

Polymorphism Lab 2

Objectives

1. Students will practice refactoring non-polymorphic code in this lab.
2. After this lab, students should be comfortable with interfaces and enumerated types.

Overview

In this lab we will refactor our various Things from a previous lab. The BoringThing and ColorfulThing classes will be added to a common class hierarchy, and the ThingContainer will be enhanced to hold any type of Thing.

Unit Test

You are expected to write tests for your methods as part of a test-driven development process.

Instructions

Copy your code from Initialization Lab #2 into a project for this lab.

Part 1:

Before we can refactor ThingContainer, BoringThing or ColorfulThing, we need to refactor our class hierarchy a bit. Create a new interface called `Thing`. For now your `Thing` interface does not need any methods. Next, refactor your `ColorfulThing` and `BoringThing` classes to implement the `Thing` interface. This will come in handy shortly.

Part 2:

Now it's time to refactor the ThingContainer to accept any type of Thing object. Adjust its implementation to accommodate this new program structure.

Searching for a ColorfulThing of a particular color is going to be a bit harder now that they will be mixed in with other types of Things. Add a `getProperties()` method to the `Thing` interface that returns a list of properties (like color) that each Thing has (it's okay if the list is empty for certain types of Things). Use this new method to update your search algorithm. Remember to write tests for your code.

Part 3: One more Thing...

Let's add another Thing to our type hierarchy. Create a `CoatedThing` class that extends `ColorfulThing`. `CoatedThing` s should have a special coating that gives them both a color and a texture. Your code should support at least four textures including: `Glossy`, `Matte`, and `Speckled`. Texture sounds an awful lot like a new property....

Update `ThingContainer` to allow searching by any type of property. When you are finished, make sure your unit tests cover all of the new features you've added and that they all pass. Write a short `main()` method to demonstrate a few of these new features.