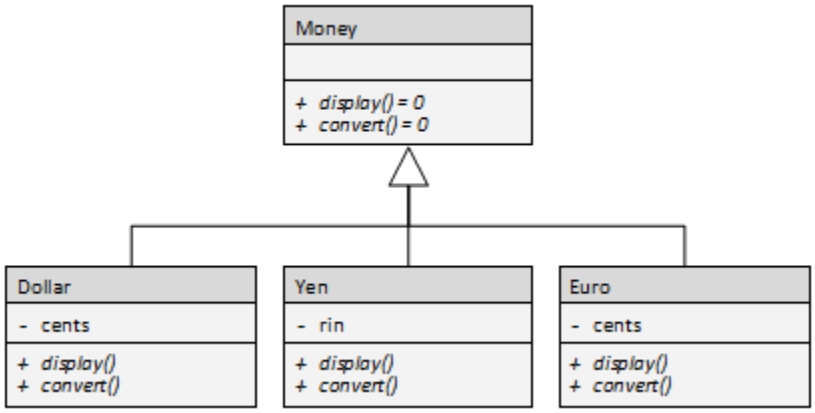
CSE 231 Problem Set 03

Problem 31.1: Level of Adaptability

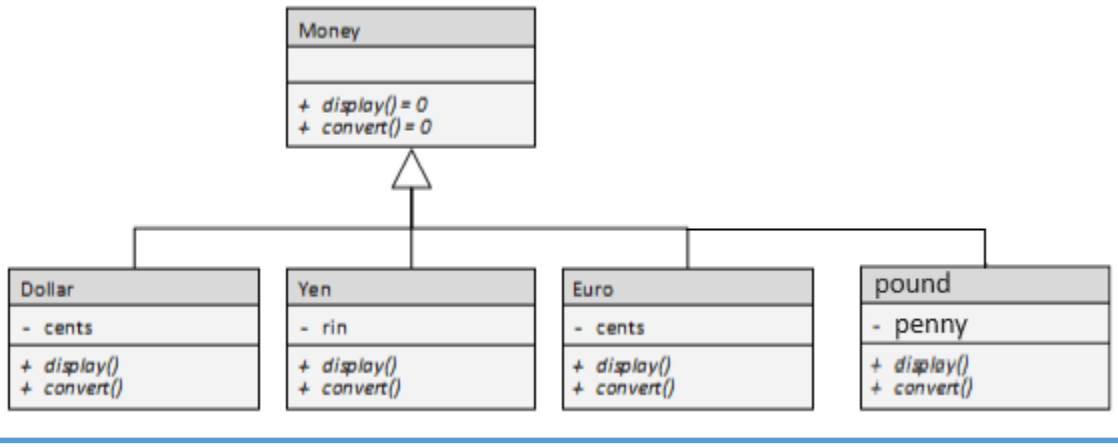
Consider the following class diagram. The base class has no member variables and two pure virtual functions. The derived classes honor the base class' interface but provide all their own functionality. The `convert()` method allows one currency type to convert to another.



Characterize the level of adaptability based on the following scenario. Provide a convincing rationale as to why it is what you say it is.

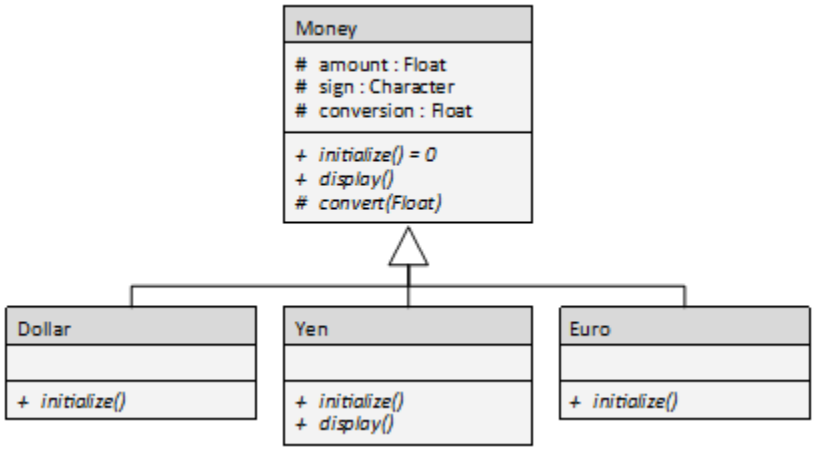
Augment the `Money` class to include an English pound.

