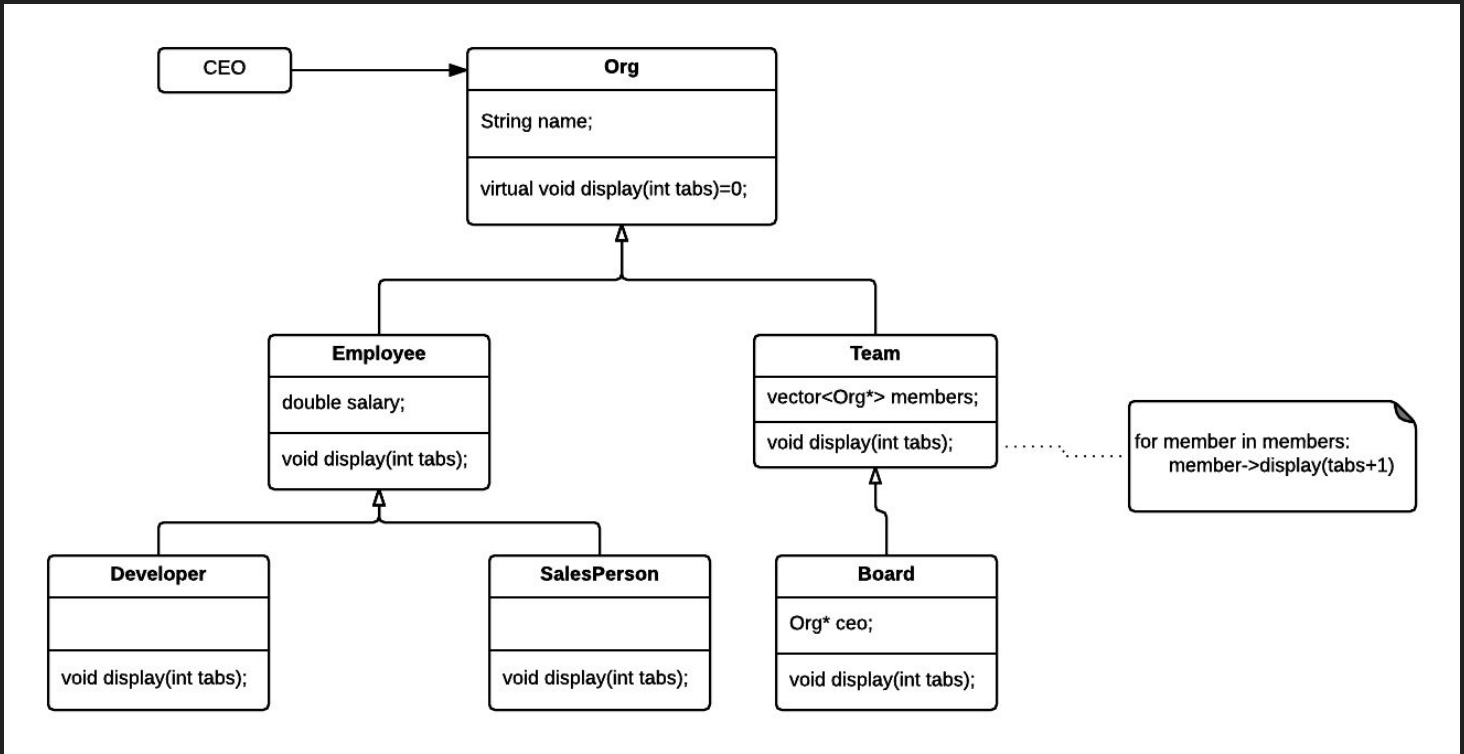
Decorator Pattern Exercise

Organization Charts

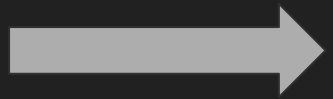
Organizational Chart (Composite Pattern)



Example:

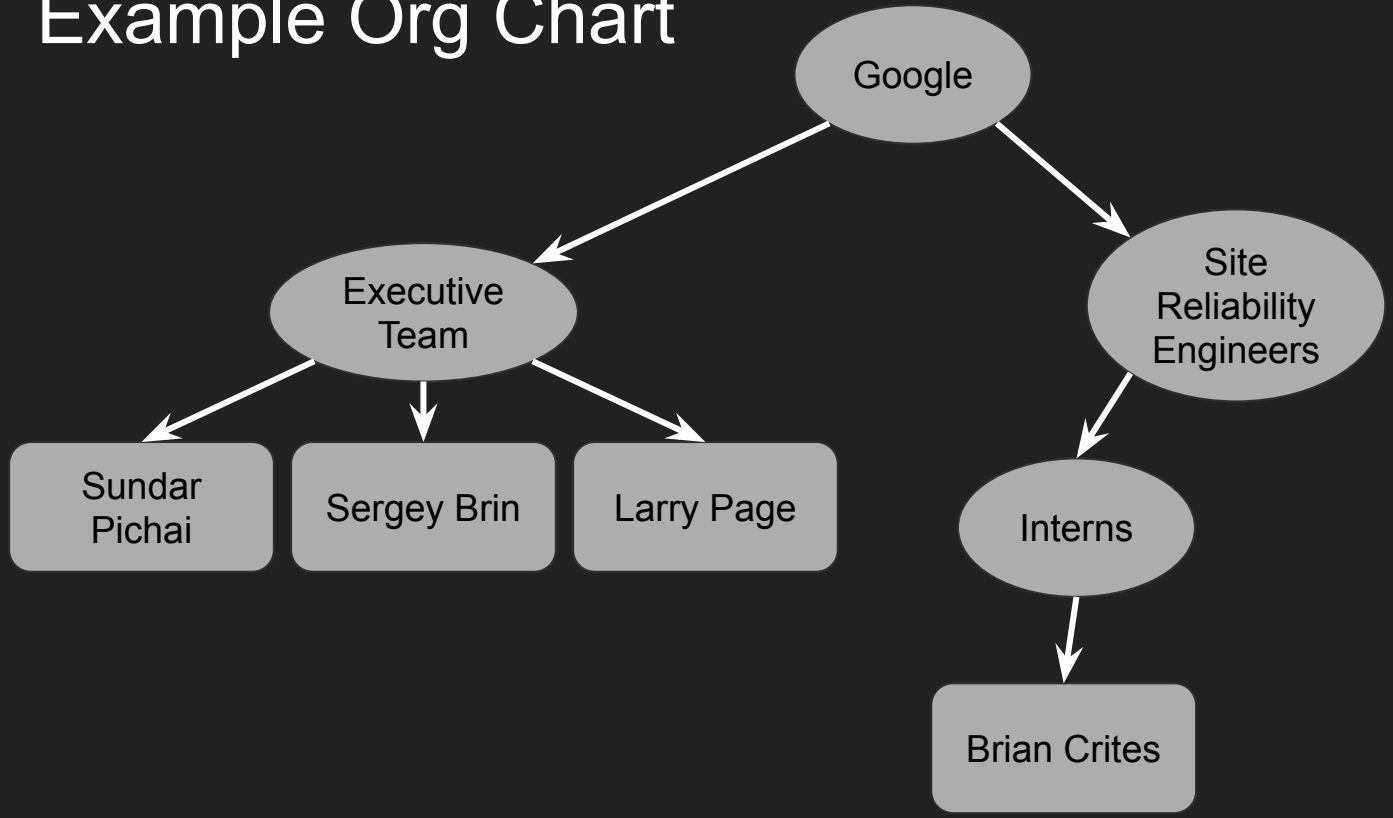
```
Org* google = new team(“Google”);
Org* exec = new board();
google->add_team(exec);
exec->add_ceo(new employee(“Sundar Pichai”);
exec->add_member(new employee(“Sergey Brin”);
exec->add_member(new employee(“Larry Page”);
Org* sre = new team(“Site Reliability Engineers”);
Org* interns = new team(“Interns”);
interns->add_member(new employee(“Brian Crites”);
sre->add_team(interns);
google->add_team(sre);


google->display();
```



Google
- Executive Team
  - Sundar Pichai
  - Sergey Brin
  - Larry Page
- Site Reliability Engineers
  - Interns
    - Brian Crites

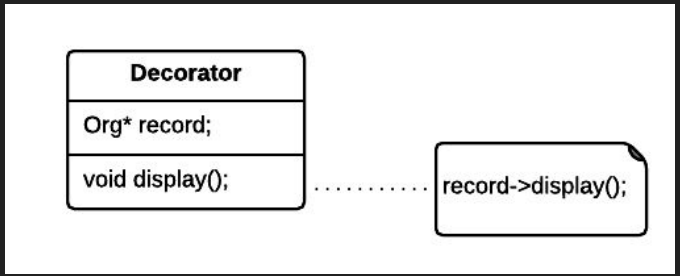
Example Org Chart



What if an employee quits?

