Even Semester (2023)



**BINUS UNIVERSITY**

**BINUS INTERNATIONAL**

**Assignment Cover Letter**

**(Individual Work)**

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | | | | |  | |  | |  |
| **Student Information**: **Surname** | | | | | **Given Names**    **Mikhael** | | **Student ID Number**  **2301890245** | |  |
| 1. | | **Salim** |  | |  |
|  |  |  |
| **Course Code** | **: COMP6510** |  |  | | **Course Name** | | **: Programming Languages** | |  |
| **Class** | **: L2AC** |  |  | | **Name of Lecturer(s)** | | **:** **: Jude Joseph Lamug Martinez** | |  |
|  |  |  |  | |  | |  | |  |
| **Major** | **: Computer Science** |  |  | |  | |  | |  |
| **Title of Assignment**  (if any) | : ATM Machine | |  |  | |  | |  | |
| **Type of Assignment**    **Submission Pattern** | **: Final Project** |  |  | |  | |  | |  |
| **Due Date** | **: 20 June 2020** |  |  | | **Submission Date** | | **: 19 June 2020** | |  |

The assignment should meet the below requirements.

1. Assignment (hard copy) is required to be submitted on clean paper, and (soft copy) as per lecturer’s instructions.
2. Soft copy assignment also requires the signed (hardcopy) submission of this form, which automatically validates the softcopy submission.
3. The above information is complete and legible.
4. Compiled pages are firmly stapled.
5. Assignment has been copied (soft copy and hard copy) for each student ahead of the submission.

# Plagiarism/Cheating

BiNus International seriously regards all forms of plagiarism, cheating and collusion as academic offenses which may result in severe penalties, including loss/drop of marks, course/class discontinuity and other possible penalties executed by the university. Please refer to the related course syllabus for further information.

# Declaration of Originality

By signing this assignment, I understand, accept and consent to Binus International terms and policy on plagiarism. Herewith I declare that the work contained in this assignment is my own work and has not been submitted for the use of assessment in another course or class, except where this has been notified and accepted in advance.

Signature of Student: (Name of Student)

1. Mikhael Angeloputra Salim



**“ATM Machine”**

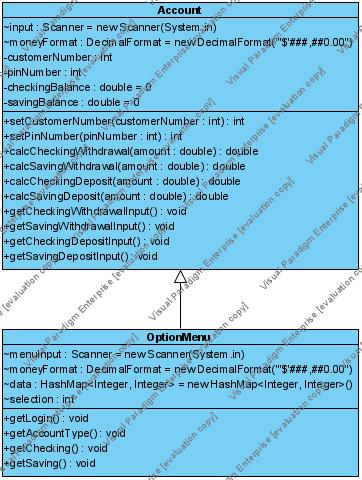
**Name : Mikhael Angeloputra Salim**

**ID : 2301890245**

1. **Project Specifications**

ATM Machine is an ATM Application which is used to enable any user to perform financial transactions, such as cash withdrawals, deposits, or account balance inquiries, at any time and without the need to interact with a person. Note that this ATM Machine only accepts a certain amount of Customer Number and PIN that has been set up inside the data. Customer Numbers and PIN numbers beside the one that has been set up will not be able to be used and will immediately tell the user that they have entered the wrong PIN / Customer Number.

