```
    public abstract void remove(Employee e);
        throws NoSuchElementException;
    public abstract Enumeration subordinates();
    public abstract Employee getChild(String s);
    public abstract long getSalaries();
    public boolean isLeaf() {
        return leaf;
    }
}
```

Our concrete Employee class stores the name and salary of each employee and allows us to fetch them as needed.

```
public Employee(String _name, float _salary) {
    name   =    _name;
    salary =    _salary
    leaf   =    true;
}
//--------------
public Employee(Employee _parent, String _name, float _salary) {
    name   =    _name;
    salary =    _salary;
    parent =    _parent;
    leaf   =    true;
}
//--------------
public float getSalary() {
    return salary;
}
//---------------
public String getName() {
    return name;
}
```

The Employee class must have concrete implementations of the add, remove, getChild, and subordinates methods classes. Since an Employee is a leaf, all of these will return some sort of error indication. For example, *subordinates* could return null, but programming will be more consistent if it returns an empty Enumeration.

```
public Enumeration subordinates () {
    return v.elements ();
}
```

The *add* and *remove* methods must generate errors, since members of the basic Employee class cannot have subordinates.

```
public boolean add(Employee e) throws NoSuchElementException {
    throw new NoSuchElementException("No subordinates");
    }
    //
```