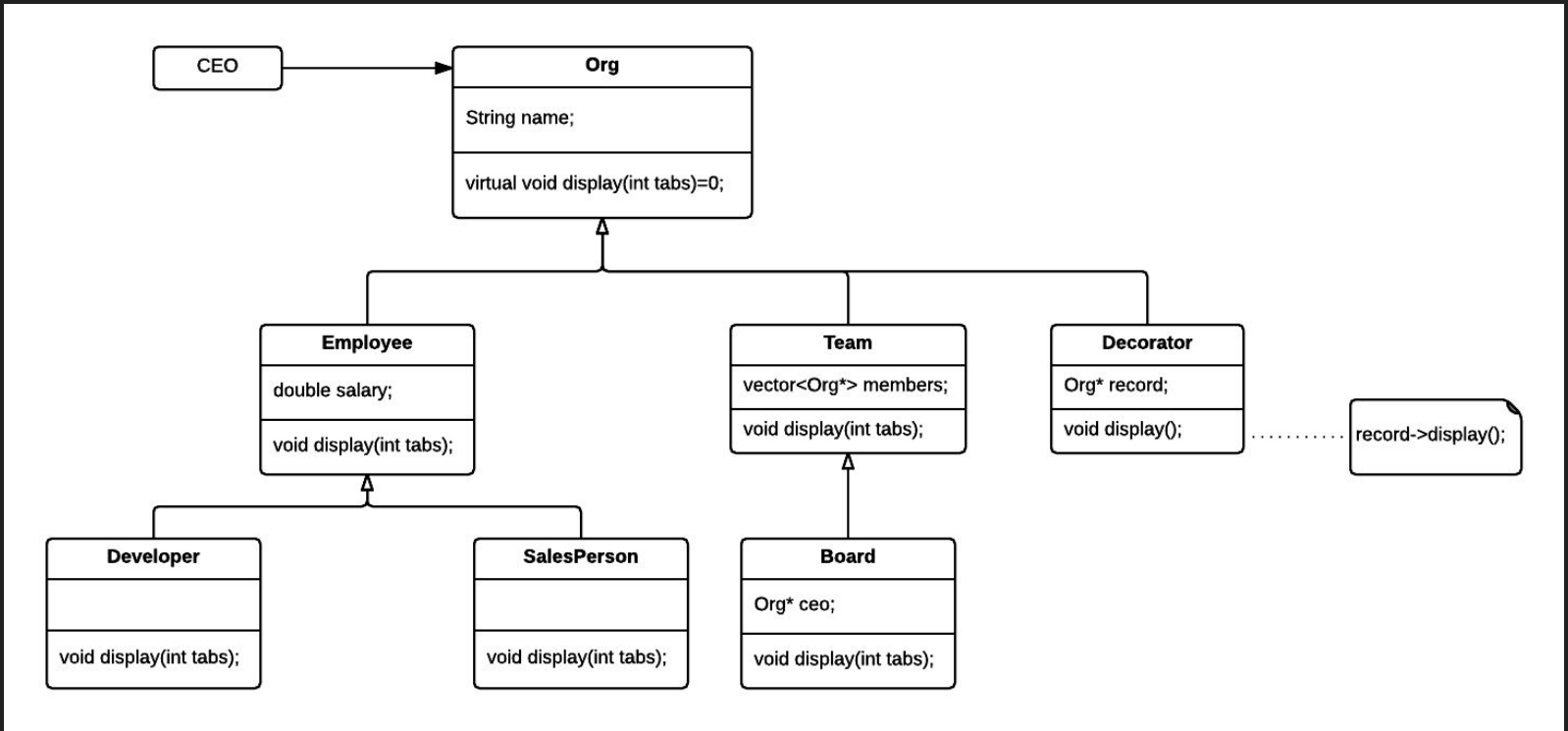
Decorator Pattern!

