Composition

It's time now to talk about composition.

Composition is another component of object-oriented programming.

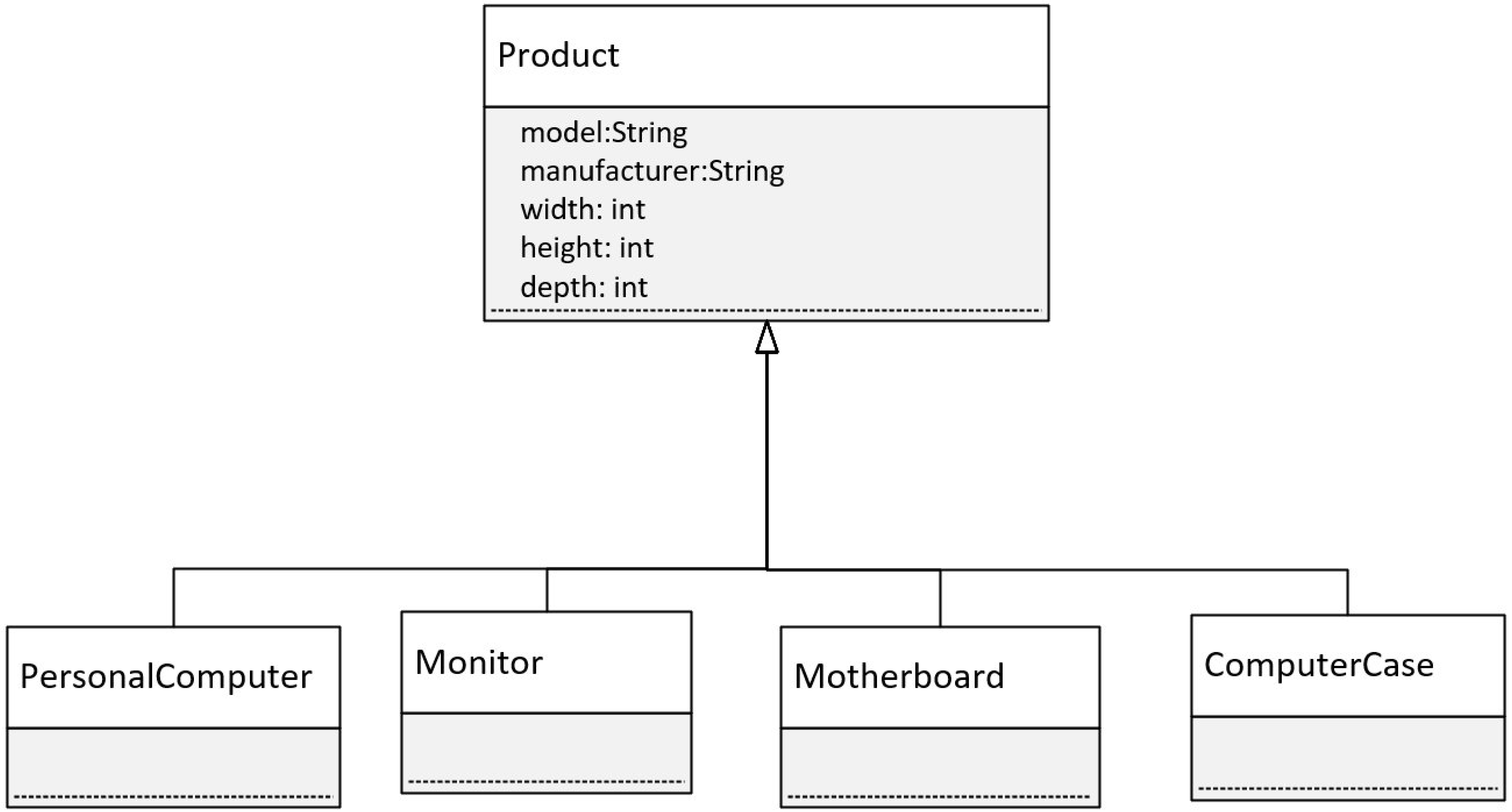
Inheritance

In this instance, I have a base class called Product.

All of my computer parts are going to inherit from Product.

