

객체지향개발방법론

Distributed Vending Machine : MS129 Vending Machine

Manual : 사용메뉴얼(v2)

201710560 컴퓨터공학부 정의재
201711315 컴퓨터공학부 신원세
201714164 컴퓨터공학부 박서영
201914173 컴퓨터공학부 김현웅

목차

0. 프로그램 실행 방법(추가)

1. 프로그램 개요

2. DVM 구성

3. DVM 선택

4. 음료 선택

5. 결제 방법 선택

6. 카드 결제

7. 선결제

8. 코드 결제

9. 데모 영상

0. 프로그램 실행 방법(jar 파일 이용시)

1. DVM 1.jar ~ DVM8.jar를 모두 각각 실행합니다.

- cmd창을 총 8개 따로따로 만들어 java -jar로 실행
- 로그에 '[DVMx] SERVER ON'이 뜨면 성공적으로 실행된 것

```
ca. 명령 프롬프트 - java -jar DVM1.jar
Microsoft Windows [Version 10.0.19041.985]
(c) Microsoft Corporation. All rights reserved.

C:\Users\Jeong-EuiJae>cd desktop

C:\Users\Jeong-EuiJae\Desktop>cd DVMs

C:\Users\Jeong-EuiJae\Desktop\DVMs>java -jar DVM1.jar
[DVM1] SERVER ON
```

```
ca. 명령 프롬프트 - java -jar DVM2.jar
Microsoft Windows [Version 10.0.19041.985]
(c) Microsoft Corporation. All rights reserved.

C:\Users\Jeong-EuiJae>cd desktop/DVMs

C:\Users\Jeong-EuiJae\Desktop\DVMs>java -jar DVM2.jar
[DVM2] SERVER ON
```

```
ca. 명령 프롬프트 - java -jar DVM3.jar
Microsoft Windows [Version 10.0.19041.985]
(c) Microsoft Corporation. All rights reserved.

C:\Users\Jeong-EuiJae>cd desktop/DVMs

C:\Users\Jeong-EuiJae\Desktop\DVMs>java -jar DVM3.jar
[DVM3] SERVER ON
```

```
ca. 명령 프롬프트 - java -jar DVM4.jar
Microsoft Windows [Version 10.0.19041.985]
(c) Microsoft Corporation. All rights reserved.

C:\Users\Jeong-EuiJae>cd desktop/DVMs

C:\Users\Jeong-EuiJae\Desktop\DVMs>java -jar DVM4.jar
[DVM4] SERVER ON
```

```
ca. 명령 프롬프트 - java -jar DVM5.jar
Microsoft Windows [Version 10.0.19041.985]
(c) Microsoft Corporation. All rights reserved.

C:\Users\Jeong-EuiJae>cd desktop/DVMs

C:\Users\Jeong-EuiJae\Desktop\DVMs>java -jar DVM5.jar
[DVM5] SERVER ON
```

```
ca. 명령 프롬프트 - java -jar DVM6.jar
Microsoft Windows [Version 10.0.19041.985]
(c) Microsoft Corporation. All rights reserved.

C:\Users\Jeong-EuiJae>cd desktop/DVMs

C:\Users\Jeong-EuiJae\Desktop\DVMs>java -jar DVM6.jar
[DVM6] SERVER ON
```

```
ca. 명령 프롬프트 - java -jar DVM7.jar
Microsoft Windows [Version 10.0.19041.985]
(c) Microsoft Corporation. All rights reserved.

C:\Users\Jeong-EuiJae>cd desktop/DVMs

C:\Users\Jeong-EuiJae\Desktop\DVMs>java -jar DVM7.jar
[DVM7] SERVER ON
```

```
ca. 명령 프롬프트 - java -jar DVM8.jar
Microsoft Windows [Version 10.0.19041.985]
(c) Microsoft Corporation. All rights reserved.

C:\Users\Jeong-EuiJae>cd desktop/DVMs

C:\Users\Jeong-EuiJae\Desktop\DVMs>java -jar DVM8.jar
[DVM8] SERVER ON
```

0. 프로그램 실행 방법(jar 파일 이용시)

2. MainController를 실행합니다.

```
Microsoft Windows [Version 10.0.19041.985]
(c) Microsoft Corporation. All rights reserved.

C:\Users\Jeong-EuiJae>cd desktop/DVMs

C:\Users\Jeong-EuiJae\Desktop\DVMs>java -jar Controller.jar
Error: Unable to access jarfile Controller.jar

C:\Users\Jeong-EuiJae\Desktop\DVMs>java -jar MainController.jar
[Controller] DVM1에게 메시지 발신(유형: 4(위치 요청))
[Controller] DVM1으로부터 메시지 수신(유형: 5(위치 응답), 내용: 101)
[Controller] DVM2에게 메시지 발신(유형: 4(위치 요청))
[Controller] DVM2으로부터 메시지 수신(유형: 5(위치 응답), 내용: 202)
[Controller] DVM3에게 메시지 발신(유형: 4(위치 요청))
[Controller] DVM3으로부터 메시지 수신(유형: 5(위치 응답), 내용: 303)
[Controller] DVM4에게 메시지 발신(유형: 4(위치 요청))
[Controller] DVM4으로부터 메시지 수신(유형: 5(위치 응답), 내용: 404)
[Controller] DVM5에게 메시지 발신(유형: 4(위치 요청))
[Controller] DVM5으로부터 메시지 수신(유형: 5(위치 응답), 내용: 505)
[Controller] DVM6에게 메시지 발신(유형: 4(위치 요청))
[Controller] DVM6으로부터 메시지 수신(유형: 5(위치 응답), 내용: 606)
[Controller] DVM7에게 메시지 발신(유형: 4(위치 요청))
[Controller] DVM7으로부터 메시지 수신(유형: 5(위치 응답), 내용: 707)
[Controller] DVM8에게 메시지 발신(유형: 4(위치 요청))
[Controller] DVM8으로부터 메시지 수신(유형: 5(위치 응답), 내용: 808)
```

| | | | |
|--------------------|--------------------|--------------------|--------------------|
| 1. DVM1 주소: 101 | 2. DVM2 주소: 202 | 3. DVM3 주소: 303 | 4. DVM4 주소: 404 |
| 5. DVM5 주소: 505 | 6. DVM6 주소: 606 | 7. DVM7 주소: 707 | 8. DVM8 주소: 808 |

| | | | |
|---|---|---|----|
| 1 | 2 | 3 | 확인 |
| 4 | 5 | 6 | |
| 7 | 8 | 9 | |
| 0 | ← | | |

0. 프로그램 실행 방법(jar 파일 이용시)

3. 이제 아래 사용법대로 사용하시면 됩니다.

