Application Design Using Java

Lecture 09

Classes

- Java uses a fully dynamic object model (all objects are in heap memory)
- Everything is inside classes
- Class (static) and instance members
- Final members
- Initialization
  - Implicit and explicit field initialization
  - Constructors
  - Initialization block
    - Object
    - Class
  - Order
    1. All data fields are initialized to their default values (0, false, or null).
    2. All field initializers and initialization blocks are executed, in the order in which they occur in the class declaration.
    3. If the first line of the constructor calls a second constructor, then the body of the second constructor is executed.
    4. The body of the constructor is executed.
- Destructors? finalize() but better use Disposable interface!

Mutable strings

Do not use String when you need to perform modifications of strings

- StringBuilder
- StringBuffer

Modifiers

- Fields, methods
  - Visible to the class only (private)
  - Visible to the world (public)
  - Visible to the package and all subclasses (protected)
  - Visible to the package—the (unfortunate) default. No modifiers are needed.
  - final
  - static
  - volatile (fields only)
  - transient (fields only)
  - abstract (methods only)
  - synchronized (methods only)
  - native (methods only)
- Classes
  - Visible to the world (public)
  - Visible to the package—the (unfortunate) default. No modifiers are needed.
  - final
  - static
  - abstract

Methods with varargs

- Variable number of parameters
- Internally represented by an array

Enumerations

- Are actually classes
- Can have constructors, fields, and methods
- Have a predefined number of objects, so implement the interning pattern
- Never need to use equals() to compare them, can use ==
- All enumerations are subclasses of the class Enum

//TODO before next lecture:

- Homework 2 due on Monday 3/1 at 11:59 pm EST. Must be submitted on Submitty.
- Homework 3 will be posted on 3/1.