All my parts will then have the same base set of attributes: manufacturer, model, and dimensions. The width, height, and depth, in other words.

All of these items are products, a particular type of Product.



Inheritance vs. Composition

Inheritance defines an **IS A** relationship.

Composition defines a **HAS A** relationship.

Inheritance vs Composition

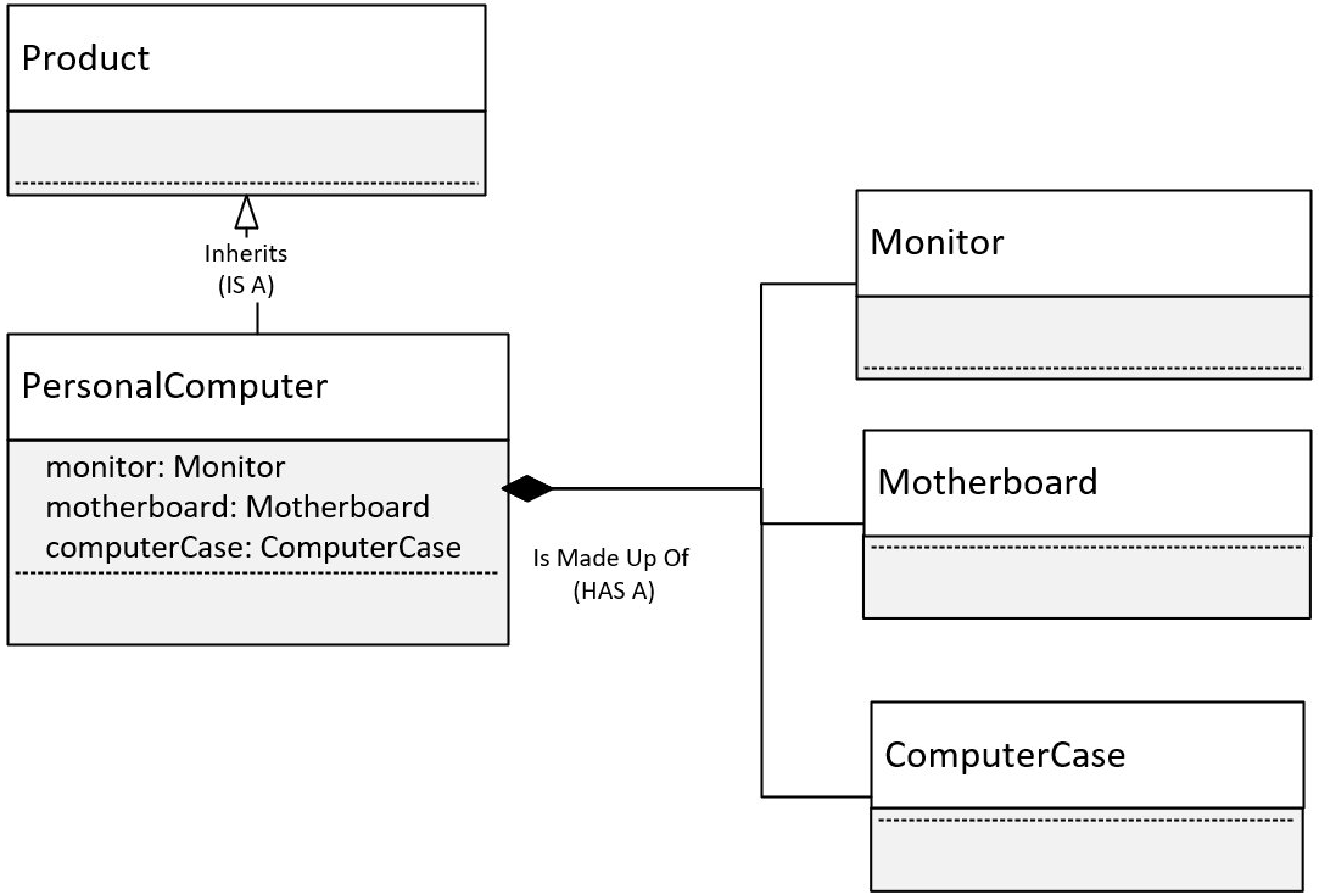
To keep this diagram simple, PersonalComputer inherits from Product.