명령 프롬프트 - java -jar MainController.jar

```
[Controller] DVM1으로부터 메시지 수신(유형: 2(재고 응답), 내용: 10)
[Controller] DVM1에게 메시지 발신(유형: 1(재고 요청), 내용: 레드불)
[Controller] DVM1으로부터 메시지 수신(유형: 2(재고 응답), 내용: 0)
[Controller] DVM1에게 메시지 발신(유형: 1(재고 요청), 내용: 몬스터드링크)
[Controller] DVM1으로부터 메시지 수신(유형: 2(재고 응답), 내용: 0)
[Controller] DVM1에게 메시지 발신(유형: 1(재고 요청), 내용: 맥텐션)
[Controller] DVM1으로부터 메시지 수신(유형: 2(재고 응답), 내용: 0)
[Controller] DVM1에게 메시지 발신(유형: 1(재고 요청), 내용: 포카리스웨트)
[Controller] DVM1으로부터 메시지 수신(유형: 2(재고 응답), 내용: 0)
[Controller] DVM1에게 메시지 발신(유형: 1(재고 요청), 내용: 게토레이)
[Controller] DVM1으로부터 메시지 수신(유형: 2(재고 응답), 내용: 0)
[Controller] DVM1에게 메시지 발신(유형: 1(재고 요청), 내용: 파워에이드)
[Controller] DVM1으로부터 메시지 수신(유형: 2(재고 응답), 내용: 0)
[Controller] DVM1에게 메시지 발신(유형: 1(재고 요청), 내용: 밀키스)
[Controller] DVM1으로부터 메시지 수신(유형: 2(재고 응답), 내용: 0)
[Controller] DVM1에게 메시지 발신(유형: 1(재고 요청), 내용: 레쓰비)
[Controller] DVM1으로부터 메시지 수신(유형: 2(재고 응답), 내용: 0)
[Controller] DVM1에게 메시지 발신(유형: 1(재고 요청), 내용: 스파클링)
[Controller] DVM1으로부터 메시지 수신(유형: 2(재고 응답), 내용: 0)
[Controller] DVM1에게 메시지 발신(유형: 1(재고 요청), 내용: 비락식혜)
[Controller] DVM1으로부터 메시지 수신(유형: 2(재고 응답), 내용: 0)
[Controller] DVM1에게 메시지 발신(유형: 1(재고 요청), 내용: 솔의눈)
[Controller] DVM1으로부터 메시지 수신(유형: 2(재고 응답), 내용: 0)
[Controller] DVM1에게 메시지 발신(유형: 1(재고 요청), 내용: 데자와)
[Controller] DVM1으로부터 메시지 수신(유형: 2(재고 응답), 내용: 0)
[Controller] DVM1에게 메시지 발신(유형: 1(재고 요청), 내용: 마운틴듀)
[Controller] DVM1으로부터 메시지 수신(유형: 2(재고 응답), 내용: 0)
[Controller] DVM1에게 메시지 발신(유형: 6(판매 확인 요청), 내용: 코카콜라)
[Controller] DVM1으로부터 메시지 수신(유형: 7(판매 확인 응답), 내용: 9)
```

카드 입력번호를 입력해주세요.
(성공 번호: 1234 1234)

Message

<음료 구매 완료>
구매 진행한 DVM: DVM2
구매한 음료: 코카콜라
음료 가격: 1500원
잔여 재고: 9개
결제 후 카드 잔고: 8500원

OK

| | | | |
|---|---|---|----|
| 1 | 2 | 3 | 확인 |
| 4 | 5 | 6 | |
| 7 | 8 | 9 | |
| 0 | ← | | |

0. 프로그램 실행 방법(intelliJ 이용시)

