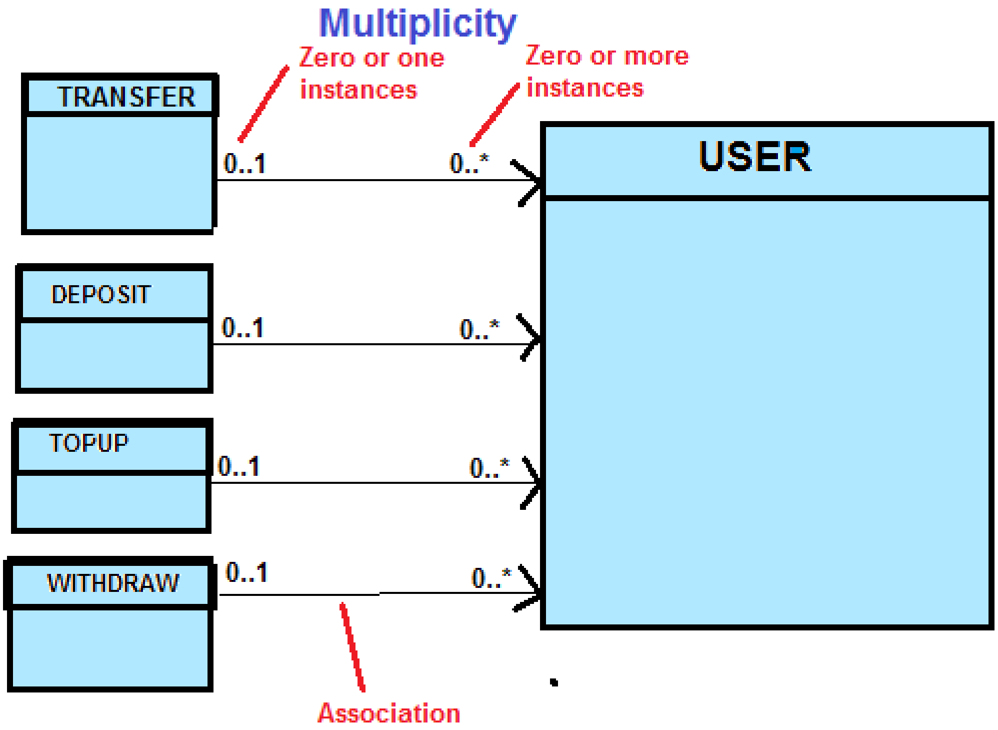
Class Diagrams

1. PC Employees

The PC Employees system in the bank is broken down into six classes. The first four classes which is transfer, deposit, top up, withdraw and under each of this classes there is a subclass which hold the database information for each function and customer.

There are two more main classes for the stock options which is Stock sell and Stock buy and under this there are two sub classes containing the variables and methods for the national and international stock.