But a Personal Computer, in addition to being a product, is actually made up of other parts.

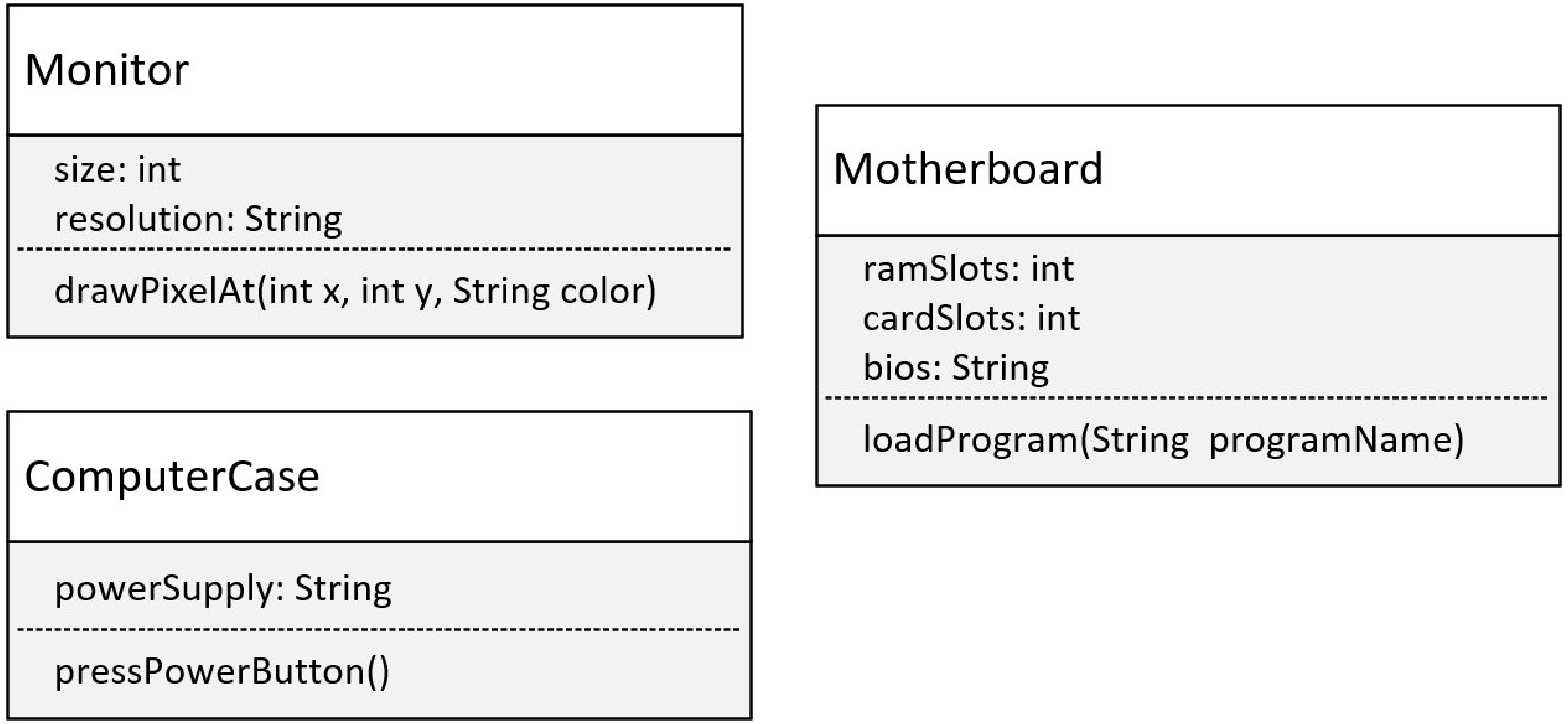
Composition is actually modeling parts, and those parts make up a greater whole.

In this case, I'm going to model the personal computer.

And I'm modeling the **has a** relationship with the motherboard, the case, and the monitor.



