



순천향대학교
SOON CHUN HYANG
UNIVERSITY

Java 프로그래밍 실습

팀프로젝트

담당교수	김석훈 교수님
학과	컴퓨터소프트웨어공학과
학번	20204005
이름	김필중
제출일	2021년 12월 15일

1. 개요

1.1 프로젝트 명

본 프로젝트는 “자판기 관리 프로그램” 개발 명칭을 지니고 있다.

1.2 프로젝트 기간

- 프로젝트 기간: 2021년 8월 31일 ~ 2021년 12월 15일

1.3 프로젝트 목적

자판기 관리 프로그램 개발을 통해 자바 언어에 대한 능력을 향상시키기 위한 2학기 팀 프로젝트이다. 학교 혹은 일상에서 쉽게 접할 수 있는 자판기는 대부분 그냥 버튼을 누르면 나온다 라는 정도의 기능만 생각을 했지만 이번 프로젝트를 통해 자판기를 구현하면서 자판기에 어떤 기능이 필요하고 그것을 코드로 구현하는 것이 목적이다.

1.4 프로젝트 기대효과

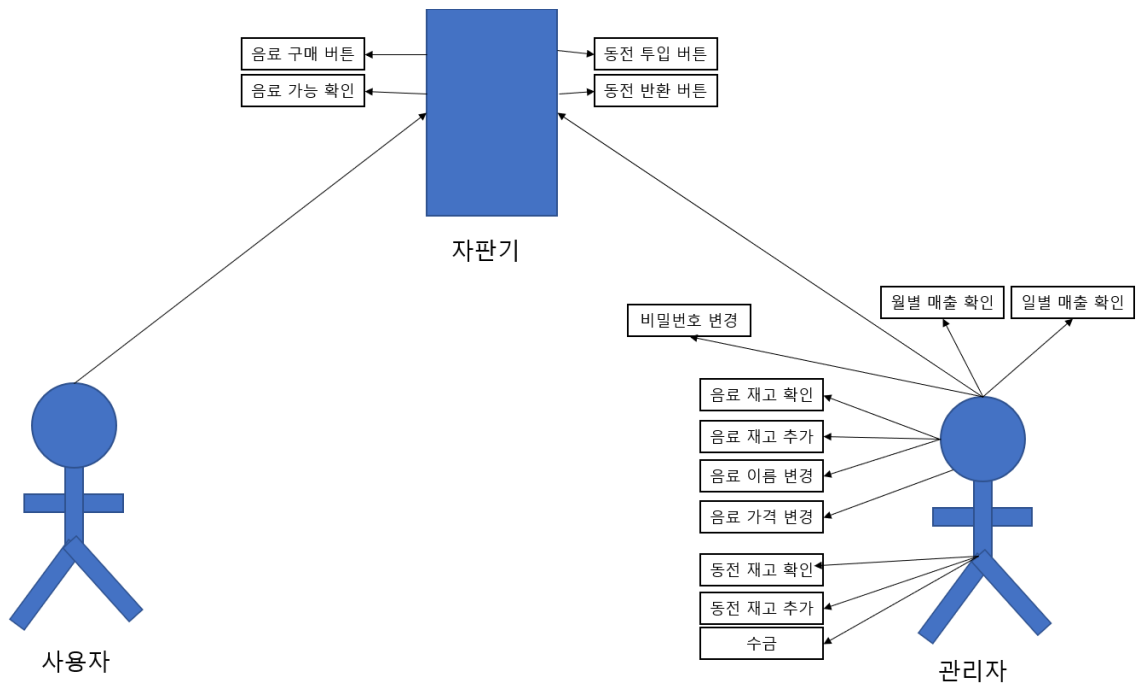
기대효과로 자바 실력 향상과 프로젝트 경험 능력을 쌓을 수 있는 기대효과가 있다. 자판기에 다양한 기능을 더 추가할 수 있는 능력을 향상 가능성이 있다. 다양한 GUI를 활용한 개발이므로 그래픽 쪽으로 비약적인 향상 가능성이 있음

