**BankAccount Class IPO chart and Algorithm Example**

**Note**: IPO chart for the Demo/Test program should be create, No IPO for the base class or the blueprint.

**IPO Chart**

|  |  |  |
| --- | --- | --- |
| **Input** | **Process** | **Output** |
| Starting Balance | Prompt user for Starting balance | Display Starting Balance |
| Deposit Amount | Store the balance | Display Updated Balance after Deposit |
| Withdrawal Amount | Prompt user for Deposit Amount | Display Updated Balance after Withdrawal |
|  | Calculate Current Balance |  |
|  | Prompt user for Withdrawal Amount |  |
|  | Calculate Current Balance |  |
|  | Display all balances |  |
|  |  |  |
|  |  |  |

**Algorithm**

**CLASS BankAccount**

DECLARE INSTANCE variable

**DEFAULT Constructor BankAccount**

INITIALIZE INSTANCE variable TO default value

**END**

**OVERLOADED Constructor BankAccount (TAKES DOUBLE as an argument)**

INITIALIZE INSTANCE variable TO the DOUBLE argument value

**END**

**OVERLOADED Constructor BankAccount (TAKES STRING as an argument)**

INITIALIZE INSTANCE variable TO the PARSED STRING argument value

**END**

**OVERLOADED deposit Method (TAKES DOUBLE as an argument)**

ADD DOUBLE value TO the INSTANCE variable

**END**

**OVERLOADED deposit Method (TAKES STRING as an argument)**

ADD PARSED STRING value TO the INSTANCE variable

**END**

**OVERLOADED withdraw Method (TAKES DOUBLE as an argument)**

SUBTRACT DOUBLE value FROM the INSTANCE variable

**END**

**OVERLOADED deposit Method (TAKES STRING as an argument)**

SUBTRACT PARSED STRING value FROM the INSTANCE variable

**END**

**OVERLOADED setBalance Method (TAKES DOUBLE as an argument)**

SET DOUBLE value TO the INSTANCE variable

**END**

**OVERLOADED setBalance Method (TAKES STRING as an argument)**

SET PARSED STRING value TO the INSTANCE variable

**END**

**getBalance method**

RETURN the INSTANCE variable value

**END**

**END of CLASS BankAccount**

-------------------------------------------------------------------------------------------------------

Below is the **AccountTest Algorithm** on the next page, continue scrolling down………..

-------------------------------------------------------------------------------------------------------

**CLASS AccountTest**

**main method**

DECLARE LOCAL variables

CREATE a DecimalFormat OJBECT to format dollar amount (currency

format)

PROMPT user for Starting Balance using DIALOG BOX

CREATE a BankAccount OBJECT CALLING the OVERLOADED String

Arg CONSTRUCTOR

PROMPT user for deposit amount using DIALOG BOX

CALL the OVERLOADED deposit method that takes STRING arg

DISPLAY deposit amount and current balance

PROMPT user for withdrawal amount using DIALOG BOX

CALL the OVERLOADED withdrawal method that takes STRING arg

DISPLAY the updated balance after withdrawal

EXIT the system

**END**

**END of CLASS AccountTest**