1. DVM 1.java ~ DVM8.java를 모두 각각 실행합니다.

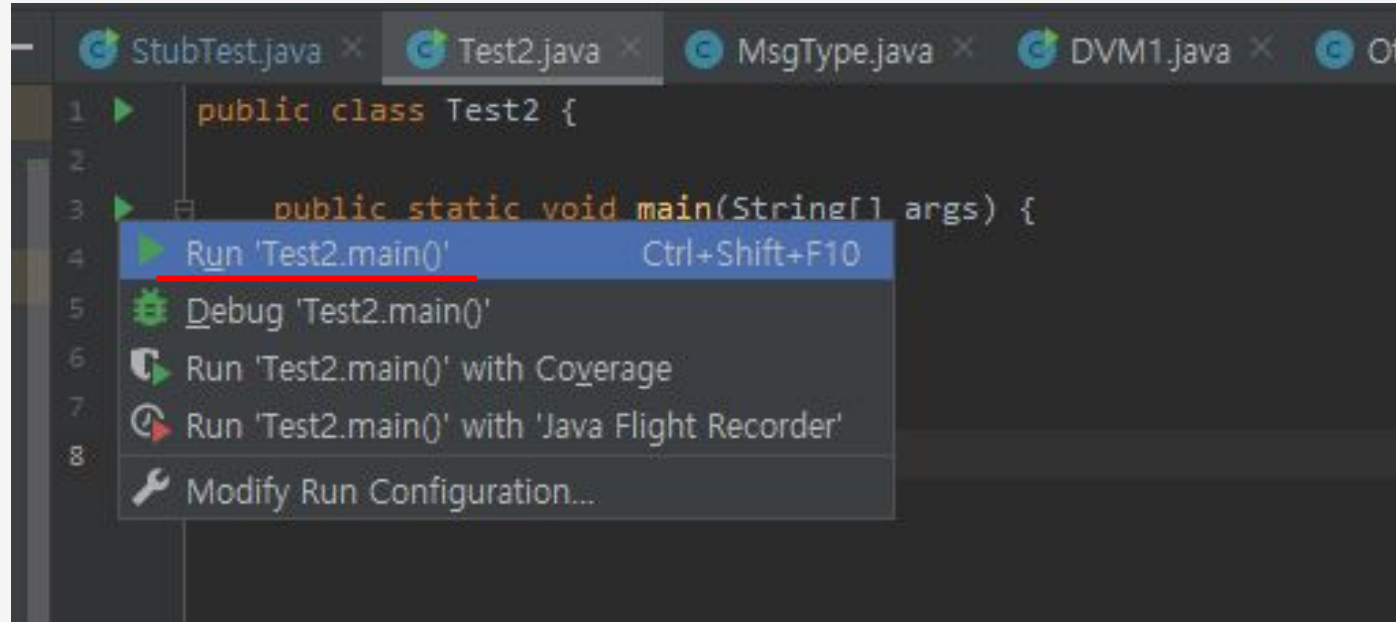
- 각각의 main함수를 실행
- 로그에 '[DVMx] SERVER ON'이 뜨면 성공적으로 실행된 것

```
1  @import ...
2
3
4
5
6
7
8  ▶ public class DVM6 extends Thread implements DVM {
9
10     private ArrayList<Drink> drink_list;
11     private int id;
12     private int address;
13     ServerSocket serverSocket = null;
14     Socket receive_socket = null;
15     ObjectInputStream objectInputStream = null;
16     ObjectOutputStream objectOutputStream = null;
17     private final int STUB_TEST_ID = 999;
18
19     public DVM6(ArrayList<Drink> drink_list, int id, int address) {
20         this.drink_list = drink_list;
21         this.id = id;
22         this.address = address;
23     }
24
25     public DVM6(ArrayList<Drink> drink_list, int id) {
26         this.drink_list = drink_list;
27         this.id = id;
28     }
29
30     ▶ public static void main(String[] args) {
31         ArrayList<Drink> drinkArrayList6 = new ArrayList<>(); // 전체 음료수 리스트
32         drinkArrayList6.add(new Drink( name: "스파클링", price: 1500, stock: 9, imgUrl: "src/main/resources/image/16.jpg"));
33         drinkArrayList6.add(new Drink( name: "레드불", price: 1500, stock: 5, imgUrl: "src/main/resources/image/8.jpg"));
34         drinkArrayList6.add(new Drink( name: "백렌션", price: 1500, stock: 10, imgUrl: "src/main/resources/image/10.jpg"));
35         drinkArrayList6.add(new Drink( name: "코카콜라", price: 1500, stock: 130, imgUrl: "src/main/resources/image/1.jpg"));
36         drinkArrayList6.add(new Drink( name: "스프라이트", price: 1500, stock: 22, imgUrl: "src/main/resources/image/4.jpg"));
37         drinkArrayList6.add(new Drink( name: "화타오렌지", price: 1500, stock: 10, imgUrl: "src/main/resources/image/5.jpg"));
38         drinkArrayList6.add(new Drink( name: "포카리스웨트", price: 1500, stock: 10, imgUrl: "src/main/resources/image/11.jpg"));
39         drinkArrayList6.add(new Drink( name: "게토레이", price: 1500, stock: 0, imgUrl: "src/main/resources/image/12.jpg"));
40         drinkArrayList6.add(new Drink( name: "비락식혜", price: 1500, stock: 0, imgUrl: "src/main/resources/image/17.jpg"));
41         drinkArrayList6.add(new Drink( name: "술의눈", price: 1500, stock: 0, imgUrl: "src/main/resources/image/18.jpg"));
42         drinkArrayList6.add(new Drink( name: "랩시콜라", price: 1500, stock: 0, imgUrl: "src/main/resources/image/2.jpg"));
43         drinkArrayList6.add(new Drink( name: "칠성사이다", price: 1500, stock: 0, imgUrl: "src/main/resources/image/3.jpg"));
44         drinkArrayList6.add(new Drink( name: "레쓰비", price: 1500, stock: 0, imgUrl: "src/main/resources/image/15.jpg"));
```

```
Object-Oriented-Analysis-and-Design111 [DVM5.main()] × Object-Oriented-Analysis-and-Design111 [DVM6.main()] ×
Object-Oriented- 19 sec 오전 12:35:46: Executing task 'DVM6.main()'...
DVM6.main() 18 sec
> Task :compileJava UP-TO-DATE
> Task :processResources UP-TO-DATE
> Task :classes UP-TO-DATE
> Task :DVM6.main()
[DVM6] SERVER ON
```

0. 프로그램 실행 방법(intelliJ 이용시)

2. Test2의 main함수를 실행합니다.



The screenshot shows the IntelliJ IDEA IDE with the Gradle console output and a GUI window. The Gradle console output shows the execution of the 'Test2.main()' task, including the starting of the Gradle Daemon and the execution of the 'Test2.main()' task. The GUI window displays a grid of 8 DVM (Device Virtual Machine) icons, each with a label and address:

- 1. DVM1 주소: 101
- 2. DVM2 주소: 202
- 3. DVM3 주소: 303
- 4. DVM4 주소: 404
- 5. DVM5 주소: 505
- 6. DVM6 주소: 606
- 7. DVM7 주소: 707
- 8. DVM8 주소: 808

The GUI window also features a numeric keypad with buttons for 0-9 and a '확인' (Confirm) button.

```
오전 12:39:10: Executing task 'Test2.main()'...

Starting Gradle Daemon...
Gradle Daemon started in 3 s 516 ms
> Task :compileJava UP-TO-DATE
> Task :processResources UP-TO-DATE
> Task :classes UP-TO-DATE

> Task :Test2.main()
[Controller] DVM1에게 메시지 발신(유형: 4(위치 요청))
[Controller] DVM1으로부터 메시지 수신(유형: 5(위치 응답), 내용: 101)
[Controller] DVM2에게 메시지 발신(유형: 4(위치 요청))
[Controller] DVM2으로부터 메시지 수신(유형: 5(위치 응답), 내용: 202)
[Controller] DVM3에게 메시지 발신(유형: 4(위치 요청))
[Controller] DVM3으로부터 메시지 수신(유형: 5(위치 응답), 내용: 303)
[Controller] DVM4에게 메시지 발신(유형: 4(위치 요청))
[Controller] DVM4으로부터 메시지 수신(유형: 5(위치 응답), 내용: 404)
[Controller] DVM5에게 메시지 발신(유형: 4(위치 요청))
[Controller] DVM5으로부터 메시지 수신(유형: 5(위치 응답), 내용: 505)
[Controller] DVM6에게 메시지 발신(유형: 4(위치 요청))
[Controller] DVM6으로부터 메시지 수신(유형: 5(위치 응답), 내용: 606)
[Controller] DVM7에게 메시지 발신(유형: 4(위치 요청))
[Controller] DVM7으로부터 메시지 수신(유형: 5(위치 응답), 내용: 707)
[Controller] DVM8에게 메시지 발신(유형: 4(위치 요청))
[Controller] DVM8으로부터 메시지 수신(유형: 5(위치 응답), 내용: 808)
```

0. 프로그램 실행 방법(intelliJ 이용시)

