

융합프로그래밍 2

평가 과제

경영정보학과 2018111369 이에나

INDEX

사례
설명

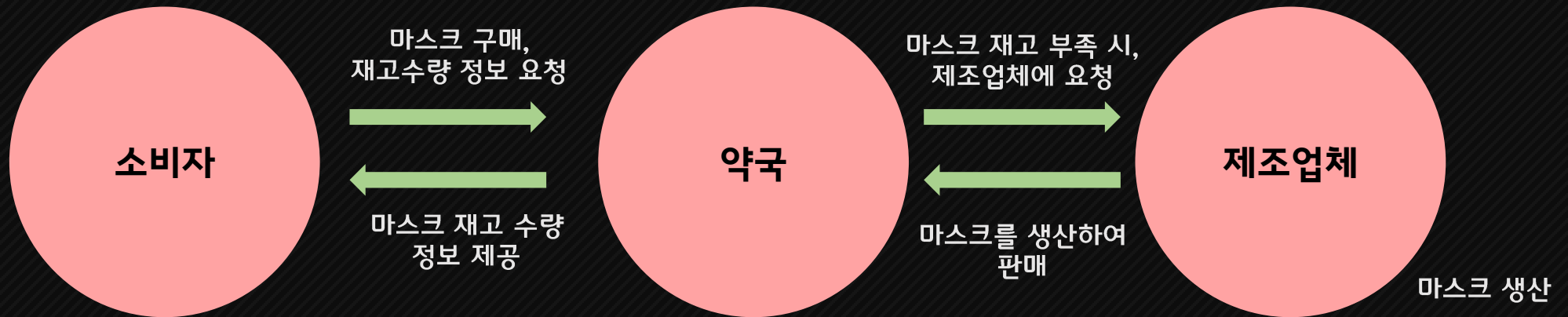
객체 지향 분석

코드 분석

사례 설명

주제 : 코로나 마스크 재고 수량 확인

저는 소비자가 약국의 마스크 재고 수량을 확인하고, 마스크를 구입하는 과정을 사례로 선정해 보았습니다.
소비자와 약국 사이의 거래 뿐만 아니라, 약국과 마스크의 제조업체와의 거래까지 다루어 보았습니다.
소비자는 약국으로부터, 마스크 재고 수량의 개수에 대한 정보를 요청하여 얻어낼 수 있으며, 약국에서 원하는 수량의 마스크를 구입할 수 있습니다.
하지만, 원하는 수량의 마스크보다, 약국이 보유하고 있는 마스크의 재고량이 적을 경우, 문제가 발생합니다.
약국은 재고가 부족할 때에, 제조업체에 요청하여 마스크를 조달할 수 있습니다.
하지만, 제조업체 또한 재고가 부족할 수 있습니다. 이때는 제조업체가 재빠르게 생산하는 과정을 거쳐야 합니다.



우선 클래스를 하나하나 소개하고, 이들의 관계를 살펴보겠습니다.

사례 연구

마스크 클래스

class Mask

```
private String type;  
private String color;
```

```
Void show() ;
```

- 1) 마스크 타입
ex) KF94, KF80 등
- 2) 마스크 색
ex) 검정색, 흰색 등

➔ 마스크의 타입 및 색을 알려줍니다.

약국 클래스

class Pharmacy

```
private String name;  
private int num;
```

```
String getName();  
int getNum();  
void setNum(int num);  
void show();  
void buy(Provider pro, int num);
```

- 1) 약국 이름
- 2) 약국이 보유하고 있는 마스크의 개수

Show()

➔ 마스크의 종류와 색, 재고수량을 보여줌.

Buy()

➔ 제조업체로부터 마스크를 구매

사례 연구

소비자 클래스

class Customer

```
private String search_place;
```

```
void search(Pharmacy pha) ;  
void buy(Pharmacy phy, int num);
```

=> 검색할 장소

Search()

→ 소비자가 원하는 약국에 대한
재고수량 검색

Buy()

→ 소비자가 약국으로부터 마스크
를 구입

제조업체 클래스

class Provider

```
private String companyname;  
private int stock;
```

```
int getStock();  
String getCompanyname();  
void provide(int num);  
void check_stock() ;  
void make(int num);
```