Create an abstract decorator class



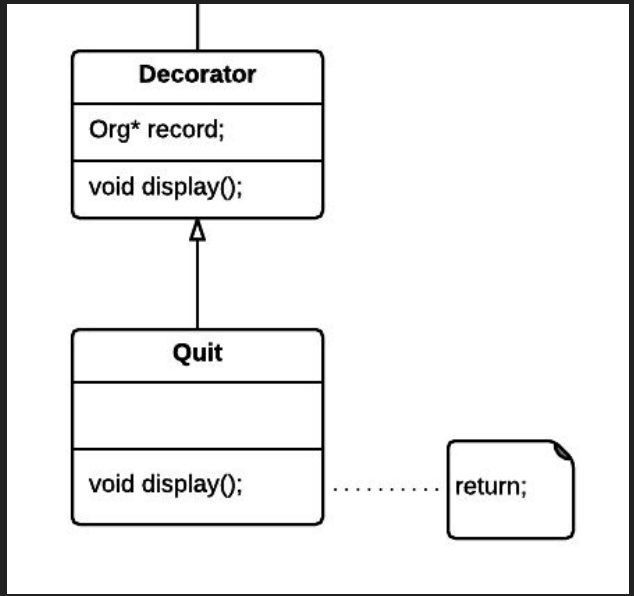
- Notice it simply extends an `Org*` object (record)
- The `record->display();` function is still called.

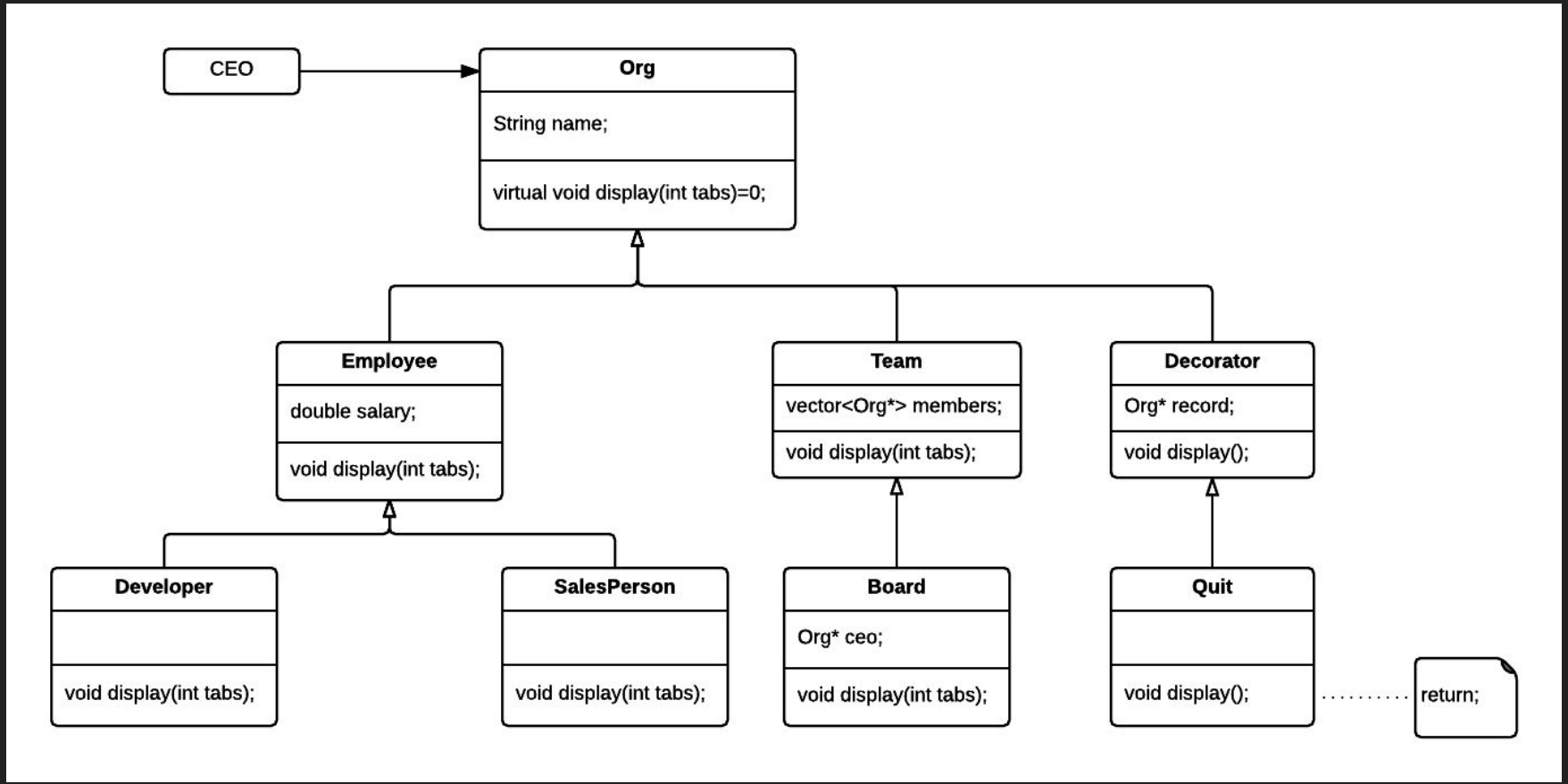
Inherit from Org (component)