3. 이제 아래 사용법대로 사용하시면 됩니다.

```
[Controller] DVM8로부터 메시지 수신(유형: 2(재고 증감), 내용: 6)
[Controller] DVM8에게 메시지 발신(유형: 1(재고 요청), 내용: 코카콜라)
[Controller] DVM8으로부터 메시지 수신(유형: 2(재고 응답), 내용: 10)
[Controller] DVM8에게 메시지 발신(유형: 1(재고 요청), 내용: 환타포도)
[Controller] DVM8으로부터 메시지 수신(유형: 2(재고 응답), 내용: 2)
[Controller] DVM8에게 메시지 발신(유형: 1(재고 요청), 내용: 포카리스웨트)
[Controller] DVM8으로부터 메시지 수신(유형: 2(재고 응답), 내용: 0)
[Controller] DVM8에게 메시지 발신(유형: 1(재고 요청), 내용: 스파클링)
[Controller] DVM8으로부터 메시지 수신(유형: 2(재고 응답), 내용: 0)
[Controller] DVM8에게 메시지 발신(유형: 1(재고 요청), 내용: 게토레이)
[Controller] DVM8으로부터 메시지 수신(유형: 2(재고 응답), 내용: 0)
[Controller] DVM8에게 메시지 발신(유형: 1(재고 요청), 내용: 비락식혜)
[Controller] DVM8으로부터 메시지 수신(유형: 2(재고 응답), 내용: 0)
[Controller] DVM8에게 메시지 발신(유형: 1(재고 요청), 내용: 스프라이트)
[Controller] DVM8으로부터 메시지 수신(유형: 2(재고 응답), 내용: 0)
[Controller] DVM8에게 메시지 발신(유형: 1(재고 요청), 내용: 환타오렌지)
[Controller] DVM8으로부터 메시지 수신(유형: 2(재고 응답), 내용: 0)
[Controller] DVM8에게 메시지 발신(유형: 1(재고 요청), 내용: 핫식스)
[Controller] DVM8으로부터 메시지 수신(유형: 2(재고 응답), 내용: 0)
[Controller] DVM8에게 메시지 발신(유형: 1(재고 요청), 내용: 몬스터드링크)
[Controller] DVM8으로부터 메시지 수신(유형: 2(재고 응답), 내용: 0)
[Controller] DVM8에게 메시지 발신(유형: 1(재고 요청), 내용: 마운틴듀)
[Controller] DVM8으로부터 메시지 수신(유형: 2(재고 응답), 내용: 0)
[Controller] DVM8에게 메시지 발신(유형: 1(재고 요청), 내용: 펄시콜라)
[Controller] DVM8으로부터 메시지 수신(유형: 2(재고 응답), 내용: 0)
[Controller] DVM8에게 메시지 발신(유형: 1(재고 요청), 내용: 칠성사이다)
[Controller] DVM8으로부터 메시지 수신(유형: 2(재고 응답), 내용: 0)
[Controller] DVM8에게 메시지 발신(유형: 1(재고 요청), 내용: 밀키스)
[Controller] DVM8으로부터 메시지 수신(유형: 2(재고 응답), 내용: 0)
[Controller] DVM8에게 메시지 발신(유형: 1(재고 요청), 내용: 파워에이드)
[Controller] DVM8으로부터 메시지 수신(유형: 2(재고 응답), 내용: 0)
```

| | | | | |
|-----------------------|------------------------|------------------------|----------------------|-------------------------|
| 1.레드불 1500원 (20개) | 2.팍텐션 1500원 (1개) | 3.솔의눈 1500원 (2개) | 4.레쓰비 1500원 (0개) | 5.데자와 1500원 (6개) |
| 6.코카콜라 1500원 (10개) | 7.환타포도 1500원 (2개) | 8.포카리스웨트 1500원 (0개) | 9.스파클링 1500원 (0개) | 10.게토레이 1500원 (0개) |
| 11.비락식혜 1500원 (0개) | 12.스프라이트 1500원 (0개) | 13.환타오렌지 1500원 (0개) | 14.핫식스 1500원 (0개) | 15.몬스터드링크 1500원 (0개) |
| 16.마운틴듀 1500원 (0개) | 17.펄시콜라 1500원 (0개) | 18.칠성사이다 1500원 (0개) | 19.밀키스 1500원 (0개) | 20.파워에이드 1500원 (0개) |
| 1 | 2 | 3 | 확인 | |
| 4 | 5 | 6 | | |
| 7 | 8 | 9 | | |
| 0 | | ← | | |

