

**Subsystem:** Financial Subsystem

**Collaborations Graphs:** *not implemented*

**Classes:** Account, Balance Inquiry, Deposit Transaction, Funds Transfer, Transaction, Withdrawal Transaction

**Description:** This subsystem implements the financial aspects of a bank customer's interactions with the ATM machine.

**Contracts**

**8. Execute a financial transaction**

**Server:** Transaction

**Subsystem:** User Interface Subsystem

**Collaborations Graphs:** *not implemented*

**Classes:** Bank Card Reader, Cash Dispenser, Deposit Slot, Display Device, Display Screen, Form, Input Device, Keypad, Menu, Output Device, Receipt Printer, Secure Form, User Interaction, User Message, User Response

**Description:** This subsystem implements the interface between the ATM machine and the bank customer.

**Contracts**

**4. Get a numeric value from the user**

**Server:** Form

**6. Get a user selection from a list of options**

**Server:** Menu

**9. Display a message and wait for some event**

**Server:** User Message

**Class:** ATM

(Concrete)

**Superclasses:** none

**Subclasses:** none

**Hierarchy Graphs:** *not implemented*

**Collaborations Graphs:** *not implemented*

**Description:** This class represents a teller machine through which bank customers can perform financial services.

**Contracts**

**Private Responsibilities**

*Create and initiate transactions*

uses Financial Subsystem(8)

*Display the greeting message*

uses User Interface Subsystem(6)

*Display the main menu*

uses User Interface Subsystem(6)

*Eject the receipt*

uses User Interface Subsystem(9)

*Eject the bank card*

uses User Interface Subsystem(9)

**Class:** Account

(Concrete)

**Superclasses:** none

**Subclasses:** none

**Hierarchy Graphs:** *not implemented*

**Collaborations Graphs:** *not implemented*

**Description:** This class represents a customer's account in the bank's database. All accesses to and modifications of bank accounts must occur through this class. Creating an instance of this class does not insure against modifications to the account from other software with access to the database.

### **Contracts**

#### **1. Access and modify the account balance**

*Accept deposits*

##### **deposit(Fixed Point)**

This method records a deposit to the account of the amount specified by the Fixed Point value. The database will not be updated until the commit method has been invoked.

Updating the database does not change the balance recorded there, but registers the deposit until the amount has been verified.

*Accept withdrawals*

##### **withdrawal(Fixed Point)**

This method records a withdrawal from the account of the amount specified by the Fixed Point value. The database will not be updated until the commit method has been invoked.

Updating the database causes a change in the account balance because the amount has already been verified. A negative account balance may result.

*Know the account balance*

##### **balance() returns Fixed Point**

This method returns the current balance as recorded in the bank's database.

#### **2. Commit the results to the database**

*Commit changes to the database*

##### **commit() returns Boolean**

This method causes any modifications to the account to be logged against the database if possible. Return true if the commit was successful. The modifications can fail if there is a transmission error.

##### **commitWith(Account) returns Boolean**

This method causes any modifications to either this or another account to be logged against the database if possible. Return true if the commit was successful. The modifications can fail if there is a transmission error.

### **Private Responsibilities**

**Class:** Balance Inquiry

(Concrete)

**Superclasses:** Transaction

**Subclasses:** none

**Hierarchy Graphs:** *not implemented*

**Collaborations Graphs:** *not implemented*

**Description:** This class represents requests by a bank customer to access the balance of an account.

**Contracts**

**8. Execute a financial transaction**

This contract is inherited from Transaction.

**Private Responsibilities**

*Access the balance*

uses Account(1)

**Class:** Bank Card Reader

(Concrete)

**Superclasses:** Input Device, Output Device

**Subclasses:** none

**Hierarchy Graphs:** *not implemented*

**Collaborations Graphs:** *not implemented*

**Description:** This class represents the hardware device capable of reading and validating a bank customer's card.

**Contracts**

**5. Accept input from the user**

This contract is inherited from Input Device.

**7. Output to the user**

This contract is inherited from Output Device.

**Private Responsibilities**

*Read bank cards*

*Eject bank cards*

*Keep bank cards whose PIN is not correctly entered*

*Inform user of unreadable cards*

uses User Message(9)

*Prompt user for PIN*

uses Secure Form(4)

**Class:** Cash Dispenser

(Concrete)

**Superclasses:** Output Device

**Subclasses:** none

**Hierarchy Graphs:** *not implemented*

**Collaborations Graphs:** *not implemented*

**Description:** This class represents the hardware device through which cash is dispensed to bank customers.

**Contracts**

**7. Output to the user**

This contract is inherited from Output Device.

