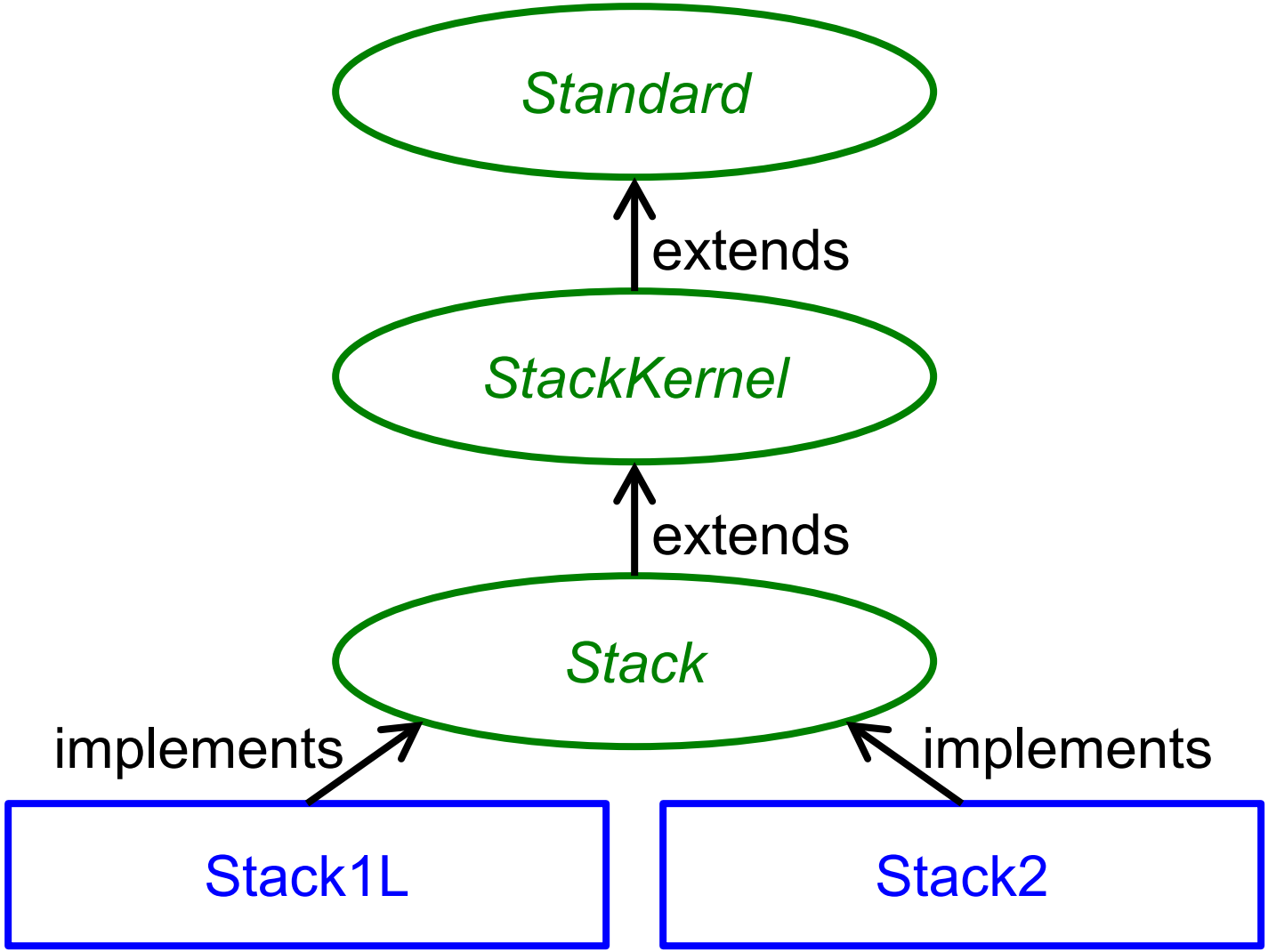
Stack

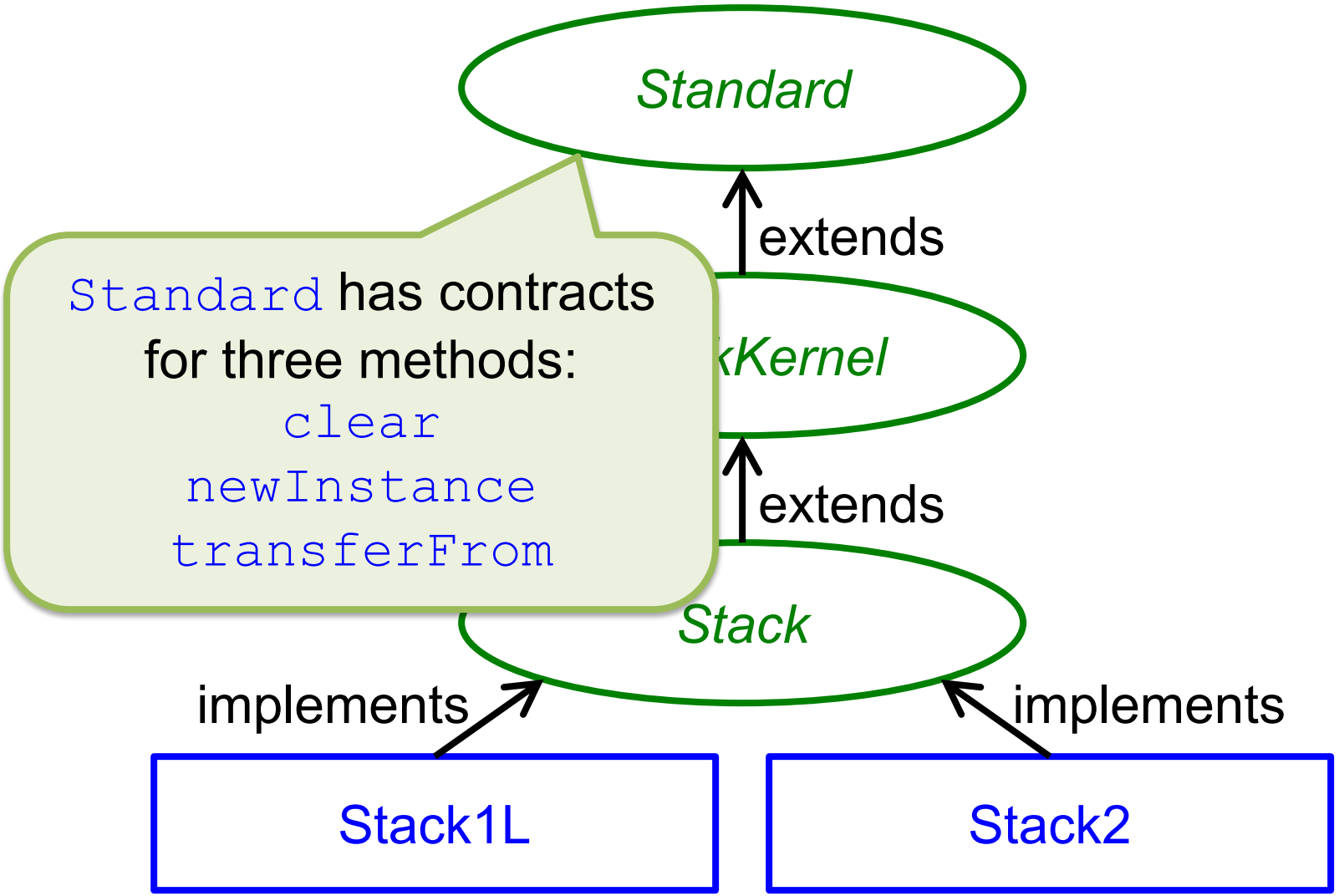
Stack

- The ***Stack*** component family allows you to manipulate strings of entries of any (arbitrary) type in ***LIFO*** (last-in-first-out) order
  - A kind of “dual” to `Queue`
  - Remember, "first" and "last" here refer to the *temporal* order in which entries are put into the string and taken out of it, not about the order in the string when it is written down

