CS 121 - Intro to Programming: Java - Lecture 3

Announcements

Introductory Survey - Please do it!
Ch 1,2 Embedded problems due Wed, noon
First programming assignment on OWL, due Wed at 5
Ch 1,2 OWL hwk due Friday at 5
MyOffice hours posted, Some TA hours posted
http://twiki-edlab.cs.umass.edu/bin/view/Moll121/WebHome

Check CourseWork link at left for due dates
A typical Java program
A very, very simple application

Class G

main
public class Howto{
    // a baby intro example
    public static void main(String args[]) {
        System.out.println("Welcome to 121");
        System.out.println("3 + 5");
        System.out.println(3 + 5);
    }
}

Prints:
Welcome to 121
3 + 5
8
Errors

- Compile-time errors - syntax, type errors
- Run-time errors - divide by 0
- Logic errors - Everything works fine - get wrong answer

System.out.println(5 + 3;  
System.out.println(5/0);  
System.out.println("area = " + " " + (3 * radius));
How + works

3 + 5 → 8

“three” + “five” → threefive

“three” + 5 → three5

“three “ + 5 → three 5

“three” + (5 + 4) → three9

“three” + 5 + 4 → three54

5 + “three” + 4 → 5three4
Java’s Object Model

This subject will occupy us for a good deal of the next month or so!
The Object Model

We model “things” as objects

Objects have attributes, and behaviors

Trip (attributes: start, end, days, distance)
  (behaviors: getDays, setDays, distPerDay ..)

Horse (attributes: name, breed, age, height)
  (behaviors: getAge, setAge ..)

Car

House

Student

Tree
Where are all of these characteristics written down?

Answer: In a class definition

A class definition for an object is different from an object, in the same way that the blueprint for a house is different from a house.

Still -

How do you make an object?

How do you invoke its behaviors?
Infant objects

Attributes

name

age (in months)

Behaviors

getName

getAge

anotherMonth (make kid one month older)
Objects have **state**

The state of an object is the **value** of its attributes

<table>
<thead>
<tr>
<th>Infant</th>
<th>Car</th>
<th>Horse</th>
</tr>
</thead>
<tbody>
<tr>
<td>name: Jill</td>
<td></td>
<td></td>
</tr>
<tr>
<td>age: 11</td>
<td></td>
<td></td>
</tr>
<tr>
<td>make: Ford</td>
<td></td>
<td></td>
</tr>
<tr>
<td>miles: 8875</td>
<td></td>
<td></td>
</tr>
<tr>
<td>HP: 190</td>
<td></td>
<td></td>
</tr>
<tr>
<td>name: Zippy</td>
<td></td>
<td></td>
</tr>
<tr>
<td>age: 4</td>
<td></td>
<td></td>
</tr>
<tr>
<td>breed: Appa</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
public class InfantTester{

public static void main (String[] args){
    Infant myKid = new Infant("Kit",4);
    System.out.println("name: " +
                         myKid.getName());
    myKid.anotherMonth();
    myKid.anotherMonth();
    System.out.println("my kid is now " +
                         myKid.getAge());
}
}
What happens here?

Perspectives for understanding..

• Object model level
• Statement level
• Flow of Control
• Code development level
Broadly, two kinds of behaviors:

**Get behaviors** -
- How old is the kid
- What’s the kid’s name
- Get behaviors do NOT change the calling object

**Mutating behaviors** - Alter calling object
- Name change
- Age change

When a mutator does what it does, the state of the calling object changes.
public class Infant {  // the Infant class definition

    private String name;
    private int age;  // in months

    public Infant(String who, int months) {
        name = who;
        age = months;
    }

    public String getName() { return name; }

    public int getAge() { return age; }

    public void anotherMonth() { age = age + 1; }
}
public class Infant{

    private String name;
    private int age;    // in months

    public Infant(String who, int months){
        name = who;
        age = months;
    }

    public String getName(){return name;}

    public int getAge(){return age;}

    public void anotherMonth(){age = age + 1;}
}

Attributes

Constructor
A crucial concept - flow of control

- Statement level flow
- Class level flow
public class Infant{
    priv String name;
    priv int age;
    public Infant(...){
        name = ..
        age = ..
    }
    pub int getName(){
        return name;
    }
}
Source Code - text files: Infant.java, InfantTester.java

Object Code - Infant.class, InfantTester.class - these are bytecode files.

Exactly one object created - how do we refer to it?

• Where is it?
• How was it created?
• What is its state when it was created?
• Does its state change? How?
• What is its final state?
• Do Infant, InfantTester communicate?
In Java all data is tagged (that is, typed)

    int number;
    double v;
    Infant kid;
Generally there are two broad kinds of data: objects (roughly, things - defined by classes) primitives (numbers, true/false values etc.)

Variables are associated with data

For primitives anyway, there is a cell model for variables and their values;

```plaintext
int num = 6;
```

```
num
```

```
6
```
Assignment Statements and Identifiers

An identifier is the name of a variable (or method, or class..)

```java
int number = 7; // number now “holds” 7
number = 4; // number now “holds” 4
number = number + 2; // number now “holds” 6
```

Assignment is NOT equality!

Assignment is an **action** operator: **Compute** the RHS,
Then **copy** the result to variable named on the LHS
For Thursday:

Plenty to do: Ch 1,2 OWL hwk, Program 1
I will lecture continue with Ch 2 material
I’ll post some more work to do later today
Trapdoors in the textbook