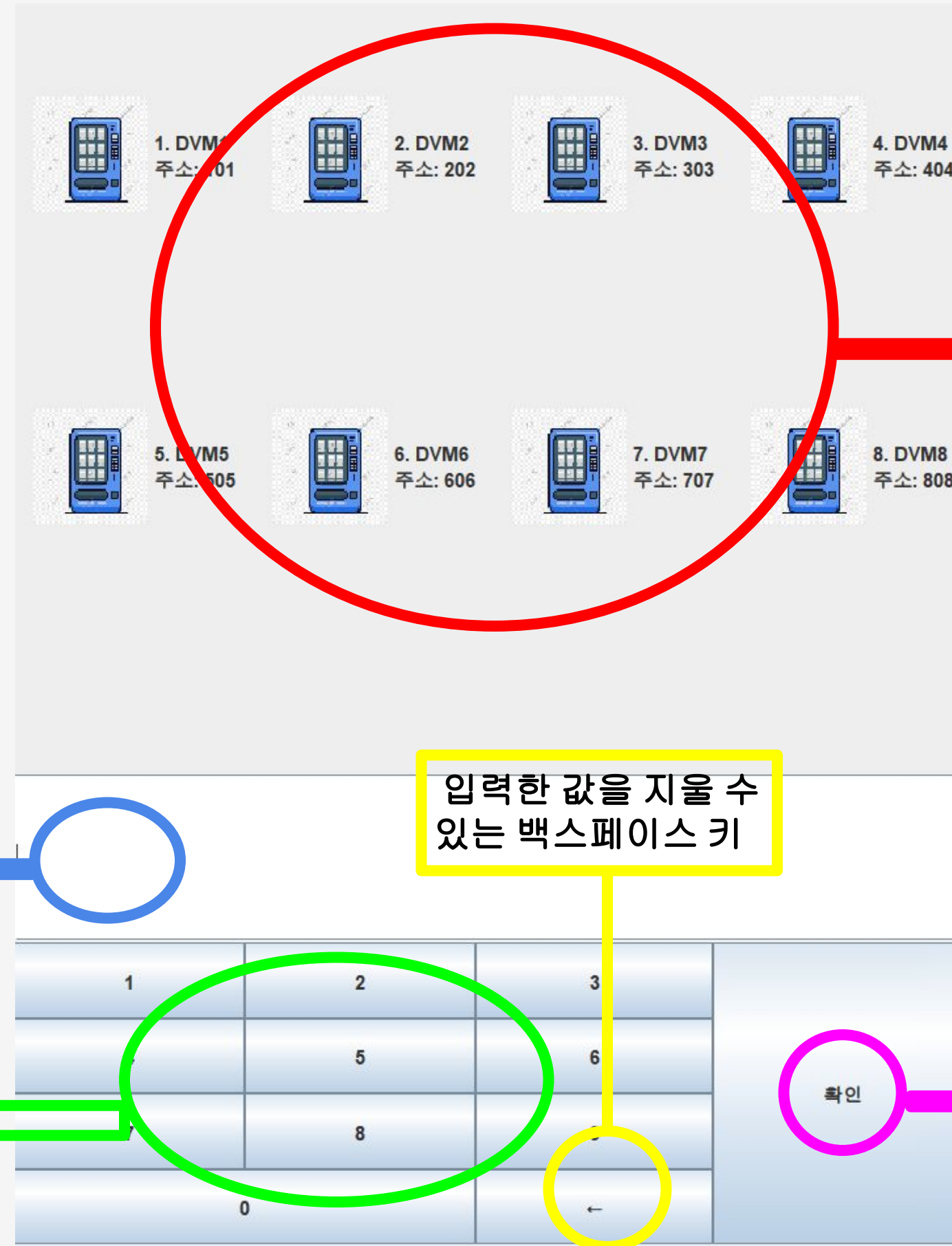
1. 개요



MS129 DVM은 기존 자판기의 불편함을 개선하기 위해 고안된 자판기로, DVM 간의 네트워크 통신을 통해 자판기가 가지고 있는 음료수의 재고 확인을 돕고 사용자에게 선결제의 편리함을 제공한다.

총 8대의 DVM이 존재하며, 각 DVM은 오직 7종류의 음료만을 가지고 있다. 사용자가 이용하고 있는 DVM이 원하는 음료를 가지고 있지 않다면 선결제를 통해 그 음료에 대한 계산을 미리 하고 발급받은 코드를 통해 다른 자판기에서 해당 음료를 구매할 수 있다.

