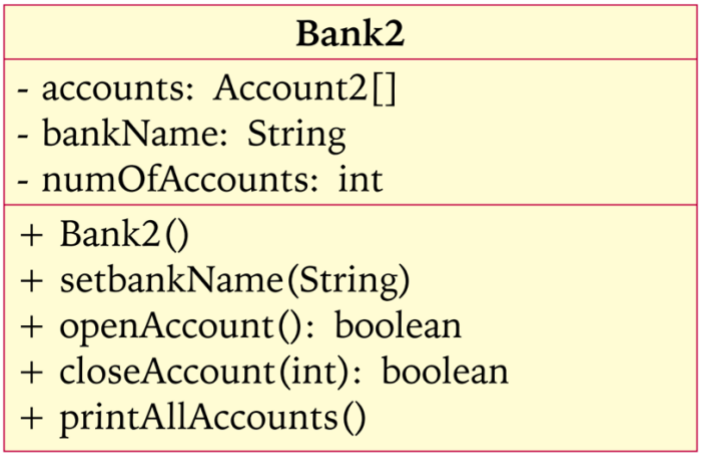
Lab 4

CST 338

Task 1

Download `Account2.java` from this link and create a Java project called **Lab4** with the file. Then, create a new Java file called `Bank2.java` that can hold up to 3 accounts with an array of the `Account2` class type. (Of course, 3 is a very small number, but it's only for the purposes of our lab.) The following presents a UML diagram of the `Bank2` class.



In the class, the `openAccount()` method should allow a user to type the account holder's name, account number, account type, and initial balance. If the account number given by the user is already taken by another account holder, your method should return false. Additionally, if there are already 3 accounts in the bank, the method should return false.

The `closeAccount()` method should get an account number and if the account number doesn't exist, your method should return false. Otherwise, the method should remove the account (=assigning null to the array element) and return true.

Task 2

Let's assume that you are the instructor of CST 338 and want to test the `Bank2` class and `Account2` class. Develop two test cases called `Bank2Test_1.java`, `Bank2Test_2.java` to test the operations of `Bank2` and `Account2` classes you

developed. For your reference, I include a sample test case called `Bank2Test_0.java` below, linked here. In your test cases, you should include explanatory comments.

```java
/**
 * Title: Bank2Test_0.java
 * Abstract: This test case is designed to verify if the Bank2
 * class can handle only one account at openAccount()
 * and closeAccount().
 * Author: XXXX
 * ID: XXXX
 * Date: MM/DD/YY
 */

public class Bank2Test_0 {
    public static void main(String[] args) {
        Bank2 testBank = new Bank2();
        testBank.setBankName("CSUMB");

        // Input: Name: "Alice", Number: 1234,
        // Type: 1, Balance: 1000.50s
        testBank.openAccount();

        // Output: The program should display the one account
        // information properly on the screen.
        testBank.printAllAccounts();

        // Output: The program should display "true" because
        // the bank has an account of 1234.
        System.out.println(testBank.closeAccount(1234));

        // Output: The program shouldn't display anything
        // because there's no account in the bank.
        testBank.printAllAccounts();
    }
}
```

Deliverable

When you have finished your lab, submit `Bank2.java`, `Bank2Test_1.java` and `Bank2Test_2.java` on iLearn.