Interfaces

In this lab, we eliminate the fat-interface smell with *Company*'s dependency on *HumanResource*. We will also implement the *Composite Pattern*

Builds On - Inheritance

Overview

Determine *Company*'s dependency and design an interface that contains the exact methods/properties required by company. Replace *Company*'s dependency on *HumanResource* with the new interface type.

Also, we will allow companies to be nested within companies.

Steps

1. Create a new interface named *IPayable*
   - single method: *void Pay()*
2. Add *IPayable* to the *HumanResource* base list
   - is there any purpose in *IEmployee* any more?
3. Modify *Company* replacing all occurances of *HumanResource* with *IPayable*
4. Update *CompanyTest* - change mocks

The composite pattern is an elegant pattern that will greatly increase the functionality of our application. How can we adjust the design so that companies can contain companies?