

Instance Variables and Container Classes

August 29, 2006

Instance Variables

- The classes we looked at last week had no instance variables or data stored with them – they only had methods.
- Normally a class will define some instance variables for holding information.
- The variables must have a type associated with them. Common types are:
 - float Holds a number with a decimal portion
 - int Holds a number without decimal places
 - String Holds a group of characters
 - boolean Holds a true/false value.

Defining a Class With Instance Variables

```
public class BaseballTeam {
    private String teamName;
    private int    numberOfWins, numberOfLosses;
    private float  winningPercentage;

    public BaseballTeam ( ) {
        teamName = "Toledo Mud Hens";
        numberOfWins = 0;
        numberOfLosses = 0;
        winningPercentage = 0.0;
    }
}
```

We Really Need More In Our Class

- Most classes have at least a method to access the instance variables. These routines are known as accessors or get methods.
- Sometimes we want to enable other classes to change the instance variables. In this case we have mutators or set methods.

Continuing with the BaseballTeam

Access methods are usually public. **Class -- accessors**

```
public int getNumberOfWins ( ) {
    return numberOfWins;
}

public int getNumberOfLosses ( ) {
    return numberOfLosses;
}

public String getTeamName ( ) {
    return teamName;
}

public float getWinningPercentage ( ) {
    return winningPercentage;
}
```

Normally the name of the method is the word get followed by the instance variable name. (But it doesn't HAVE to be)

The return type should match the variable type.

Let's add Mutators for numberOfWins and numberOfLosses

```
public void setNumberOfWins (int wins) {
    numberOfWins = wins;
    winningPercentage = (float) wins / (float)(wins + losses);
}

public void setNumberOfLosses (int losses) {
    numberOfLosses = losses;
    winningPercentage = (float) wins / (float)(wins + losses);
}
```

This is a cast – it tells the computer to treat wins as if it were a value with decimal places rather than as an int.

We will add 2 more routines...

- Two other routines that are common are a "complete" constructor which takes a value for each of the instance variables and a "toString" function that sets up a String to be printed for the object.
- We will do the constructor first since we already know about those.

The Second Constructor

```
public BaseballTeam (String name, int wins,
                    int losses,
                    float percentage) {
    teamName = name;
    numberOfWins = wins;
    numberOfLosses = losses;
    winningPercentage = percentage;
}
```



I've avoided using the same name for the arguments and the instance variables – That would be confusing.

The toString Method

```
public String toString ( ) {
    return teamName + " " +
        numberOfWins + " wins" + " " +
        numberOfLosses + " losses" +
        " " + winningPercentage;
}
```

The "+" operator adds numbers but it will also tack strings of characters together.

A Main Program to Test Our Class

```
public static void main (String arguments[ ]) {
    BaseballTeam one, two;
    one = new BaseballTeam( );
    two = new BaseballTeam("Durham Bulls", 59,
                          76, 0.437);
    System.out.println(one);
    System.out.println(two);
    System.out.println(two.getTeamName( ));
    System.out.println(two.getNumberOfWins( ));
    System.out.println(two.getNumberOfLosses( ));
    System.out.println(two.getWinningPercentage( ));
    one.setNumberOfWins( 83 );
    one.setNumberOfLosses( 53 );
    System.out.println( one );
    System.exit( 0);
}
```

These 2 statements are here to test the constructors.

Print out the items to check if it worked.

Test the accessors

Details

- It is considered good practice to limit access to instance variables as much as is practical. Think of it as your room – do you want anybody to be able to do anything they want to it?
- Variable names do have some definite naming rules. However, the computer could care less what names you select. (I definitely do care though!)

More Details

- Accessor and Mutator functions (gets and sets) are normally done with the word set or get and the Variable name. This is also a convention but should be followed.
- I've left out comments on my slides. I DO **NOT** consider them optional. The version of the code available on the web page includes the comments.

Variable Naming Rules

- A variable name must start with a letter (a-z and A-Z), an underscore (_) or a currency symbol (\$).
- Can be followed by any number of letters, digits (0-9), underscores, and currency symbols.
- By convention – avoid currency symbols – they are used by the compiler for special purposes.
- The variable name must not be one of Java's reserved words.

Java's Reserved Words

abstract	default	if	package	synchronized
assert	do	implements	private	this
boolean	double	import	protected	throw
break	else	instanceof	public	throws
byte	extends	int	return	transient
case	false	interface	short	true
catch	final	long	static	try
char	finally	native	strictfp	void
class	float	new	super	volatile
const	for	null	switch	while
continue	goto			

What is a Container Class

- A container class, in its simplest form, is a class that holds something. What we have just made is a container class to hold some information on a baseball team.
- Why Bother?
 - It makes good sense to group all the information about an OBJECT together into the class and to have specific methods to access the instance variables.

This Week's Lab

- This week's lab will run you through the process of building a Java program in the eclipse environment.
- We are going to use the Sun lab but anything we will do can also be done in a PC lab. It's one of the nice things about Java – it is platform independent.

Disks With Software

- I've got some disks with the software available. I'll burn more if we need them.
- The software on the disks includes:
 - The java development kit (available from java.sun.com)
 - The eclipse environment (available from www.eclipse.org)
 - The Java API documentation (available from java.sun.com)