Habib University

CS 224 – Object Oriented Programming and Design Methodologies
Fall’ 2023

Lab# 11

UML Diagrams

Objectives: In this lab, we will create Class UML diagrams on http://Draw.io

Exercise # 1: Bank Accounts Management System

Create UML diagram of the following scenario:

In this question we are going to develop a little simulation of banking accounts. You need to think in terms of objects encapsulated with attributes and their functions.

The first object is a Bank which have a vector of Accounts. Each Account object have certain balance and it can be manipulated with **two types of transactions**: Deposit and Withdrawal. To maintain record, the Account needs to store all the transactions executed on it, hence vectors of transactions can be part of an Account object. After executing all the transactions any account with less than 5000 available balance will become dormant, otherwise it is active. A Date class can be created that handles all tasks related to it.

You need to model all the classes can be found in this scenario e.g. Bank, Account, Transactions, Deposit, Withdraw, Date... You may create any other class/- functions/attributes as deemed appropriate or necessary.

Input File Processing

A sample input file is as follows:

```
Create Joe_Root JR230 70000

Deposit GH090 30000 11-Mar

Withdrawal GH090 4000 12-April-21
```

Please refer to file bankinput.txt which contains certain entries to process. You can find three different types of entries in the file:

- Create Title Code InitialDeposit: This entry creates an account with account title, code and initial deposit.
- Deposit Code Amount Date: This entry deposits the given amount in the particular account as per code. Date can vary in different forms; you have to adjust all of the forms. For example, default year is 2021, and it should accept all forms of the months.

- Withdrawal Code Amount Date: This entry withdraws the amount from given account successfully if the available balance is sufficient to 23 withdraw the money, else the transaction should fail. Date can vary in different forms, you have to adjust all of the forms. For example, default year is 2021, and it should accept all forms of the months.

Output File

After executing all the transactions from input file, your program should generate an output bankoutput.txt file, as per following sample it should print information of all accounts.

```
Account Title: Saeed Ghani

Account Code: GH090

Initial Balance: 2000

Available Balance: 28000

Current Status: Active

Deposits:

    1. 30000 on 11/03/21

Withdrawals:

    1. 4000 on 12/04/21 Successful
    2. 40000 on 15/06/21 Denied
```

Exercise # 2: HU Mania

Create UML diagram for the following scenario from HW 4.

You are required to:

- Create a class Unit, which deals with drawing of an object. It has a draw method, completely implemented, and it draws one of the three states of object. It also has a fly method, since fly behavior is different for different types of objects, therefore it’s not fully implemented rather it’s made virtual.

- Create a Pigeon class inherited from Unit. It over-rides the fly function, that flies the pigeon gradually to right side, and rotates through the screen.
- Create a Butterfly class inherited from Unit. It over-rides the fly function, that should take the butterfly right-down direction. Once a butterfly reaches to bottom of the screen, it starts flying right-up direction. Once it reaches top of the screen it moves right-down. Similar to the pigeon, it should rotate through the screen.
- Create a Bee class inherited from Unit. It over-rides the fly function, that should make it fly towards right only. During fly it should hover (doesn’t move forward) for a while over a random interval. You may choose 1% probability in every frame to decide whether it starts hover, and it keeps hovering for 10 frames. As a bee reaches to right most border of screen, it exits from the game, hence the object must be removed from the bees vector.
- Every object animates three of the images provided in the assets file. The draw function is only drawing the object.
- As you click on the screen, one of the above objects is created randomly. Let’s create a class ObjectCreator, it has only one function Unit* getObject(). This function creates one of the objects (Pigeon, Butterfly, Bee) randomly, and returns the pointer to that object. You’ll maintain just a single list of Unit* to store objects of all the types. Simply iterate over the list and call draw and fly functions of each object.
- You have to create objects dynamically with new operator, hence the list should hold pointers to all of the objects. Remember to delete the objects when game is ended, and when the bee objects are removed from vector.
- As you are maintaining a single list to store all types of objects, therefore it would be essential to find out the Bee objects, because it’s required to remove Bee objects when they leave the window. For this task you have to identify the type of objects. Please refer to this page to find type.