Shay Snyder

Honors 1260 – Spring 2020

Credit Card Manager Design Document

Problem: Spring 2020 Honors 1260 Project 1: Credit Card Manager

### List of Inputs, Outputs, and Processing Required

INPUTS

* Inputs will be menu driven using JOptionPane
* Allow the user to enter credit card information
  + Card holder’s name
  + Credit card number
  + Expiration date

OUTPUTS

* A report summerizing the information entered about a credit card

PROCESSING

* Store the information input from the credit card in separate attributes
* Determine if the card number is valid
* Determine if the card is expired
* Determine the credit card type (Visa, Mastercard, etc.)
* Generate a report summarizing the credit card

### Identification of Classes and Their Responsibilities

**Class name:** **CreditCardManagerDriver**

Responsibilities:

* Use JOptionPane to make I/O
* Input the card holder’s name
* Input the card number
* Input the card expiration date
* Outputting a report of the card

**Class name:** **CreditCard**

Responsibilities:

* knowing the card holder’s name
* knowing the card number
* knowing the card expiration data
* determine if the entered card number is valid
* determine if the card if valid
* generate a report of the credit card

### UML Class Diagram

A screenshot of a cell phone

Description automatically generated

### Algorithms

Class: CreditCard

private void checkCardNumber()

{

*// create a variable to hold the sum as required by the algorithm*

int sum = 0;

boolean isSecondDigit = false;

*// loop through every 2nd integer of the credit card number*

for(int i = this.cardNumber.length() -1 ; i >= 0; i--)

{

*// use a substring to access every digit*

*// int n = Integer.parseInt(this.cardNumber.substring(i, i + 1));*

int n = Character.getNumericValue(this.cardNumber.charAt(i));

*// if the loop is at a second digit, multiply by 2*

if(isSecondDigit)

{

n \*= 2;

*// if the integer is greater than 9, use the sum of indivudal digits*

if( n > 9 ) n -= 9;

}

*// add the value of n to the total sum*

sum += n;

*// flip the toggle*

isSecondDigit = !isSecondDigit;

} *// END: looping through every digit and summation*

*/\**

*\* if the total sum can be evenly divided by 10, the card is valid*

*\* if the total sum isn't evenly divisble by 10, the card is invalid*

*\*/*

if(sum % 10 == 0 && sum > 0)

{

this.isCardNumberValid = true;

}

*/\**

*\* no statement is made to set the isCardValid boolean to false because*

*\* of the values defined in the default constructor*

*\*/*

} *// END: checkCardNumber method*

### Test Cases

|  |  |  |  |
| --- | --- | --- | --- |
| **Card Holder** | **Card Number** | **Expiration Date** | **Expected Output** |
| Amex | 374245455400126 | 05/2023 | Success |
| Discover | 60115564485789458 | 12/2023 | Success |
| Mastercard | 5425233430109903 | 04/2023 | Success |
| Other | 5425233430109903 | 12/2004 | Invalid Expiration Date |
| Visa | 4918484589897107 | 01/2023 | Failure |
| Other | 6034932528973614 | 06/2020 | Success |