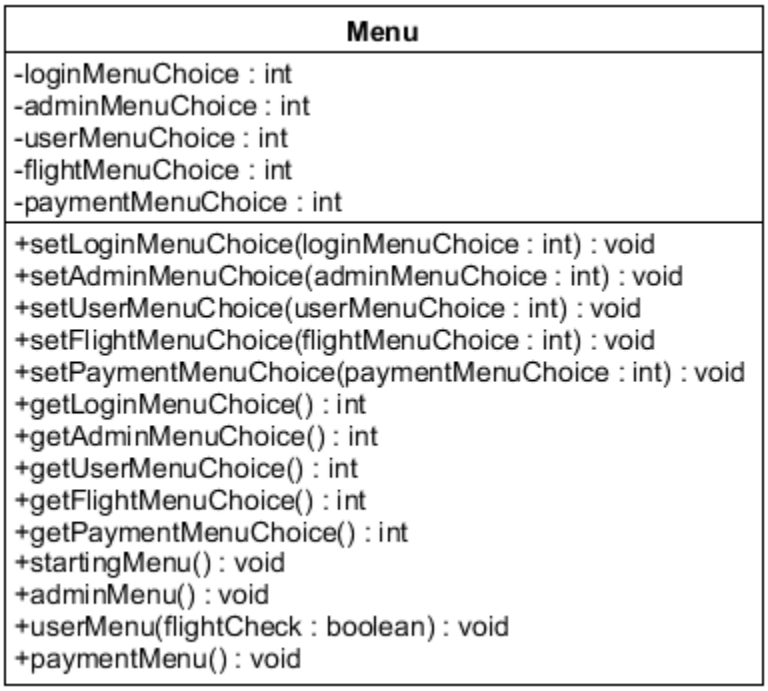
Finals (Airlines)

Airlines has 7 classes:

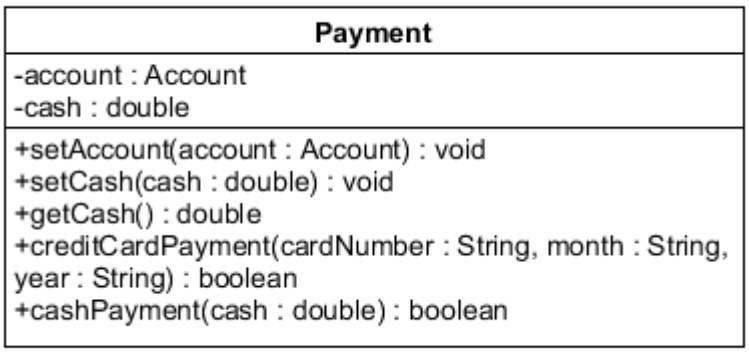
Menu class



Menu class has all the choice variables and some menus you will need throughout the program.

- startingMenu – has the Login and Exit choices.
- adminMenu – has the Edit Flight Details and Logout choices.
- userMenu – will take a Boolean parameter. Has the View Account, Edit Account, View Flight, Book Flight/Edit Flight, Payment and Logout choices. Book flight will occur if the argument you passed on is false, otherwise Edit Flight will occur at the menu.
- paymentMenu – has the Credit Card Payment, Cash Payment and a Back choice.

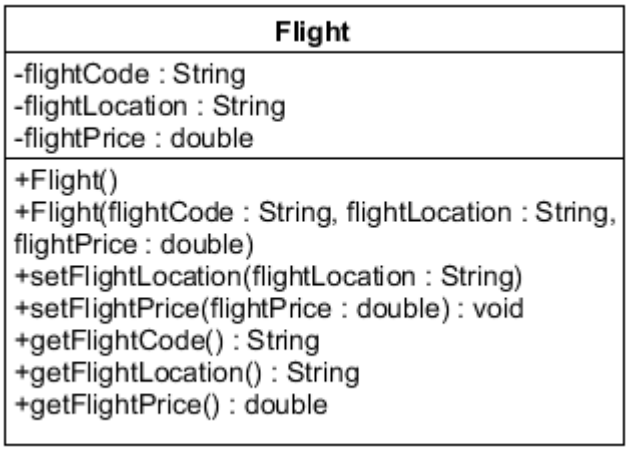
Note: I didn't include the Flight Menu since you will be changing the flight details throughout the program.



Payment class will have a link to the account class. Also has a cash variable for the cash payment feature.

