2022Fall-T-CSE460-70519 Homework 2

Wei Hng Yeo

TOTAL POINTS

92 / 100

QUESTION 1

34 pts

1.1 a 14 / 16

✓ - 2 pts The abstractions per Object Model for the loan account should not be too simple given the problem description, explanations during lectures, and discussion board.

✓ - 0 pts One class is specified. Proper relationships such as Part-Of and Is-A should be used.

✓ - 0 pts Relationship(s) should include cardinality and navigation as appropriate.

The class is too simple with only primitive data types.

1.2 b 18 / 18

✓ - 0 pts Correct

QUESTION 2

21 pts

2.1 a 7 / 7

✓ - 0 pts Correct

2.2 b 6 / 7

✓ - 1 pts The benefit identified is mostly understandable

Idea is to have separate classes, with each having its own responsibility. Thus making our system more modular. Since there is no modularity in your diag, I am deducting one point.

2.3 c 7 / 7

✓ - 0 pts Correct

QUESTION 3

45 pts

3.1 a 20 / 20

✓ - 0 pts Correct

3.2 b 20 / 25

✓ - 5 pts The choices of active abstractions should be consistent with the classes identified

incorrect classification for StartButton, PauseButton and ResetButton classes as the trigger function will be called only when the user wants to perform those functions.

1.1 a **14 / 16**

✓ **- 2 pts** The abstractions per Object Model for the loan account should not be too simple given the problem description, explanations during lectures, and discussion board.

✓ **- 0 pts** One class is specified. Proper relationships such as Part-Of and Is-A should be used.

✓ **- 0 pts** Relationship(s) should include cardinality and navigation as appropriate.

The class is too simple with only primitive data types.

1.2 b **18 / 18**

✓ **- 0 pts** Correct

2.1 a **7 / 7**

✓ **- 0 pts** Correct

2.2 b **6 / 7**

✓ **- 1 pts** **The benefit identified is mostly understandable**

Idea is to have separate classes, with each having its own responsibility. Thus making our system more modular. Since there is no modularity in your diag, I am deducting one point.

2.3 C **7 / 7**

✓ **- 0 pts** Correct

3.1 a **20 / 20**

✓ **- 0 pts** Correct

3.2 b 20 / 25

✓ **- 5 pts** **The choices of active abstractions should be consistent with the classes identified**

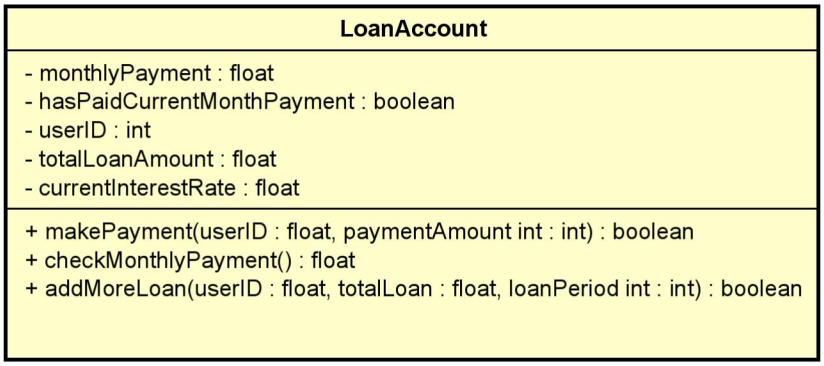
incorrect classification for StartButton, PauseButton and ResetButton classes as the trigger function will be called only when the user wants to perform those functions.

CSE 460
Software Analysis and Design
(Fall 2022)

Homework #2
Assigned: August September 5, 11:59 pm
Due: September 16, 11:59 pm

Posting ID 9816-379

1) (a)



The monthlyPayment is a floating-point attribute that will be used to store the amount of monthly payment.

The hasPaidCurrentMonthPayment is a Boolean stating whether has the owner paid the currently monthly payment or not.

The userID is the integer id of the account the user is currently holding.

The totalLoanAmount is the total amount of load the user has with this loan company.

The currentInterestRate is the amount of interest for the user loan.

The makePayment will provide the transaction process which will deduct the monthly payment from the user bank account or his credit card and then verified that the payment has gone through successfully. If the payment went through successfully then the makePayment will then set hasPaidCurrentMonthPayment to be true. If the payment is not successful, the makePayment will roll back to the original state of the system. It also returns a Boolean of whether the transaction is successful or not.

