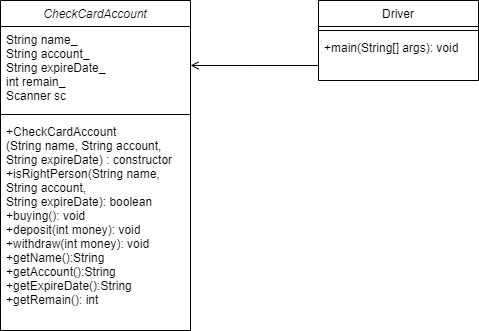
1. 필요한 클래스들

(1) CheckCardAccount: 계좌의 정보를 확인하고, 입출금을 한다.

(2) Driver: CheckCardAccount 클래스를 이용한다

클래스 다이어그램



2. 각 클래스 설계

(1) CheckCardAccount 클래스

-변수

- String name\_: 예금주 이름

- String account\_ = 계좌번호

- String expireDate\_ = 만기연월

- int remain\_: 잔고

- Scanner sc: 입력

- 메소드

- 생성자 메소드

- 목적: CheckCardAccount 객체를 만들 때 변수를 설정 및 초기화 한다.

- 매개변수: String name, String account, String expireDate

- 알고리즘

1. this.name\_ = name을 실행한다.

2. this.account\_ = account을 실행한다.

3. this.expireDate\_ = expireDate을 실행한다.

4. this.remain\_ = 0을 실행한다.

- isRightPerson 메소드

- 목적: 신원을 확인한다.

- 매개변수: String name, String account, String expireDate

- 반환값: this.name\_.equals(name) && this.account\_.equals(account)

&& this.expireDate\_.equals(expireDate)

- 알고리즘

1. this.name\_.equals(name) && this.account\_.equals(account)

&& this.expireDate\_.equals(expireDate) 을 반환한다

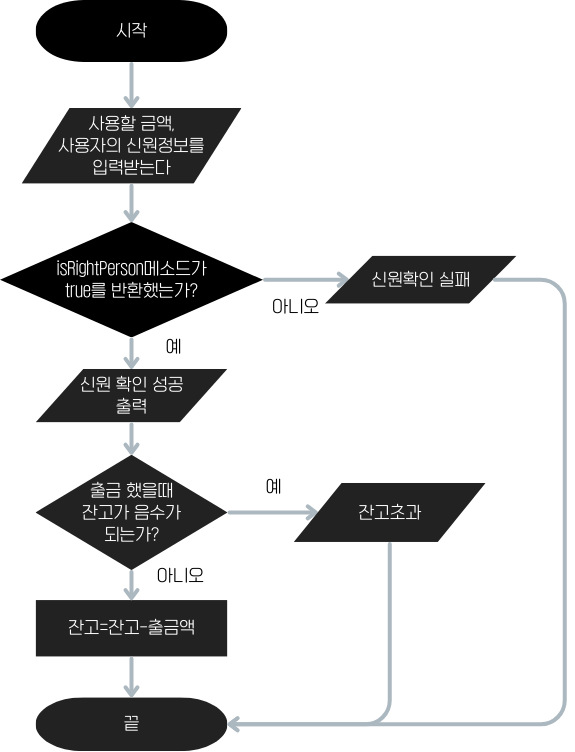
- buying 메소드

- 목적: 물건을 구매하고 신원을 물어본다

- 매개변수: 없음

- 반환값: 없음

- 알고리즘(순서도)



- deposit 메소드

- 목적: 계좌에 돈을 입금한다

- 매개변수: int money

- 반환값: 없음

- 알고리즘

1. 만약 money 가 0 보다 크면 money + "원을 입금한다." 를 출력하고,

this.remain\_ += money를 한다

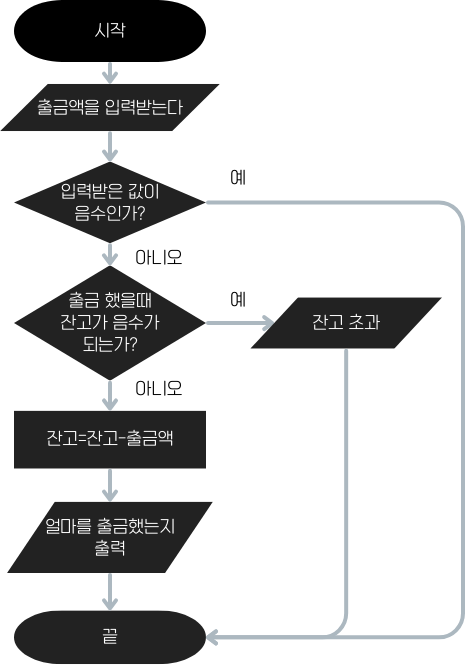
- withdraw 메소드

- 목적: 현금을 출금한다

- 매개변수: int money

- 반환값: 없음

- 알고리즘(순서도)



-stat 메소드

- 목적: 계좌의 정보를 출력한다

- 매개변수: 없음

- 반환값: 없음

- 알고리즘

1. "\n예금주 이름: " + getName()을 출력한다

2. "계좌번호: " + getAccount()을 출력한다

3. "만기연월: " + getExpireDate()을 출력한다

4. "잔고: " + getRemain()을 출력한다

- getName 메소드

- 목적: name\_을 반환한다

- 매개변수: 없음

- 반환값: this.name\_

- 알고리즘

1. this.name\_을 반환한다

- getAccount 메소드

- 목적: account\_반환한다

- 매개변수: 없음

- 반환값: this.account\_

- 알고리즘

1. this.account\_을 반환한다

- getExpireDate 메소드

- 목적: expireDate\_를 반환한다

- 매개변수: 없음

- 반환값: this. expireDate\_

- 알고리즘

1. this. expireDate\_을 반환한다

- getRemain 메소드

- 목적: remain\_을 반환한다

- 매개변수: 없음

- 반환값: this. remain\_

- 알고리즘

1. this. remain\_을 반환한다

(2) Driver 클래스

- 변수 : 없음

- main 메소드

- 지역 변수: card = CheckCardAccount를 가르키는 참조변수, scan = Scanner객체를 가르키는 참조변수

- 알고리즘(순서도)

