CSCI 136: Data Structures & Advanced Programming

Lecture 3:

Interfaces & Invariants

Morgan McGuire

February 8, 2017

Review

- Members for state
- Methods for computation
- Constructor: special initializing computation

- **Extend** classes to specialize
- **this**, **super**
- **instanceof** and Casting
- **==** versus .equals(), floating-point equality

Today

1. Invariants
2. Protection mechanisms
3. Accessors
4. Arrays of dimensions 1 and 2
5. [Pseudo-] random numbers
6. Interfaces
7. final & static

RPG Class Hierarchy

- **Entity** *(extends Object)*
  - int x, y
- **Item** extends Entity
  - double weight
  - pickUp()

- **Club** extends Item
  - damage
  - attack()
- **Emerald** extends Item
  - int value

- **Broccoli** extends Item
  - double energy
  - bite()
- **Monster** extends Entity …

We need to keep these synchronized

Implementing the Map

- 1D Array of entities
- 2D Array of squares
  - Array of arrays
  - Wrapped 1D array
    - Row major
    - Column major
    - Indexing math
- Tradeoffs?
- **Invariants,** if we use both?

Random Object Placement

- java.util.Random is considered a joke/bug by computer scientists who work with randomized algorithms
- But it will be really convenient for our purposes
- Awesome!
- https://docs.oracle.com/javase/8/docs/api/java/util/Random.html

```
Random rng = new Random()
x = rng.nextInt(map.getWidth()); y = rng.nextInt(map.getHeight())
```

Interfaces

- Noun = state [class, member]

  Entity, position, energy

- Verb = computation [constructor, method]

  pickUp, run, swing, heal

- **Adjective = shared property of state [interface]**

  Flammable, Edible, Wearable, Dangerous, …

- Java classes may:
  - **extend** only one super **class**
  - **implement** many **interfaces**

Advanced Syntax

- **final** classes cannot be **extended** further
- **final** methods cannot be overridden in subclasses
- **final** variables are constants whose value cannot be changed
  - *Good programmers make variables final by default*
- **static** members are shared among all instances
  - Example: System.out
- **static** methods are invoked directly on the class
  - Example: Math.ceil()
- **static** inner classes do not retain a pointer to the outer class that instantiated them
  - *Good programmers do this by default*

Summary

- **Invariants**
  - Enforced by **protected** state, **accessors**, and **final**
- **Interfaces** are stateless abstractions of properties
- **java.util.Random**

Next Time

- Histogram
- Cumulative distribution function
- Methods on String
- Pre and Post-conditions

Lab #1 today & tomorrow…solution due Monday night!

Essential String Methods

- https://docs.oracle.com/javase/7/docs/api/java/lang/String.html
- int length()
- char charAt(int)
- String substring(int, int)
- Int indexOf(String, int)