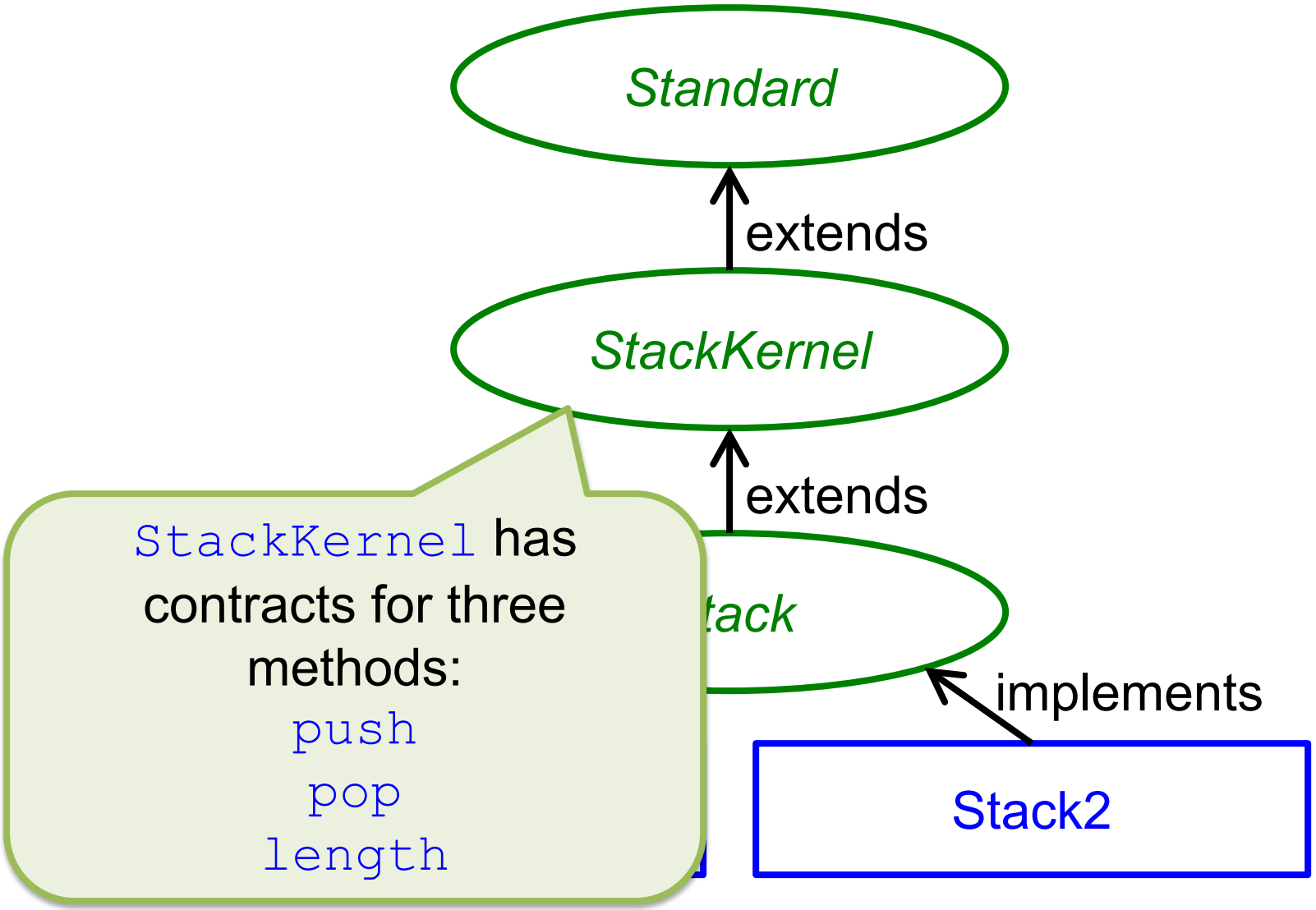
Interfaces and Classes



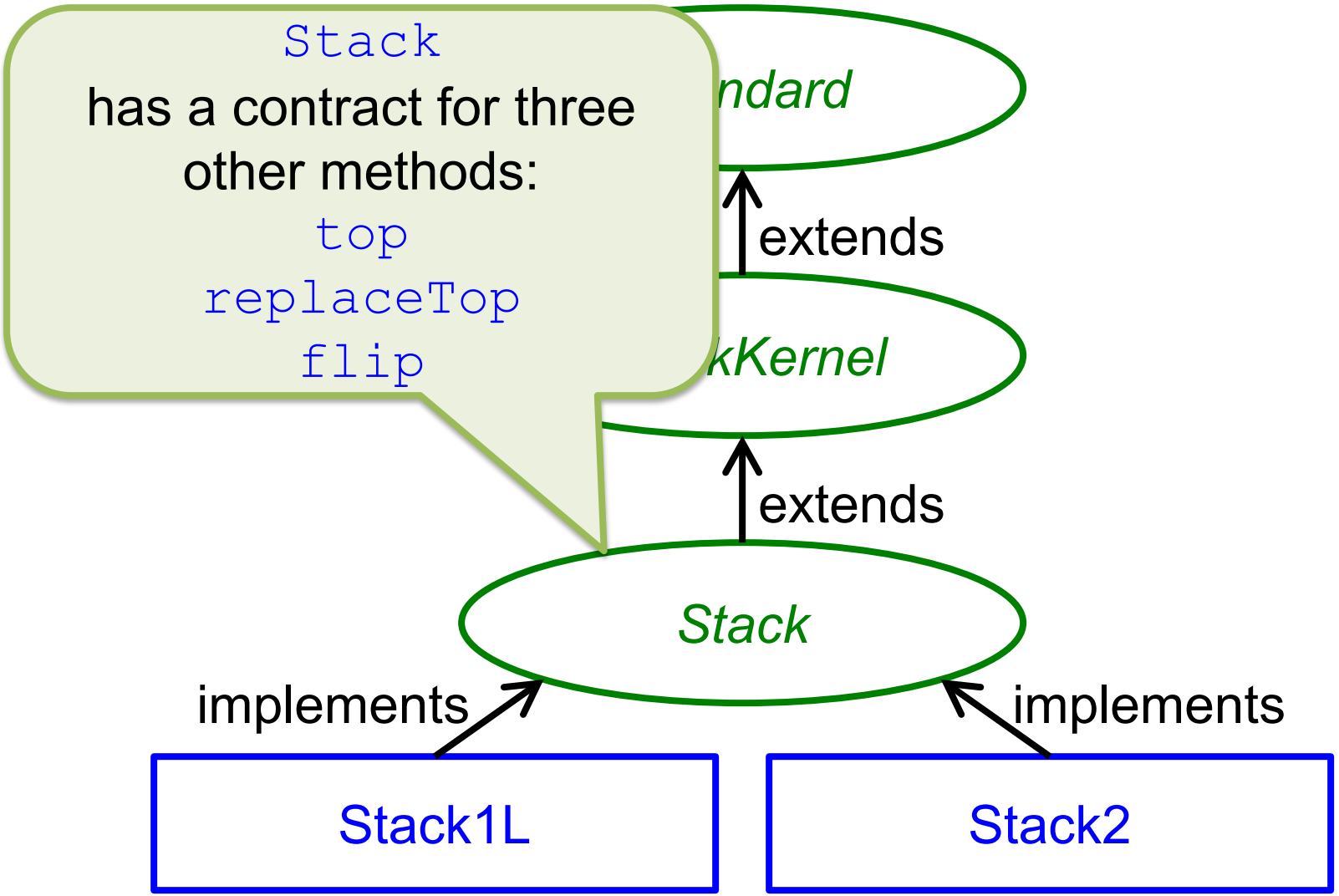
Interfaces and Classes



Interfaces and Classes



Interfaces and Classes



Mathematical Model

- The value of a `Stack` variable is modeled as a string of entries of type `T`
- Formally:

```
type Stack is modeled by
    string of T
```