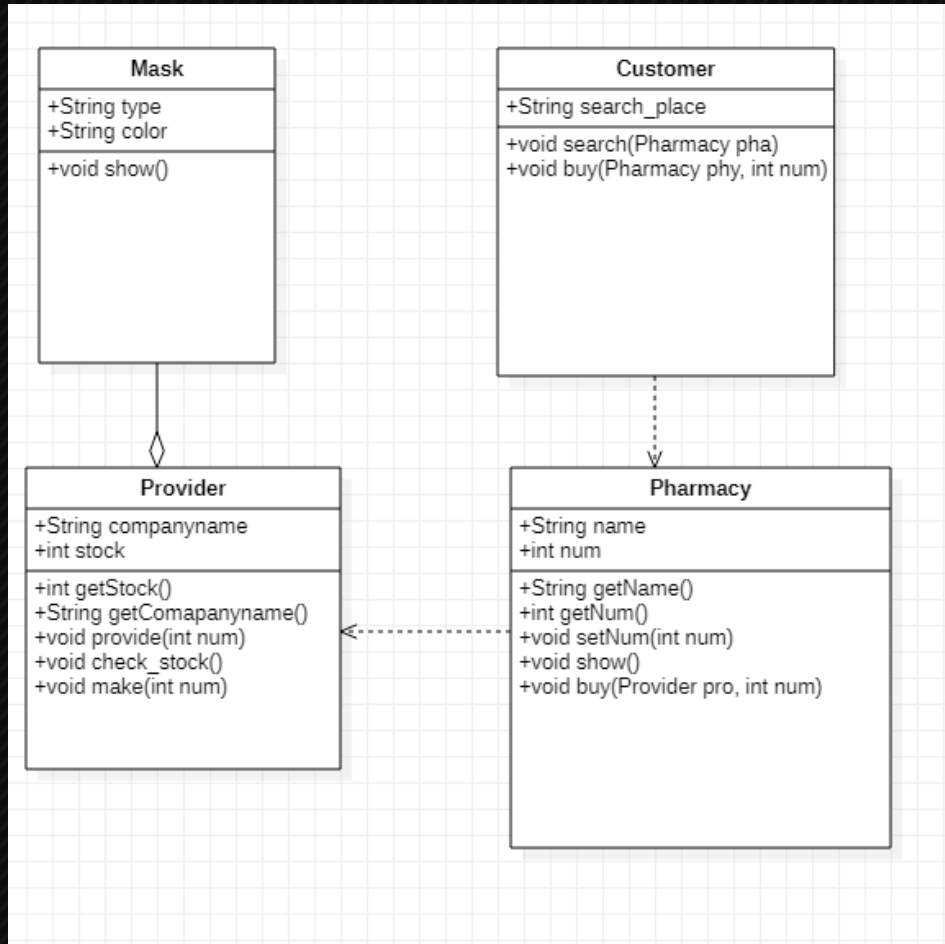
provide(int num)
→ 제조업체가 약국에 마스크 제공

check_stock()
→ 제조업체의 재고수량 확인

make(int num)
→ 제조업체의 마스크 제작

1) 회사명
2) 마스크
재고수량

객체 지향 분석



**Mask와 Provider는 구성관계
(Provider has Mask)**

**Customer와 Pharmacy는
message passing 관계**

**Pharmacy와 Provider는
message passing 관계**

코드 분석

Class Mask

```
class Mask{
    private String type;
    private String color;

    Mask(){};
    Mask(String type, String color){
        this.type = type;
        this.color = color;
    }
    void show() {
        System.out.print("마스크의 종류 : "+type+", 색 : "+color);
    }
}
```

코드 분석

class Pharmacy

```
//2018111369
class Pharmacy{
    private String name;
    private int num;
    Pharmacy(String name, int num){
        this.name = name;
        this.num = num;
    }
    String getName() {
        return name;
    }

    int getNum() {
        return num;
    }

    void setNum(int num) {
        this.num = num;
    }

    void show() {
        if (num<1) {
            System.out.println("현재 "+name+"에 남은 마스크의 재고가 없습니다. \n");
        }
        else {
            System.out.println("현재 "+name+"에는 "+num+"개의 마스크가 있습니다. \n");
        }
    }

    void buy(Provider pro, int num) {
        System.out.println("["+pro.getCompanyname()+" -> "+name+"]");
        if(pro.getStock()<=0) {
            System.out.println(name+"은 제조업체로부터 마스크를 구입할 수 없습니다. \n");
        }
        else {
            if(pro.getStock()<num) {
                int stock = pro.getStock();
                pro.provide(stock);
                System.out.println(name+"은 제조업체로부터 "+stock+"개의 마스크만 구입할 수 있습니다. 이만큼만 구입합니다. \n");
                this.num = this.num + stock;
            }
            else {
                pro.provide(num);
                System.out.println(name+"은 "+pro.getCompanyname()+"로부터 "+num+"개의 마스크를 구입합니다. \n");
                this.num = this.num + num;
            }
        }
    }
}
```