Create a concrete decorator (Quit)

- Inherits from abstract decorator class
- Overrides display() to simply return without displaying the *quit* decorated Org* object

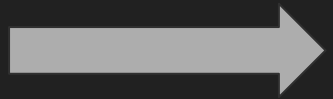



CEO
Org
String name;
virtual void display(int tabs)=0;
Employee
double salary;
void display(int tabs);
Team
vector<Org*> members;
void display(int tabs);
Decorator
Org* record;
void display();
Developer
void display(int tabs);
SalesPerson
void display(int tabs);
Board
Org* ceo;
void display(int tabs);
Quit
void display();
return;

Example:

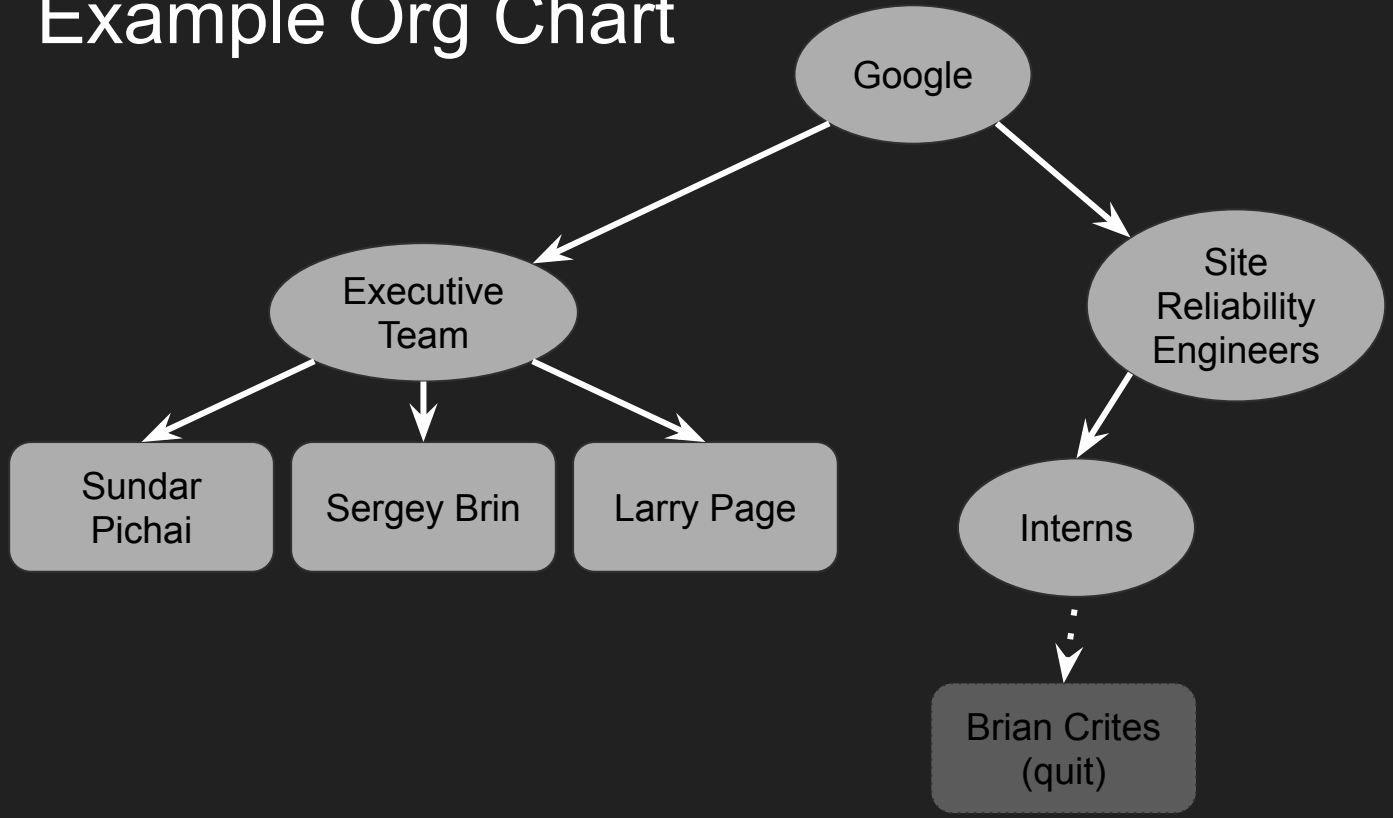
```
Org* google = new team(“Google”);
Org* exec = new board();
google->add_team(exec);
exec->add_ceo(new employee(“Sundar Pichai”);
exec->add_member(new employee(“Sergey Brin”);
exec->add_member(new employee(“Larry Page”);
Org* sre = new team(“Site Reliability Engineers”);
Org* interns = new team(“Interns”);
interns->add_member(new employee(“Brian Crites”);
sre->add_team(interns);
google->add_team(sre);
//Brian Crites quits/terminated

google->display();
```



