Defining Variable Types

Chris Piech

CS106A, Stanford University
Can you do this?
Interactors
Button

![Image of a button with the text 'Press me']
JButton
JButton button = new JButton("Press me");
JButton button = new JButton("Press me");
JButton button = new JButton("Press me");
add(button, SOUTH);
```java
JButton button = new JButton("Press me");
add(button, SOUTH);
addActionListeners();
```
public void actionPerformed(ActionEvent e) {
    String actionCmd = e.getActionCommand();
    if(actionCmd.equals("Press me")) {
        println("Tehehe");
    }
}
public void actionPerformed(ActionEvent e) {
    String actionCmd = e.getActionCommand();
    if (actionCmd.equals("Press me")) {
        println("Tehehe");
    }
}
public void actionPerformed(ActionEvent e) {
    String actionCmd = e.getActionCommand();
    if(actionCmd.equals("Press me")) {
        println("Tehehe");
    }
}
public void actionPerformed(ActionEvent e) {
    String actionCmd = e.getActionCommand();
    if(actionCmd.equals("Press me")) {
        println("Tehehe");
    }
}
End review
Some *large* programs are in Java
How?
Define New Variable Types

Inbox Database

Email Sender

Login Manager

Email

User

Inbox

Piech, CS106A, Stanford University
Even small programs define new variable types
You Have Been *Using* Variable Types

What would it take to define your own?
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 different voices, 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.

“This class is so meta” – Anonymous Stanford Student
Classes are like blueprints

**class**: A template for a new type of variable.

A blueprint is a helpful analogy
Classes define new variable types
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.
public class BankAccount {
    // the instance variable define what makes up the class
    public String name;
    public double money;
}

Instance variables have a special meaning
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 mehran = new BankAccount();
        chris.name = “Chris”;
        chris.money = 100;
        mehran.name = “Mehran”;
        mehran.money = 999999;
    }
}
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 mehran = new BankAccount();
        chris.name = "Chris";
        chris.money = 100;
        mehran.name = "Mehran";
        mehran.money = 999999;
    }
}
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 mehran = new BankAccount();
        chris.name = "Chris";
        chris.money = 100;
        mehran.name = "Mehran";
        mehran.money = 999999;
    }
}
```java
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 mehran = new BankAccount();
        chris.name = "Chris";
        chris.money = 100;
        mehran.name = "Mehran";
        mehran.money = 999999;
    }
}
```
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 mehran = new BankAccount();
        chris.name = "Chris";
        chris.money = 100;
        mehran.name = "Mehran";
        mehran.money = 999999;
    }
}
**Classes: Take 1**

```java
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 mehran = new BankAccount();
        chris.name = "Chris";
        chris.money = 100;
        mehran.name = "Mehran";
        mehran.money = 999999;
    }
}
```

[Diagram showing the instantiation and attribute setting of BankAccount instances chris and mehran]
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 mehran = new BankAccount();
        chris.name = "Chris";
        chris.money = 100;
        mehran.name = "Mehran";
        mehran.money = 999999;
    }
}
What is a class?
A class defines a new variable type
• **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.
This goes in its own file!

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

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

Piech, CS106A, Stanford University
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;
    }
}
Venmo

The easiest way to pay your friends.
You must define three things

1. What **variables** does each instance store?

2. What **methods** can you call on an instance?

3. What happens when you make a **new** one?
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 is a class?
A class defines a new variable type
Bouncing Balls
1. What **variables** does each instance store?
   - Each ball has its own Goval (let's call it shape)
   - Each ball has its own dx
   - Each ball has its own dy

2. What **methods** can you call on an instance?
   - `heartbeat();`
   - `getShape();`

3. What happens when you make a **new** one?
   - Sets initial values for all the "instance" vars

*details on how to define these three things coming soon*
What classes?
What classes?