Message Passing

제조업체 (provider) 로 부터 마스크를 구입하기 위해서는, 그 제조업체의 재고현황을 고려해야 합니다. 따라서, message passing을 통해, 제조업체의 이름과, 그 제조업체의 재고수량에 따라 약국이 구매할 수 있는 마스크의 수량이 결정됩니다.

1) 제조업체의 재고수량이 0인 경우

→ 약국 : 마스크 구입 불가

2) 제조업체의 재고수량이 약국에서 요청한 마스크의 개수보다 적은 경우

→ 약국 : 요청한 마스크의 개수보다 적은 양의 마스크 구입 (재고수량 전부를 구입)

3) 약국에서 요청한 마스크의 개수보다 많은 양의 재고를 제조업체가 보유하고 있는 경우

→ 요청한 마스크의 개수만큼 구입

코드 분석

class Customer

```
class Customer{
    private String search_place;

    void search(Pharmacy pha) {
        search_place = pha.getName();
        System.out.println("[ "+search_place+"의 마스크 재고를 확인합니다.]");
        pha.show();
    }

    void buy(Pharmacy phy, int num) {
        Scanner sc = new Scanner(System.in);
        System.out.println("[소비자의 구매]");
        if(phy.getNum() <= 0) {
            System.out.println("마스크의 재고가 없어 구매할 수 없습니다. ");
        }
        else {
            if(phy.getNum() < num) {
                System.out.println("현재 마스크의 재고는 "+phy.getNum()+"개로 주문하신 양보다 적습니다. ");
                System.out.print("남아있는 전체 재고 "+phy.getNum()+"만큼이라도 구매하시겠습니까?(y/n)");
                String c = sc.next();
                if(c.equals("y")) {
                    System.out.println("현재 마스크의 재고는 "+phy.getNum()+"개 이기 때문에 "+phy.getNum()+"
                        "개의 마스크만 구매합니다. 나머지 "+(num-phy.getNum())+"개의 마스크는 다른 약국에서 구매하겠습니다.\n");
                    phy.setNum(0);
                }
                if(c.equals("n")) {
                    System.out.println("구매하지 않겠습니다.");
                }
            }
            else {
                System.out.println("마스크를 "+num+"개 구매합니다.\n");
                phy.setNum(phy.getNum()-num);
            }
        }
    }
}
```

Message Passing

소비자가 약국에서 마스크를 구매할 때,
약국의 재고 현황을 고려해야 합니다.
따라서 message passing을 통해 이를 구현해
보았습니다.

1) 약국의 재고수량이 없는 경우

→ 구매 불가

2) 소비자의 희망 구매 개수 < 약국의 재고수량

→ 1) 적은 수량이라도 구매하기

→ 2) 구매하지 않기 (다른 약국 가기)

3) 소비자의 희망 구매 개수 < 약국의 재고수량

→ 원하는 수량만큼 구매 가능

코드 분석

class Provider

```
class Provider{ // 구성관계 . 회사별로 한 종류의 마스크만 판매한다고 가정.
private String companyname;
private int stock;

Mask mask;
Provider(String companyname, String type,String color, int stock){ //처음에 회사가 가지고 있는 마스크의 종류와 마스크의 색, 재고.
    mask = new Mask(type,color);
    this.stock = stock;
    this.companyname = companyname;
}

void setStock(int stock) {
    this.stock = stock;
}

int getStock(){
    return stock;
}

String getCompanyname() {
    return companyname;
}

void provide(int num) {
    if (stock<1) {
        System.out.println(companyname+"은 현재 재고가 부족하여 약국에 마스크를 제공할 수 없습니다. 마스크의 제조가 필요합니다.\n");
    }
    else {
        stock = stock - num;
        mask.show();
        System.out.println(" , 제공개수 : "+ num+"개");
    }
}

void make(int n) {
    stock = stock + n;
}

void check_stock() {
    System.out.println("[제조업체"+companyname+"의 재고]");
    if (stock<1) {
        System.out.println(companyname+"는 마스크 재고가 없습니다.\n");
    }
    else {
        System.out.println(companyname+"의 마스크 재고는 "+stock+"입니다.\n");
    }
}
}
```

