Today

- Math API
- More objects
- For loops
- ArrayLists

Short Questions?

Quiz

Project 1: Code Reviews

- Only conducted on code that has been submitted to D2L
- For evaluation of your project, not to help you do your project (don't waste your code review slot in this way)
- In general: if you miss your code review appointment, then you must do a walk-in review
  - For this project: we have no open code review slots left. If you missed your appointment over the last few days, then you have been assigned a grade of zero
  - For those who have not been able to sign up because all of the slots were occupied, we will free up some time Thursday and Monday to finish this off [note that we will not extend the code review deadline again]

Project 2

- Is available on the main course web page
- Due on Tuesday

Object Example I (Last Time)

Write a method that returns the past tense of a verb

- Assume regular verbs
- Verbs can end with any letter

Object Example II

Write a method that ***turns*** a verb into the past tense of itself

- Assume regular verbs
- Verbs can end with any letter
- The original object should be modified

Loop Example I

Write a method that will indicate whether all of the characters in a string are lower case

- What is the method prototype?

Loop Example I

Write a method that will indicate whether all of the characters in a string are lower case

- What is the method prototype?
- What is the method implementation?

Loop Example II

Write a method that takes two positive integer inputs and returns their greatest common divisor.

- You may assume that the first integer is greater than or equal to the second integer

Loop Example III

Write a method that takes two positive integer inputs and returns their greatest common divisor.

- There are no guarantees about the input values

ArrayList Class

We often want to represent not just one “thing”, but an ordered list of things. An ArrayList object allows us to do this:

```
ArrayList<String> list = new ArrayList<String>();
```

- ArrayLists are ***generic***: they can represent a list of any type of object
- The type in this example is ArrayList<String>: it is a list of strings

ArrayList Class

Type: `ArrayList<E>`

- Add an object of type E:

  ```
  add(E o)
  ```

  - E must match the type of the ArrayList

- Get the object at position i in the list:

  ```
  E get(int i)
  ```

  - Zero indexed!

ArrayList Example

ArrayList Problem I

Write a program that takes as input a list of integers (each integer is placed into a list). When input is complete, report the median of the integers.

ArrayList Problem I

Write a program that takes as input a list of integers (each integer is placed into a list). When input is complete, report the median of the integers.

- How is the ArrayList declared?
- How do we find the median number from a list?

ArrayList Problem II

Write a program that takes as input a list of names (each name is placed into a list). When input is complete, print out the names in alphabetical order.

Summary

- Object classes allow us to group primitive data types and other objects into some meaningful whole
- The Java libraries provide a large number of interesting and useful classes
- “Generic” classes determine their true type only at compile time. Hence, they can be used in many different ways

Wrap Up

Due:

- Project 2: Due next Tuesday
- HW 5: released today and due in one week

Next time:

- More on ArrayLists
- Nesting control structures