CSE8B Discussion

Wednesday, May 1, 2013
Primitive Data Types vs Reference Data Types

RECAP

Primitive DataTypes
- Included in the language
- Reserved keywords
- Do not need "new"
- Initialized to a default value automatically when declared

Reference Data Types
- Created using defined constructors of the classes. They are used to access objects.
- Class objects, and various type of array variables come under reference data type.
- Default value of any reference variable is null.

What about String?
- Array of chars
- Immutable
Primitive Data Types vs Reference Data Types

Passing Primitive Data Type Arguments

Primitive arguments, such as an int or a double, are passed into methods by value. This means that any changes to the values of the parameters exist only within the scope of the method. When the method returns, the parameters are gone and any changes to them are lost.

Passing Reference Data Type Arguments

Reference data type parameters, such as objects, are also passed into methods by value. This means that when the method returns, the passed-in reference still references the same object as before. However, the values of the object's fields can be changed in the method, if they have the proper access level.
Primitive Data Types vs Reference Data Types

LET'S TRY IT!

PassReferenceType.java
PassPrimitiveType.java

DRAW MEMORY MODEL!!!
Primitive Data Types vs Reference Data Types

Assignments

Primitive Data Types: Value get copied

Reference Data Types: Reference to the object is copied not the values inside of the object
Class Variables vs Local Variables

- Member variables in a class—these are called *fields*.
- Variables in a method or block of code—these are called *local variables*.
- Variables in method declarations—these are called *parameters*

```java
class myClass{
    int field;
    int memberVariable;

    public void method(int parameter, int localVariableParameter)
    {
        int localVariable;
    
    }
}
```

- Shadow (see example)
Class design

Real World Example

Manager:
Design a class that represents a cell phone.
I want to know the manufacturer of the cell phone.
The level of the battery and if it is currently charging or not.
When instantiate a cell phone object battery should be 0 and it should be not charging.
Access Levels

<table>
<thead>
<tr>
<th>Modifier</th>
<th>Class</th>
<th>Package</th>
<th>Subclass</th>
<th>World</th>
</tr>
</thead>
<tbody>
<tr>
<td>public</td>
<td>Y</td>
<td>Y</td>
<td>Y</td>
<td>Y</td>
</tr>
<tr>
<td>protected</td>
<td>Y</td>
<td>Y</td>
<td>Y</td>
<td>N</td>
</tr>
<tr>
<td>no modifier</td>
<td>Y</td>
<td>Y</td>
<td>N</td>
<td>N</td>
</tr>
<tr>
<td>private</td>
<td>Y</td>
<td>N</td>
<td>N</td>
<td>N</td>
</tr>
</tbody>
</table>
Manager:

I don't want any other developer to be able to change the properties of the cell phone unless they do it using our provided methods. Create a way to modify the battery value and set charging state. Now I want to have a list with the names of all the applications installed in a cell phone. By default each cell phone cannot have more than 10 applications installed.
Inheritance
public class <class name> extends <base class>

public class People => public class People extends Object

public class Engineer extends Employee