No-argument Constructor

- Ensures:

  this = < >

Example

Code	State
`Stack<Integer> si = new Stack1L<>();`	

Example

Code	State
`Stack<Integer> si = new Stack1L<>();`	
	si = < >

push

void push(T x)

- Adds x at the top (left end) of **this**.
- Aliases: reference x
- Updates: **this**
- Ensures:

  this = <x> * #***this***

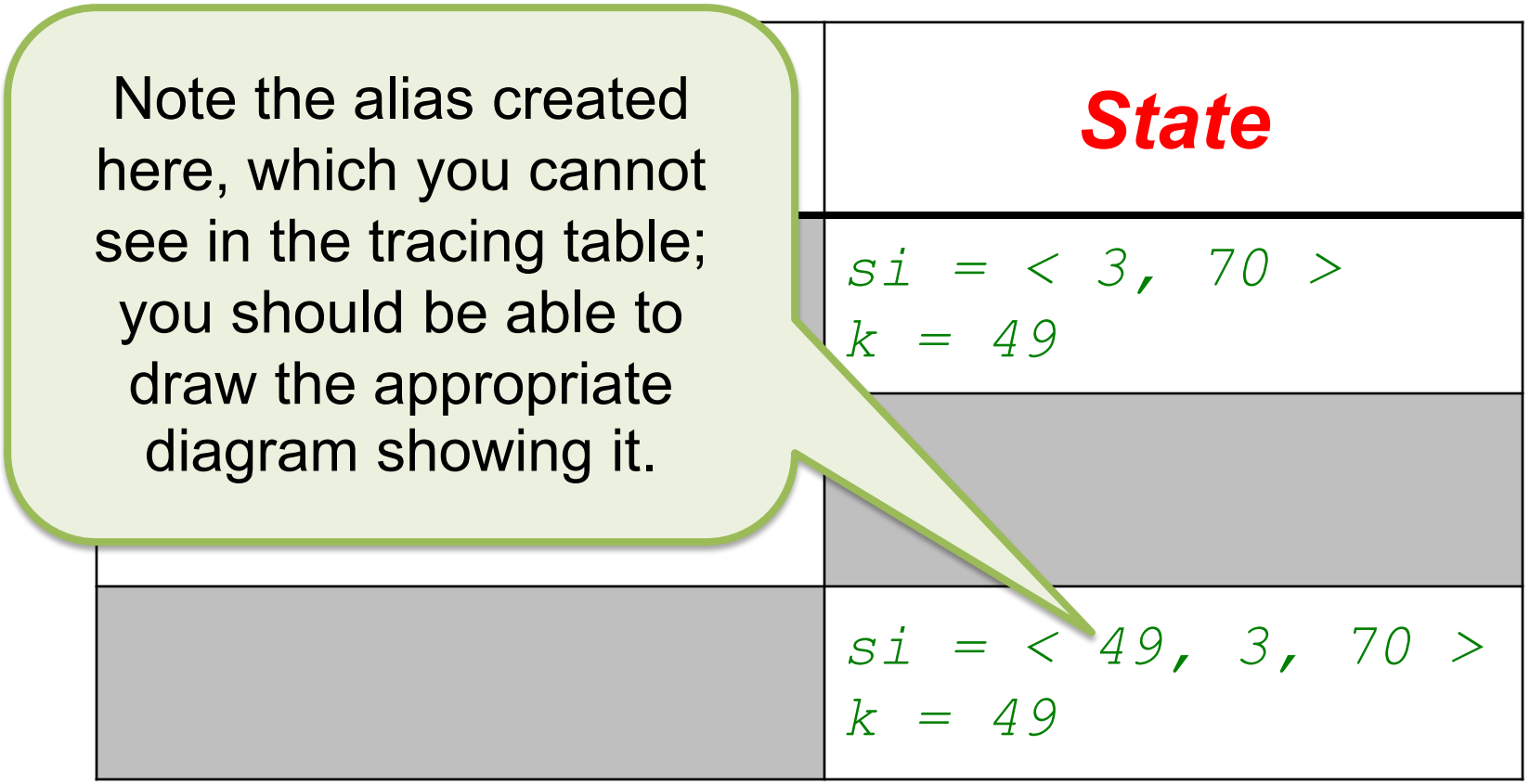
Example

Code	State
	si = < 3, 70 > *k = 49*
`si.push(k);`	

Example

Code	State
	si = < 3, 70 > *k = 49*
`si.push(k);`	
	si = < 49, 3, 70 > *k = 49*

Example


State
si = < 3, 70 >
k = 49
si = < 49, 3, 70 >
k = 49
Note the alias created here, which you cannot see in the tracing table; you should be able to draw the appropriate diagram showing it.

pop

`T pop()`

- Removes and returns the entry at the top (left end) of **`this`**.
- Updates: **`this`**
- Requires:

  this `/= < >`
- Ensures:

  #this `= <pop> *` ***this***

Example

Code	State
	si = < 49, 3, 70 > *z = -584*
`z = si.pop();`	

Example

Code	State
	si = < 49, 3, 70 > *z = -584*
`z = si.pop();`	
	si = < 3, 70 > *z = 49*

`length`

`int` `length()`

- Reports the length of **`this`**.
- Ensures:

  `length` = |***`this`***|

top

`T top()`

- Returns the entry at the the top (left end) of **`this`**.
- Aliases: reference returned by `top`
- Requires:

  this `/= < >`
- Ensures:

  <top> ***is prefix of this***

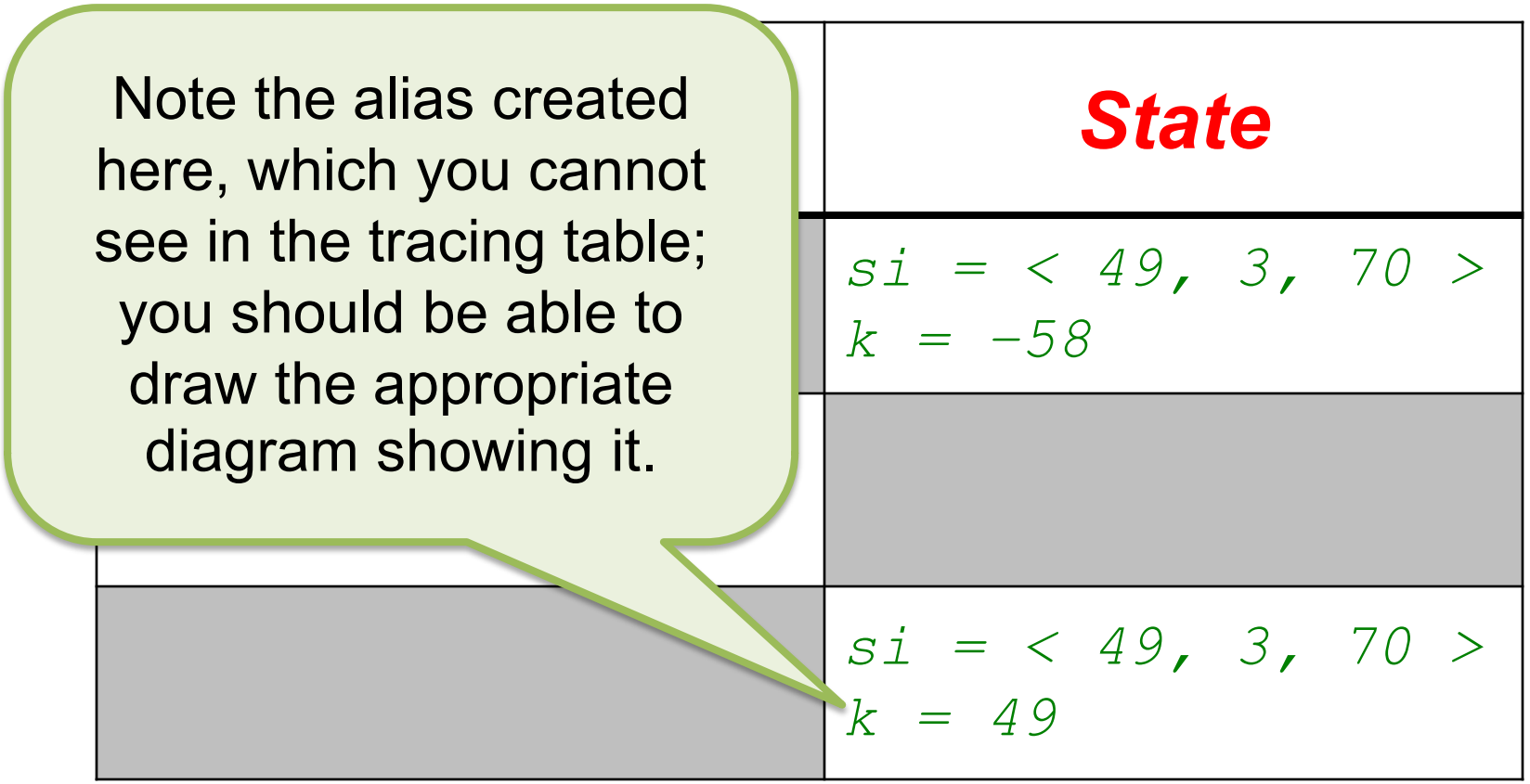
Example

Code	State
	si = < 49, 3, 70 > *k = -58*
`k = si.top();`	

Example

Code	State
	si = < 49, 3, 70 > *k = -58*
`k = si.top();`	
	si = < 49, 3, 70 > *k = 49*

Example


State
si = < 49, 3, 70 >
k = -58
si = < 49, 3, 70 >
k = 49
Note the alias created here, which you cannot see in the tracing table; you should be able to draw the appropriate diagram showing it.

replaceTop

`T replaceTop(T x)`

- Replaces the top of **`this`** with `x`, and returns the old top.
- Aliases: reference `x`
- Updates: **`this`**
- Requires:

  this */= < >*
- Ensures:

