

Mathmodels contracts

ABSTRACT-STACK[G] *

```
feature -- model
  model: SEQ[G] -- abstraction function

feature -- queries
  top: G
  require ¬empty
  ensure Result = model[1]

  count: INTEGER
  ensure Result = model.count

  empty: BOOLEAN
  ensure Result ≡ (count = 0)

feature -- commands
  pop(x:G)
  require ¬empty
  ensure model = old model.tail

  push(x:G)
  ensure model = x ◁ (old model)

invariant
  count >= 0
```

executable

implementation

MY_STACK[G] +

```
feature {NONE}
  imp: LIST[G]
  make ...

feature -- model
  model: SEQ [G] -- abstraction function
  do
    create Result.make_empty
    from imp.start
    until imp.after
    loop
      Result.append (implementation.item)
      imp.forth
    end
  ensure  $\forall i \in 1..count: \text{Result}[i] = \text{imp}[i]$ 
  end

feature -- commands
  push(x:G)
  do
    imp.extend(x)
    count := count + 1
    item := x
  ensure -- inherit mathmodels contracts
  end
  ...
invariant
  count >= 0
  model.count = count
```