This is a straightforward because to add to the implementation all you need to do is add a child class



Problem 31.2: Level of Adaptability

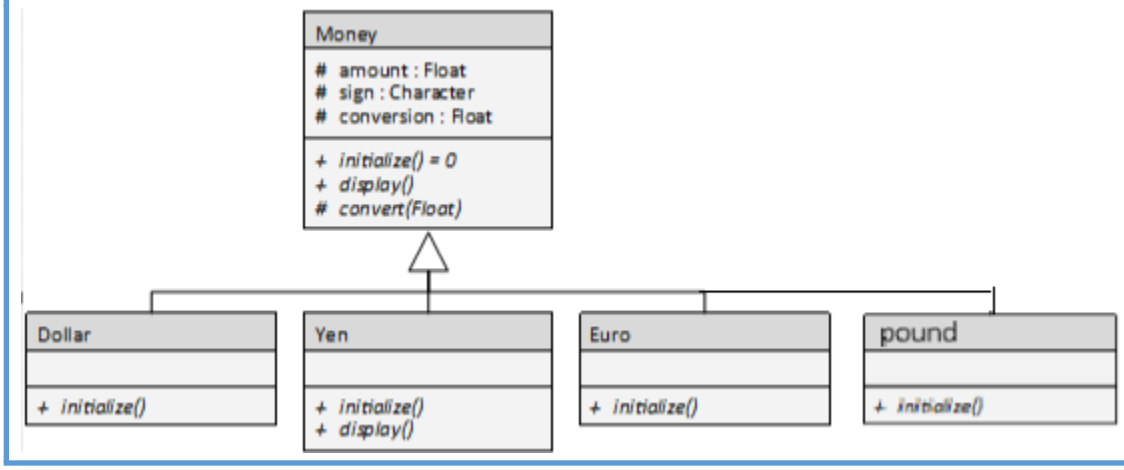
Consider the following class diagram. The `initialize()` function sets the sign (used in the `display()` method) and conversion value (used in the `convert()` method) in the base class.



Characterize the level of adaptability based on the following scenario:

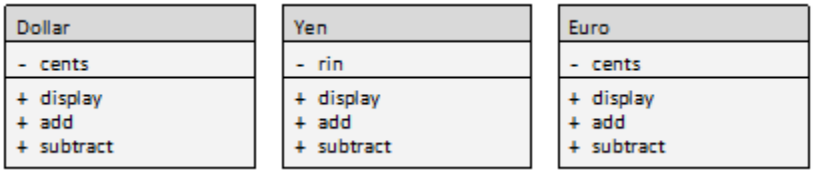
> Augment the `Money` class to include an English pound, which behaves much like the American dollar and the Euro.

This is enabling because anything that needs to be added with the child class that needs to be created is very trivial, for many money values only needing the initialize function, and some needing display as well.



Problem 31.3: Level of Adaptability

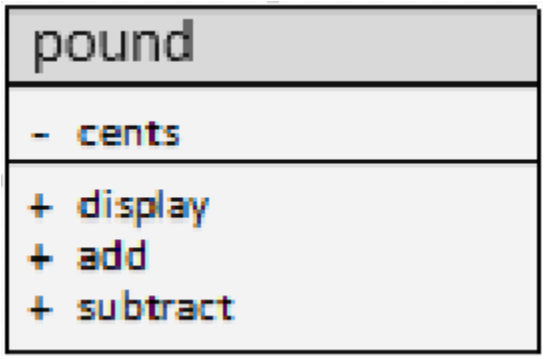
Consider the following class diagram:



Characterize the level of adaptability based on the following scenario:

Create a class to represent an English pound

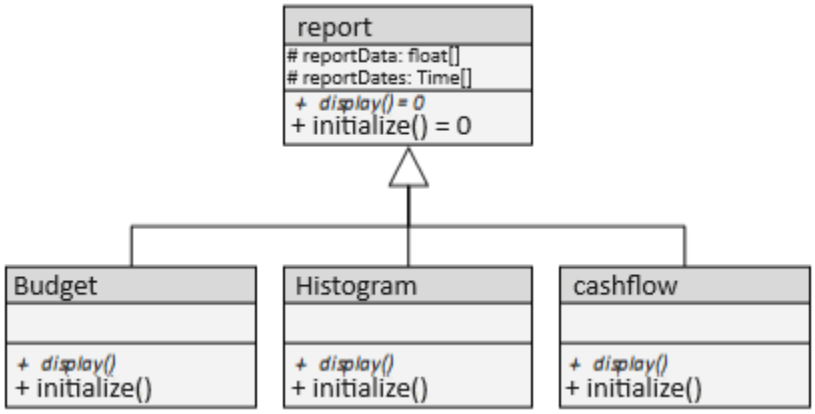
closed because there is no allotment made to extend the classes. These are all individual classes not linked together in any way.



