Week 2: Invariants, Dynamic arrays, and Sequences

Friday, September 11, 2020 8:40 AM

We'll start at about 10:05, or when most get here.

Unit Tests

→ test public behaviors
→ make sure public methods return correct information
→ public-facing parts are correct
→ outside of the class
  * cannot access private info

Invariants

→ make sure data structure is correct
→ set of rules data structure must follow
→ test invariant at the beginning of every public method
→ test at end of any public method that changes the data structure
  ex: constructor
  add/remove
  clone

* catches bugs related to data that unit tests cannot catch
** always do this first & write descriptive messages for yourself

Dynamic Arrays

Specification

→ container for objects
→ (almost) no restriction on size
→ behaviors:
  → add
  → remove
  → clone

Implementation

→ data structure:
  → array
  → effective size
→ if capacity is reached, double size
  → ensure capacity

ex add:
  → invariant
  → ensure capacity
  → add element
  → increment manyItems

Invariant Rules

→ manyItems ≤ array length
  (manyItems: size; array length: capacity)
→ manyItems ≥ 0
→ array cannot be null

Sequence

Specification

- ordered collection/container
- essentially no limit on size
- * can have elements/nulls *
- behaviors:
  - add → before/after/all
  - removeCurrent
  - clone
  - getCurrent
  - size
  - start/advance
  - atEnd → after all elements

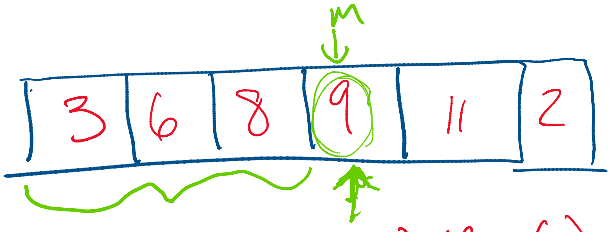
implementation

- Data Structure:
  - Dynamic array
    - array
    - manyItems
  - cursor
    - isCurrent
    - currentIndex
      - if isCurrent is false: no current

Invariant Rules

- dynamic array rules
- currentIndex ≥ 0
  - currentIndex ≤ manyItems
  - *if currentIndex == manyItems
    isCurrent must be false
    *no current element

manyItems: 2



manyItems - 3
currentIndex=2
isCurrent = true

addAfter(5)
→aftercurrent, after removed, end

addBefore(3)

removeCurrent

advance