankAccount();
      BankAccount nick = new BankAccount();
      chris.name = "Chris";
      chris.money = 100;
      nick.name = "Nick";
      nick.money = 50;
   }
}
```



Classes: Take 1

```
public class BankAccount {
   // the instance variable define what makes up the class
   public String name;
   public double money;
}
```

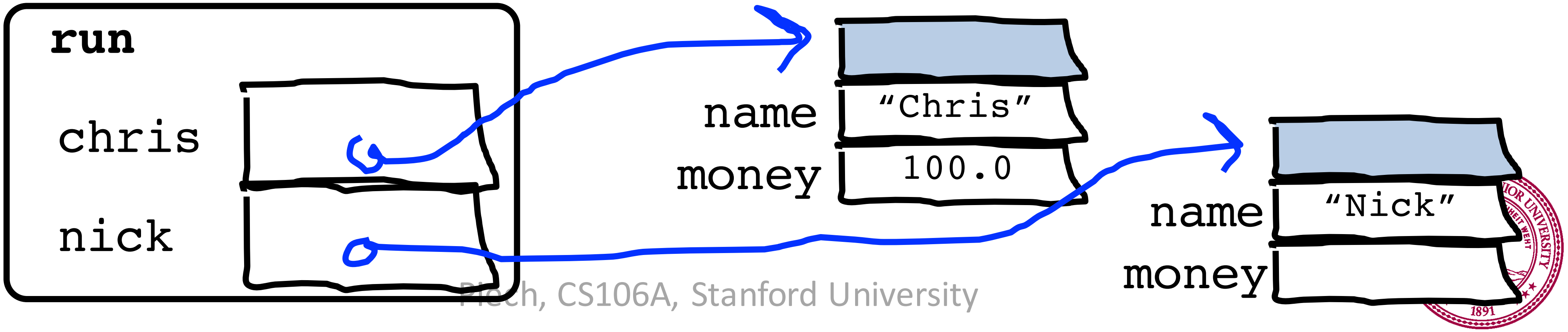
```
public class Benmo{
   public void run() {
      BankAccount chris = new BankAccount();
      BankAccount nick = new BankAccount();
      chris.name = “Chris”;
      chris.money = 100;
      nick.name = “Nick”;
      nick.money = 50;
   }
}
```



Classes: Take 1