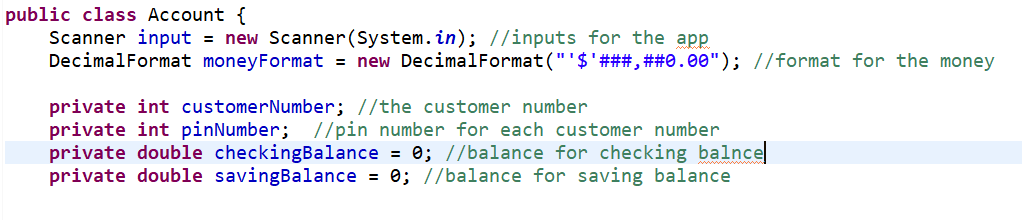
1. **Solution Design**

**Class Diagram**

****

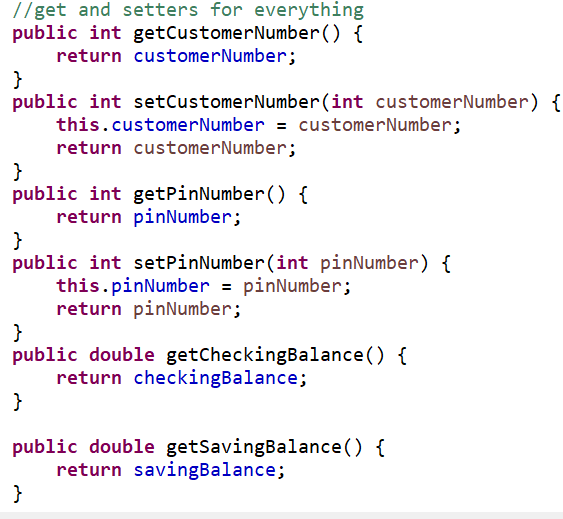
1. **A Discussion about what is Implemented and How it works**



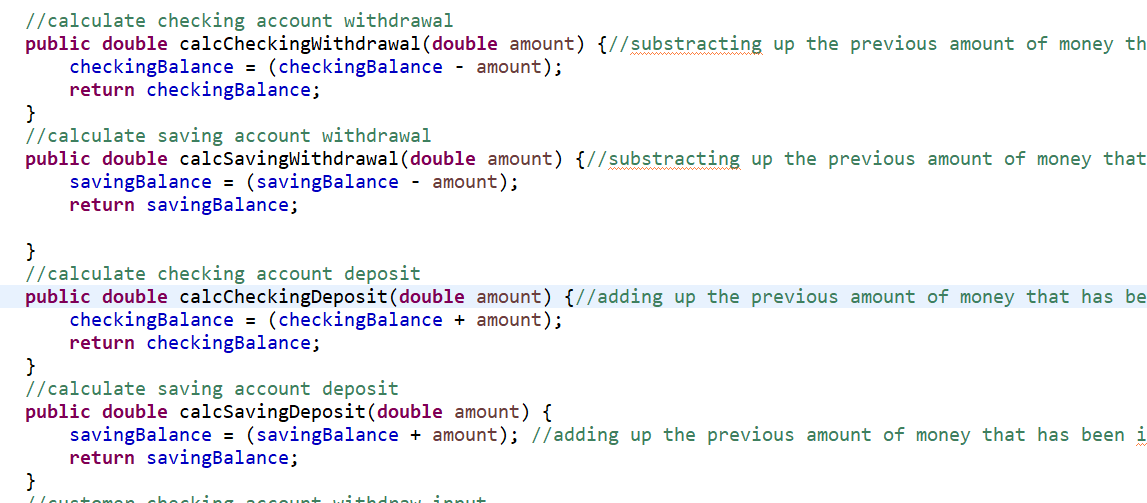
Allows the app to have a decimal format.

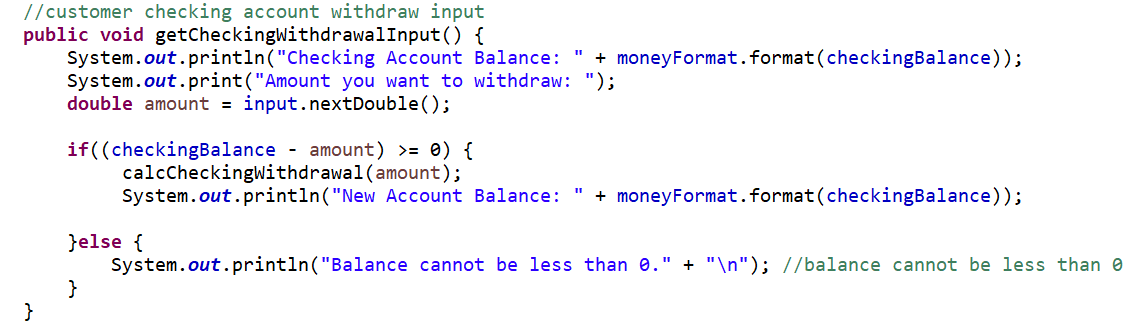
The account class is the super class of this program, inside the account class there is 4 instance variables which is, customerNumber, pinNumber, checkingBalance, and savingBalance. The class also has a decimal format which allows the app to have a money format of its own.

checkingBalance and savingBalance has the value set to 0 (as for any balance account would).