The checkMonthlyPayment will query for the owner's monthly payment through the database then print it onto the user's screen.

The addMoreLoan allows user to apply for more loan.

(b) The 2 pre-conditions for makePayments() method is that the owner must already be authenticated and the owner have not make the loan payment for the current month yet.

The post-condition method for makePayments() method is that the hasPaidCurrentMonthPayment Boolean attribute must be true after executing makePayments() method if it executes successfully.

The 2 pre-conditions for checkMonthlyPayment() method is that the owner must already be registered in the database for loan and the owner must be already be authenticated.

The post-condition for checkMonthlyPayment() method is that the current monthly payment amount must be returned to the caller and then perhaps be printed on screen for the owner to see.

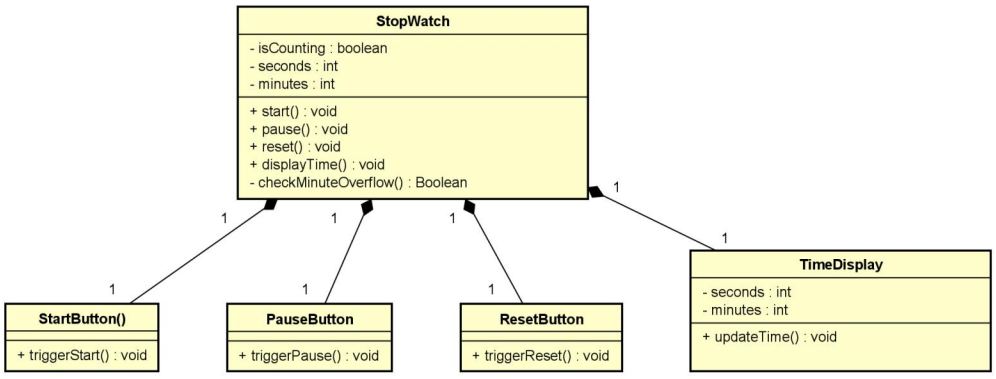
2) (a) With encapsulation of the monthlyPayment and hasPaidCurrentMonthPayment attribute, these attribute cannot be altered without the use of makePayment() method and checkMonthlyPayment() method.

Another benefit of encapsulation is that the code for the LoanAccount class will looks cleaner and more flexible. This is because we can change the code to be read-only or write-only through the getter and setter methods which in this case is makePayment() method and checkMonthlyPayment() method.

(b) One benefit of the code modularity for LoanAccount class is that the methods are separated into independent modules which do not rely on any other methods but depends on only its own attributes.

(c) The importance of both encapsulation and modularity in the LoanAccount class is that the code will be more maintainable and flexible. It is more maintainable since the methods are independent of each other. It is flexible since the attributes are being encapsulated such that if we want to change it, we can only use the getters and setters to do it. Thus, it is flexible in the sense that we only need to change the getters and setters when we want to add or remove functionality from the LoanAccount class.

3) (a)



(b) The active class are StopWatch, StartButton, PauseButton and ResetButton.

StopWatch class is the main class that controls the flow of the stopwatch on whether it is running, paused, or being reset. It has all the methods which control what the state of the stopwatch will be in. It also checks if the minute has exceeded 60, if it is exceeded, the StopWatch will be paused permanently until the reset button is depressed.

StartButton is the class that contains the button listening method (triggerStart()) that will signal StopWatch class to start running when trigger. So when the button is depressed, it generates an interrupt and triggers triggerStart() method which will call the start() method of the StopWatch. Since StartButton can alter the flow of StopWatch class program, it is an active class.

Pause button is the class that will signal the StopWatch to Pause the running of the stopwatch in the event when it is depressed. Since it can alter the flow of StopWatch class program, it is an active class.

Reset button is the class that will signal the seconds and minutes attribute of StopWatch class to be zero when depressed. It has the ability to change the flow of program since when the stopwatch is at 60 minutes, only depressing the reset button will allow StopWatch to continue to start counting again.

The passive class is TimeDisplay.

TimeDisplay class is only responsible for taking whatever input that the StopWatch class has updated to it and then displaying it on the display. It does not alter any flow of the program.