Problem 31.4: Design a Class

Create a class diagram matching the following scenario:

> A personal finance application can display several types of reports: a budget report, a histogram showing account balances over time, and a cash flow graph depicting income vs outgo.



Characterize the level of adaptability for this or for your class diagram.

This is straightforward because the any new function can easily be added with the addition of a child class

After your initial characterization of adaptability, make the following change:

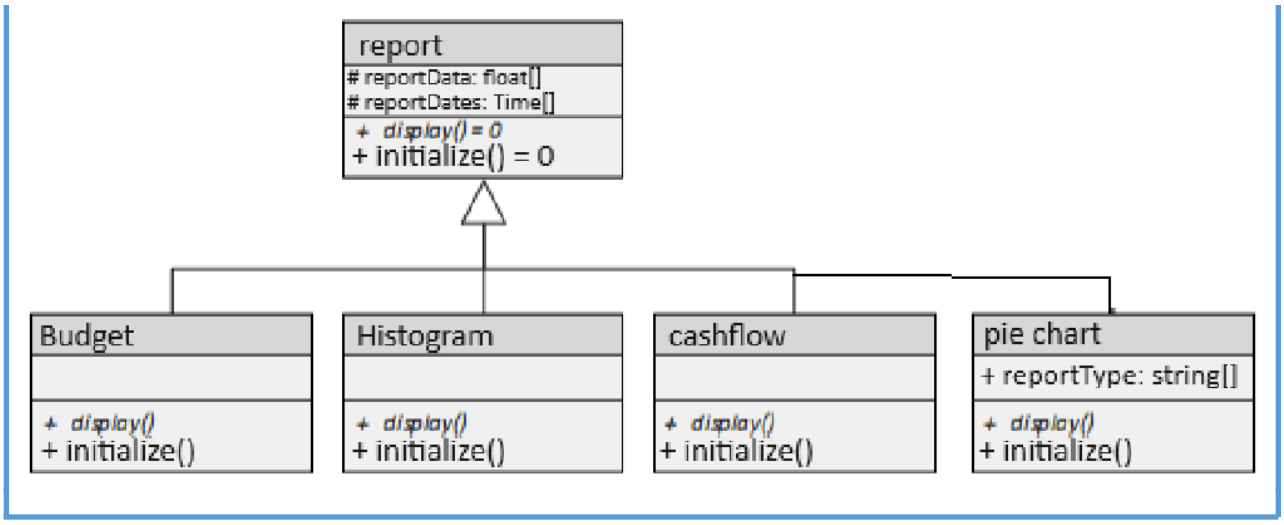
> Create a new report type: a pie chart. This will display the percentage of spending associated with each budget category.

Describe in English what you would need to do to accommodate this change:

I would need to initialize a child class and implement the display and data type

After making this change, characterize again the level of adaptability. Was your predicted and realized levels of adaptability comparable?

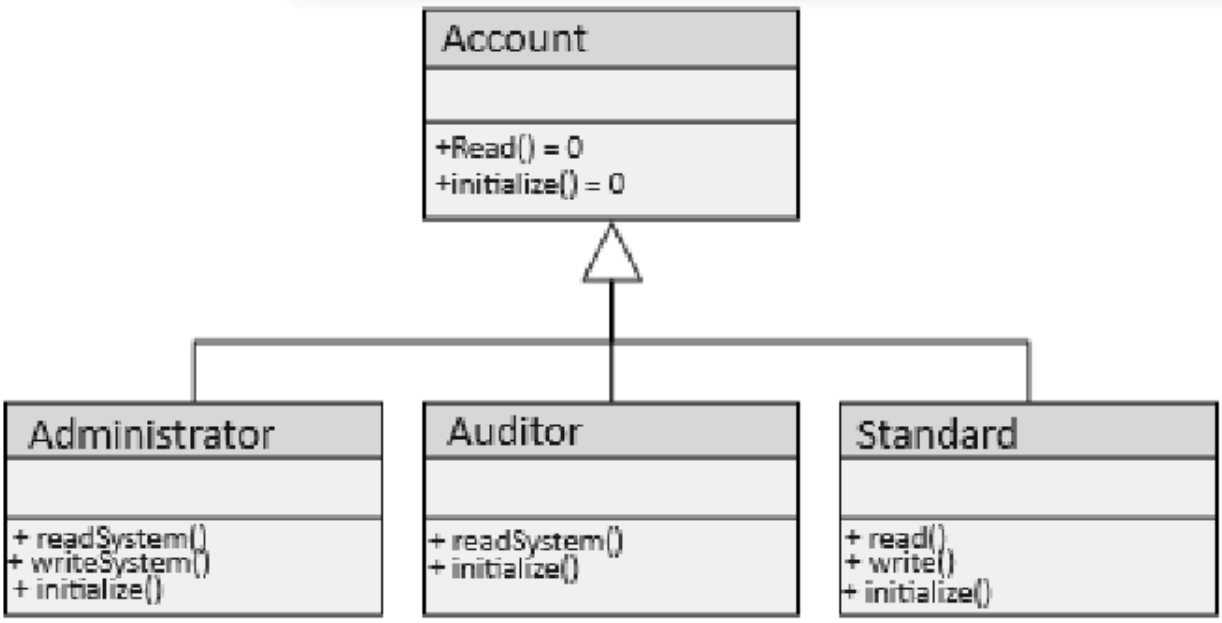
This is straightforward because the any new function can easily be added with the addition of a child class. I believe my predicted and realized levels of adaptability are comparableThis is straightforward because the any new function can easily be added with the addition of a child class


