CSE 2221 Lecture Notes

Professor Bucci

SP17

When you trace a recursive method, and you get to the recursive call, treat it like a pre-existing method. i.e. just plug in what the method contract says the method call should return/do.

Subclasses vs superclasses. The compiler can't see methods declared in the subclass.

Lecture: Tuesday, April 4th

Sequence This component family allows you to mess with strings of entries of any type T through direct access by position.
It is an alternative to an array, and is a generic container type like Queue and Set.

Standard still has clear, newInstance, and transferFrom. The sequence type also has a SequenceKernel.
The mathematical model is a string of T. You can use a sequence as a Queue with full functionality, you would prefer a Queue over a Sequence if you wanted anyone reading your code to understand that you only wanted them to modify elements at the ends.
No-argument constructor yields <>.

void add(int pos, T x) :
Adds x and position pos of this.
x is aliased!
Requires: $0 \leq pos$ and $pos \leq |this|$.

T remove(int pos) :
removes and returns the entry at position pos of this.
Updates this.
Requires: $0 \leq pos$ and $pos \leq |this|$.

T entry(int pos) :
Reports entry at pos
Aliases the reference returned by entry.
Requires: $0 \leq pos$ and $pos \leq |this|$.
ensures $< entry >= \#this[pos, pos + 1)$.

T replaceEntry(int pos, T x) :
replaces the entry at pos with x
aliases reference x.

Stack Model is a string of T's again. Queue gave us first in first out behavior. Stack exhibits LIFO last-in-first-out behavior, i.e. like a stack of pancakes, you can't mess with the bottom. No-argument constructor yields <>.

void push(T x) :
Adds x at the top (left end) of this.
Aliases x again.
Updates this.
Ensures: $this =< x > +\#this$

T pop() :
Removes and returns the entry at the top of the stack.
Require $this \neq \emptyset$.

int length() :
self-explanatory.

T top() :
Returns the entry at the top of this.
Aliases whatever it returns.
Require $this \neq \emptyset$.
Also restores the Stack.

T replaceTop(T x) :
replaces the top of x with this and returns the old top.

More Queue Let's look at the secondary method "sort" for Queue.

void sort(Comparator¡T¿) :
This interface contains a header for only one method:
int compare(T o1,T o2) When we implement the interface, we are allowed to define any ordering so long as the method returns a negative integer if the first is less than the second, positive if greater, and 0 if equal. Where less, greater, equal are defined by the method and specified by the type T being used.