**Private Responsibilities**

*Dispense funds*

**Class:** Deposit Slot

(Concrete)

**Superclasses:** Display Device, Input Device

**Subclasses:** none

**Hierarchy Graphs:** *not implemented*

**Collaborations Graphs:** *not implemented*

**Description:** This class represents the hardware device through which a bank customer's deposit envelope is inserted.

**Contracts**

**3. Display information**

This contract is inherited from Display Device.

**5. Accept input from the user**

This contract is inherited from Input Device.

**Private Responsibilities**

*Accept a deposit envelope*



**Class:** Deposit Transaction

(Concrete)

**Superclasses:** Transaction

**Subclasses:** none

**Hierarchy Graphs:** *not implemented*

**Collaborations Graphs:** *not implemented*

**Description:** This class represents a request from a bank customer to deposit funds into an account.

**Contracts**

**8. Execute a financial transaction**

This contract is inherited from Transaction.

**Private Responsibilities**

*Prompt for the amount*

uses User Interface Subsystem(4)

*Deposit funds*

uses Account(1), User Interface Subsystem(9)

**Class:** Device

(Concrete)

**Superclasses:** none

**Subclasses:** none

**Hierarchy Graphs:** *not implemented*

**Collaborations Graphs:** *not implemented*

**Description:**

**Contracts**

**Private Responsibilities**

**Class:** Display Device

(Concrete)

**Superclasses:** none

**Subclasses:** Display Screen, Receipt Printer, Deposit Slot

**Hierarchy Graphs:** *not implemented*

**Collaborations Graphs:** *not implemented*

**Description:** This class defines the behavior common to all devices that can display information for a bank customer.

**Contracts**

### 3. Display information

*Display text and graphics*

**display(Text)**

This method displays the Text on the device in the default location.

**display(Text, Point)**

This method displays the Text on the device at the specified Point.

**display(Graphic)**

This method displays the Graphic on the device in the default location.

**display(Graphic, Point)**

This method displays the Graphic on the device at the specified Point.

**Private Responsibilities**

**Class:** Display Screen

(Concrete)

**Superclasses:** Display Device

**Subclasses:** none

**Hierarchy Graphs:** *not implemented*

**Collaborations Graphs:** *not implemented*

**Description:** This class represents a screen on which text and graphics information can be displayed.

**Contracts**

### **3. Display information**

This contract is inherited from Display Device.

**Private Responsibilities**

**Class:** Form

(Concrete)

**Superclasses:** User Interaction

**Subclasses:** Secure Form

**Hierarchy Graphs:** *not implemented*

**Collaborations Graphs:** *not implemented*

**Description:** This class represents an interaction with the user for the purpose of obtaining a numeric value.

**Contracts**

**4. Get a numeric value from the user**

*Ask the user for information*

uses Display Screen(3), Keypad(5)

**getNumber(String) returns User Response**

This method displays the Text prompt on the screen to inform the user what type of value is expected, reads numeric key presses, and assembles those key presses into a value which it returns. The number representing a key press is echoed on the display to provide visual feedback to the customer.

**Private Responsibilities**

*Know if user has responded*

*Know the user's response*

uses Keypad(5)

*Provide feedback on input*

uses Display Screen(3)

**Class:** Funds Transfer

(Concrete)

**Superclasses:** Transaction

**Subclasses:** none

**Hierarchy Graphs:** *not implemented*

**Collaborations Graphs:** *not implemented*

**Description:** This class represents a request from a bank customer to have funds transferred from one account to another.

**Contracts**

**8. Execute a financial transaction**

This contract is inherited from Transaction.

**Private Responsibilities**

*Prompt for the amount*

uses User Interface Subsystem(4)

*Transfer funds*

uses Account(1)

**Class:** Input Device

(Abstract)

**Superclasses:** none

**Subclasses:** Bank Card Reader, Deposit Slot, Keypad

**Hierarchy Graphs:** *not implemented*

**Collaborations Graphs:** *not implemented*

**Description:** This class defines the behavior of devices from which input can be obtained.

**Contracts**

**5. Accept input from the user**

*Get user input*

**input() returns User Response**

This method waits until the device to which it was sent has received the expected type of input, then returns an indication of whether the input was received, and when appropriate, the value of that input.

