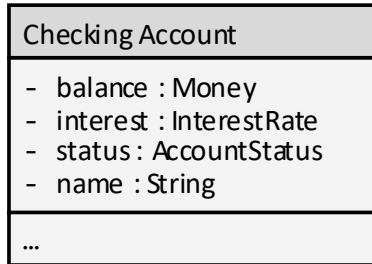


# CSE 230 Problem Set 06

## Problem 22.1: Checking Account Analysis

Identify the level of fidelity from the following class diagram meant to represent a checking account. This includes an account balance, interest rate, and status. You can assume that the enclosed classes (Money, InterestRate, and AccountStatus) have complete fidelity.



Rationale for whether this class is **Extraneous**:

This class is not extraneous. It does not have any states that are not used for the design concern of a checking account.

Rationale for whether this class is **Partial**:

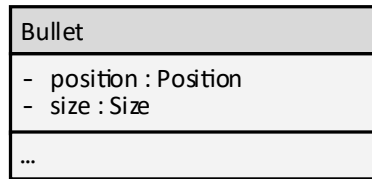
This class could use a member variable object of a transaction history class, along with routing/account numbers as well.

Level of fidelity:

This class is partial due to the missing member variables needed to cover all information for this design concern.

## Problem 22.2: Bullet Analysis

Identify the level of fidelity from the following class diagram meant to represent a bullet in a 3D game. You can assume that the enclosed classes (Position and Size) have complete fidelity.



Rationale for whether this class is **Extraneous**:

This class is not extraneous. It already entails any possible position and any size possible for a bullet.

Rationale for whether this class is **Partial**:

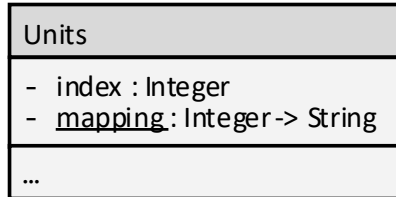
The bullet class is missing a member variable to handle a bullet's speed.

Level of fidelity:

This class has partial fidelity. It has most details, but could benefit from having a member variable to cover speed.

## Problem 22.3: Units Analysis

Identify the level of fidelity from the following class diagram meant to represent a unit of an ingredient that is part of a recipe program. Here the static member variable mapping contains the following collection: {1:cups, 2:teaspoons, 3:tablespoons, 4:ounce, 5:pound}.



Rationale for whether this class is **Extraneous**:

Currently, this class is extraneous just due to the fact that the index member variable isn't restricted to the 5 possible states of the design concern.

Rationale for whether this class is **Partial**:

This class is not partial, due to the fact that it covers all required information contained in the design concern.

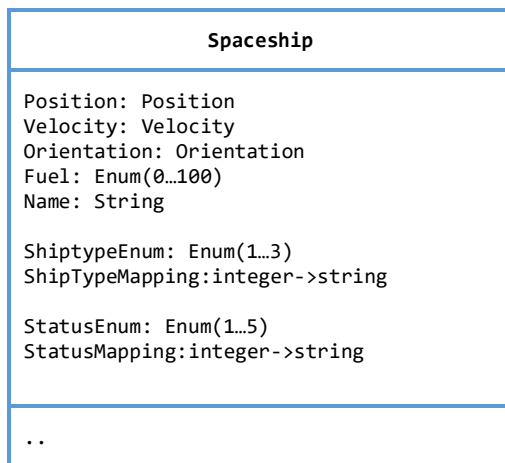
Level of fidelity:

This class would be classified as extraneous. It could have bugs or misfunctions with the index member variable.

## Problem 22.4: Spaceship Design

Create a class diagram exhibiting the highest possible degree of fidelity to match the following design concern:

A spaceship in a 3D game has several attributes: position, velocity, orientation. It also as a ship type (one of 3 types: Beginner, Intermediate, and Advanced), a status (one of 5 types: Unharmd, Light Damage, Heavy Damage, Critical Damage, Destroyed), fuel (value from 0...100), and name.



## Problem 22.5: Recipe Design

Create a class diagram exhibiting the highest possible degree of fidelity to match the following design concern:

A recipe consists of several things: a collection of ingredients, a collection of process steps, a name, and a description. It also has an estimated prepare time, estimated cook time, cost, and number of calories. The user can categorize the recipe (one of 5: Appetizer, Salad, Soup, Main Course, Dessert), specify its use (one of 4: Breakfast, Lunch, Dinner, Snack), and rate it according to quality (up to 5 stars).

| Recipe   |
|--|
| Ingredients: array<Ingredients><br>Steps: array<string><br>Name: string<br>Description: string<br>cookTime: Short<br>cost: Short<br>calories: Short<br>TypeEnum: Enum(1...5)<br>TypeMapping: integer->string<br><br>MealEnum: Enum(1...4)<br>MealMapping: integer->string<br><br>Rating: Enum(1...5) |
| ...  |

## Problem 22.6: Financial Institution Design

Create a class diagram exhibiting the highest possible degree of fidelity to match the following design concern:

A financial institution has several properties: a name, a web site address, a type (Bank, Credit Card, Investment, Other), a street address, and a phone number. The user can also store a username and password combination. We will keep track of the date that the first account was opened, the date of the last usage, and a list of accounts associated with this institution.

| Financial Institution  |
|--|
| Name: String<br>siteAddress:String<br>accountTypeEnum: Enum(1...5)<br>accountTypeMap:integer->string<br>streetAddress: String<br>phoneNumber: PhoneNumber<br>credentialMapping:Username->Password<br>firstAccountStartDate: Date<br>lastUsage: DateTime<br>accounts:array<Account> |
| ...  |