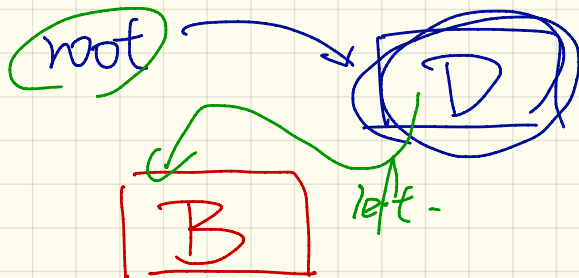
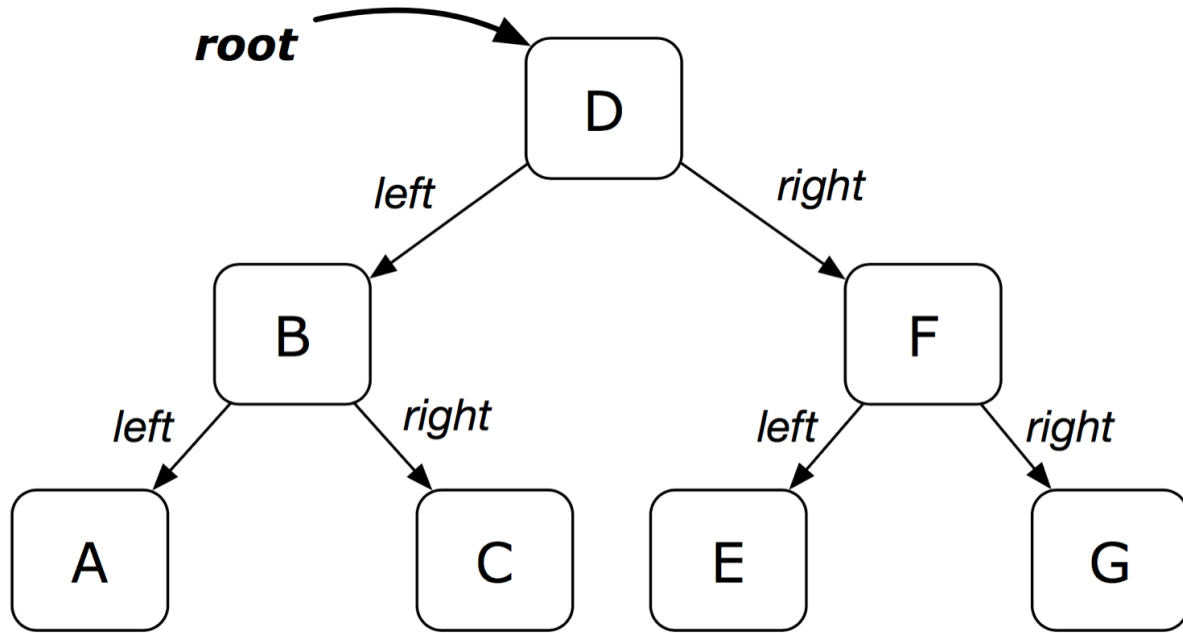


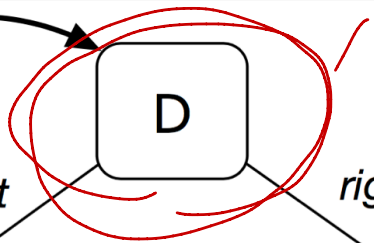
Lecture 20

Thursday Nov. 16



bt.addToLeft(root, "B")
↳ root.setLeft(new
BTNode("B"))

root

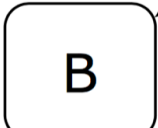


left

right

3

3

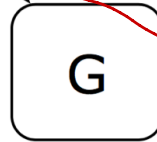
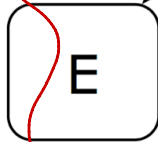
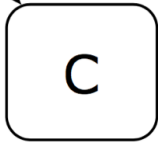
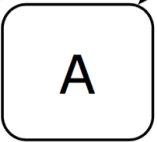


left

right

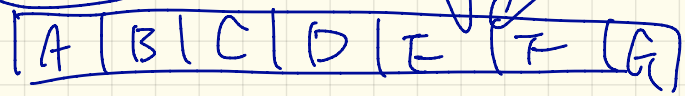
left

right

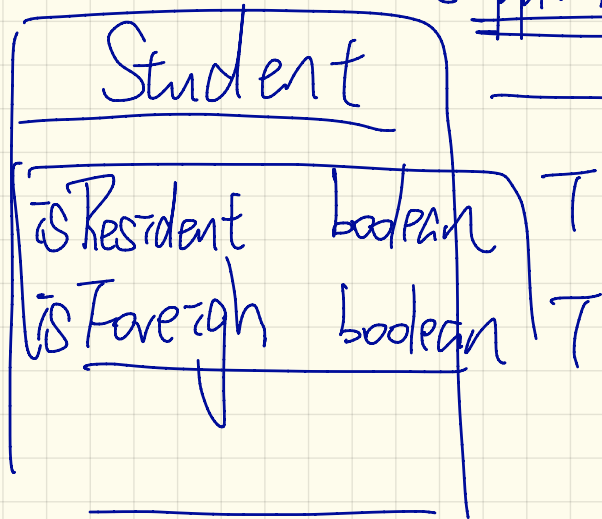


$$\text{size}(D) = \text{size}(D.\text{left}) + \text{size}(D.\text{right}) + 1$$