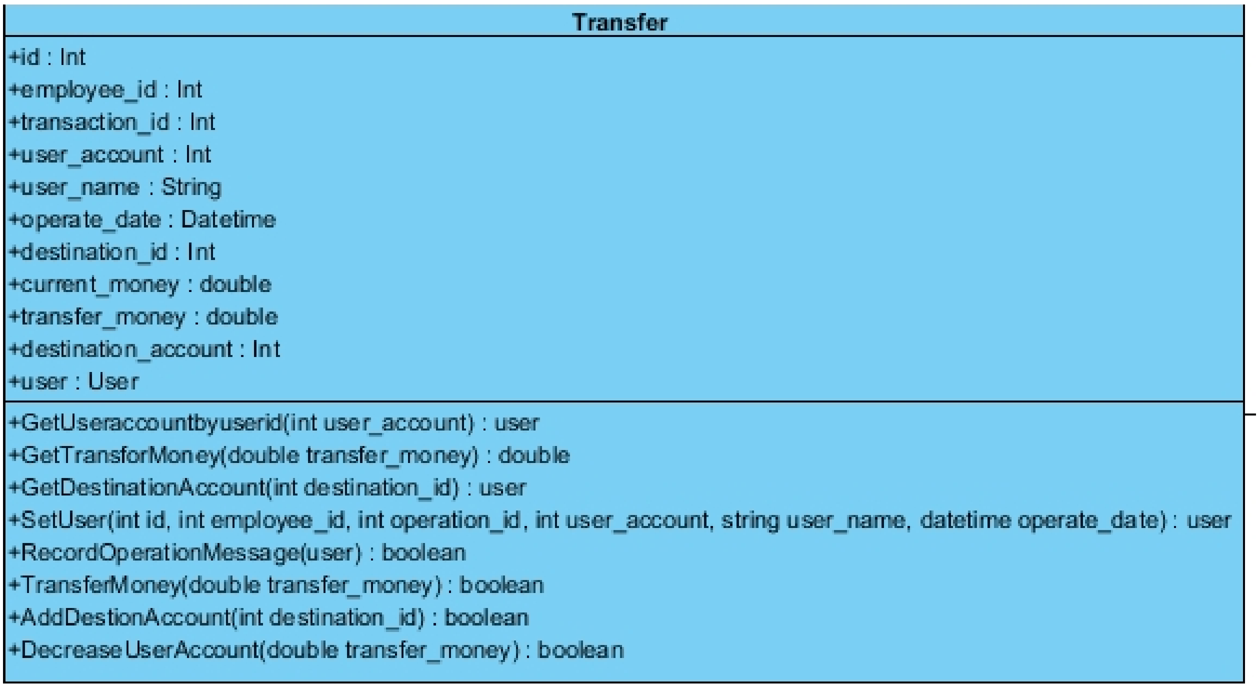
Between these classes there is an association relationship, where one class uses another class but without inheritance between the two classes.   The relationship is represented with a line between two classes with attributes both end of the line.



So in this case all the classes in PC Employees uses and contains an instance of user class.

Here are the two classes to transfer money and deposit money through the PC Employees. In this case, the system can have 0 or more transfer, deposit, top up and withdraw for the user. This relationship is a multiplicity association where many transfer, deposit, top up and withdraw can be processed from the user account by the bank. The plus sign denotes the visibility which means other classes can view this information.

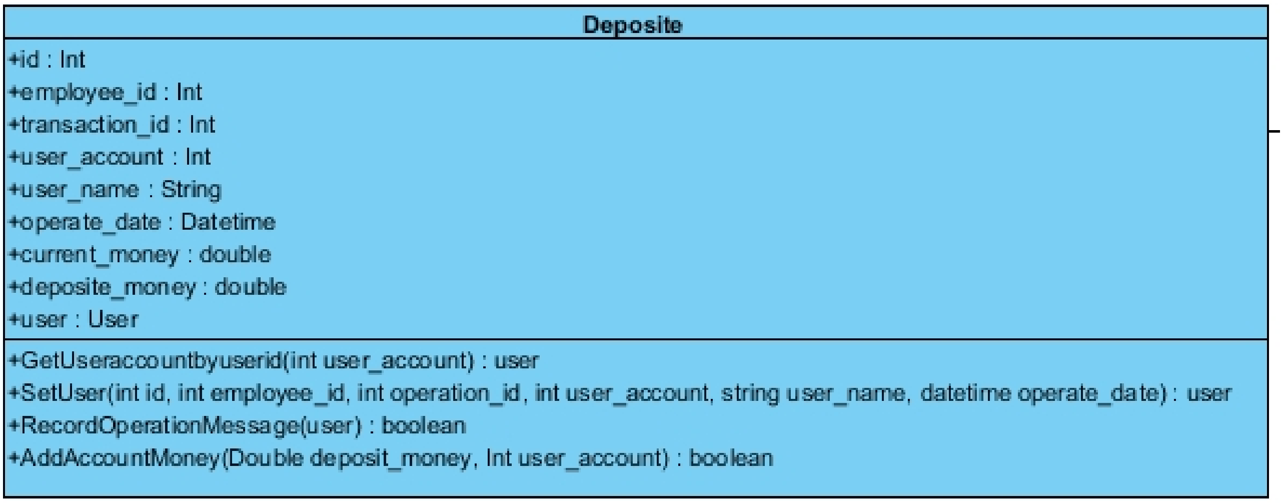
The class name, attributes and the methods for this class are represented as follows:



Transfer

+id : Int
+employee_id : Int
+transaction_id : Int
+user_account : Int
+user_name : String
+operate_date : Datetime
+destination_id : Int
+current_money : double
+transfer_money : double
+destination_account : Int
+user : User

+GetUseraccountbyuserid(int user_account) : user
+GetTransforMoney(double transfer_money) : double
+GetDestinationAccount(int destination_id) : user
+SetUser(int id, int employee_id, int operation_id, int user_account, string user_name, datetime operate_date) : user
+RecordOperationMessage(user) : boolean
+TransferMoney(double transfer_money) : boolean
+AddDestionAccount(int destination_id) : boolean
+DecreaseUserAccount(double transfer_money) : boolean



Deposite

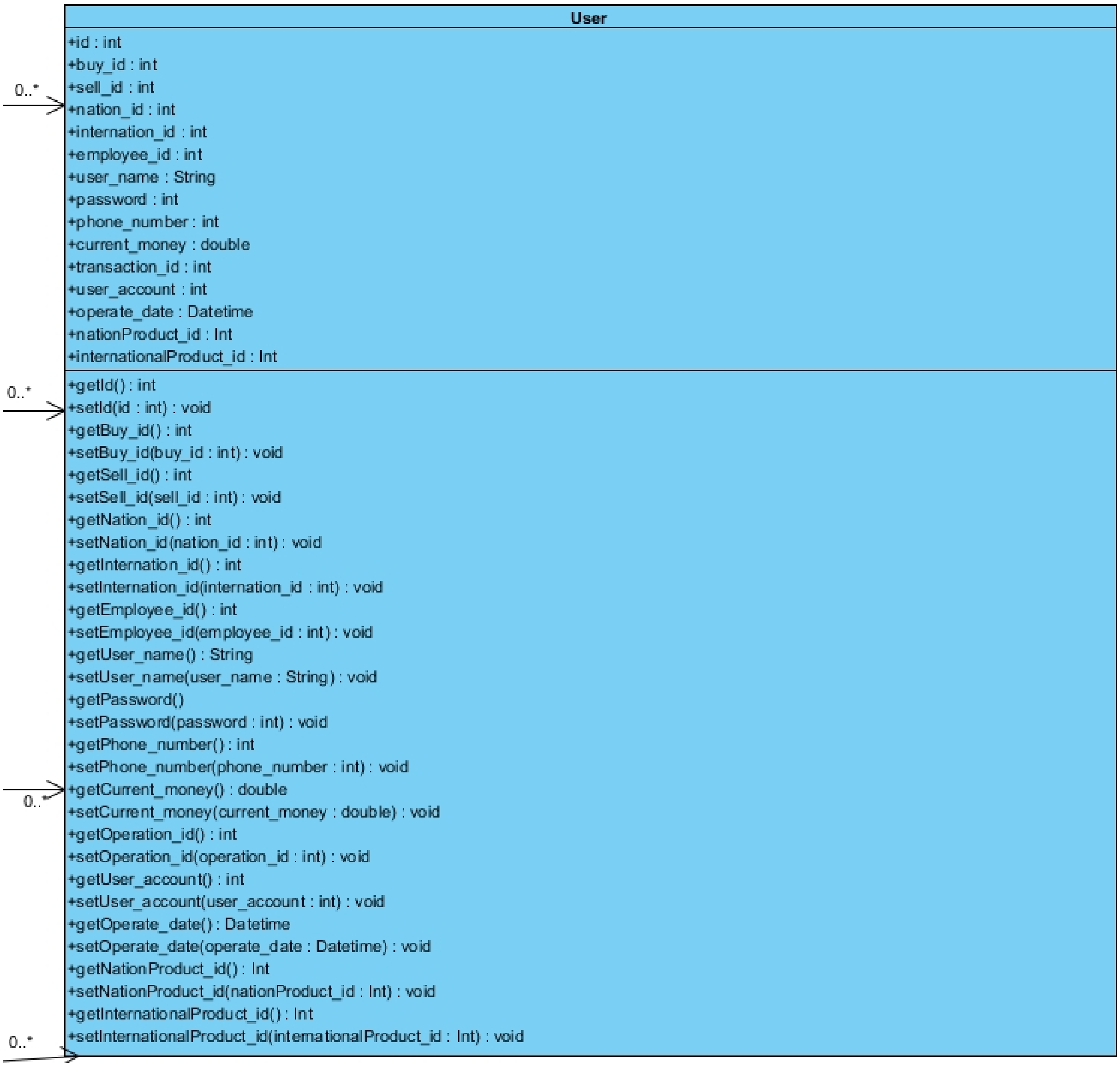
+id : Int
+employee_id : Int
+transaction_id : Int
+user_account : Int
+user_name : String
+operate_date : Datetime
+current_money : double
+deposite_money : double
+user : User

