Seneca College

Feb 24, 2019

Applied Arts & Technology
SCHOOL OF COMPUTER STUDIES

JAC444 **Demo Due date: Mar 15, 2019**
Final Code Submission Date: Mar 15, 2019

Workshop 5

Notes:

i. Each task should be presented during the lab, demo worth 70% of the workshop marks and code uploading worth the other 30%.
ii. Make sure you have all security and check measures in place (with proper use of Exceptional Handling where ever needed), like wrong data types etc.
iii. Make your project in proper hierarchy; introduce proper class coherence in your project. Proper packages and **your project should be handled by only one main method which should be in a TesterClass**.
iv. Given output structure is just for student to have a glimpse what the output can look, students are free to make the output better in any way.

Other inputs can be given during demo, so make sure you test your program properly.

Task 1: (Part A) Design a class named **Account** which has the following:

- A private int data field named id for the account (default 0).
- A String for First name and another String for last name.
- A private double data field named balance for the account (default 0).
- A private double data field named annualInterestRate that stores the current interest rate (default 0). Assume all accounts have the same interest rate.
- A private Date data field named dateCreated that stores the date when the account was created.
- A no-arg constructor that creates a default account.
- A constructor that creates an account with the specified id and initial balance.
- The accessor and mutator methods for id, balance, and annualInterestRate.
- The accessor method for dateCreated.
- A method named getMonthlyInterestRate() that returns the monthly interest rate.
- A method named getMonthlyInterest() that returns the monthly interest.
- A method named withdraw that withdraws a specified amount from the account.
- A method named deposit that deposits a specified amount to the account.

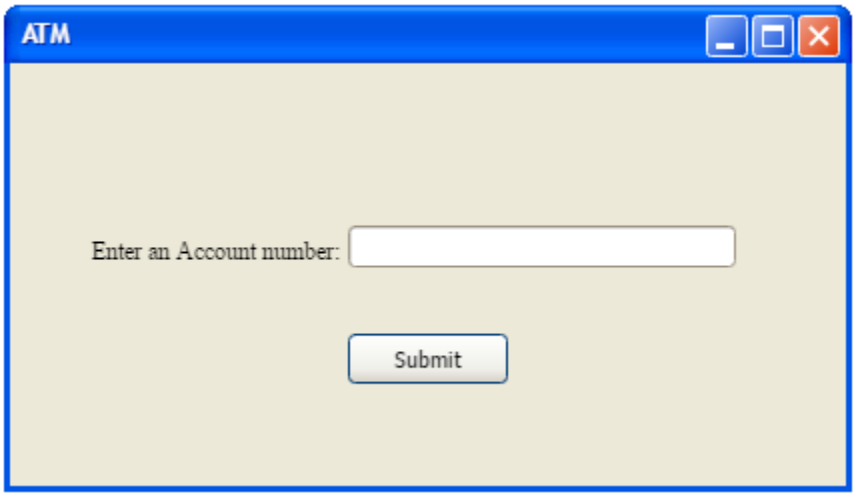
Create an Array of Objects for 10 Accounts with initial balance $100. All the objects should be serialized properly and written on to the file account.dat

Your program should de-serialize all the objects then to show all accounts with their numbers and balances, and account holders first and last names.

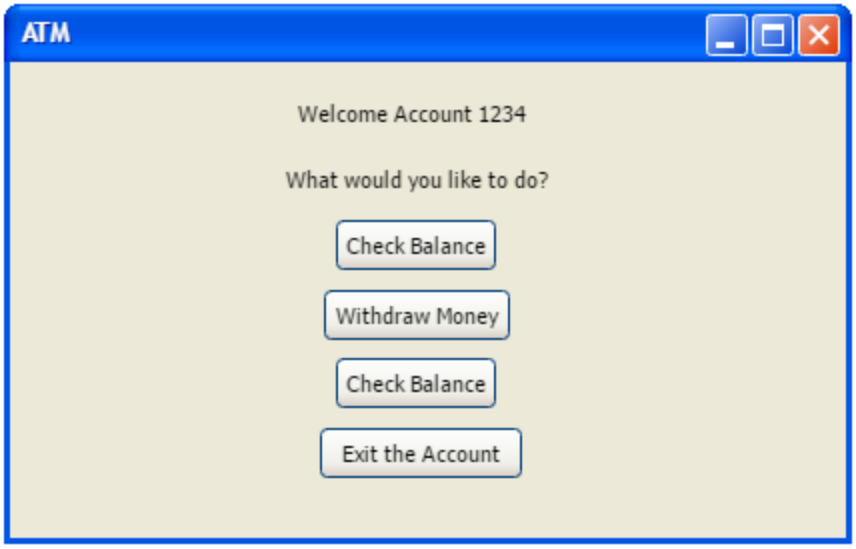
(Part B): Now create a GUI for an ATM which is going to use the Account class we created in Part A. we can update our class with another field,