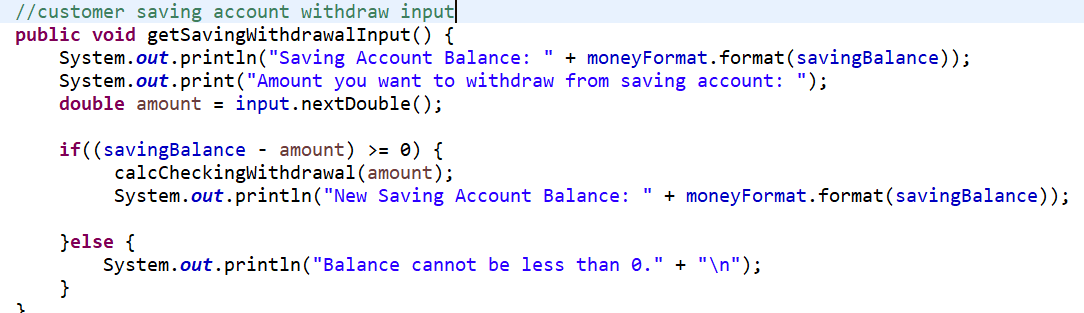


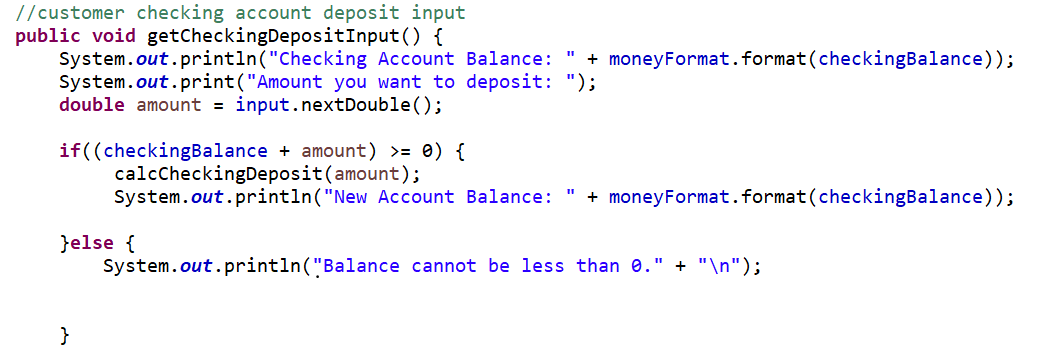
A get and setters for every instance variable there is available in the Account class.

Calculate methods for each saving and checking account and for each withdrawal(taking out an amount of money from either saving/checking account) and deposits(adding an amount of money to either saving/checking account) that has been done by the user.

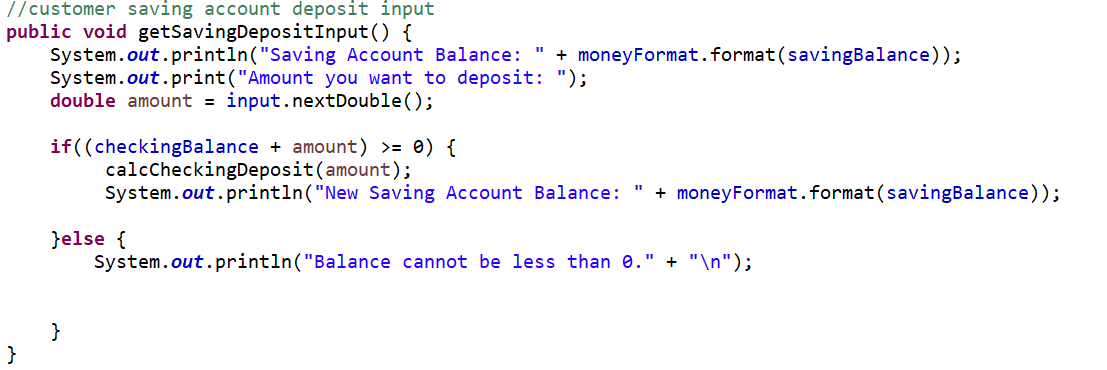


Method to check whether the user withdrawed/inputted the correct amount of money inside the checking account and the amount cannot be less than 0 (which is reasonable since the user is there to withdraw an amount of money).