**Private Responsibilities**

**Class:** Key

(Concrete)

**Superclasses:** none

**Subclasses:** none

**Hierarchy Graphs:** *not implemented*

**Collaborations Graphs:** *not implemented*

**Description:**

**Contracts**

**Private Responsibilities**

*Know if it has been pressed*



**Class:** Keypad

(Concrete)

**Superclasses:** Input Device

**Subclasses:** none

**Hierarchy Graphs:** *not implemented*

**Collaborations Graphs:** *not implemented*

**Description:** This class represents the keys on the face of the ATM machine that can be pressed.

**Contracts**

**5. Accept input from the user**

This contract is inherited from Input Device.

**Private Responsibilities**

**Class:** Menu

(Concrete)

**Superclasses:** User Interaction

**Subclasses:** none

**Hierarchy Graphs:** *not implemented*

**Collaborations Graphs:** *not implemented*

**Description:** This class represents a form of user interaction in which a bank customer is asked to choose from one option from a small number of choices.

**Contracts**

**6. Get a user selection from a list of options**

*Present user with choices*

uses Display Screen(3), Keypad(5)

**getChoice(Text) returns User Response**

This method presents the user with the list of choices specified (see addItem). When the user responds, the value associated with the selected choice is returned. The list of choices is labeled with Text describing the options and the purpose for the question.

**addItem(Text, any)**

This method adds a choice to the menu. The choice is represented to the user with the parameter Text. If this item is chosen, the second argument will be returned. If more items are added than can be displayed on the screen, the items will be divided into groups that will fit, with the last item of each group being a "next page" item. The last item is "return to start of menu."

**Private Responsibilities**

*Know if user has responded*

*Know user's response*

uses Keypad(5)

**Class:** Numeric Input Key

(Concrete)

**Superclasses:** none

**Subclasses:** none

**Hierarchy Graphs:** *not implemented*

**Collaborations Graphs:** *not implemented*

**Description:**

**Contracts**

**Private Responsibilities**

**Class:** Output Device

(Abstract)

**Superclasses:** none

**Subclasses:** Bank Card Reader, Cash Dispenser, Receipt Printer

**Hierarchy Graphs:** *not implemented*

**Collaborations Graphs:** *not implemented*

**Description:** This class defines the behavior common to all classes that can send some physical output to the bank customer.

**Contracts**

**7. Output to the user**

*Output something physical*

**eject()**

This method causes something physical to be ejected from the machine.

**Private Responsibilities**

**Class:** Receipt Printer

(Concrete)

**Superclasses:** Display Device, Output Device

**Subclasses:** none

**Hierarchy Graphs:** *not implemented*

**Collaborations Graphs:** *not implemented*

**Description:** This class represents the hardware device that can print information on a paper receipt, and eject that receipt.

**Contracts**

**3. Display information**

This contract is inherited from Display Device.

**7. Output to the user**

This contract is inherited from Output Device.

**Private Responsibilities**

*Print receipt of transactions*

*Eject the receipt*

**Class:** Secure Form

(Abstract)

**Superclasses:** Form

**Subclasses:** none

**Hierarchy Graphs:** *not implemented*

**Collaborations Graphs:** *not implemented*

**Description:** This class represents a type of user interaction in which the user is prompted for numeric input, but in which the feedback includes only the number of digits entered, not the value of those digits.

**Contracts**

**4. Get a numeric value from the user**

This contract is inherited from Form.

**Private Responsibilities**

**Class:** Transaction

(Abstract)

**Superclasses:** none

**Subclasses:** Balance Inquiry, Deposit Transaction, Funds Transfer, Withdrawal Transaction

**Hierarchy Graphs:** *not implemented*

**Collaborations Graphs:** *not implemented*

**Description:** This class defines the behavior common to all requests from a bank customer to perform some financial transaction.

**Contracts**

**8. Execute a financial transaction**

*Execute a financial transaction*

**execute()** returns **Boolean**

This method executes a transaction, returning true if the transaction was completed, false if the user canceled the transaction before completion.

**Private Responsibilities**

*Check if cancel key has been pressed*

uses User Interface Subsystem(4), User Interface Subsystem(6), User Interface Subsystem(9)

*Commit the transaction to the database*

uses Account(2)

*Prompt for an account*

uses User Interface Subsystem(6)

*Gather information*

uses User Interface Subsystem(4), User Interface Subsystem(6), User Interface Subsystem(9)

*Remember data relevant to the transaction*

*Print a record of the transaction*

uses User Interface Subsystem(9)

**Class:** User Interaction

(Abstract)

**Superclasses:** none

**Subclasses:** Form, Menu, User Message

**Hierarchy Graphs:** *not implemented*

**Collaborations Graphs:** *not implemented*

**Description:** This class describes the behavior common to all interactions with the bank customer.

**Contracts**

**Private Responsibilities**

*Check to see if cancel key has been pressed*  
uses Keypad(5)



**Class:** User Message

(Concrete)

**Superclasses:** User Interaction

**Subclasses:** none

**Hierarchy Graphs:** *not implemented*

**Collaborations Graphs:** *not implemented*

**Description:** This class represents a type of user interaction in which the user is prompted to take some action. The message will not return to its caller until the action has been taken.