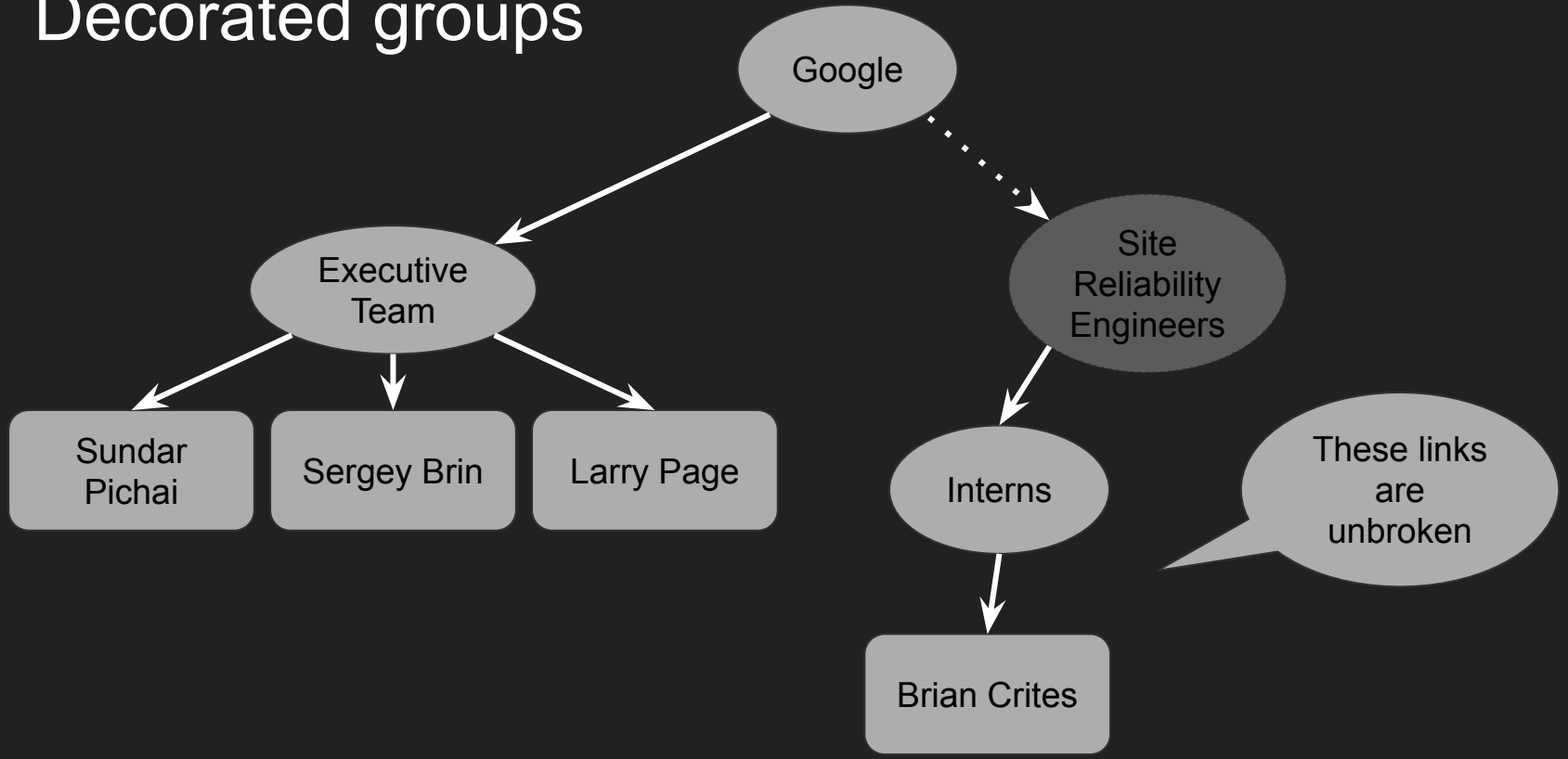
Google
- Executive Team
  - Sundar Pichai
  - Sergey Brin
  - Larry Page
- Site Reliability Engineers
  - Interns