2. DVM 구성



DVM의 메인 스크린으로 DVM 선택, 음료 선택, 결제 방법 선택 등의 시스템 상 진행상황과 사용자에게 알리는 메시지를 출력하는 화면

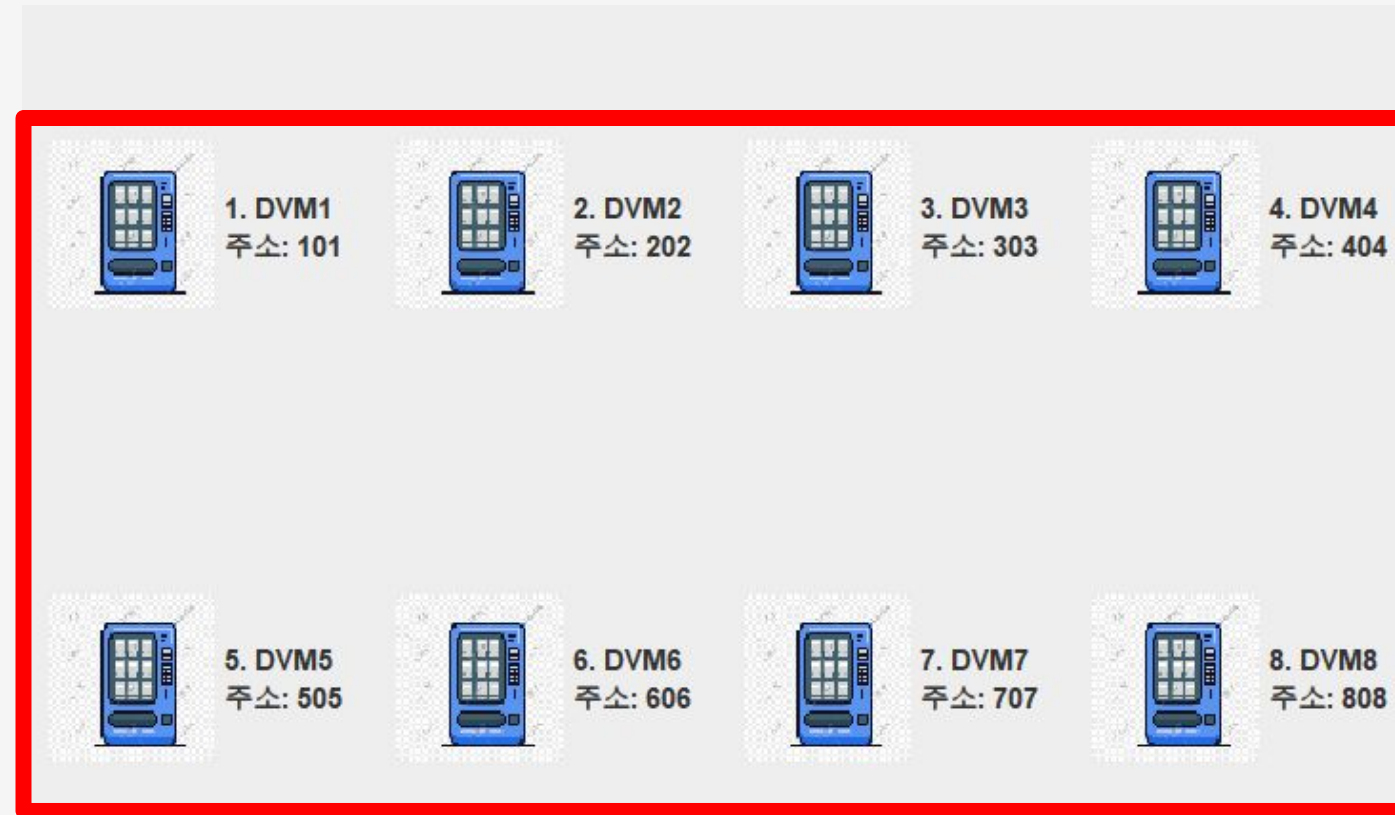
사용자가 다이얼패드를 통해 입력한 값이 나타나는 화면

입력한 값을 지울 수 있는 백스페이스 키

사용자가 값을 입력할 수 있는 다이얼패드

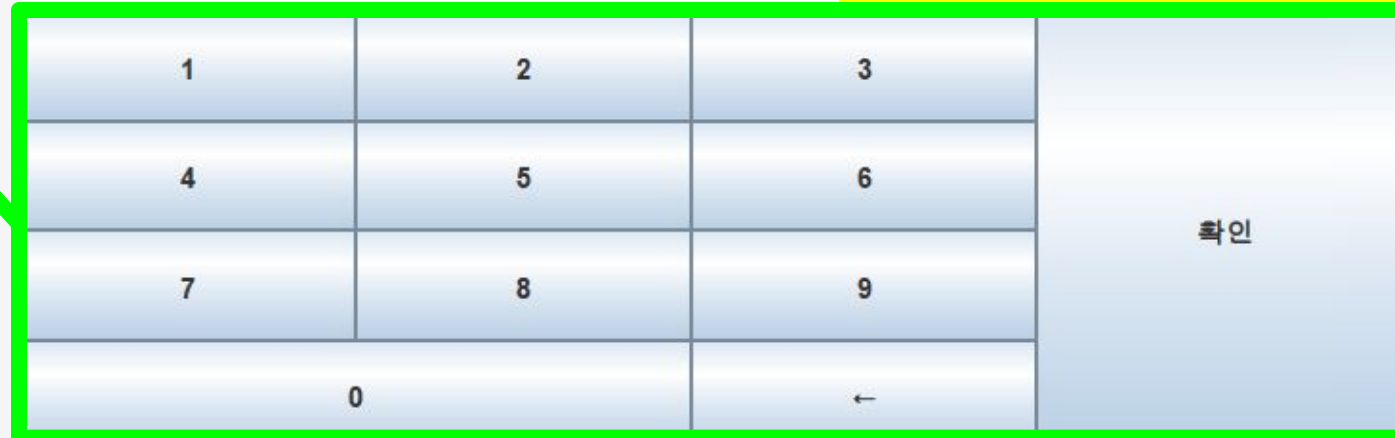
사용자가 다이얼패드를 통해 입력한 값을 저장하는 키

3. DVM 선택



1) 총 8개의 DVM들이 DVM 이름, 주소와 함께 출력돼있다. 사용자들은 이 DVM들 중 하나의 DVM을 선택하여야 한다.

2) 사용자가 DVM을 선택할 수 있는 키패드이다. 사용자는 키패드를 터치하여 번호를 입력할 수 있다. 번호 입력후 확인을 누른다.



3) 사용자가 어떤 DVM을 선택하였는지 DVM의 번호와 함께 출력해주는 메시지

4. 음료 선택

1) 1번 - 7번
현재 DVM
판매메뉴 제공

1.코카콜라 1500원 (10개) 2.펄시콜라 1500원 (11개) 3.칠성사이다 1500원 (0개) 4.스프라이트 1500원 (10개) 5.환타오렌지 1500원 (8개)

6.환타포도 1500원 (1개) 7.핫식스 1500원 (10개) 8.레드불 1500원 (0개) 9.몬스터드링크 1500원 (0개) 10.락텐션 1500원 (0개)

11.포카리스웨트 1500원 (0개) 12.게토레이 1500원 (0개) 13.파워에이드 1500원 (0개) 14.밀키스 1500원 (0개) 15.레쓰비 1500원 (0개)

16.스파클링 1500원 (0개) 17.비락식혜 1500원 (0개) 18.솔의눈 1500원 (0개) 19.데자와 1500원 (0개) 20.마운틴듀 1500원 (0개)

1) 8번 - 20번 까지
현재 DVM 에서
판매하지 않는 메뉴제공
: 선택 가능

2) Dial Button 클릭으로
음료 번호 버튼 선택 후,
확인 버튼 클릭

| | | |
|---|---|---|
| 1 | 2 | 3 |
| 4 | 5 | 6 |
| 7 | 8 | 9 |
| 0 | ← | |

확인

Message

i 1번 음료를 선택하셨습니다.

