Programming Languages and Techniques (CIS120)

Lecture 36

April 20, 2016

Resizable Arrays

Chapter 32

Game project grading

- Final Program Due: (88 points)

  Tuesday April 26th at 11:59pm

  - Submit zipfile online, submission *only* checks if your code compiles
- Grade based on demo with your TA during reading days
  - Make sure that you test your program in Moore 100, especially if you use outside libraries
  - Grading rubric on the assignment website
  - Recommendation: don't be too ambitious.

- *NO LATE SUBMISSIONS PERMITTED*

How is the Game Project going so far?

1. not started
2. got an idea, submitted design proposal
3. it's somewhat working
4. it's mostly working
5. debugging / polishing
6. done!

Final exam

- Monday, May 9th at 9AM
  - Use form on course website if you have multiple exams that day
  - Only reason for make up
  - http://www.upenn.edu/registrar/pdf_main/provost-rules.pdf
- Old exams will be available on course website
  - Exam will cover the entire semester (through Friday's lecture)
  - More emphasis on Java part of the course
- Lab this week: final exam review

Design Exercise: ResizeableArray

Arrays that grow without bound

Step 1: Understand the problem

- Say we want to create an abstract data structure, like a Map, that contains associations from keys to values.
- Both keys and values will be ints
- The domain of the map should include *all* integers greater than or equal to 0. Each int k will be mapped to 0 by default.
- We also want to be able to find the largest key that has a nonzero value

Step 2: Define the interface

```
public class ResArray {
    /** Constructor, takes no arguments. */
    public ResArray() {    … }

    /** Access position i. If position i has not yet
     * been initialized, return 0. */
    public int get(int idx) { …   }

    /** Update index i to contain the value v. */
    public void set(int idx, int val) { … }

    /** Return the extent of the array. i.e.
    one past the index of the last nonzero value in the array. */
    public int getExtent() { … }

}
```

Step 3: Write tests

```
ResArray a = new ResArray();
a.set(17, 120);
int result = a.get(17);
```

What should be the result?

1. 0
2. 17
3. 120
4. ArrayIndexOutOfBoundsException
5. NullPointerException

```
ResArray a = new ResArray();
int result = a.get(17);
```

What should be the result?

1. 0
2. 17
3. 120
4. ArrayIndexOutOfBoundsException
5. NullPointerException

```
ResArray a = new ResArray();
a.set(17, 120);
int result = a.getExtent();
```

What should be the result?

1. 0
2. 16
3. 17
4. 18
5. 120
6. ArrayIndexOutOfBoundsException
7. NullPointerException

```
ResArray a = new ResArray();
a.set(17, 120);
a.set(17, 0);
int result = a.getExtent();
```

What should be the result?

1. 0
2. 16
3. 17
4. 18
5. 120
6. ArrayIndexOutOfBoundsException
7. NullPointerException

Demo: Steps 3 & 4

ResArray.java

ResArrayTest.java