- Class definition 26.03.20

class name $< T_1, \dots, T_r >$ extends $t_1, \dots, t_s$ {

var field$_1$: type$_1$;

...

var field$_n$: type$_n$;

function / method / constructor declarations

}

declared as traits

- Creating objects

o := new C // no constructor

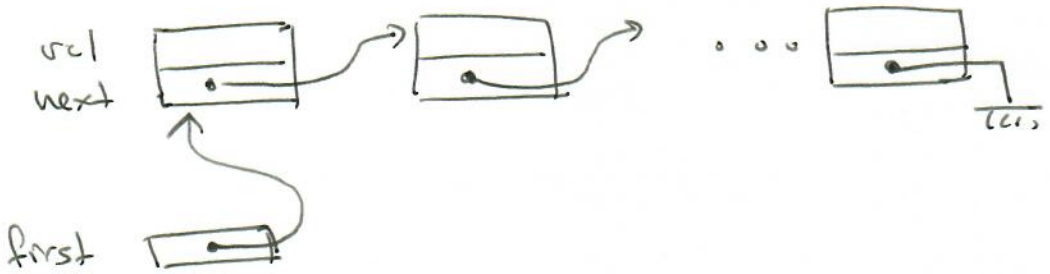
o := new C $(a_1, \dots, a_m)$ // anonymous constructor

o := new C.name $(a_1, \dots, a_m)$ // named constructor

- frame theory
  - a method may only write on the part of the heap defined in a modifies clause → its footprint
  - a method may create fresh objects. These may be declared in its postcondition

    ensures fresh $(o_1, \dots, o_r)$
  - a function must declare in a reads clause the part of the heap its result depends on
  - expression old(e) in a postcondition refers to 'e' evaluated in the heap state before calling the method

- a linked list



- recommended methodology for verifying object oriented programs

  - explicitly define the heap footprint of an object

    ghost var repr : set<object>

  - explicitly define the invariant of the footprint

    predicate Valid()
    reads this, repr

  - include Valid() in
    - postcondition of each class constructor
    - precondition of each class function
    - pre- and post- condition of each class method

  - include an abstract model of the class and use it in the specifications

    function model() : t
    requires Valid()
    reads repr