구성 관계
(Provider has a Mask)

코드 분석

Main 클래스

```
public class Main {

    static void menu() {
        System.out.println("*****\n");
        System.out.println(" 0.menu");
        System.out.println(" 1. 소비자 : 마스크 구매");
        System.out.println(" 2. 소비자 : 약국의 마스크 재고 확인");
        System.out.println(" 3. 약국 : 제조업체로부터 마스크 구매");
        System.out.println(" 4. 제조업체 : 제조업체의 재고 확인 ");
        System.out.println(" 5. 제조업체 : 마스크 생산");
        System.out.println(" 6. 프로그램 종료");
        System.out.println("\n*****");
    }

    public static void main(String[] args) {
        // TODO Auto-generated method stub

        menu();

        Pharmacy ph1 = new Pharmacy("금호약국", 20); // 약국 마스크 재고 처음에 20개로 설정.

        int stock;
        stock = (int)(20+Math.random()*200);

        Provider company = new Provider("A회사", "kf94", "white", stock); // 제조업체 마스크 재고 랜덤으로 설정.
        System.out.println("초기"+company.getCompanyname()+"의 재고 : " + stock);
        System.out.println("초기 "+ph1.getName()+"의 재고 : "+ph1.getNum());

        Customer yaena = new Customer();

        Scanner sc = new Scanner(System.in);
```

Menu를 활용하여,
시나리오를 보여줄 수 있는 코드를 작성했습니다.

0. menu
1. 소비자 : 마스크 구매
2. 소비자 : 약국의 마스크 재고 확인
3. 약국 : 제조업체로부터 마스크 구매
4. 제조업체 : 제조업체의 재고 확인
5. 제조업체 : 마스크 생산
6. 프로그램 종료

```
int mode = -1;
while(mode!=6) {
    System.out.print("Menu 선택 > ");
    mode = sc.nextInt();
    switch(mode) {

        case 0:
            menu();
            break;

        case 1:
            System.out.println("***** 소비자 : 마스크 구매 *****");
            System.out.print("희망 구매 개수 : ");
            int num = sc.nextInt();
            yaena.buy(ph1, num);
            break;

        case 2:
            System.out.println("***** 소비자 : 약국의 마스크 재고 확인 *****");
            yaena.search(ph1);
            break;

        case 3:
            System.out.println("***** 약국 : 제조업체로부터 마스크 구매 *****");
            System.out.print("약국이 제조업체로부터 구매할 마스크의 개수를 입력하세요 > ");
            int buy_num = sc.nextInt();
            ph1.buy(company, buy_num);
            break;

        case 4:
            System.out.println("***** 제조업체 : 제조업체의 재고 확인 *****");
            company.check_stock();
            break;

        case 5:
            System.out.println("***** 제조업체 : 제조업체의 마스크 생산 *****");
            System.out.print("제조업체가 생산할 마스크의 개수를 입력하시요. > ");
            int make = sc.nextInt();
            company.make(make);
            System.out.println(company.getCompanyname()+"은 "+make+"개의 마스크를 생산합니다.");
            break;

        case 6:
            System.out.println("프로그램 종료");
            break;

    }
}
sc.close();

}
```

실행 예시

초기 A회사의 재고 : 36

초기 금호약품의 재고 : 20

Menu 선택 > 2

***** 소비자 : 약국의 마스크 재고 확인 *****

[금호약품의 마스크 재고를 확인합니다.]

현재 금호약품에는 20개의 마스크가 있습니다.

Menu 선택 > 1

***** 소비자 : 마스크 구매 *****

희망 구매 개수 : 15

[소비자의 구매]

마스크를 15개 구매합니다.