- Private int pin. (which will hold the pin for each account)
- Your program should check against each Account number for the pin, if the pin is not assigned then ask your ATM should give a welcome message to register your account. Then assign a pin number (should be only numbers).
- If your account is registered, then ATM should check the pin verify it and display the next screen.
- Main menu of your ATM consists of
    - Check Balance
    - Withdraw Money
    - Deposit Money
    - Exit the account (which will display again the main Menu of the ATM)

Below are some of the screen shots for the ATM

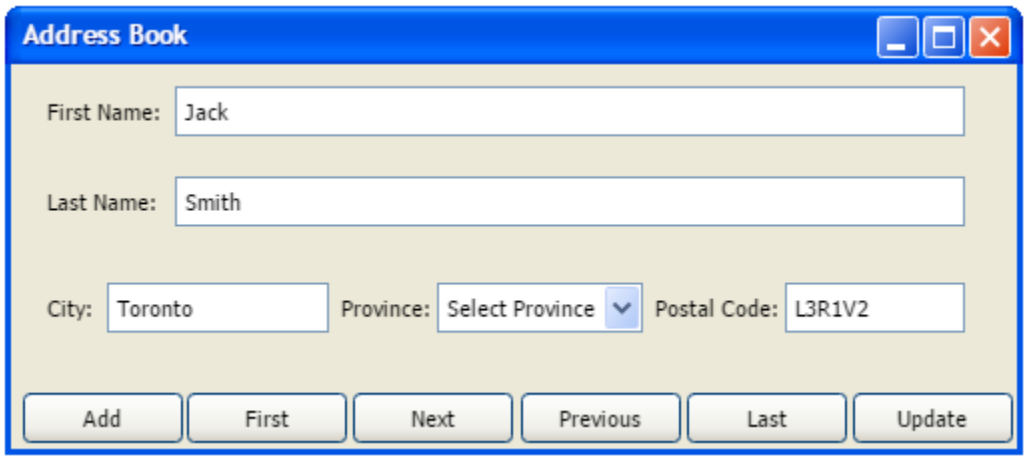


ATM start page (Students are welcome to make it better)



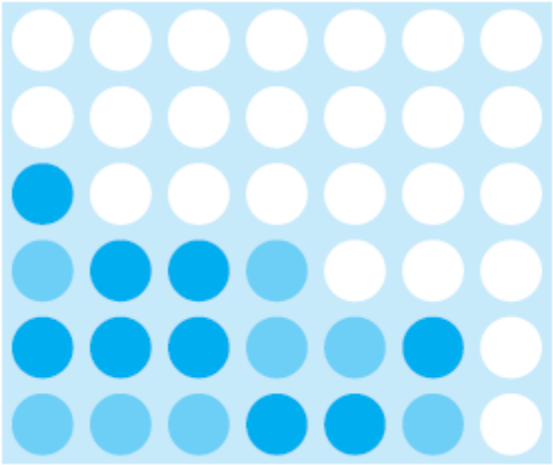
ATM Screen after the confirmed account

Task 2: Write a program that stores, retrieves, adds and updates addresses as shown in the figure below. Use ***random access file*** for reading and writing an address. When the program starts your files should be empty with no addresses.
Hint: Use fixed-length string for storing each attribute in the address.