+GetUseraccountbyuserid(int user_account) : user
+SetUser(int id, int employee_id, int operation_id, int user_account, string user_name, datetime operate_date) : user
+RecordOperationMessage(user) : boolean
+AddAccountMoney(Double deposit_money, Int user_account) : boolean

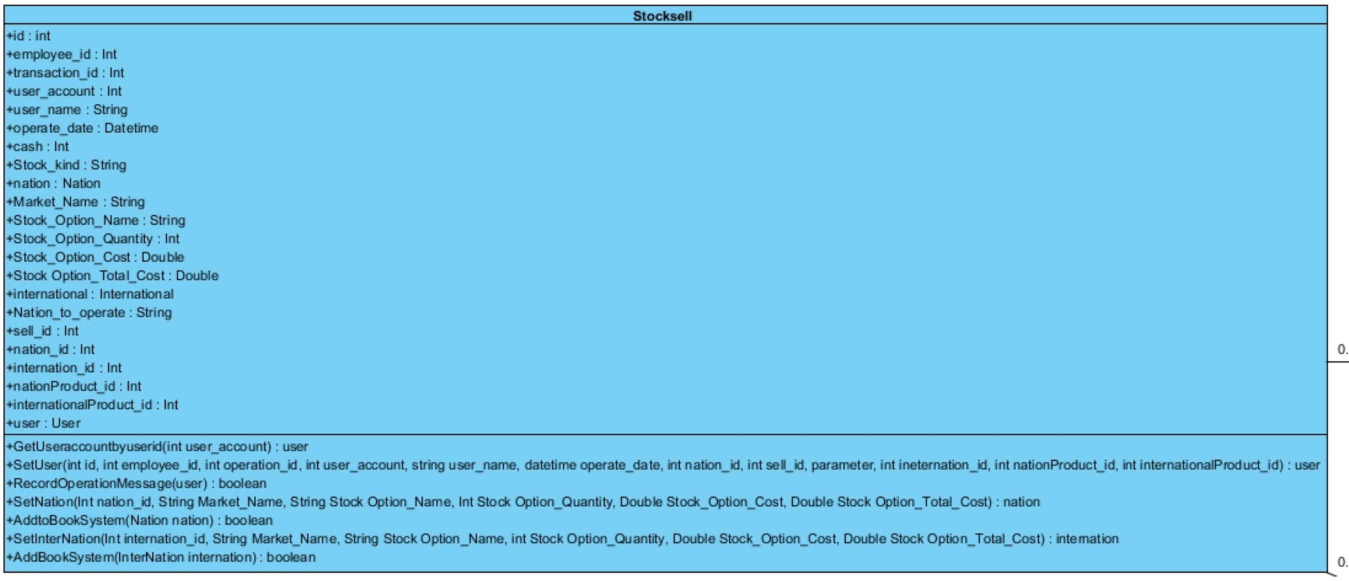
In transfer class what we assumed here is that the system communicates with users and obtain user account information, including current account balances and the user result returned here is passed in the next method. The Record operation message method keep records of relevant information such as employee operation, employee and customer, operation time, etc.

We have used a similar idea in deposit class as well.

These four main classes contain a subclass which is user holding all the corresponding details of the customers. This user class is related to the four other class mentioned above. Below you can see the user class in our class diagram.