report
reportData: float[]
reportDates: Time[]
+ display() = 0
+ initialize() = 0
Budget
+ display()
+ initialize()
Histogram
+ display()
+ initialize()
cashflow
+ display()
+ initialize()
pie chart
+ reportType: string[]
+ display()
+ initialize()

Problem 31.5: Design a Class

Create a class diagram matching the following scenario:

> A personal finance system can have three types of users: an administrator who has complete access to the entire system, an auditor who has read-only access to the entire system, and a standard user who can only read/write those accounts specifically assigned to her.



Characterize the level of adaptability for this or for your class diagram.

This is closed because although the accounts are all linked to one parent class, they are not implemented together in a way where they share much in common

After your initial characterization of adaptability, make the following change:

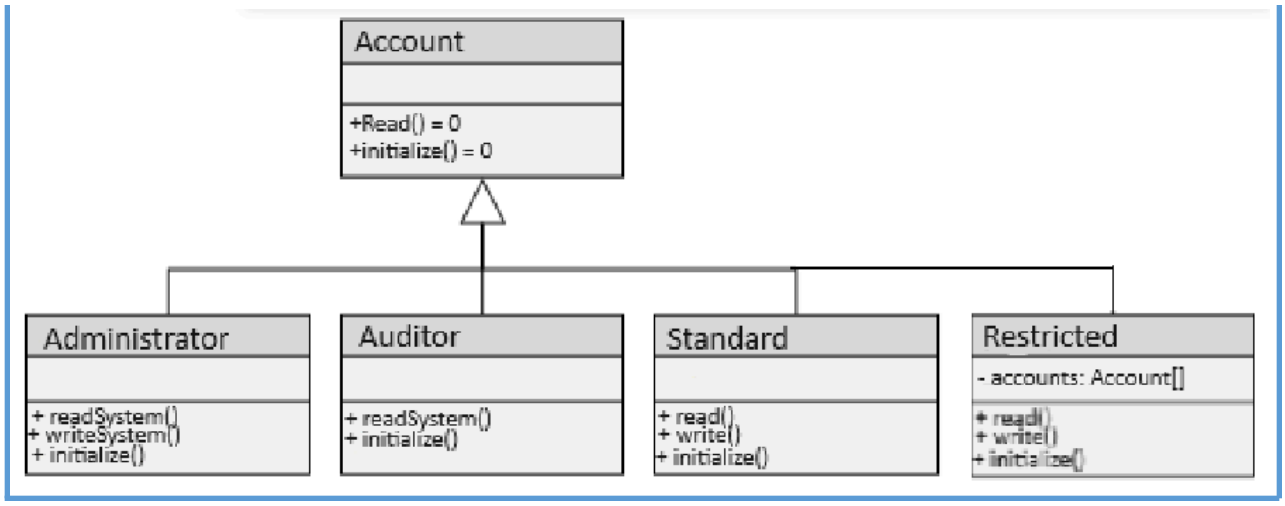
> Add a new type of user: restricted. This user can have read-only access to only her specifically assigned accounts and no other.

Describe in English what you would need to do to accommodate this change:

i need to have a read, write, and initialize function. i will also need to store the assigned accounts,

After making this change, characterize again the level of adaptability. Was your predicted and realized levels of adaptability comparable?

This is closed because although the accounts are all linked to one parent class, they are not implemented together in a way where they share much in common. i believe my prediction and realized levels of adaptability are comparable.


Account
+Read() = 0
+initialize() = 0
Administrator
+ readSystem()
+ writeSystem()
+ initialize()
Auditor
+ readSystem()
+ initialize()
Standard
+ read()
+ write()
+ initialize()
Restricted
- accounts: Account[]
+ read()
+ write()
+ initialize()