```
public class BankAccount {
   // the instance variable define what makes up the class
   public String name;
   public double money;
}
```

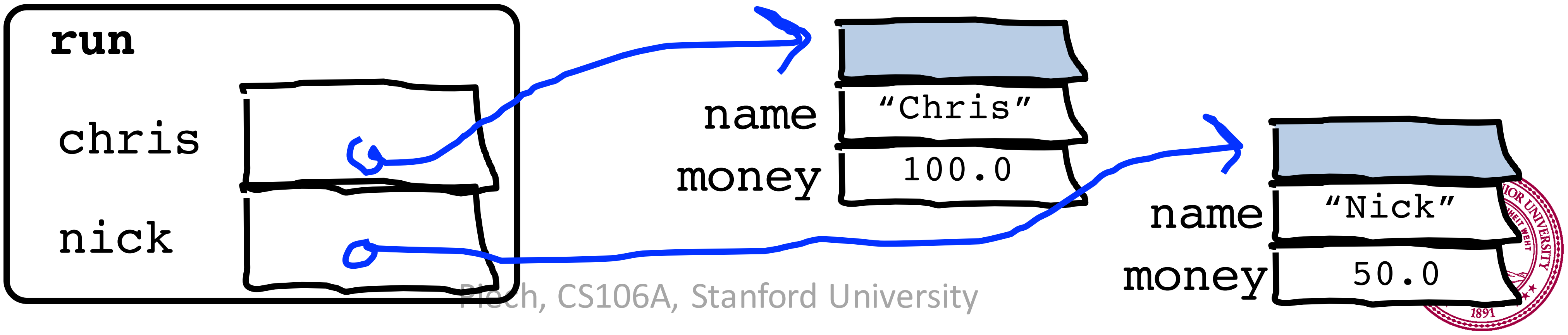
```
public class Benmo{
   public void run() {
      BankAccount chris = new BankAccount();
      BankAccount nick = new BankAccount();
      chris.name = "Chris";
      chris.money = 100;
      nick.name = "Nick";
      nick.money = 50;
   }
}
```



Classes: Take 1

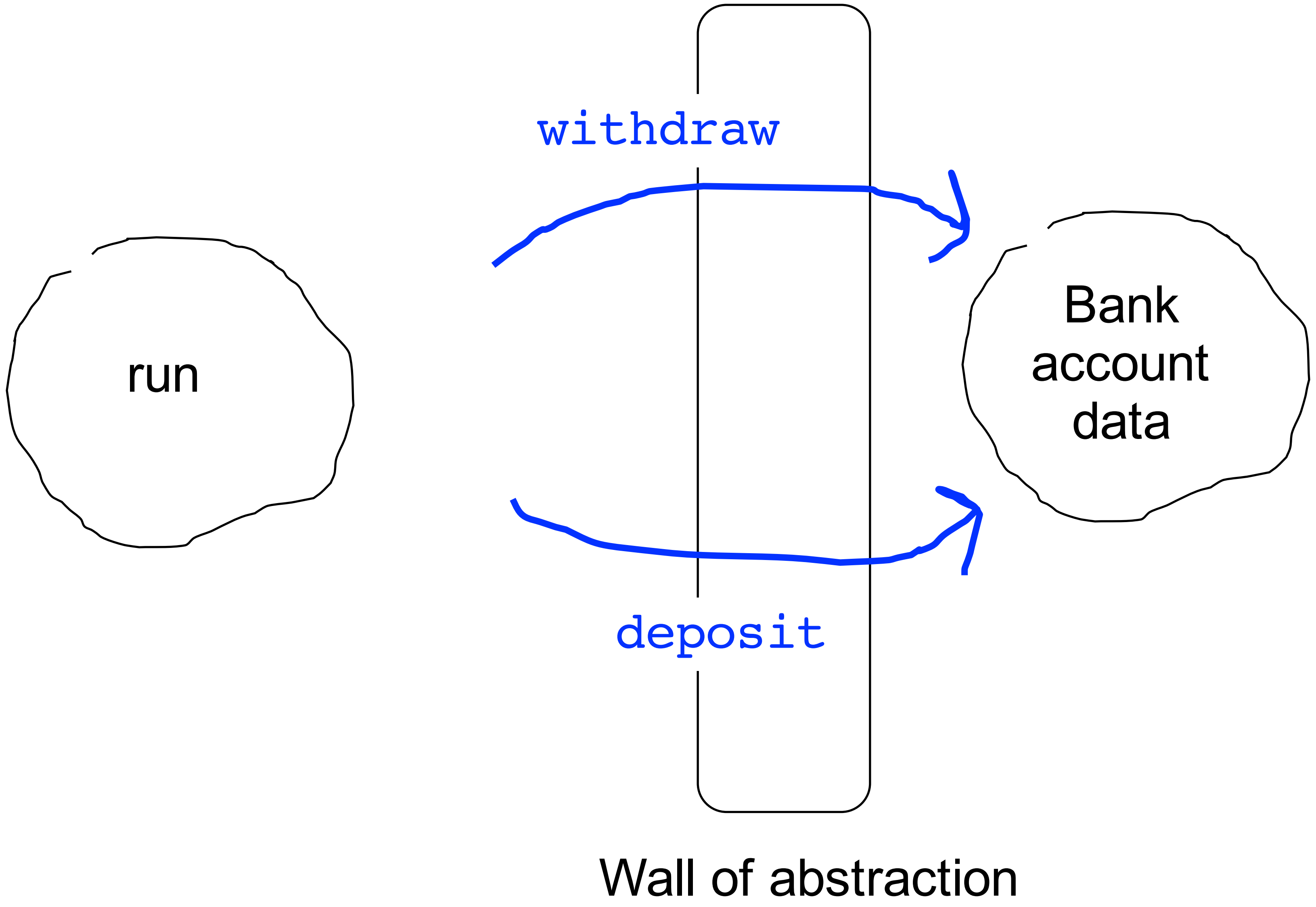
```
public class BankAccount {
   // the instance variable define what makes up the class
   public String name;
   public double money;
}
```

