

# Design Patterns

## Group Exercise

### **Composite**

Organizations often are made up of suborganizations. For example, the government of the State of Illinois includes the University of Illinois. The University has three campuses, which have many colleges. Each college is made up of many departments.

This makes it difficult when you are trying to negotiate a site license with an organization. A site license can be for an entire university, or campus, or college, or department. Of course, sometimes people just buy licenses for a single machine or a single person. The price of a site license depends on the size of an organization.

Design a system that can help an employee tell whether there is a site license that covers a particular piece of software, and that can help the organization know how many people will be covered by a particular agreement, and thus how much to pay for it.

Make a Party class with subclasses Individual and Organization. Organization is a composite party. Because it is a composite you'll have to design the normal composite operations like add, remove, and enumerate children. Each party must be able to tell you how many individuals are in it. It will do this by communicating with its children.

Give each party a set of licenses. A party should be able to add and remove licenses.

### **Chain of Responsibility**

We also want our system to determine whether a party is covered for a license for a particular piece of software. It will do this by communicating with its parent.

Assume that each individual is in only one organization. If you have time, you might discuss how you would deal with an individual being in several organizations.

## Strategy

Extend your University Licensing Design so that parties can have different licensing policies. The simplest policy is to acquire a single seat license. However, some parties will want to take advantage of quantity license discounts and may obtain policies for acquiring a full-site license. Assume for simplicity that a quantity discount is available for 10-seat licenses or more by purchasing a site license.

The default license policy for parties purchasing licenses is to first see whether they are covered under an existing license. If not, they will then look to the parent organization and see if the parent organization will purchase a license for them. If one of their parent organizations will purchase it for them, then they have them do it and they are done. If not, the party will use its own license policy to complete the licensing process itself.

For example, a department can save money by purchasing a site license when 10 or more parties under its jurisdiction have licensed a product. Its policy will be that once it gets 10 or more licenses under its jurisdiction it will cancel the single seat licenses, and acquire a site license.

When a new party is created, they will be given a `NoLicensingPolicy`, which will not purchase a license. Of course, when a party gets added to an organization, its policy will likely change. Also, your organization might change its licensing policy for example from `IndividualLicensingPolicy` to `SiteLicensingPolicy`.

So your solution should have three different licenses policies: `NoLicensingPolicy`, `IndividualLicensingPolicy`, and `SiteLicensingPolicy`.

Whenever you use the Strategy pattern, there are some standard questions that you ask yourself.

- 1) Who invokes the strategy? In this case, this happens when you license new products.
- 2) What is the interface of the strategy?
- 3) What information does the strategy need to carry out its responsibility?
- 4) How are policies established and when do policies change?

In other words, you should think about the *context* of the strategy as well as the strategy itself. To answer this question, you will need to make up a few sample strategies and test them. Figure out what input a strategy needs, and what it produces.

Here are some questions to consider after you have made your first solution:

- 1) How might you use the Template Method pattern in this example?
- 2) A lot of Parties will have the same `LicensingPolicy`. How can you keep from wasting space by duplicating them?