**Student name: Karem Mostafa ibrahim ID: 20190387 Group: S17**

**1**. **Which Program (Idea for task 2) did you choose?**

Problem-3

**2**. **Which of the following Java / OOP features did you use in your program?**

* 1. **How many classes did you create and their names?**

**Five Classes:**

* Account
* SpecialAccount
* Client
* CommercialClient
* Bank
  1. **How many different access specifiers did you use and their names?**

Two “Public-Private”

1. **How many Java coding style rules did you use and which ones?**

Three (Method Names, Variables and object Names, Class and interface Names)

1. **How many Javadoc tags did you use and which ones?**

Five “@author-@version-@since-@param-@return”

1. **Did you use inheritance? When and why?**

Yes used many time to inherit the Properties of the classes to each other “Special Account-CommercialClient”

1. **Did you use method overriding? When and why?**

Yes, to overriding the toString () method and override withdraw () in SpecialAccount.

1. **Did you use method composition? When and why?**

 Yes, in class Bank and Client.Because if the aggregate is destroyed, then the parts are destroyed

1. **Did you use method polymorphism? When and why?**

Yes to give the method toString more than one implement.

UML:

|  |
| --- |
| **Bank** |
| **-name : string**  **-address : string**  **-phone : int** |
| **+Bank()**  **+Bank(name : string, address : string, phone : int)**  **+addNewClient(client : Client , account : Account) : void**  **+displayClients() : void** |

**1**

**\***

**\***

**\***

|  |
| --- |
| **Account** |
| **-balance : float**  **-accountNumber : string** |
| **+Account()**  **+toString():string**  **+deposit(BalanceDeposit:float):void**  **+withdraw(BalanceWithdraw:float):void** |

|  |
| --- |
| **Client** |
| **-name : string**  **-nationalID : string**  **-address : string**  **-phone : string**  **-account : Account** |
| **+Client()**  **+Client(obj:Account)**  **+toString() : string** |

**1**

**1**

|  |
| --- |
| **SpecialAccount** |
| **-balance : float** |
| **+SpecialAccount ()**  **+SpecialAccount(balance:float , accountNumber:int)**  **+toString() : string**  **+withdraw(BalanceWithdraw:float):void** |

|  |
| --- |
| **CommercialClient** |
| **- commercialID : int** |
| **+ CommercialClient(obj:Account)**  **+toString() : string** |