The Parts



Monitor

size: int
resolution: String

drawPixelAt(int x, int y, String color)

ComputerCase

powerSupply: String

pressPowerButton()

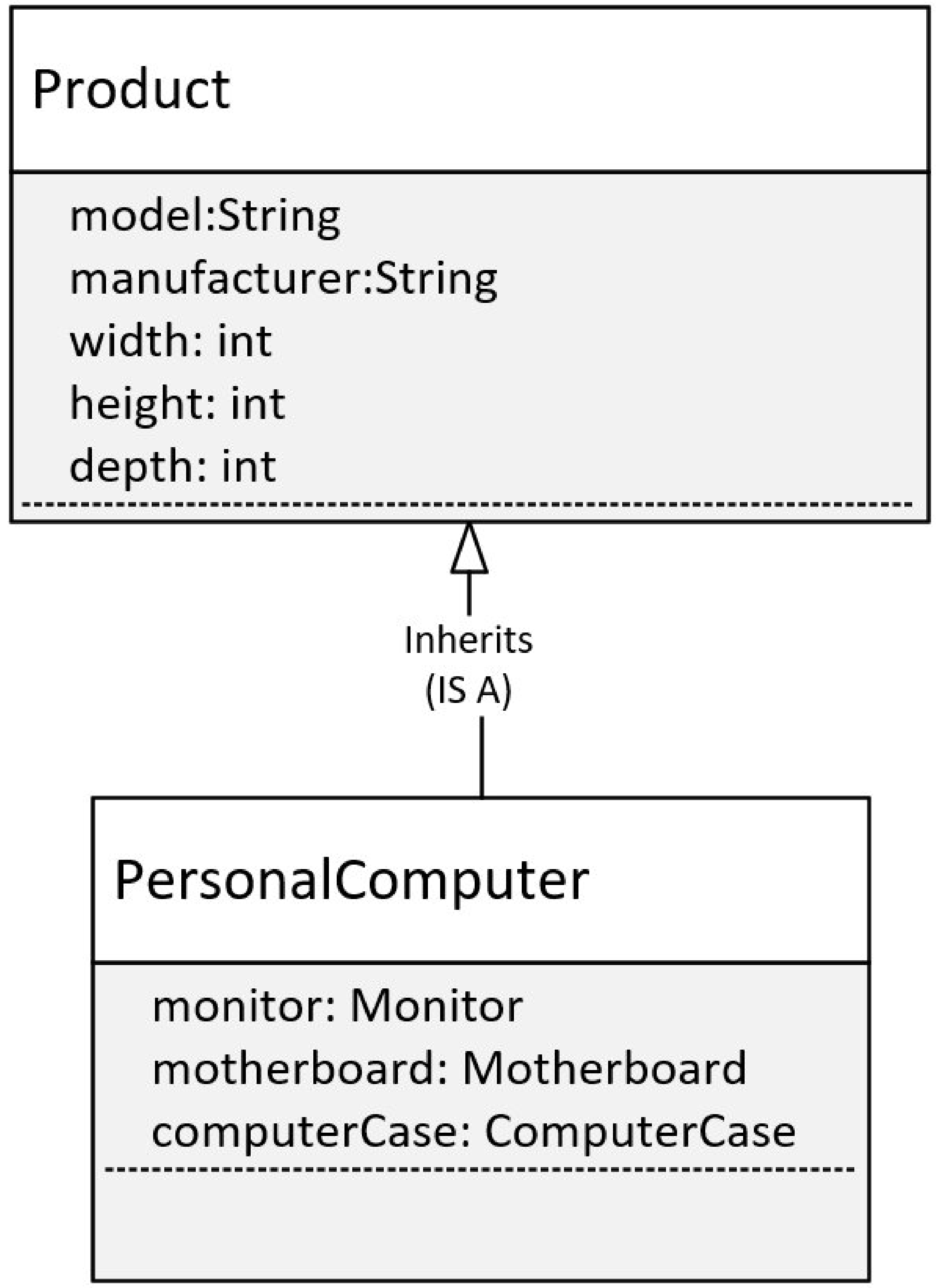
Motherboard

ramSlots: int
cardSlots: int
bios: String

loadProgram(String programName)

This diagram shows the 3 classes that will make up the personal computer.

PersonalComputer



This will be my personal computer, and I've said it inherits from Product.

But it also has 3 fields, which are classes: these are Monitor, Motherboard, and ComputerCase.