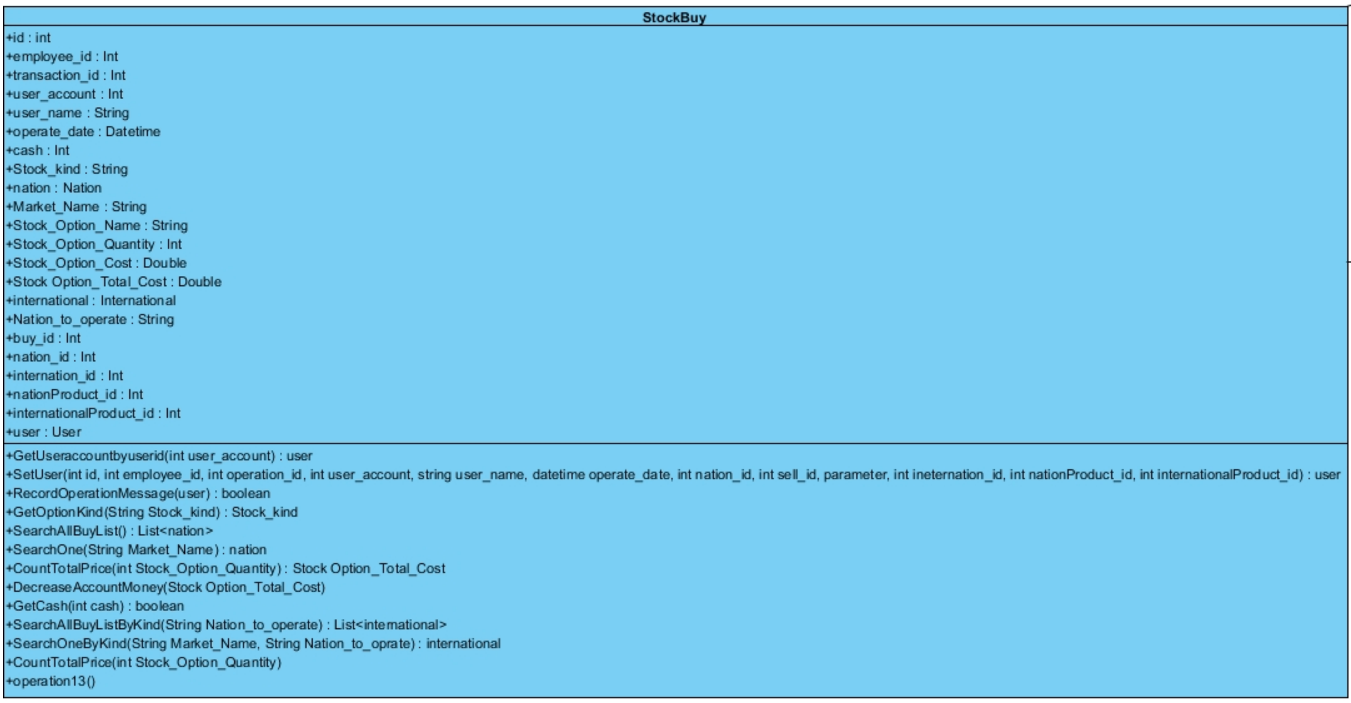

User
+id : int
+buy_id : int
+sell_id : int
+nation_id : int
+internation_id : int
+employee_id : int
+user_name : String
+password : int
+phone_number : int
+current_money : double
+transaction_id : int
+user_account : int
+operate_date : Datetime
+nationProduct_id : Int
+internationalProduct_id : Int
+getId() : int
+setId(id : int) : void
+getBuy_id() : int
+setBuy_id(buy_id : int) : void
+getSell_id() : int
+setSell_id(sell_id : int) : void
+getNation_id() : int
+setNation_id(nation_id : int) : void
+getInternation_id() : int
+setInternation_id(internation_id : int) : void
+getEmployee_id() : int
+setEmployee_id(employee_id : int) : void
+getUser_name() : String
+setUser_name(user_name : String) : void
+getPassword()
+setPassword(password : int) : void
+getPhone_number() : int
+setPhone_number(phone_number : int) : void
+getCurrent_money() : double
+setCurrent_money(current_money : double) : void
+getOperation_id() : int
+setOperation_id(operation_id : int) : void
+getUser_account() : int
+setUser_account(user_account : int) : void
+getOperate_date() : Datetime
+setOperate_date(operate_date : Datetime) : void
+getNationProduct_id() : Int
+setNationProduct_id(nationProduct_id : Int) : void
+getInternationalProduct_id() : Int
+setInternationalProduct_id(internationalProduct_id : Int) : void
0..*
0..*
0..*
0..*