2. 요구분석

2.1 유스케이스

사용자	최종 유스케이스	기타
사용자	음료 구매	'구매 버튼' 유스케이스의 이벤트 흐름
		'동전 투입' 유스케이스의 이벤트 흐름
자판기	동전 입출력	'동전 반환' 유스케이스의 이벤트 흐름
	음료 출력	'음료 구매 가능 표시' 유스케이스의 이벤트 흐름
관리자	음료 관리	'음료 재고 조회' 별도의 유스케이스
		'음료 추가' 별도의 유스케이스
		'음료 가격 변경' 별도의 유스케이스
		'음료 이름 변경' 별도의 유스케이스
	동전 관리	'동전 재고 확인' 별도의 유스케이스
		'동전 개수 추가' 별도의 유스케이스
		'수금' 별도의 유스케이스
	장부 관리	'월별 매출 확인' 별도의 유스케이스
		'일별 매출 확인' 별도의 유스케이스
	관리자 설정	'비밀번호 변경' 별도의 유스케이스

2.2 유스케이스 다이어그램



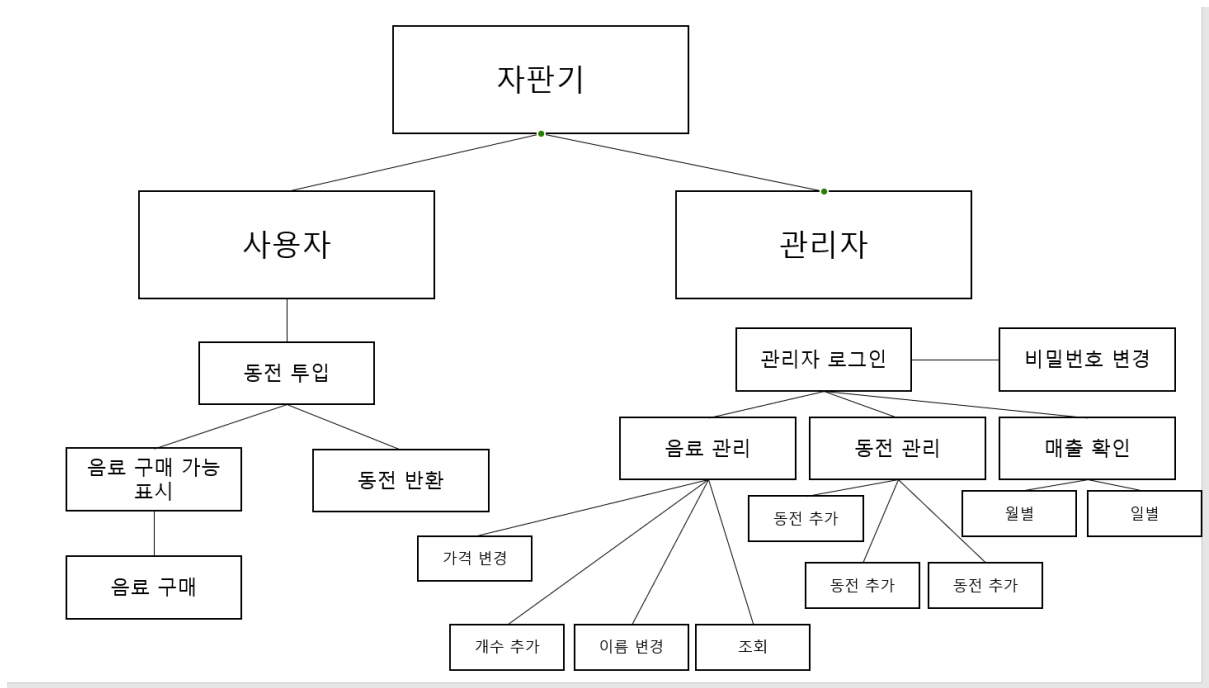
3. 목표

3.1 목표

현재 일상에서 볼 수 있는 자판기의 기능을 구현하면서 비슷하게 제작을 하는 것을 목표로 하고, 관리자 메뉴도 예상을 하면서 제작하는 것이 목표이다