Menu 선택 > 1

***** 소비자 : 마스크 구매 *****

희망 구매 개수 : 6

[소비자의 구매]

현재 마스크의 재고는 5개로 주문하신 양보다 적습니다.

남아있는 전체 재고 5만큼이라도 구매하시겠습니까?(y/n)y

현재 마스크의 재고는 5개 이기 때문에 5개의 마스크만 구매합니다.

Menu 선택 > 2

***** 소비자 : 약국의 마스크 재고 확인 *****

[금호약품의 마스크 재고를 확인합니다.]

현재 금호약품에 남은 마스크의 재고가 없습니다.

Menu 선택 > 4

***** 제조업체 : 제조업체의 재고 확인 *****

[제조업체 A회사의 재고]

A회사의 마스크 재고는 36입니다.

Menu 선택 > 3

***** 약국 : 제조업체로부터 마스크 구매 *****

약국이 제조업체로부터 구매할 마스크의 개수를 입력하세요 > 30

[A회사 -> 금호약품]

마스크의 종류 : kf94, 색 : white, 제공개수 : 30개

금호약품은 A회사로부터 30개의 마스크를 구입합니다.

초기 A회사의 재고는 랜덤 값으로 설정했었고, 이 예시의 경우 36개입니다.
초기 금호약품의 재고는 20으로 설정했었습니다.

시나리오 1)

소비자가 약국의 마스크 재고수량을 확인합니다.

시나리오 2)

소비자가 약국에서 마스크를 15개 구매합니다.

시나리오 3)

소비자가 약국에서 마스크 6개를 구매하고자 합니다.

하지만 현재 약국의 마스크 재고량은 5개뿐입니다.

따라서, 소비자가 계속해서 구매를 진행할 것인지 선택권을 줍니다.

이 경우, 소비자가 적은량이라도 구입하겠다고 대답했고

이에, 5개만 구입하게 됩니다.

시나리오 4)

약국의 재고량을 확인합니다.

소비자가 전부다 구입했기 때문에 현재 재고량은 0입니다.

시나리오 5)

제조업체의 마스크 재고량을 확인합니다.

아직 약국에 판매한 마스크가 없기 때문에 초기 재고량과

일치한 36개입니다.

시나리오 6)

약국에 재고량이 없기 때문에, 제조업체로부터 구매하고자

합니다. 30개를 구입합니다.

Menu 선택 > 2

***** 소비자 : 약국의 마스크 재고 확인 *****

[금호약품의 마스크 재고를 확인합니다.]

현재 금호약품에는 30개의 마스크가 있습니다.

시나리오 7)

약국의 마스크 재고를 확인합니다.

약국은 전 단계에서 30개를 확보했으므로, 재고가 30개입니다.

Menu 선택 > 3

***** 약국 : 제조업체로부터 마스크 구매 *****

약국이 제조업체로부터 구매할 마스크의 개수를 입력하세요 > 10

[A회사 -> 금호약품]

마스크의 종류 : kf94, 색 : white, 제공개수 : 6개

금호약품은 제조업체로부터 6개의 마스크만 구입할 수 있습니다. 이만큼만 구입합니다.

시나리오 8)

약국이 제조업체로부터 마스크를 구매하고자 합니다.

10개를 구매하고자 하였으나, 현재 제조업체의 재고가 6개 뿐이라

6개만 구매하게 되었습니다.

Menu 선택 > 4

***** 제조업체 : 제조업체의 재고 확인 *****

[제조업체 A회사의 재고]

A회사는 마스크 재고가 없습니다.

시나리오 9)

제조업체의 재고를 확인합니다. 이 예시의 경우 남은 재고가 없습니다

Menu 선택 > 5

***** 제조업체 : 제조업체의 마스크 생산 *****

제조업체가 생산할 마스크의 개수를 입력하십시오. > 100

A회사는 100개의 마스크를 생산합니다.

Menu 선택 > 4

시나리오 10)

제조업체의 재고가 없으므로 생산을 진행합니다. 100개를 생산합니다.

Menu 선택 > 4

***** 제조업체 : 제조업체의 재고 확인 *****

[제조업체 A회사의 재고]

A회사의 마스크 재고는 100입니다.

시나리오 11)

제조업체의 재고를 확인합니다. 전 단계에서 100개를 생산했기에, 현재 재고는 100개입니다.

Menu 선택 > 6

프로그램 종료