1) A good O-O design would include a deferred class FIGURE or something similar, with subclasses RECTANGLE and LINE_SEGMENT. SQUARE should inherit from RECTANGLE. In FIGURE deferred features `move (p: POINT)` and `name: STRING` could be declared. Strict command-query separation should be used. In class APPLICATION, feature `make` should be altered to call `move` and `name` on the FIGUREs. Feature `move_and_get_name` should be removed. The resulting code should include contracts, e.g. RECTANGLE’s `move` could ensure `upper_left.x = old upper_left.x + p.x`. The principles applied include abstraction (to form the FIGURE hierarchy) and command-query separation. The language mechanisms facilitating the solution include inheritance, Design by Contract, polymorphic attachment and dynamic binding.

2) Class EVENT_CHANNEL [G] should maintain a private list of subscribers of type `LINKED_LIST[PROCEDURE[ANY,G]]`. The `subscribe` procedure should extend this list, and `publish` should traverse it and `call` every element with a one-element TUPLE as argument.

3) C.f
   A.g
   A.f
   A.f
   C.f