from bankcard import \*

from authentication import \*

from authentication import Authentication

from bankcard import BankCard

def main():

card\_num\_list = [4319473018453644, 5555555555554444, 372357888597453, 5487573018453644]

cnum = input("card number")

print(cnum)

card\_num = card\_num\_list[3]

card\_num=int(cnum)

auth = Authentication()

card = BankCard(card\_num)

verify = auth.verify\_card(card\_num)

print(verify)

details = card.get\_card\_details()

print(details)

portion = input("enter the first ")

Checksum=str(fportion)

checksum = auth.get\_checksum(fportion)

print("Checksum: ", checksum)

vendor = "MasterCard, visa, American\_express"

if vendor == 'Visa':

first\_digit = '4'

elif vendor == 'MasterCard':

first\_digit = '5'

elif vendor == 'American\_express':

first\_digit = '3'

checksum = auth.get\_checksum(first\_digit)

print("Checksum: ", checksum)

if \_\_name\_\_ == '\_\_main\_\_':

main()

import random

class Authentication:

def \_\_init\_\_(self):

self.sum = 0

def verify\_card(self, card\_num):

total\_even\_i = 0

total\_odd\_i = 0

str\_list = str(card\_num)

for i in range((len(str\_list) - 1), -1, -1):

if i % 2 == 0:

num\_even\_i = int(str\_list[i])

num\_even\_i = num\_even\_i \* 2

if num\_even\_i > 9:

num\_even\_i = (num\_even\_i - 9)

total\_even\_i += num\_even\_i

else:

total\_odd\_i += int(str\_list[i])

self.sum = total\_even\_i + total\_odd\_i

if self.sum % 10 == 0:

return "Valid"

else:

return "Invalid"

def get\_checksum(self, first\_portion):

first\_portion = str(first\_portion)

valid\_card\_lenght = 16

for x in range((valid\_card\_lenght - 1) - len(first\_portion)):

other\_portion = str(random.randint(0, 9))

first\_portion += other\_portion

self.verify\_card(int(first\_portion))

if self.sum % 10 == 0:

checksum = 0

else:

checksum = 10 - (self.sum % 10)

full\_card = first\_portion + str(checksum)

print("Generated Card: ", full\_card)

print(self.verify\_card(full\_card))

return checksum (checksum is returned)

from authentication import Authentication

class BankCard:

def \_\_init\_\_(self, card\_num):

self.card\_num = card\_num

self.vendor = self.get\_vendor(self.card\_num)

def get\_vendor(self, card\_num):

validity = Authentication().verify\_card(card\_num)

if validity == 'Valid':

first\_digit = str(card\_num)[0]

industry = ''

if first\_digit == '1' or first\_digit == '2':

industry = 'Airline'

elif first\_digit == '3':

industry = 'Travel & '

issuer = 'American Express'

elif first\_digit == '4' or first\_digit == '5' or first\_digit == '3':

industry = 'Banking'

if first\_digit == '4':

issuer = 'Visa'

elif first\_digit == '5':

issuer = 'MasterCard'

else:

issuer = 'American\_express'

return '\tIndustry: {0} \n\tIssuer: {1}'.format(industry, issuer)

else:

self.vendor = "Invalid card has no Vendor"

return self.vendor

def get\_card\_details(self):

return "\nCARD DETAILS\nCard No.: {0} \nVendor Info: \n{1}".format(self.card\_num, self.vendor)