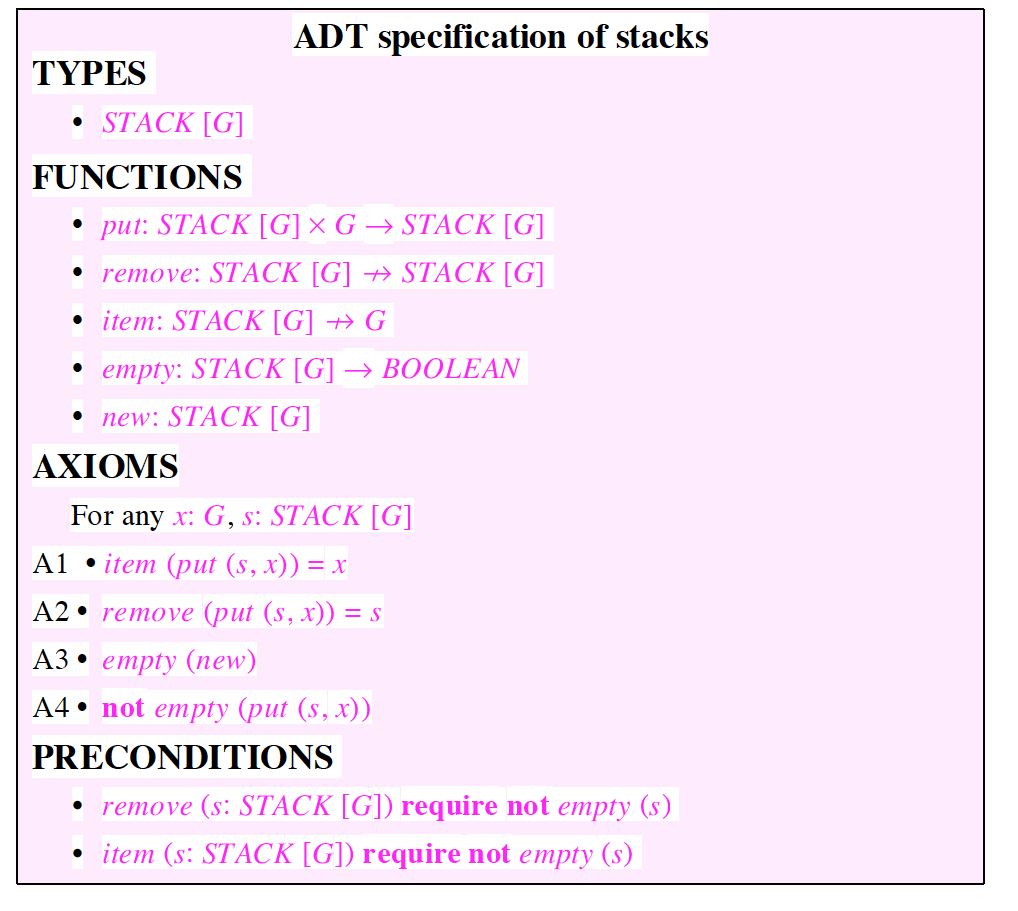
Stack ADT using a model SEQ[G]

Class SEQ[G] is part of library **mathmodels**.

In OOSC2, the ADT specification of a stack is given as follows:



We will use SEQ[G] as a mathematical model for an abstract stack class, allowing not only classical contracts but also complete contracts.

# Chart View of SEQ[G]

|  |
| --- |
| **class**  SEQ [G -> **attached** ANY]  **General**  cluster: mathmodels  description:   "(1) Model contracts for finite sequences for elements of type G.  (2) Queries are side effect free and return sequences via   deep\_twin; thus inefficient (use only for contracts).  (3) Commands change the current sequence.   (4) A valid index is 1..count.  (5) Array notation can be used, as well as iteration (across).  (6) Empty sequences can be created, or creation can be from an array.   (7) Sequences have a first item (the head), a tail and last item.  (8) Infix notation for prepended\_by(x:G):  seq1 |< x  (9) Infix notation for appended\_by(x:G):  seq1 |> x  (10) For concatenation may use infix: seq1 |++| seq2  (11) For queries, to assert that the state is not changed,   the postcondition is  Current ~ old Current.deep\_twin"  create: make\_empty, make\_from\_array  **Ancestors**  DEBUG\_OUTPUT\*  ITERABLE\* [G]  **Queries**  appended **alias** "|->" (v: G): [**like** **Current**] **attached** SEQ [G]  as\_array: ARRAY [G]  as\_function: FUN [INTEGER\_32, G]  comprehension **alias** "|" (exp: PREDICATE [PAIR [INTEGER\_32, G]]):  [**like** **Current**] **attached** SEQ [G]  concatenated **alias** "|++|" (other: [**like** **Current**] **attached** SEQ [G]):  [**like** **Current**] **attached** SEQ [G]  count **alias** "#": INTEGER\_32  debug\_output: STRING\_8  first: G  front: [**like** **Current**] **attached** SEQ [G]  has (v: G): BOOLEAN  hold\_count (exp: PREDICATE [PAIR [INTEGER\_32, G]]): INTEGER\_32  inserted (v: G; i: INTEGER\_32): [**like** **Current**] **attached** SEQ [G]  is\_contiguous\_subseq\_of (other: [**like** **Current**] **attached** SEQ [G]): BOOLEAN  is\_empty: BOOLEAN  is\_equal (other: [**like** **Current**] **attached** SEQ [G]): BOOLEAN  is\_subsequence\_of **alias** "|<:" (other: [**like** **Current**] **attached** SEQ [G]): BOOLEAN  item **alias** "[]" (i: INTEGER\_32): G  last: G  lower: INTEGER\_32  new\_cursor: ITERATION\_CURSOR [G]  out: STRING\_8  overriden (v: G; i: INTEGER\_32): [**like** **Current**] **attached** SEQ [G]  prepended **alias** "|<-" (v: G): [**like** **Current**] **attached** SEQ [G]  removed (i: INTEGER\_32): [**like** **Current**] **attached** SEQ [G]  reversed: [**like** **Current**] **attached** SEQ [G]  slice (a\_start, a\_end, a\_step: INTEGER\_32): [**like** **Current**] **attached** SEQ [G]  subsequenced (i, j: INTEGER\_32): [**like** **Current**] **attached** SEQ [G]  tail: [**like** **Current**] **attached** SEQ [G]  twin2: [**like** **Current**] **attached** SEQ [G]  upper: INTEGER\_32  valid\_position (pos: INTEGER\_32): BOOLEAN  **Commands**  append (v: G)  concatenate (other: [**like** **Current**] **attached** SEQ [G])  insert (v: G; i: INTEGER\_32)  make\_empty  make\_from\_array (a: ARRAY [G])  override (v: G; i: INTEGER\_32)  prepend (v: G)  remove (i: INTEGER\_32)  reverse  subsequence (i, j: INTEGER\_32)  **Constraints**  value semantics  value semantics  value semantics  value semantics  properties:  **not** is\_empty **implies**  **Current** ~ tail**.**prepended (first) **and** **Current** ~ front**.**appended (last) |

