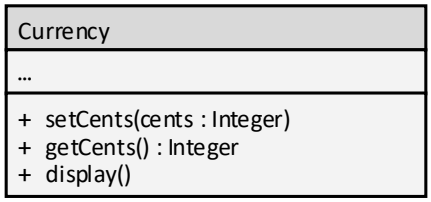
CSE 230 Problem Set 05

Problem 21.1: Currency

Consider the following scenario:

> A financial package uses a `Currency` class to manage money. This class cannot have rounding errors so, for fidelity reasons, uses an integer as the internal representation. Note that the software only works with dollars and cents, does not need to make coin change or anything like that. The software would add a dollar to a total, or find the difference between two values.

Consider the following class diagram



Classify the level of convenience for this class. Justify your answer.

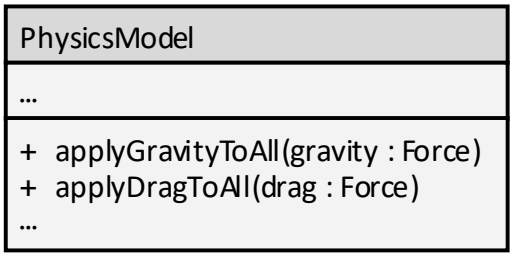
This class seems to have a straightforward level of convenience. The only work necessary for interfacing with this class would be converting into cents from dollar amounts. This is a simple amount of work and would not require much code to do fulfill.

Problem 21.2: Physics Model

Consider the following scenario:

> A video game uses one class to manage all applications of physics. Everything in the game uses a `Force` object to monitor force: force due to gravity, force due to thrusting, force due to drag, and force due to collisions. Note that the `Force` class is used throughout the project and we can assume that it is fully flushed out and has seamless convenience.

Consider the following class diagram:



Classify the level of convenience for this class. Justify your answer.

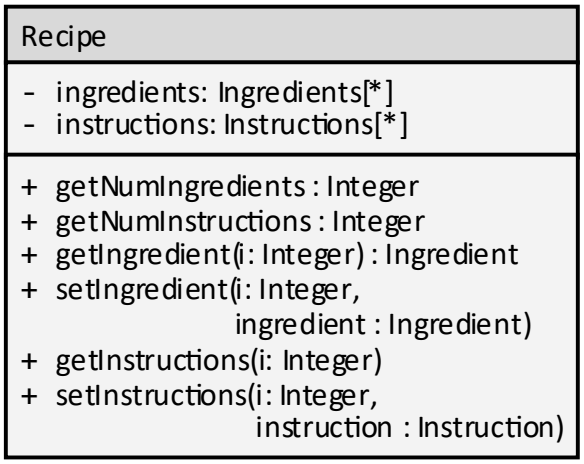
This class becomes complicated due to the fact that drag force doesn't apply to every object the same way as applyDragToAll would suggest. Even with the Force class being seamless, this class seems to be convoluted due to the work needed to adjust drag for each object in the game based on various levels of motion or lack thereof.

Problem 21.3: Recipe I

Consider the following scenario:

> A recipe program has a scaling option where the user can specify the number of servings needed for a given meal. For example, you should be able to scale a taco recipe originally designed for three people to accommodate a party of seven.

Consider the following class diagram:



Classify the level of convenience for this class. Justify your answer.

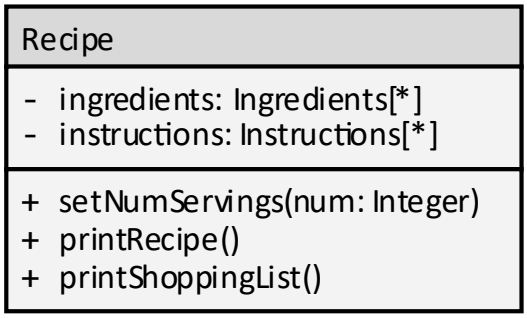
This class is not clear enough to work with in terms of scaling recipes. There doesn't seem to be any properties or methods involved with scaling options within the class. Even if using the setter methods does this, most of the work to do so would be handled outside the class. Because of this, this class has a convoluted level of convenience.

Problem 21.4: Recipe II

Consider the following scenario:

> A recipe program has a scaling option where the user can specify the number of servings needed for a given meal. For example, you should be able to scale a taco recipe originally designed for three people to accommodate a party of seven.

Consider the following class diagram:



Classify the level of convenience for this class. Justify your answer.

This class seems to have a seamless level of convenience. It seems to draw template recipe info from another source, making it easy for the user to pull data. It has simple methods to fulfill the purposes of the class, and requires minimal effort to do so.