Method to check whether the user withdrawed/inputted the correct amount of money inside the saving account and the amount cannot be less than 0 (which is reasonable since the user is there to withdraw an amount of money).

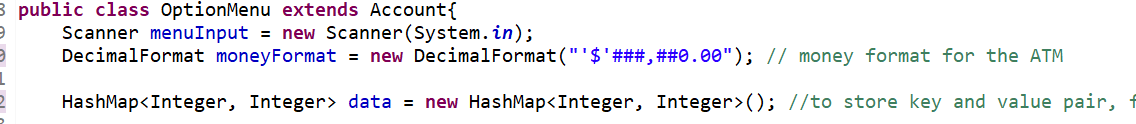


Method to check whether the user deposited/inputted the correct amount of money inside the checking account and the amount cannot be less than 0 (which is reasonable since the user is there to deposit an amount of money).

 Method to check whether the user deposited/inputted the correct amount of money inside the saving account and the amount cannot be less than 0 (which is reasonable since the user is there to deposit an amount of money).

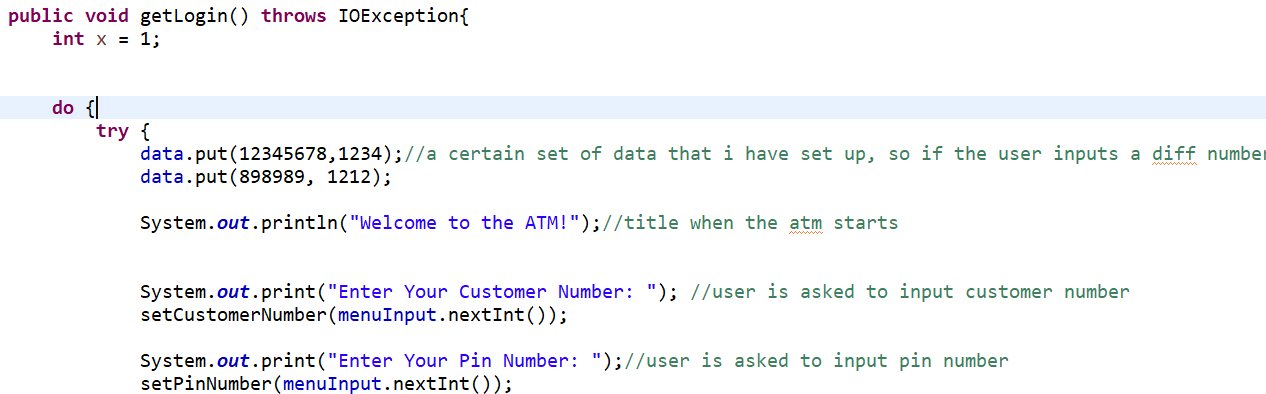
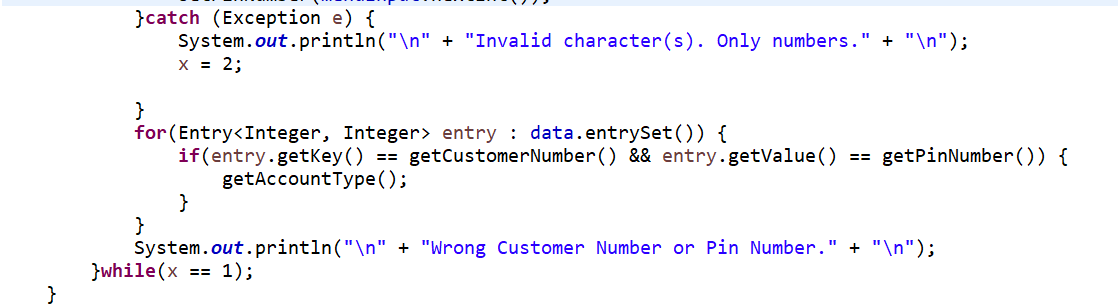


This import(this is inside the option menu class which is a subclass of account class) is used for when there is any input / output file operation issues while application performing certain tasks accessing the files.