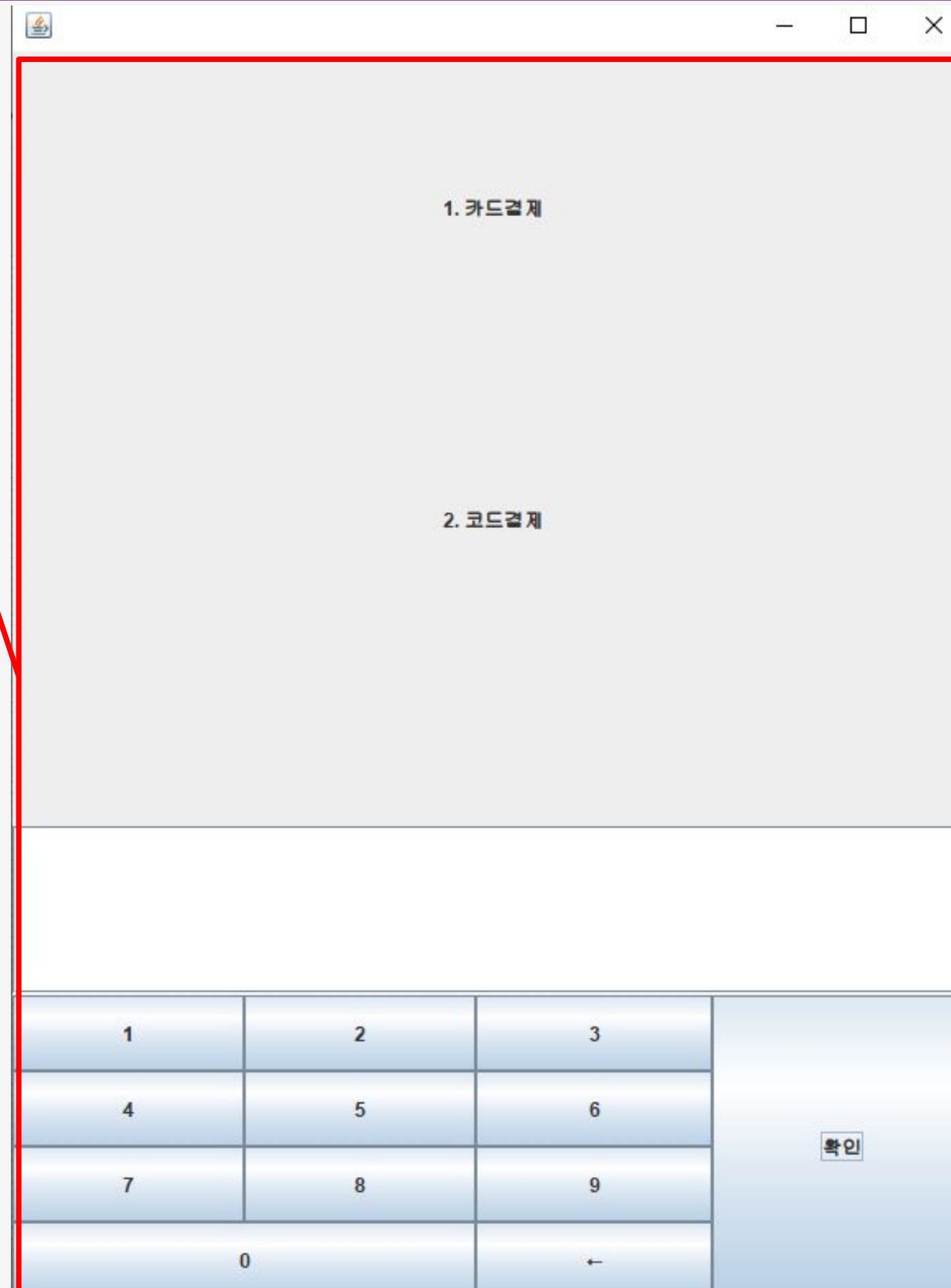
OK

3) 음료 선택 시, 팝업
OK 버튼 클릭-> 다음 화면 이동

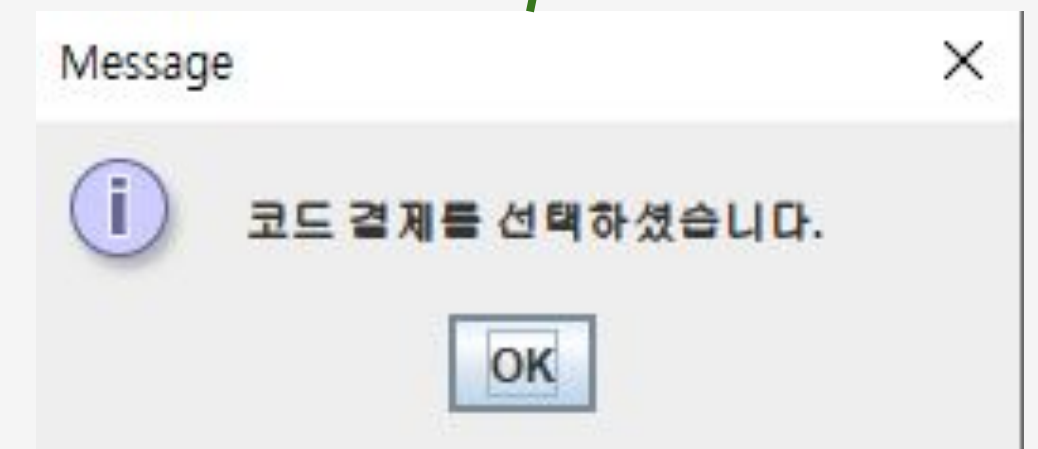
5. 결제 방법 선택

1)결제 방법 메뉴
2가지 제공

: Dial Button사용하여,
1 또는 2 버튼 클릭 후
확인 버튼 클릭



2) 결제 방법 선택 후
팝업 발생 시,
OK 버튼 클릭 -> 다음화면 이동



6. 카드 결제

Dial Button 사용하여
결제할 카드의 번호 버튼 선택 후,
확인 버튼 클릭

결제 가능 카드번호
: 12341234 (잔액 10000원)
: 11111111 (잔액 0원)
: 10000000 (잔액 10000원)

카드 입력번호를 입력해주세요.
(성공 번호: 1234 1234)

| | | | |
|---|---|---|----|
| 1 | 2 | 3 | 확인 |
| 4 | 5 | 6 | |
| 7 | 8 | 9 | |
| 0 | ← | | |