Example Org Chart



What else can we decorate?

Decorated groups



Additional Decorators

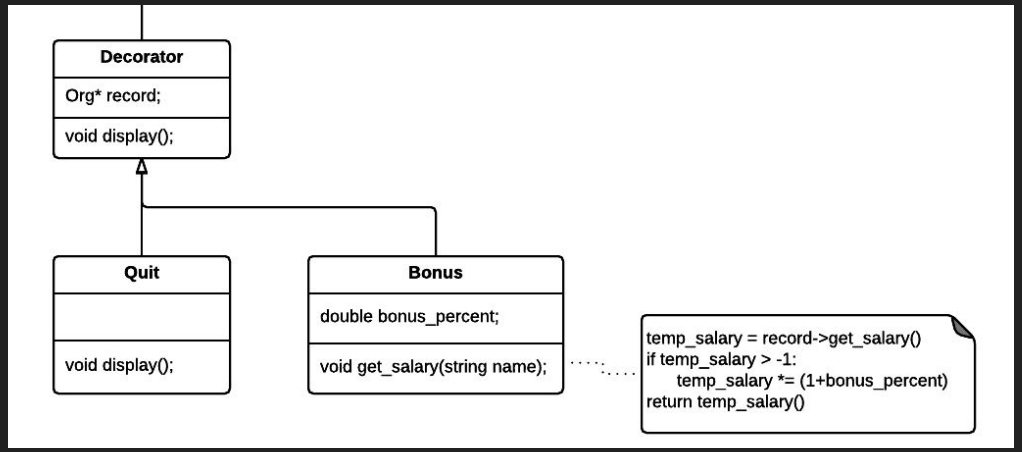
Modify `Employee` to have a salary:

```
class Employee: public org {
    int get_salary(string name) {
        if (name == this->name) {
            return this->salary;
        } else {
            return -1;
        }
    }
};
```