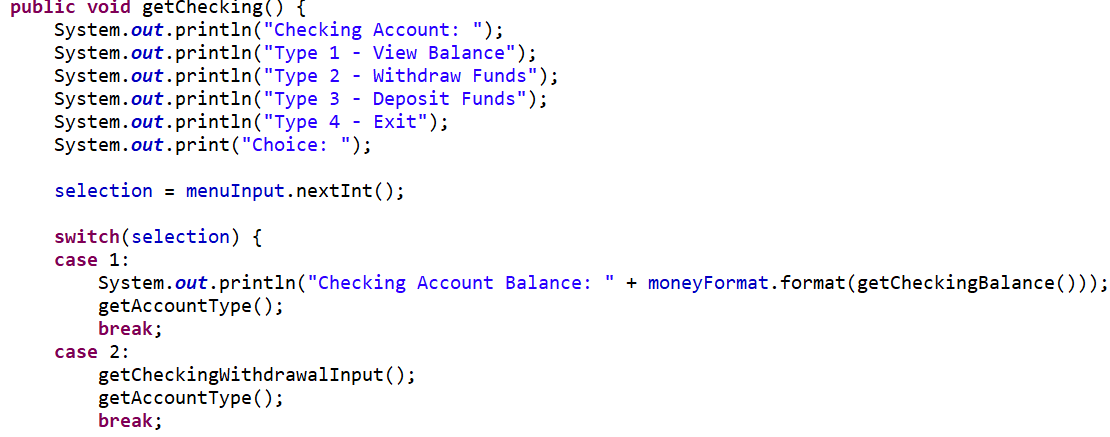


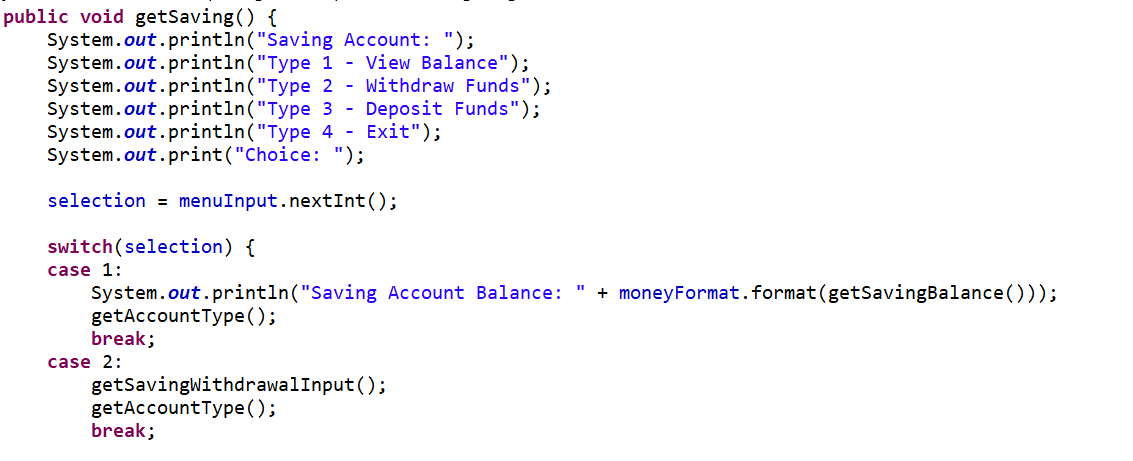
OptionMenu class is a subclass of Account class which has all the option menu for the ATM and it also runs when it meets a certain condition and has cases for each method inside the class.

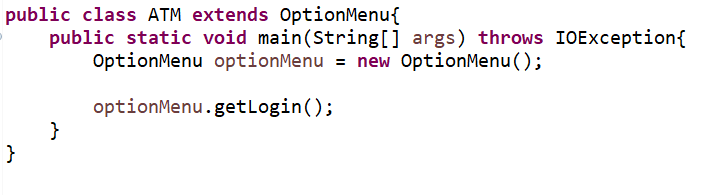
The HashMap I used is to store a set of data or a key and value pair for the data inputs inside the ATM such as customer number and pin number for each ATM account (that I put in the data). The HashMap has the value (integer, integer) so it only accepts numbers as its input.