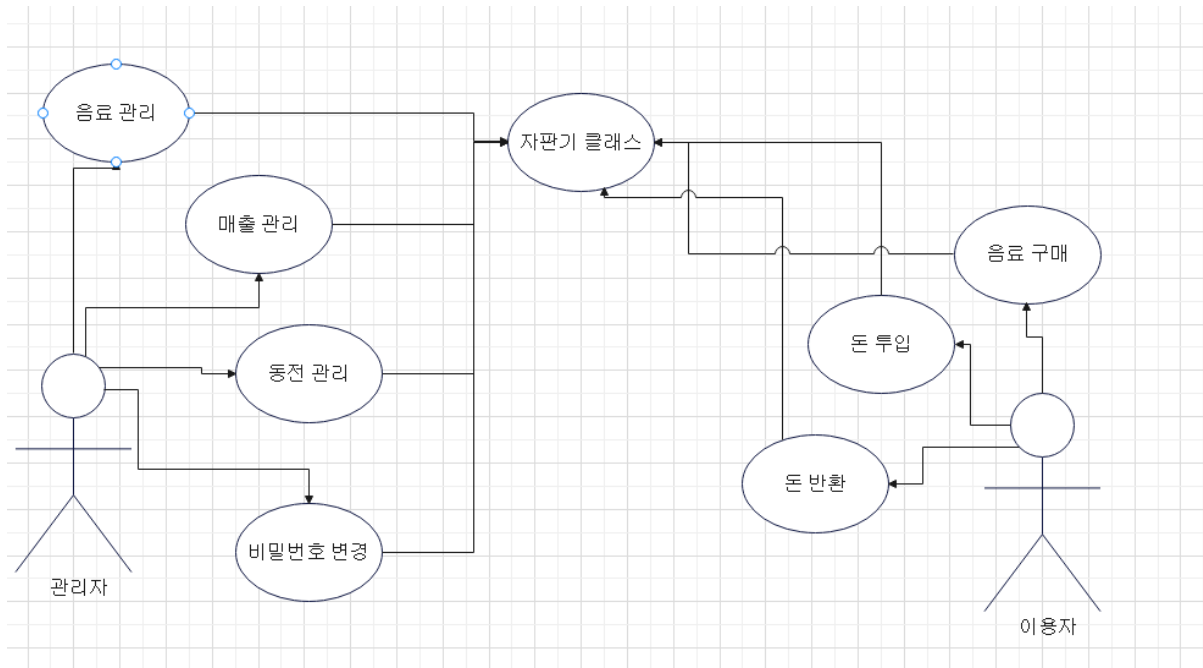
3.2 기능

- 사용자가 동전을 넣는 기능
- 사용자가 동전을 반환하는 기능
- 음료 구매 기능
- 가격에 맞게 음료 구매를 표시해주는 기능
- 관리자 로그인 기능
- 음료 재고 조회, 이름 변경, 가격 변경, 음료 추가 기능
- 잔돈 재고 조회, 잔돈 추가, 수금 기능
- 비밀번호 변경 기능
- 일별 매출액, 월별 매출액 확인 기능