  <replaceTop> ***is prefix of #this and***

  this *= <x> ** ***#this****[1, |****#this****|)*

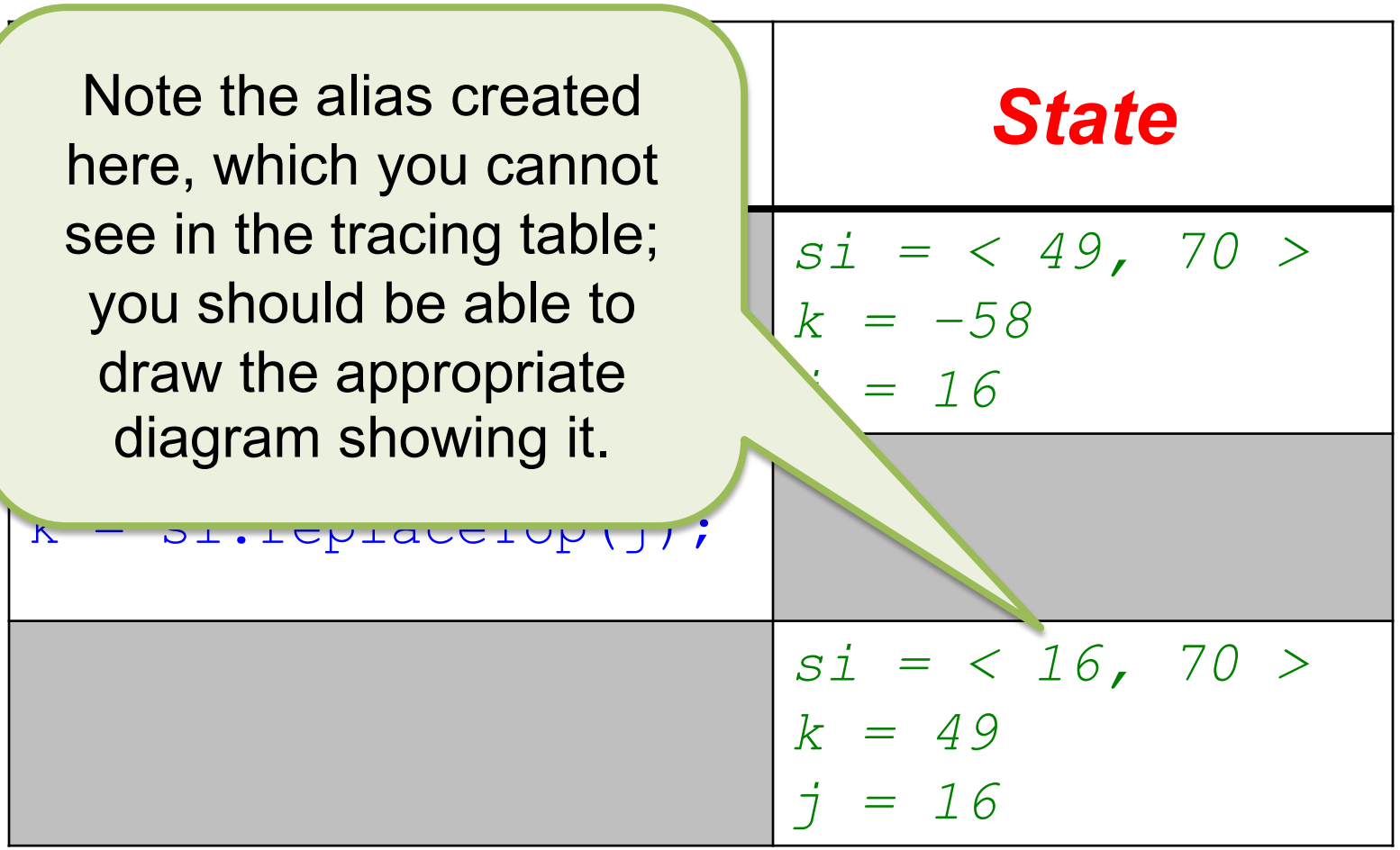
Example

Code	State
	si = < 49, 70 > *k = -58* *j = 16*
`k = si.replaceTop(j);`	

Example

Code	State
	si = < 49, 70 > *k = -58* *j = 16*
`k = si.replaceTop(j);`	
	si = < 16, 70 > *k = 49* *j = 16*

Example


State
si = < 49, 70 >
k = -58
= 16
si = < 16, 70 >
k = 49
j = 16
Note the alias created here, which you cannot see in the tracing table; you should be able to draw the appropriate diagram showing it.

Another Example

Code	State
	si = < 49, 70 > *j = 16*
`j = si.replaceTop(j);`	

Another Example

Code	State
	si = < 49, 70 > *j = 16*
`j = si.replaceTop(j);`	
	si = < 16, 70 > *j = 49*

Another Example

This use of the method avoids creating an alias: it ***swaps*** `j` with the entry previously at the top.

	State
	si = < 49, 70 > *j = 16*
`j = si.replaceTop(j);`	
	si = < 16, 70 > *j = 49*

flip

void flip()

- Reverses ("flips") **this**.
- Updates: **this**
- Ensures:

this = ***rev***(#***this***)

Example

Code	State
	s1 = < 18, 6, 74 >
`s1.flip();`	

Example

Code	State
	s1 = < 18, 6, 74 >
`s1.flip();`	
	s1 = < 74, 6, 18 >

Resources

- OSU CSE Components API: `Stack`
  - http://web.cse.ohio-state.edu/software/common/doc/