There are two more main classes in   PC Employees which is Stock sell and stock buy and these two classes has the same sub class which is Nation and International which contains the variables and methods that has to include during the International and National stock sell and buy operation. The stock sell and stock buy classes also have a similar association relationship between the two sub classes which   is Nation and Internation.



```
Stocksell
+id : int
+employee_id : Int
+transaction_id : Int
+user_account : Int
+user_name : String
+operate_date : Datetime
+cash : Int
+Stock_kind : String
+nation : Nation
+Market_Name : String
+Stock_Option_Name : String
+Stock_Option_Quantity : Int
+Stock_Option_Cost : Double
+Stock Option_Total_Cost : Double
+international : International
+Nation_to_operate : String
+sell_id : Int
+nation_id : Int
+internation_id : Int
+nationProduct_id : Int
+internationalProduct_id : Int
+user : User
+GetUseraccountbyuserid(int user_account) : user
+SetUser(int id, int employee_id, int operation_id, int user_account, string user_name, datetime operate_date, int nation_id, int sell_id, parameter, int ineternation_id, int nationProduct_id, int internationalProduct_id) : user
+RecordOperationMessage(user) : boolean
+SetNation(Int nation_id, String Market_Name, String Stock Option_Name, Int Stock Option_Quantity, Double Stock_Option_Cost, Double Stock Option_Total_Cost) : nation
+AddtoBookSystem(Nation nation) : boolean
+SetInterNation(Int internation_id, String Market_Name, String Stock Option_Name, int Stock Option_Quantity, Double Stock_Option_Cost, Double Stock Option_Total_Cost) : internation
+AddBookSystem(InterNation internation) : boolean
```

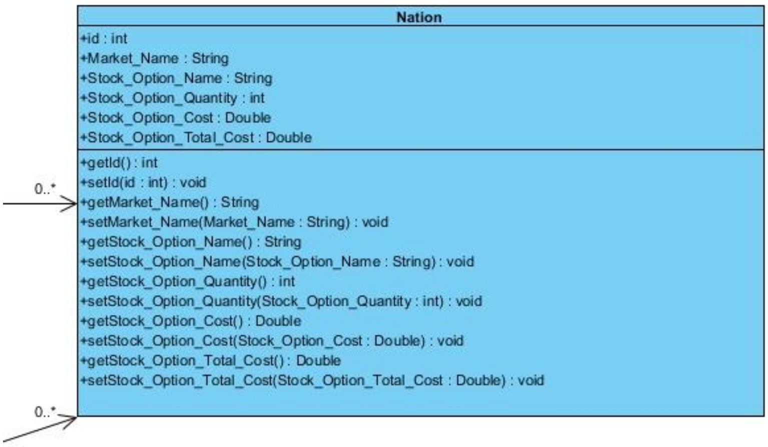
This is the other main class for the PC Employee with the variable, methods and attribute needed for stock buy operation.