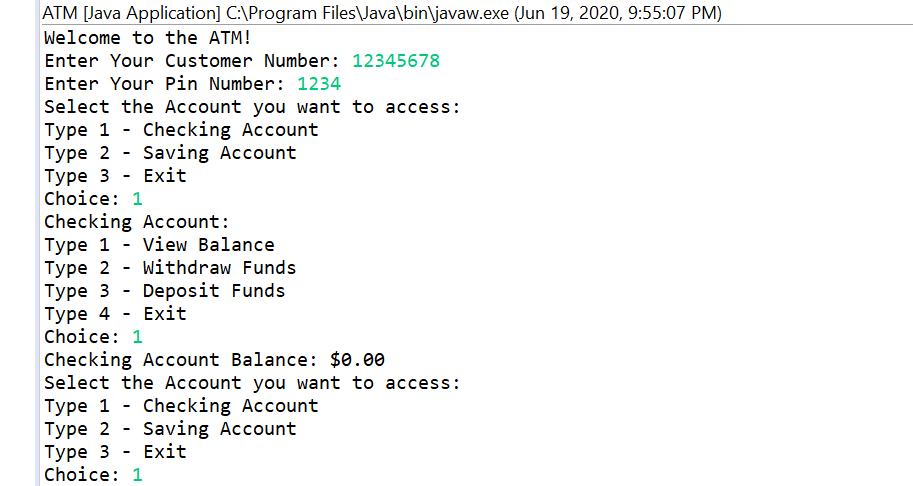
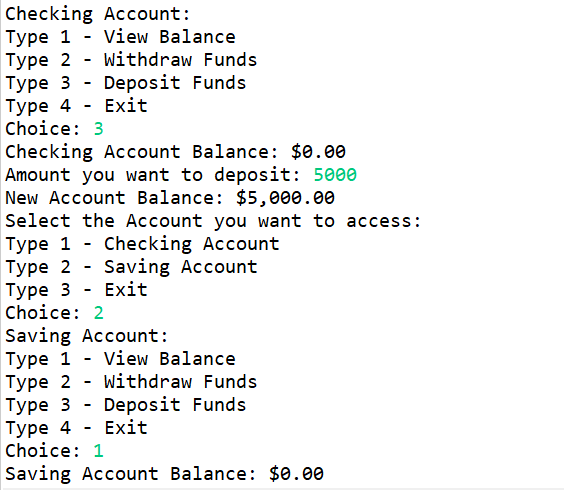
This method getLogin is to get the user to login into the ATM with a certain customer number and custom pin that has been set by me (the data ive set up so if the number the user inputted is different the ATM will not work unless it’s the same as the data ive set). I also made a catch where if the user inputs a different thing besides number, then the ATM will show you the caption “invalid characters, only numbers”. So if the user inputs anything besides number and the data above the ATM will not be able to work.

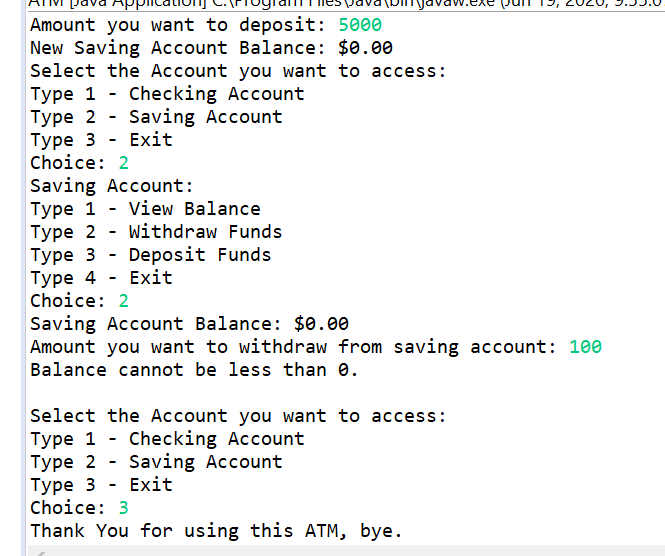
This method getChecking is for when the user chooses the checking account option, they have the option to view the balance inside the checking account, withdraw funds, deposit funds inside the checking account and then exit. I used a switch case for the options so its easier to access in between options.

Same as the method getChecking, getSaving is for the saving account option.

This is the driver class ATM and this class is for the user to use the ATM and to test out the ATM.

1. **Evidence and Screenshots of the working app**





1. **Resources**

* <https://www.youtube.com/watch?v=WsUuqlTgBK0&t=312s>
* <https://www.javatpoint.com/java-hashmap>
* <https://stackoverflow.com/>