```
public class Benmo{
   public void run() {
      BankAccount chris = new BankAccount();
      BankAccount nick = new BankAccount();
      chris.name = "Chris";
      chris.money = 100;
      nick.name = "Nick";
      nick.money = 50;
   }
}
```



What is a class?

A class defines a new variable type


withdraw
run
Bank account data
deposit
Wall of abstraction

Adding Privacy

```
private double money;
```

- **encapsulation**: Hiding implementation details of an object from its clients.
  - Encapsulation provides *abstraction*.
    - separates external view (behavior) from internal view (state)
  - Encapsulation protects the integrity of an object's data.

- A class's instance variables should be declared *private*.
  - No code outside the class can access or change it.

Classes: Take 2

This goes in its own file!

```
public class BankAccount {
   // the instance variable define what makes up the class
   public String name;
   public double money;
}
```

Instance variables have a special meaning

Classes: Take 2

```
public class BankAccount {
   // 1. What variables make up the class
   public String name;
   public double money;
}
```

Classes: Take 2

```
public class BankAccount {
   // 1. What variables make up the class
   private String name;
   private double money;
}
```

Classes: Take 2

```
public class BankAccount {
   // 1. What variables make up the class
   private String name;
   private double money;

   // 2. What methods can a user call on a bankAccount?
   public void deposit(double amount) {
       ...
   }

   public boolean withdraw(double amount) {
       ...
   }

}
```

Classes: Take 2

```
public class BankAccount {
   // 1. What variables make up the class
   private String name;
   private double money;

   // 2. What methods can a user call on a bankAccount?
   public void deposit(double amount) {
      this.money += amount;
   }

   public boolean withdraw(double amount) {
      ...
   }

}
```

`this`

Classes: Take 2

```
public class BankAccount {
   // 1. What variables make up the class
   private String name;
   private double money;

   // 2. What methods can a user call on a bankAccount?
   public void deposit(double amount) {
      this.money += amount;
   }

   public boolean withdraw(double amount) {
      ...
   }

}
```

Classes: Take 2

```java
public class BankAccount {
   // 1. What variables make up the class
   private String name;
   private double money;

   // 2. What methods can a user call on a bankAccount?
   public void deposit(double amount) {
      this.money += amount;
   }

   public boolean withdraw(double amount) {
      if(amount <= this.money) {
         this.money -= amount;
         return true;
      }
      return false;
   }

}
```

Classes: Take 2

```java
public class BankAccount {
   // 1. What variables make up the class
   private String name;
   private double money;

   // 2. What methods can a user call on a bankAccount?
   public void deposit(double amount) {
      this.money += amount;
   }

   public boolean withdraw(double amount) {
      if(amount <= this.money) {
         this.money -= amount;
         return true;
      }
      return false;
   }

    // 3. How do you make a new one?
   public BankAccount(String name, double amount) {
      this.money = amount;
      this.money = name;
   }
}
```

Jay Parekh paid Jin-Young Kim 22h
For food Saturday from 🚌 :)
Like Comment

Chris Maddern paid Holly Goldin 23h
Demo!
Like Comment - $10.00

Venmo Apps paid Chris Maddern 1d
to refund a payment to szimonka@yahoo.com Welcome to venmo! from Sep 14, 2013 since it hadn't been claimed
Like Comment + $5.00

Venmo Rewards paid Robert C 1d
Lawrence Kim just used Venmo for the first time! Thanks for spreading the word: venmo.com/referrals
Like Comment

Classes on one slide

1. What variables make up this new super variable type?

    Instance variables

2. What methods can you call on a variable of this type?

    It’s public methods

3. What happens when the user makes a **`new`** instance?

    The “constructor”

* Don’t forget that all methods and constructors have access to a **`this`** reference

What classes?

What is a class?

A class defines a new variable type