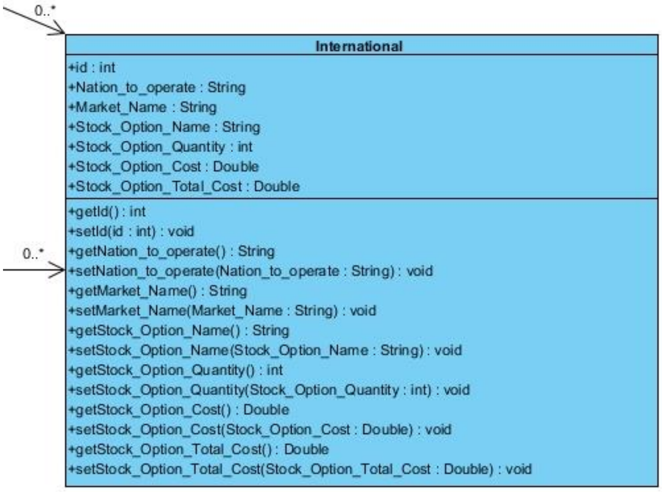


```
StockBuy
+id : int
+employee_id : Int
+transaction_id : Int
+user_account : Int
+user_name : String
+operate_date : Datetime
+cash : Int
+Stock_kind : String
+nation : Nation
+Market_Name : String
+Stock_Option_Name : String
+Stock_Option_Quantity : Int
+Stock_Option_Cost : Double
+Stock Option_Total_Cost : Double
+international : International
+Nation_to_operate : String
+buy_id : Int
+nation_id : Int
+internation_id : Int
+nationProduct_id : Int
+internationalProduct_id : Int
+user : User
+GetUseraccountbyuserid(int user_account) : user
+SetUser(int id, int employee_id, int operation_id, int user_account, string user_name, datetime operate_date, int nation_id, int sell_id, parameter, int ineternation_id, int nationProduct_id, int internationalProduct_id) : user
+RecordOperationMessage(user) : boolean
+GetOptionKind(String Stock_kind) : Stock_kind
+SearchAllBuyList() : List<nation>
+SearchOne(String Market_Name) : nation
+CountTotalPrice(int Stock_Option_Quantity) : Stock Option_Total_Cost
+DecreaseAccountMoney(Stock Option_Total_Cost)
+GetCash(int cash) : boolean
+SearchAllBuyListByKind(String Nation_to_operate) : List<international>
+SearchOneByKind(String Market_Name, String Nation_to_oprate) : international
+CountTotalPrice(int Stock_Option_Quantity)
+operation13()
```

In this class for stock buy operation in pc employees we have mentioned a nation id which will be set to 0 according to the selection of stock option. We have assumed that the nation and international stock options have the same choice.

These are the Nation and Internation class that is related to the stock sell and buy class.


Nation
+id : int
+Market_Name : String
+Stock_Option_Name : String
+Stock_Option_Quantity : int
+Stock_Option_Cost : Double
+Stock_Option_Total_Cost : Double
+getId() : int
+setId(id : int) : void
+getMarket_Name() : String
+setMarket_Name(Market_Name : String) : void
+getStock_Option_Name() : String
+setStock_Option_Name(Stock_Option_Name : String) : void
+getStock_Option_Quantity() : int
+setStock_Option_Quantity(Stock_Option_Quantity : int) : void
+getStock_Option_Cost() : Double
+setStock_Option_Cost(Stock_Option_Cost : Double) : void
+getStock_Option_Total_Cost() : Double
+setStock_Option_Total_Cost(Stock_Option_Total_Cost : Double) : void
0..*
0..*


0..*
International
+id : int
+Nation_to_operate : String
+Market_Name : String
+Stock_Option_Name : String
+Stock_Option_Quantity : int
+Stock_Option_Cost : Double
+Stock_Option_Total_Cost : Double
+getId() : int
+setId(id : int) : void
+getNation_to_operate() : String
+setNation_to_operate(Nation_to_operate : String) : void
+getMarket_Name() : String
+setMarket_Name(Market_Name : String) : void
+getStock_Option_Name() : String
+setStock_Option_Name(Stock_Option_Name : String) : void
+getStock_Option_Quantity() : int
+setStock_Option_Quantity(Stock_Option_Quantity : int) : void
+getStock_Option_Cost() : Double
+setStock_Option_Cost(Stock_Option_Cost : Double) : void
+getStock_Option_Total_Cost() : Double
+setStock_Option_Total_Cost(Stock_Option_Total_Cost : Double) : void
0..*

In this two classes there is a very small difference between the variables and attributes. We have included extra variables and methods in international stock option. For instance, the nation to operate in this class will choose the country to operate whereas for national stock option we didn’t include that since we assumed its only for china’s stock.