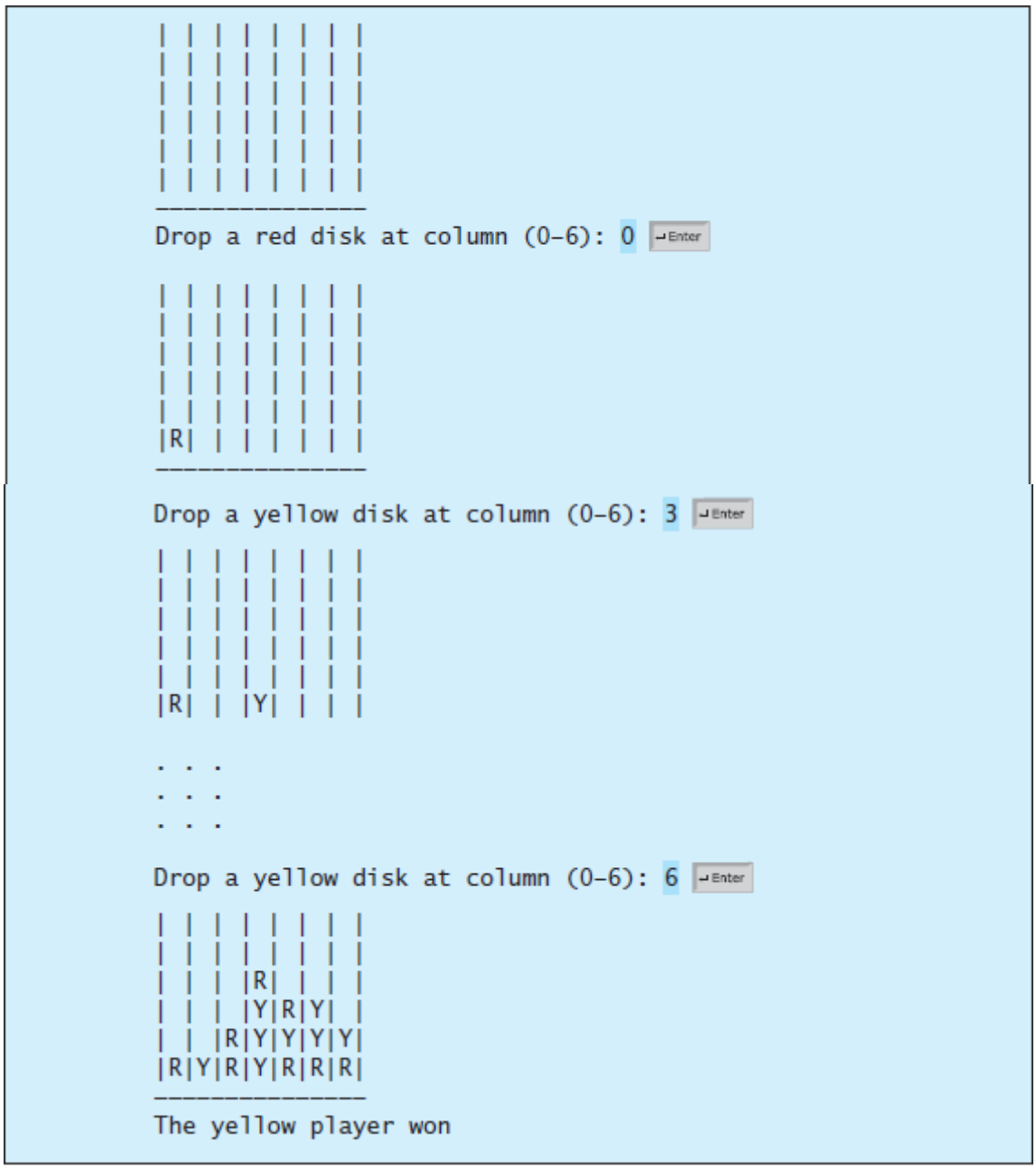


Task 3: (Game – Connect four)

Connect four is a two-player board game in which the players alternately drop colored disks into a seven-column, six-row vertically suspended grid, as shown below.



The objective of the game is to connect four same-colored disks in a row, a column, or a diagonal before your opponent can do likewise. The program prompts two players to drop a red or yellow disk alternately. In the preceding figure, the red disk is shown in a dark color and the yellow in a light color. Whenever a disk is dropped, the program redisplays the board on the console and determines the status of the game (win, draw, or continue). Here is a sample run:



Drop a red disk at column (0–6): 0 Enter
R						

Drop a yellow disk at column (0–6): 3 Enter
R			Y			
. . .
. . .
. . .
Drop a yellow disk at column (0–6): 6 Enter
			R			
			Y	R	Y	
		R	Y	Y	Y	Y
R	Y	R	Y	R	R	R

The yellow player won