**Contracts**

### 9. Display a message and wait for some event

*Display message text*

uses Display Device(3), Input Device(5), Output Device(7)

**InsertValidCard() returns User Response**

This method displays a message asking the user to insert a card, and waits until a valid card has been inserted.

**InsertDepositEnvelope() returns User Response**

This message displays a message asking the user to insert a deposit envelope and waits until an envelope has been inserted. If no envelope is inserted within five minutes, the transaction is considered canceled, and that User Response is returned.

**removeCard() returns User Response**

This method displays a message asking the user to remove the bank card and waits until the card has been removed. If the card has not been removed after five minutes, a User Response is returned signifying that the user did not collect the bank card.

**removeReceipt() returns User Response**

This method displays a message asking users to remove the printed receipt of their transactions. If the receipt has not been removed after five minutes, a User Response is returned signifying that the user did not remove the receipt.

**removeCash() returns User Response**

This method displays a message asking users to remove the cash just withdrawn. If the cash has not been removed after five minutes, a User Response is returned signifying that the user did not remove the cash.

**Private Responsibilities**

*Wait for the appropriate user response*

**Class:** User Response

(Concrete)

**Superclasses:** none

**Subclasses:** none

**Hierarchy Graphs:** *not implemented*

**Collaborations Graphs:** *not implemented*

**Description:** This class represents a response made by a user when asked to perform some task. The response can be invalid if the user pressed the cancel key or failed to perform the task. If the response is valid, there may be a value associated with the response indicating the results of the user's actions.

**Contracts**

**Private Responsibilities**

*Know a user's response*

**isValid() returns Boolean**

This method returns true if the response is valid, false if not.

**value() returns any**

This method returns the value associated with the response, if there is one. If there is no value, or if the response is invalid, a null object is returned.

*Remember a user's response*

**setValid()**

This method sets the response to be a valid response.

**setInvalid()**

This method sets the response to be an invalid response.

**value(any)**

This method sets the value associated with the response to be the value of the parameter.

**Class:** Withdrawal Transaction

(Concrete)

**Superclasses:** Transaction

**Subclasses:** none

**Hierarchy Graphs:** *not implemented*

**Collaborations Graphs:** *not implemented*

**Description:** This class represents a request by a bank customer to withdraw funds from an account in the form of cash.

**Contracts**

**8. Execute a financial transaction**

This contract is inherited from Transaction.

**Private Responsibilities**

*Prompt for the amount*

uses User Interface Subsystem(4)

*Withdraw funds*

uses Account(1), User Interface Subsystem(9)

**Contract 1:** Access and modify the account balance

**Server:** Account

**Clients:** Balance Inquiry, Deposit Transaction, Funds Transfer, Withdrawal Transaction

**Description:** This contract defines the way an account can be accessed and modified.

**Contract 2:** Commit the results to the database

**Server:** Account

**Clients:** Transaction

**Description:** This contract supports committing account changes to the database.

**Contract 3:** Display information

**Server:** Display Device

**Clients:** User Message, Form, Menu

**Description:** This contract supports the display of text and graphics to either the screen or receipt printer.

**Contract 4:** Get a numeric value from the user

**Server:** Form

**Clients:** Deposit Transaction, Funds Transfer, Transaction, Withdrawal Transaction, Bank Card Reader

**Description:** This contract supports prompting the user for numeric input.

**Contract 5:** Accept input from the user

**Server:** Input Device

**Clients:** User Message, Form, Menu, User Interaction

**Description:** This contract supports the ability to determine when the user has input some information.



**Contract 6:** Get a user selection from a list of options

**Server:** Menu

**Clients:** Transaction, ATM

**Description:** This contract supports prompting the user to choose among a finite set of choices.

**Contract 7:** Output to the user

**Server:** Output Device

**Clients:** User Message

**Description:** This contract supports the output of something physical.

**Contract 8:** Execute a financial transaction

**Server:** Transaction

**Clients:** ATM

**Description:** This contract supports executing financial transactions.

**Contract 9:** Display a message and wait for some event

**Server:** User Message

**Clients:** Bank Card Reader, Deposit Transaction, Transaction, Withdrawal Transaction, ATM

**Description:** This contract supports prompting the user to perform some action, such as inserting or removing a bank card.