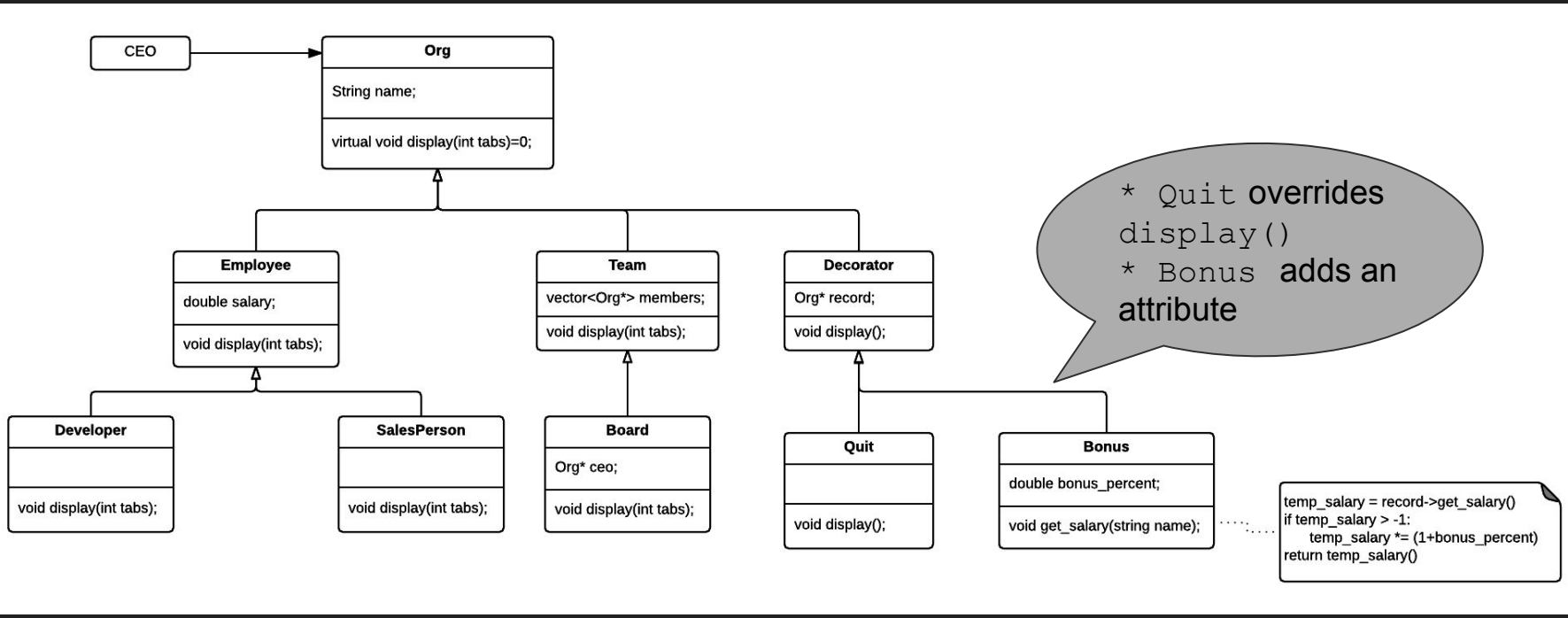
Modify `Team` to have a `get_salary()`:

```
class Team: public org {
    int get_salary(string name) {
        double temp_salary = -1;
        for (unsigned i = 0;i < this->members.size();i++) {
            if (this->members.at(i)->get_salary(name) != -1) {
                Temp_salary = this->members.at(i)->get_salary(name);
            }
        }
        Return temp_salary;
    }
};
```

Now let's create a `Bonus` decorator



Notice the added `double bonus_percent;` attribute


CEO
Org
String name;
virtual void display(int tabs)=0;
Employee
double salary;
void display(int tabs);
Team
vector<Org*> members;
void display(int tabs);
Decorator
Org* record;
void display();
* Quit overrides display()
* Bonus adds an attribute
Developer
void display(int tabs);
SalesPerson
void display(int tabs);
Board
Org* ceo;
void display(int tabs);
Quit
void display();
Bonus
double bonus_percent;
void get_salary(string name);
temp_salary = record->get_salary()
if temp_salary > -1:
temp_salary *= (1+bonus_percent)
return temp_salary()

Decorator Example Summary

- Decorators **add** **responsibility** to objects
  - Override existing methods
  - Add additional attributes
- Uses **transparent enclosure**
  - Compatible interface by inheriting from `Org` (component)
  - Single-child composition