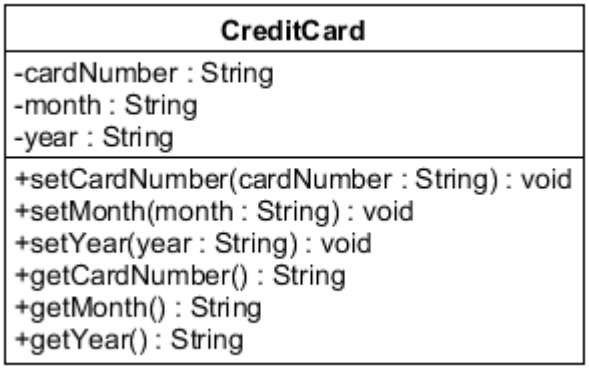
- creditCardPayment – takes three parameters and will return a Boolean value. Return true if all the arguments match to the account's credit card details. Print a success payment, otherwise return false.
- cashPayment – takes one parameter and will also return a Boolean value. Upon selecting the cash payment, you will be printing out the account holder's name and the flight details. Afterwards, input for the amount you will be paying. Return true if the amount is equal or higher to the flight price. Print a change if there's any. Return false if the input is less than 0 and if the input amount is less than the flight price.

Note: Don't forget to put messages for every actions made, valid or invalid.



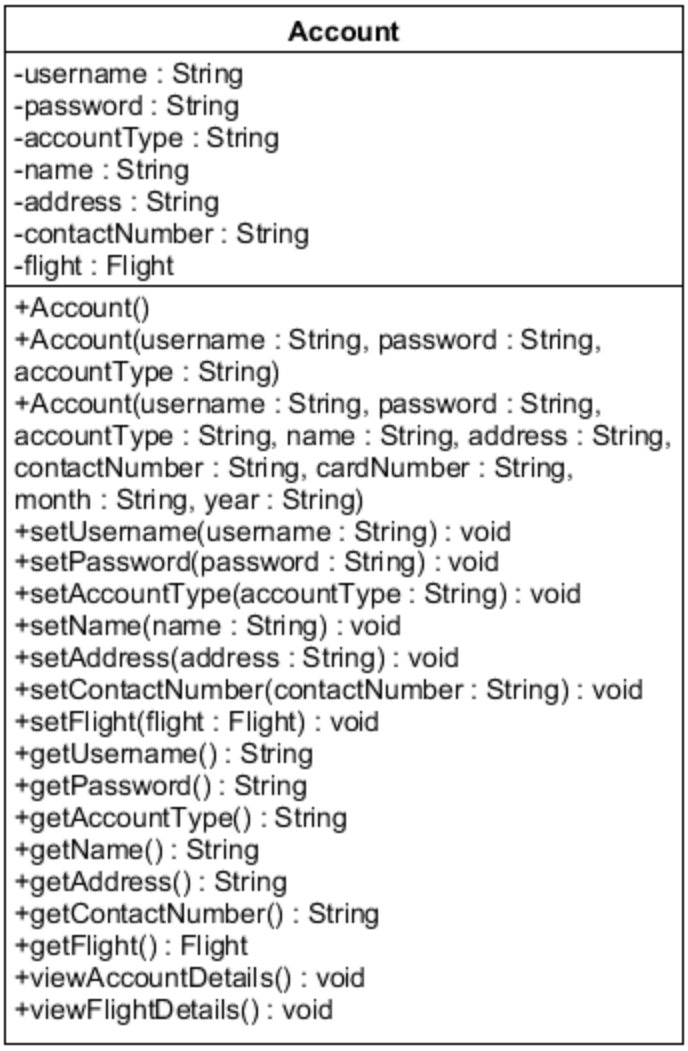
Flight class is the parent for the CreditCard class.

Flight class has the flight details for the flight. Create an object through the parameterized constructor. For the no parameter constructor, just leave it blank.



CreditCard class is the child class of Flight and the parent class of Account.

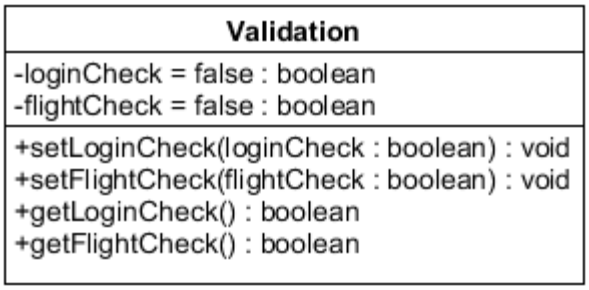
CreditCard class has the details of the cc.



