CS 2334: Project 2 Objectives

By the end of this project, you should be able to:

- Extend classes to meet new requirements
- Design and use enumerated data types
- Employ *Java generics* to write type safe generic code
- Employ a generic sorting algorithm for arbitrary lists of objects

Focus for Today

- Get started on project 2
- You should understand what we want for project 2 by the end of lab (if you don't, ask questions!)
- Project 1 demos can happen today, but project 2 is the priority

Milestone 1: FinchAction Addition

int priority: priority level for an action

- we will use this for sorting purposes

Milestone 2: New Child Classes

- FinchMoveGuarded: spin wheels at specified velocity until an obstacle is detected
- FinchOrientationGuarded: wait until the Finch is placed in a particular orientation
  - Up, down, left, right, upside down and level
- FinchObstacleGuarded: wait until the Finch's path is blocked (or unblocked) by an obstacle
  - Left, Right, Either, Neither

Input Files: An Example

NOSE	seek	12	0	0	255
GMOVE	seek	15	30.	0	30.0
NOSE	seek	1	255	0	0
MOVE	seek	19	500	−10.0	−10.0
ORIENT	seek	6	level		
NOSE	seek	4	0	255	0
OBSTACLE	seek	9	leftblocked		
MOVE	seek	27	2000	20.0	−20.0
ORIENT	seek	2	beakup		
GMOVE	seek	38	30.0	30.0	
OBSTACLE	seek	13	bothunblocked		

Milestone 3: FinchAction implements Comparable

- compareTo() should order by name and **then** by priority
- Specify Comparable implementation so that only FinchActions can be compared to other FinchActions

Milestone 4: Sorting

- Must use an adapted version the generic sort method provided in the book (listing 14.10)
  - Adaptation: use the generic form of the method declaration so that the array does not consist of any Comparable objects, but those of a specific type
- FinchActionList supplies a sort() method, which calls your generic sort()
- After reading a FinchActionList from a file, it must be sorted

Driver Classes

For each milestone, implement the main method of the milestoneX class

- Implement the class as you complete one milestone and before you move on to the next one
- Test the key components of what you have implemented for the given milestone
  - In particular: constructors, mutators, accessors and toString()

Deadlines

- September 30th @5:00pm: design
- October 7th @5:00pm: final version, including demonstration
  - If all elements are completed by October 5th @5:00pm, a 5% bonus will be awarded