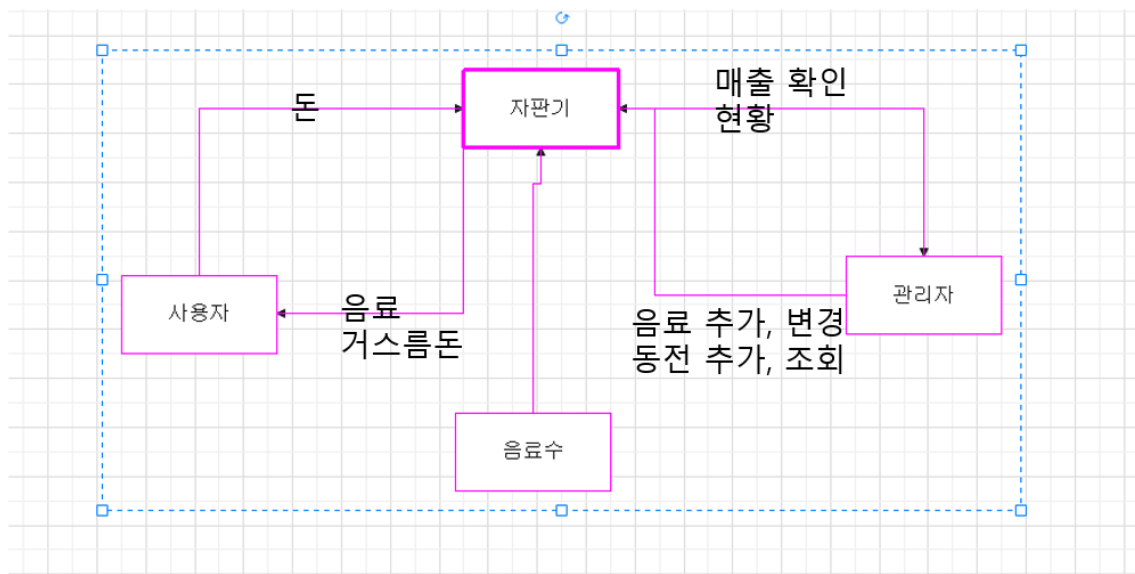


4. 설계

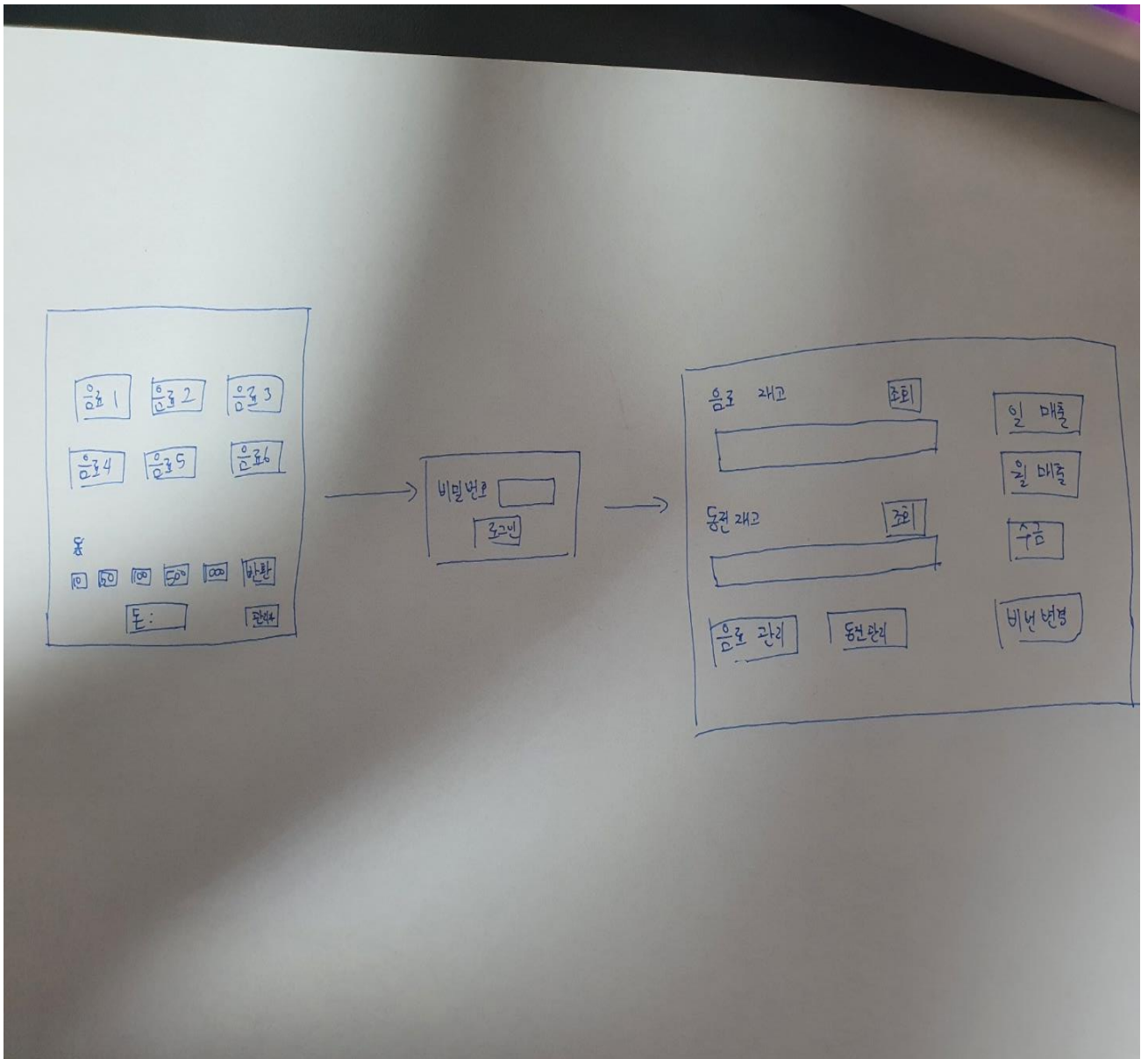
- 상위 설계



- 모델



- 대략적인 인터페이스



- 상세 설계

1. 클래스

Main_Vendig_Class: 메인 틀을 구현하기 위한 클래스

Vending_Drink: 음료 관리 클래스

Vending_Money: 동전 관리 클래스

Vending_Machine_Panel: 자판기의 GUI를 구현하기 위한 클래스

Admin_Login: 관리자 로그인을 구현하기 위한 클래스

Vending_Admin_Setting: 관리자 기능 GUI를 구현, 관리자 기능에 대한 클래스

Admin_Drink_Change: 음료 이름 변경 클래스

Admin_Drink_Change_Price: 음료 가격 변경 클래스

Admin_Change_Pw: 관리자 비밀번호 변경 구현 클래스

Admin_Drink: 음료 관리 gui 구현 클래스

Admin_File: 월별, 일별 관리 구현 클래스

Admin_Money_get: 수금 관리 구현 클래스

Admin_Money: 동전 관리 gui와 관리 기능 클래스

2. 클래스 설계

Main_Vendig_Class

```
class Vending_Frame extends JFrame {

    // 자판기 프레임 생성자 선언(비밀번호를 입력)
    public Vending_Frame(String pppw) {
        // 돈을 동적할당을 해준다
        Money_Array.MoneyL.add(new Vending_Money("10원", 10, 5));
        Money_Array.MoneyL.add(new Vending_Money("50원", 50, 5));
        Money_Array.MoneyL.add(new Vending_Money("100원", 100, 5));
        Money_Array.MoneyL.add(new Vending_Money("500원", 500, 5));
        Money_Array.MoneyL.add(new Vending_Money("1000원", 1000, 5));

        // 음료를 동적할당을 해준다