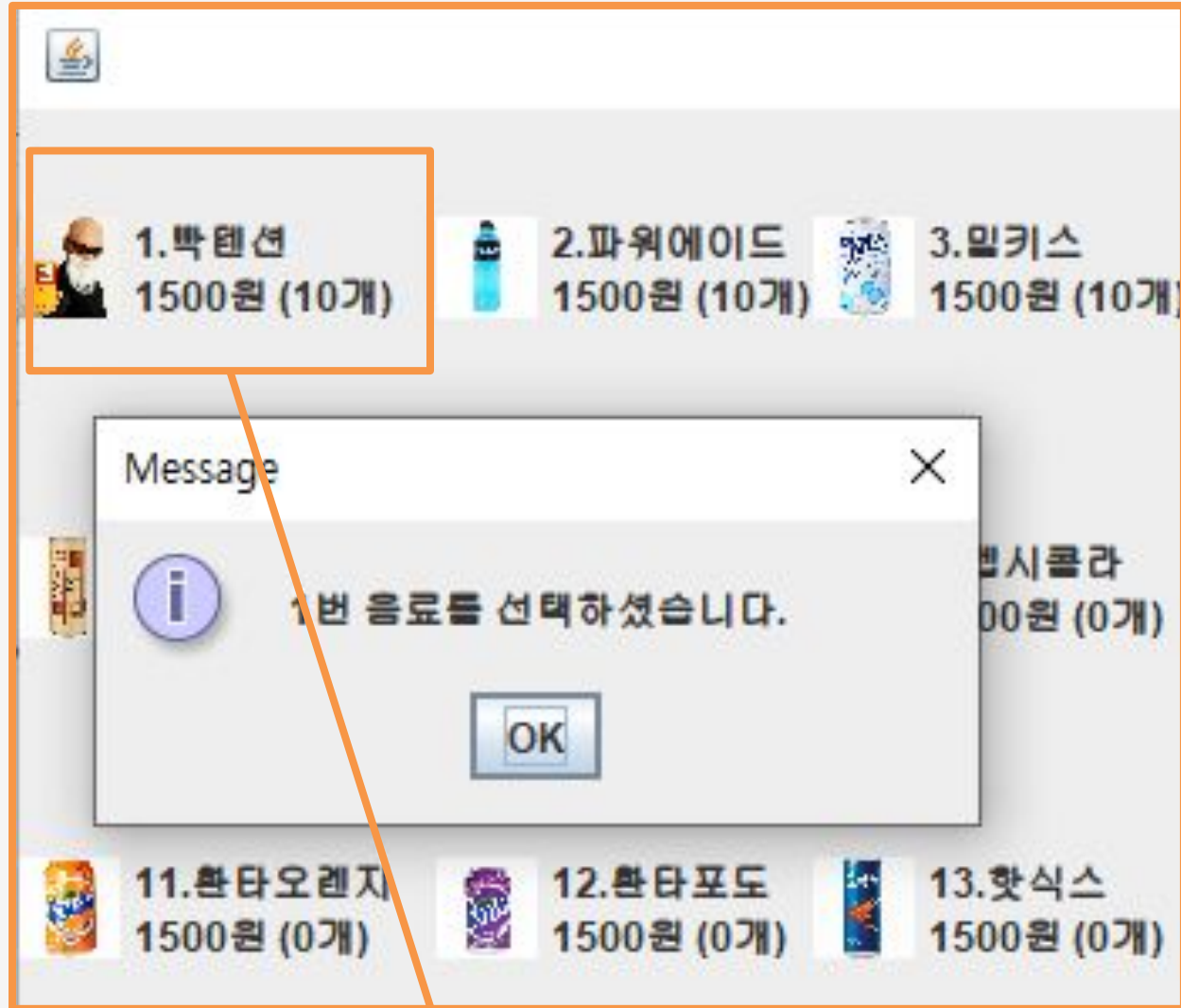
7. 선결제

1) (8~20)번 음료를 선택시,
네트워크를 통해 다른 DVM에
재고 확인 후 재고가 있다면
선결제로 넘어감

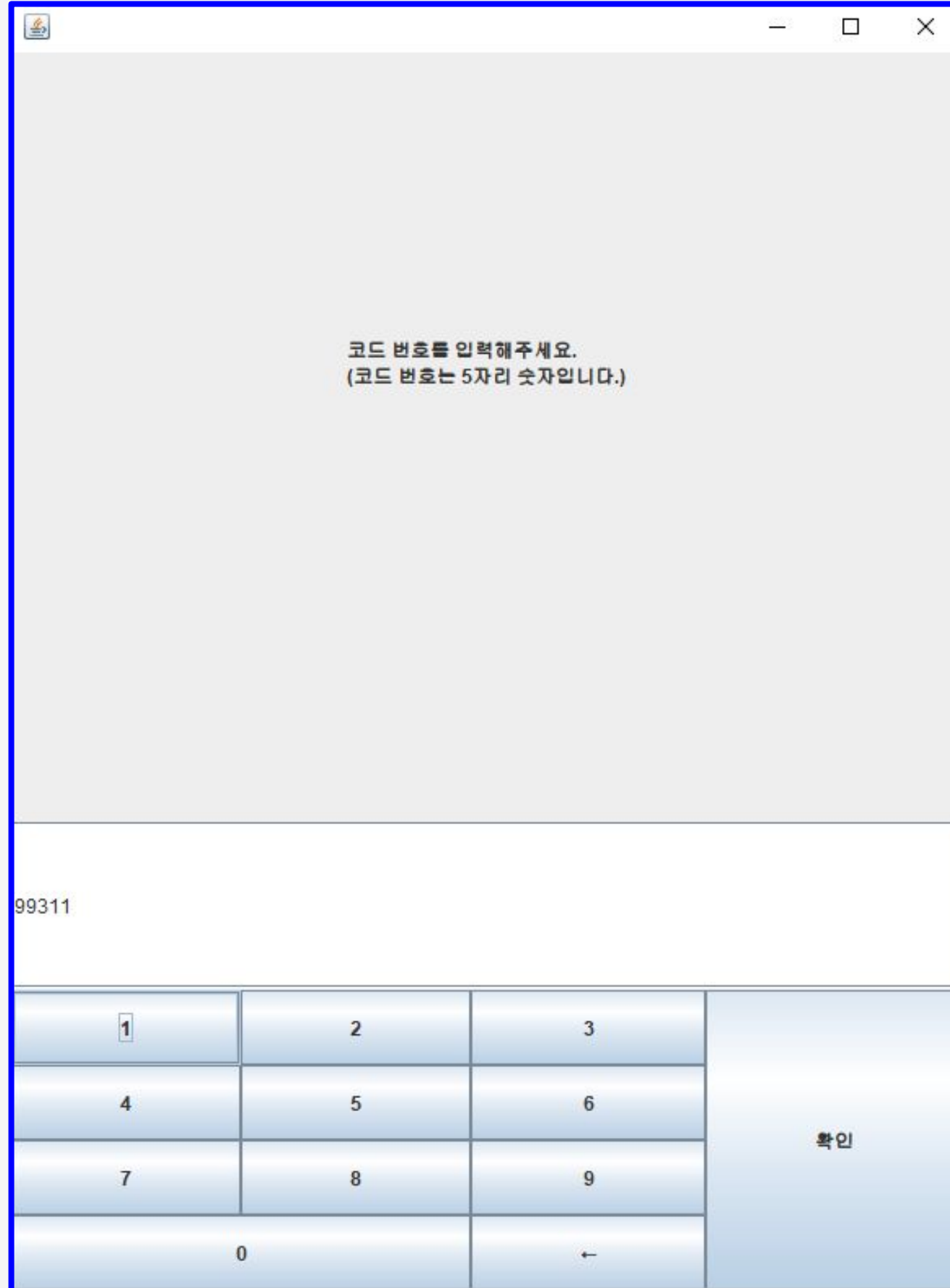
The screenshot shows a Vending Machine (VDM) interface. At the top, a message dialog box is displayed with the text: "현재 DVM에 해당 음료의 재고가 없지만 다른 DVM에 재고가 존재합니다. 선결제로 넘어갑니다." (The current VDM does not have the selected beverage in stock, but it is available at other VDMs. Proceed to pre-payment.) Below this, a card payment screen is shown with the prompt "카드 일련번호를 입력해주세요." (Please enter the card number.) and a list of available VDMs for purchase: DVM2 (202), DVM3 (303), DVM4 (404), DVM5 (505), DVM6 (606), DVM7 (707), and DVM8 (808). The card payment screen also displays: "선결제 진행 DVM: 1", "선결제 한 음료수: 박테션", "음료 가격: 1500", "선결제 후 카드 잔고: 8500원", and "발급 코드: '99311'". At the bottom of the interface is a numeric keypad with buttons for digits 1-9, 0, and a left arrow, along with a "확인" (Confirm) button.

2) 카드결제 진행 후 발급된
인증코드와 네트워크를
통해 구매가능한 DVM들의
위치 출력

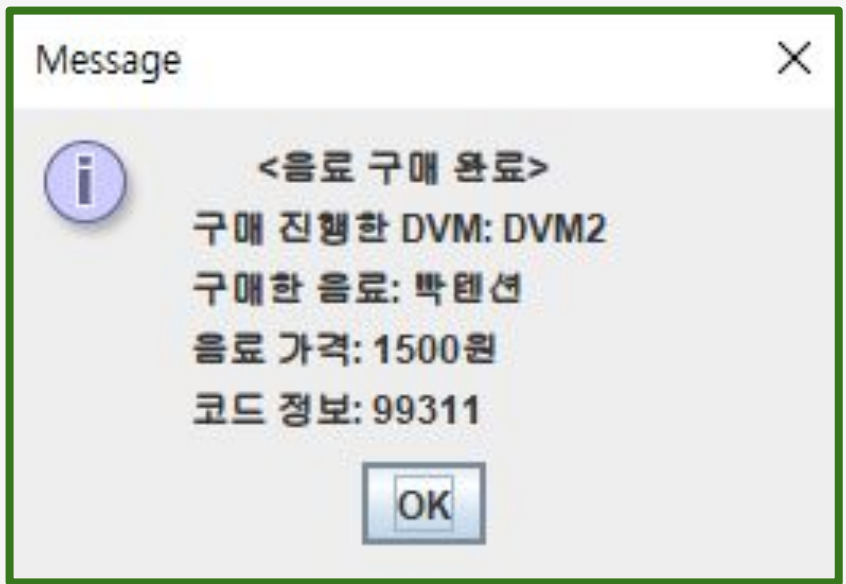
8. 코드 결제



1) 위치정보를 바탕으로 DVM2로 이동해 선결제 했던 음료수를 찾음



2) 선결제 때 발급받았던 코드 '99311'를 입력함



3) 성공적으로 구매 완료, 사용된 코드는 삭제됨

9. 데모영상



<https://youtu.be/fuhzgaLJMx4>