Account class is the child class of CreditCard and the parent class of Validation.

Account class has the details of the account. Has 3 constructor methods. For the no parameter, leave it blank. The 3-parameter constructor is for the admin account. The 9-parameter constructor is for the user account.

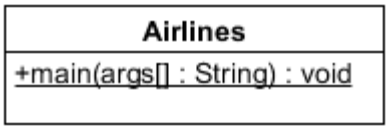
- viewAccountDetails – prints out some details of the account holder: username, name, address, and contact number.
- viewFlightDetails – prints out the details of the booked flight: flight code, flight location, and flight price.



Validation class is the child class of Account inheriting all the public methods from the very root or the first parent.

Validation class holds all the variables responsible for checking, matching, or validating details of the account, credit card, and flight classes.

It has unique attributes like loginCheck and flightCheck that are both in Boolean for efficient validation of the program.



Airlines

Airlines starts on invoking the login menu, wherein it will loop till the user chooses to logout. Print out a thank you message after choosing the exit option.

Note: Put Invalid messages for invalid character inputs, including pressing of enter.

Upon choosing login, the program will proceed into checking the username and password of the account.

Note: There are two kinds of account in this program: an admin account, and a user account. The checking of account is only available for three tries. Put an error message for every wrong login and put a going back to the login menu message signaling that you have tried logging in for three times.

If the user logs in an admin account, it will invoke the admin menu, and loginCheck state change to true.

Inside the admin menu, the admin has features for changing the flight details: flight location and flight price. Flight code won't be change throughout the program. Upon choosing the edit flight details, you will proceed to a menu letting you choose what flight to change.

Note: Entering a blank space will not change the details of the flight.

Choosing the logout option will let you go back to the login menu, and changing the loginCheck state to false.

Note: Put Invalid messages for invalid character inputs, including pressing of enter key.

If the user login in a user account, it will invoke the user menu, and loginCheck state change to true.

Inside the user menu, there are 5 options and 1 option for logout.

- First option: View Account – will invoke the viewAccountDetails method.
- Second option: Edit Account – will let you change the username, password, name, address, or contact number of the account. **Note: Entering just a blank space won't change the details of the account.**
- Third option: View Flight – will invoke the viewFlightDetails if the flightCheck is true. If flightCheck is false, print a message of no booked flight yet.
- Fourth option: Book Flight – occurs if the flightCheck is false. It will invoke a flight menu selecting what flight you want to book. Should print out a message of you have chosen the route xxxxxxx. Has also a back button to let you go back to the user menu. **Note: Don't forget to print out invalid input of character. When you have chosen a flight, flightCheck state will change to true.**
- Fourth option: Edit Flight – occurs if the flightCheck is true. Same as the Book Flight, letting you change your current chosen flight. Has a back button.
- Fifth option: Payment – will invoke the payment menu if the flightCheck is true. If flightCheck is false, print out a "can't proceed. No booked flight yet."

- Sixth option: Logout – will change the state of the loginCheck to false and print out a going back to login menu message.

Payment Menu – will have two options for credit card payment and cash payment. Also has a back button.

- Credit Card Payment – will print out the account holder's name then ask for the credit card details: cc number, month and year. Month can be in English format or a number format. Invoke the creditCardPayment method.
- Cash Payment – will print out the account holder's name then will ask how much you will be paying for the flight. Invoke the cashPayment method.

Note: Check for possible terminating characters or errors. Program should be running smoothly avoiding major or minor errors.

For default values:

One object for menu class.

One object for the validation/checker class.

One object for the payment class.

Two objects for the account class.

First object – will be the admin account having:

Username: admin
Password: admin
Account Type: admin

Second object – will be the user account having:

Username: user
Password: user
Account Type: user
Name, Address and Contact Number depends on you
Credit Card Number: 1234567890
Month: February or 2
Year: 2021

Three objects for the flight class.

First object – Flight Code: L43Z6F – Flight Location: Seoul, South Korea – Flight Price: 5000.00

Second object – Flight Code: 5PNB61 – Flight Location: Akihabara, Japan – Flight Price: 4500.00

Third object – Flight Code: 1KL78H – Flight Location: Beijing, China – Flight Price: 4000.00

One static variable for tries although if you can make it an OOP, it's a bonus.

One object for the Scanner class.

Note: To make an inheritance parent-child relationship:

public class child extends parent

MORE NOTES: DON'T FORGET TO MAKE A FLOWCHART OF THIS PROGRAM. YOU CAN USE ANYTHING YOU WANT.