

## DESIGN BY CONTRACT

### • DOCTRINE AS CONTRACT

→ PRECONDITION DESCRIBES INPUT

→ POSTCONDITION DESCRIBES OUTPUT

→ ERROR

→ SIDE EFFECTS : INTERACT WITH THE COMPUTER AT LARGE  
PART OF THE FUNCTION,  
BUT NOT BOOLEAN

→ TYPE CONTRACT A FORM OF PRE-IMPOSED CONDITIONS

### USE DOCTRINES TO:

- DEFINE THE PUBLIC INTERFACE
- HELP WRITE THE IMPLEMENTATION
- ARGUE THAT THE CODE IS CORRECTLY CORRECT

→ CHECK AUTOMATICALLY PRE/POST-CONDITIONS USING ASSERT

EG. SUBTREES OF ABSTRACT TREE ARE  
SINGLE-STANDING TREES AT FIRST

FOR SUBTREE IN SUBTREES:

→ ASSERT INSTANCE (SUBTREE, AbstractTree)

IF ASSERT TRUE  $\Rightarrow$  ASSERTED

ASSERT FALSE  $\Rightarrow$  RAISE ERROR

POST CONDITION:

- EVERY SUBTREE MUST HAVE A PARENT SET PROPERLY,
- EVERY LAST RECTANGLE GOES TO THE BOUNDARY

→

BREAK UP FUNCTIONS

NOT NECESSARILY BY WRITING  
HELPER FUNCTION, BUT ALSO  
WITH ASSERTS