Kyoungjin Yoon

IT 206 005

Assignment 1

Purpose of GRP Class

The GRP Class collects basic inputs such as name, a choice of whether the retirement party will make a speech or not, a number of purchases, and purchase amount per each purchase. This class has mutator and accessor of each input and validates each input. This class also has a few functions to set the default budget at $1000,00 and subtracting purchase amount to the budget.

Purpose of GRPClient Class

The GRPClient Class provides user interface to have the user type name, a choice of whether the retirement party will make a speech or not, a number of purchases, and purchase amount per each purchase. If any error occurs, the program prints error messages and re-prompts user for the input. It prints output such as name, a choice of whether the retirement party will make a speech or not, a number of purchase, and total amount of purchases.

UML Class Diagram

|  |  |
| --- | --- |
| GRP Class | |
| Instance Variables | The variables are instantiated after the object is created in the RetirementImplementation class. |
| (-) clientName: String | A name of the retirement party |
| (-) speecChoice : String | Whether the retirement part will make a speech or not |
| (-) noItemsPurchased: Integer | A number of purchase to make |
| (-) purchaseAmount: double | Amount spent per each number of purchase |
| (-) clientBudget: double | Default budget value is set to 1000 when the constructor is instantiated. |
| Constructor |  |
| (+) Retirement () | This constructor will run and default budget will be 1000.00 when the budget is not specified in the implementation class. |
| (+) Retirement (budget: double) | This constructor will run when the budget is specified in the implementation class. |
| Accessor |  |
| (+) getClientName () String | Returns name |
| (+) getSpeechChoice () String | Returns a choice of speech |
| (+) getNoItemsPurcahsed () Integer | Returns a number of purchases |
| (+) getPurchaseAmount () double | Returns purchase amount per a number of purchases |
| Mutator |  |
| (+) setClientName(clientName: String) Boolean | Returns false if the user did not put the name input. Otherwise it sets the name |
| (+) setSpeechChoice(speechChoice: String) Boolean | Returns false if the input is not ‘Yes’ or ‘No’. Otherwise it sets the speech decision. |
| (+) setNoItemsPurcahsed(noItemsPurchase: Integer) Boolean | Returns false if the purchase number is less than 0 or a letter. Otherwise it sets a number of purchases. |
| (+) setPurcahseAmount(purchaseAmount: double) Boolean | Returns false if the input is negative value. Otherwise it sets purchase amount. |
| Special Methods |  |
| (+) spendMoney (): double | Deducts the purchase amount to the remaining budget |
| (+) exceedingMoney () Boolean | Verifies whether the purchase amount is more than the remaining budget. |
| (+) toString() String | Returns string value of whether the retirement party will make a speech or not. |
|  |  |
|  |  |
|  |  |
|  |  |

Pseudocode of Retirement Class

Begin GRP

Set name

Set speech

Set purchaseNumber

Set purchase

Set budget

Set Retirement

Set Retirement(budget)

Begin setName (theName: String)

If name’s number of letters is less than 0 then

Return false

Else

Set name to theName

Endif

End

Begin GRP()

Set clientBudget to 1000.00

End

Begin GRP(clientBudget: double)

Set clientBudget to clientBudget parameter

End

Begin setClientName

Set clientName to clientName parameter

End

Begin setSpeechChoice

If speechChoice equals “Yes” or “No” then

Set clientChoice to ClientChoice parameter

Return true

Else

Return false

Endif

End

Begin setNoItemsPurchased

If noItemsPurchased is less than and equal to zero then

Return false

Else

Set noItemsPurchased to noItemsPurchased

Return true

Endif

End

Begin setPurchaseAmount

If purchaseAmount is less than and equal to 0 then

Return false

Else

Set purchaseAmount to purchaseAmount Parameter

Endif

End

Begin getClientName

Return clientName

End

Begin getSpeechChoice

Return speechChoice

End

Begin getNoItemsPurchased

Return noItemsPurchased

End

Begin getPurchaseAmount

Return purchaseAmount

End

Begin getClientBudget

Return clientBudget

End

Begin spendMoney

If (clientBudget – purchaseAmount) less than 0 then

Return clientBudget

Else

Return clientBudget = clientBudget – purchaseAmount

Endif

End

Begin exceedingMoney

If (clientBudget – purchaseAmount) is less than 0 then

Return true

Else

Return false

Endif

End

Begin toString

If speechChoice is equal to “Yes” then

Return “this retirement party will make a speech.”

Else

Return “this retirement party will not make a speech.”

Endif

End

Pseudocode of GRPClient

Begin GRPClient

Set DEFAULT\_BUEGET TO 2000.00

Create an object of GRP called customer1 with DEFAULT\_BUDGET parameter

createInformation

calculateBudgetAmount

printResult

end

begin createInformation

prompt user for customer1. setClientName

set customer speech to “”

repeat

prompt and read user for customer speech

if customer1.setSpeechChoice(customer speech: String) is false then

generate an error and reprompt (customer speech: String)user for another input

endif

until customer1.setSpeechChoice is true

end

begin calculateBudgetAmount

set purchaseQuantity to 0

repeat

prompt and read user for purchaseQuantity

if customer1.setNoItemsPurchased (purchaseQuantity: int) is false and letter then

generate an error and repormpt user for another input

endif

until customer1.setNoItemsPurchased (purchaseQuantity: int) is true

set spending to 0.00

do 1 to customer1.getNoItemsPurchased

if customer1.getClientBudget is equal to 0 then

break the do loop

endif

repeat

prompt the user for and read spending

if customer1.setPurchaseAmount(spending: double) is false then

generate an error and reprompt user for another input

endif

if customer1.getClientBudget – customer1.getPurchaseAmount is less than 0 then

generate an error and reprompt user for another input

customer1.spendMoney

if customer1.getClientBudget – customer1.getPurcahseAmount is less than 0 then

decrement loop count

endif

if customer1.getClientBudget is less than customer.getPurachseAmount

increment loop count

endif

if customer1.getClientBudget is equal to 0 then

generate an error message: “the party spent all of the budget.”

Break the do loop

Endif

Until customer1.setPurchaseAmount(spending: double) OR customer.exceedingMoney are true

End

Begin printResult

Print customer1.getClientName, customer1.getNoItemsPurchased, customer1.toString, (DEFAULT\_BUDGET – customer1.getClientBudget), and customer1.getClientBudget.

end

end