# ADT-STACK

|  |
| --- |
| **note**  description: "[  Abstract Data Type for a Stack, with value semantics.  Classic contracts vs. Complete contracts with a model.  ]"  author: "JSO"  **deferred** **class**   ADT\_STACK [G -> **attached** ANY]  **inherit**  ANY  **undefine**  is\_equal  **end**  **feature** -- model   model: SEQ [G]  -- abstract mathematical description of stack  -- abstraction function  **deferred**  **end**   **feature** -- queries   count: INTEGER\_32  -- number of items in stack  **deferred**  **ensure**  complete: **Result** = model**.**count  **end**   is\_empty: BOOLEAN  -- is the queue empty?  **deferred**  **ensure**  complete: **Result** = model**.**is\_empty  **end**   item: G  -- Top element of stack ("peek")  **require**  **not** is\_empty  **deferred**  **ensure**  complete: **Result** ~ model**.**last  **end**  is\_equal (other: **like** **Current**): BOOLEAN  -- Is `other' attached to an object considered  -- equal to current object?  **deferred**  **ensure then**  complete: model ~ other**.**model  **end**   **feature** -- commands   make\_empty  -- Initialization for `Current'.  **deferred**  **ensure**  classic: is\_empty  **end**   put (x: G)  -- push 'x' on top of stack ("push")  **deferred**  **ensure**  classic: count = **old** count + 1  classic: item ~ x  complete: model ~ ((**old** model) |-> x)  **end**   remove  -- pop off top of stack ("pop")  **require**  **not** is\_empty  **deferred**  **ensure**  classic: count = **old** count - 1  complete: (**old** model) ~ (model |-> (**old** item))  **end**   **feature** -- axioms   axiom (x: G)  **do**  **Current.**put (x)  **Current.**remove  **ensure**  model ~ (**old** model)  **end**   **end** -- class ADT\_STACK |

# MY\_STACK text

|  |
| --- |
| **class**   MY\_STACK [G -> **attached** ANY]  **inherit**  ADT\_STACK [G]  **create**   make\_empty  **feature** {ADT\_STACK} -- Initialization   imp: LINKED\_LIST [G]  -- implementation   make\_empty  -- Initialization for `Current'.  **do**  **create** imp**.**make  imp**.**compare\_objects  **end**   **feature** -- model   model: SEQ [G]  -- abstract mathematical description of stack  -- abstraction function  **do**  **create** **Result.**make\_empty  **across**  imp **as** cr  **loop**  **Result.**append (cr**.**item)  **end**  **end**   **feature** -- queries   count: INTEGER\_32  -- number of items in stack  **do**  **Result** := imp**.**count  **end**   is\_empty: BOOLEAN  -- is the queue empty?  **do**  **Result** := imp**.**is\_empty  **end**   item: G  -- Top element of stack ("peek")  **do**  **Result** := imp**.**last  **end**   is\_equal (other: **like** **Current**): BOOLEAN  -- Is `other' attached to an object considered  -- equal to current object?  **do**  **Result** := imp ~ other**.**imp  **end**  **feature** -- command   put (x: G)  -- push 'x' on top of stack ("push")  **do**  imp**.**extend (x)  **end**   remove  -- pop off top of stack ("pop")  **do**  imp**.**go\_i\_th (imp**.**count)  imp**.**remove  **end**   **end** -- class MY\_STACK  -- Generated by ISE Eiffel --  -- For more details: http://www.eiffel.com -- |