(bt. inorder()) \rightarrow ArrayList



Supplier



```
m() {  
    if (s.isForeign) {  
        if (s.isResident) {  
            } else {  
            }  
        }  
    }  
}
```

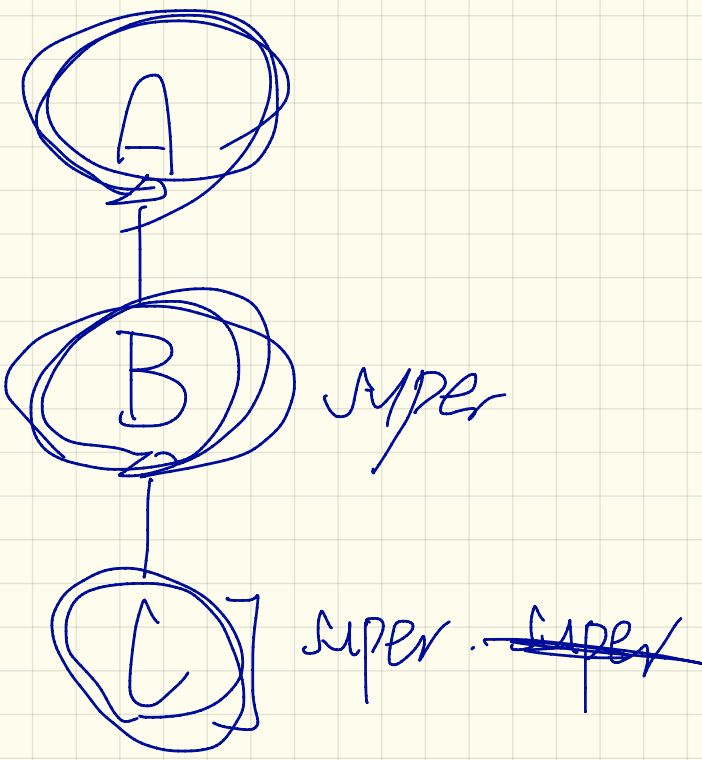
Use

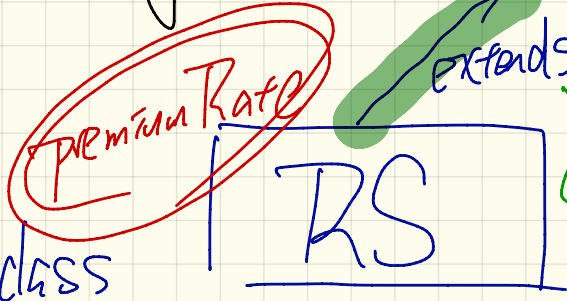
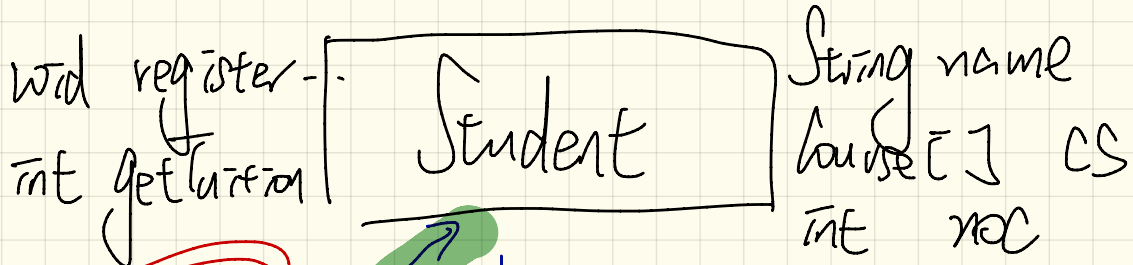
Student s = _____

m(s);

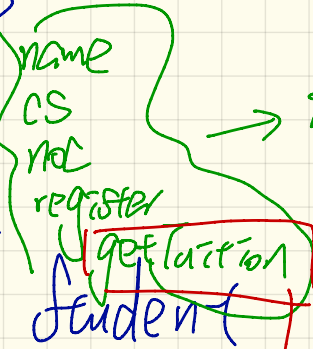
Single Choice Principle

↳ when there's a change,
there should be only one
place (class, method) to apply the change.





class
Resident Student extends



inherited,
no need to
~~repeat~~ them.

override/redefine

Resident Student vs ;

Student

S =
expectations of S

STATIC type

defines the list of
EXPECTATIONS on 'S'

- S. NAME
- S. NOC
- S. YCS
- S. REGISTER (..)
- S. GETLATION (..)

noe expectations
of S. [S. Pr
S. dv

inherits
All expectations
from
Student

address
is pr → expect.