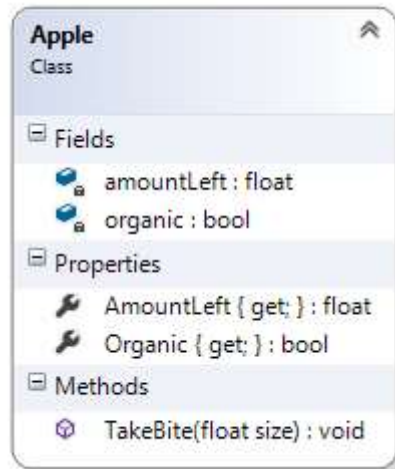


A Classy Apple Abstraction Solution

There are various designs that are reasonable for the given problem. Here's my Apple class diagram:



Fields

I included the `organic` field because the `Main` method will only eat apples that haven't had pesticides applied to them. We need to know how much of the apple is left so we can tell when there's no more apple to eat; that's what the `amountLeft` field is for.

Properties

The `Organic` property lets the `Main` method find out whether the apple is organic or not.

The `AmountLeft` property returns how much of the apple is left to eat. I thought about exposing a `SomeLeft` property that returns a `bool` instead, but when the `Main` method takes a bite (coming soon!) it tells how big a bite it wants to take. If it doesn't know how much of the apple is left, it might try to bite more apple than there is left to eat.

Methods

The `TakeBite` method takes a bite of the given size out of the apple.

Constructors

I don't include constructors in my class diagrams, but I'll definitely implement a constructor for the `Apple` class. My constructor parameters will be whether or not the apple is organic and how large the apple is (which will be my initial value for how much is left to eat).