**과제 #1 : 명함 인쇄**

**객체지향 설계와 패턴**

**2022.3.14**

**2018112178 임연지**

**프로그램 설명**

프로그램은 다음의 5개의 클래스로 이루어 진다.

Name, CardOrder, CardInteraction, MainTest, MainCard

각각의 클래스의 설명은 다음과 같다.

**1. Name**

명함 주인의 이름을 저장하고, 그 이름을 여러가지 포맷으로 변경하는 기능을 가진다.

firstName : String

첫번째 이름

middleName : String

중간 이름

lastName : String

마지막 이름

Name(String fName, String mName, String lName)

첫번째 이름(fName), 중간 이름(mName), 마지막 이름(lName)을 저장한다.

Name(String fName, String lName)

첫번째 이름(fName), 마지막 이름(lName)을 저장하고,

중간 이름은 빈 문자열을 저장한다. (“”)

getFirstName(), getMiddleName(), getLastName() : String

각 변수(firstName, middleName, lastName)을 반환한다.

setFirstName(String fn), setMiddleName(String mn), setLastName(String ln) : void

각 변수(firstName, middleName, lastName)를 설정한다.

getFirstAndLastName() : String

firstName과 lastName을 차례로 공백을 사이에 둔 문자열을 반환한다.

getLastCommaFirst() : String

lastName과 firstName을 차례로 “, “을 사이에 둔 문자열을 반환한다.

getInits()

이름의 이니셜을 반환한다.

middleName의 길이가 0이라면 두글자, 아니면 3글자가 반환된다.

**2. CardOrder**

이름, 테두리 문자, 카드 개수를 입력 받아서 명함을 제작하고, 올바른 가격을 반환한다.

name : Name

Name 클래스를 이용해서 이름을 저장한다.

border : char

테두리 문자를 저장한다. 기본값은 ‘\*’이다.

numCard : int

주문한 명함의 개수를 저장한다.

space : int

공백 문자(‘ ‘)를 저장한다.

Initials : String

이름의 이니셜을 저장한다.

CARD\_LEN : int

카드의 기본 너비(32)를 저장한다.

CardOrder(String fullName)

사용자가 입력한 이름(fullName)을 name에 저장하고, initials, border, numCards를 초기화한다.

저장하기 전에 nameIsValid를 통해서 유효한 이름인지 확인하고 유효하지 않으면 NameException 예외를 발생시킨다.

유효한 이름인 경우 공백을 기준으로 fullName을 나눈다.

중간 이름이 있는지 여부에 따라 Name의 적절한 생성자를 사용하여 name에 저장한다.

Name의 getInits()메소드를 하용하여 이니셜을 initals에 저장한다.

border는 ‘\*’, numCards는 0으로 초기화한다.

getInitials() : String

initials를 반환한다.

getFormattedName() : String

명함에 들어갈 포맷의 이름 문자열을 반환한다.

nameIsValid(String fullName) : boolean

fullName이 유효한 이름인지 여부를 반환한다.

getBorder() : char

border를 반환한다.

setBorder(char ch) : void

border를 설정한다.

getName() : Name

name을 반환한다.

setName(Name \_name) : void

name을 설정한다.

getNumCards() : int

numCards를 반환한다.

setNumCards(int n) : void

numCards를 설정한다.

getSampleCard() : String

완성된 명함을 문자열로 반환한다.

getTopLine() : String

명함의 가장 위(아래)의 라인에 해당하는 문자열을 반환한다.

모서리에 이니셜이 출력되어야 하므로 문자열의 앞, 뒤에 이니셜이 있다.

getBlankLine() : String

명함의 이름과 가장 위(아래)가 아닌 라인에 해당하는 문자열을 반환한다.

getLineWithName() : String

명함의 이름이 들어가는 라인에 해당하는 문자열을 반환한다.

이때 이름은 getFormattedName()메소드를 이용하여 중간이름이 이니셜로 바뀐 포맷의 문자열을 이용한다.

getCardPrice() : double

명함의 장당 가격을 반환한다.

명함에 들어가는 이름(getFormattedName())의 길이가 12를 넘어가면 장당 가격에 10을 더한다.

getFinalCost() : double

명함의 개수에 장당 가격을 곱하여 최종 가격을 반환한다.

이때 hasDiscount()메소드를 사용해서 할인이 들어가는 경우, 할인을 적용한 가격을 반환한다.

hasDiscount() : Boolean

할인 여부를 반환한다.

**3. CardInteraction**

프로그램 사용자로부터 필요한 입력 값을 입력 받아서 CardOrder에게 전달한다.

scanner : Scanner

콘솔을 통해서 입력을 받기 위한 Scanner 객체를 저장한다.

card : CardOrder

입력 값을 적용할 CardOrder 객체를 저장한다.

run : void

getNameFromUser()메소드를 사용해서 사용자로부터 이름을 입력 받은 후, card에 해당 이름 문자열로 CardOrder 객체를 생성하여 저장한다.

완성된 명함을 보여주고 “OK”라는 명령어가 들어올 때까지 테두리 문자를 설정한다.

카드의 개수를 getNumberFromUser()메소드를 통해서 입력받고 해당하는 가격과 할인 여부를 출력한다.

getNameFromUser() : String

사용자로부터 이름 문자열을 입력 받아 반환한다.

이때, CardOrder의 nameIsValid()메소드를 이용하여 유효한 이름이 아닌 경우, 유효한 이름을 입력 받을 때까지 입력 받는다.

getNumberFromUser() : int

사용자로부터 명함의 개수를 입력 받아 반환한다.

이때, 입력 받은 값이 1과 1000사이의 정수가 아니라면 해당하는 정수를 입력 받을 때 까지 입력 받는다.

**4. MainTest**

Name, CardOrder의 메소드들을 테스트한다.

**5. MainCard**

CardInteration 클래스의 run() 메소드를 사용하여, CardInteration을 동작시킨다.