        Drink_Array.DrinkL.add(new Vending_Drink("물", 450, 3));
        Drink_Array.DrinkL.add(new Vending_Drink("커피", 500, 3));
        Drink_Array.DrinkL.add(new Vending_Drink("이온음료", 550, 3));
        Drink_Array.DrinkL.add(new Vending_Drink("고급커피", 700, 3));
        Drink_Array.DrinkL.add(new Vending_Drink("탄산음료", 750, 3));
        Drink_Array.DrinkL.add(new Vending_Drink("비싼음료", 1000, 3));

        setTitle("자판기"); // 프레임 이름을 설정한다
        setPreferredSize(new Dimension(650, 600)); // 크기를 설정한다
        setLocationRelativeTo(null); // 창이 가운데에 뜨도록 함
        setLocation(300, 300); // 위치를 설정한다
        setDefaultCloseOperation(EXIT_ON_CLOSE); // 창을 끄면 프로그램을 종료
        setVisible(true); // 창을 눈에 보이도록 함
        setSize(640, 480); // 사이즈는 640*480
        setResizable(false); // 사이즈 재조정 불가능

        Container contentPanel = getContentPane(); // contentPanel을 Container로 getContentPane을 한다

        JPanel Panel1 = new Vending_Machine_Panel1(); // 판넬을 선언한다

        contentPanel.add(Panel1, BorderLayout.CENTER); // contentPanel에 판넬을 추가한다

        JPanel Panel2 = new JPanel(); // 판넬을 선언한다
        JButton click = new JButton("관리자 로그인"); // 버튼 로그인을 만든다
        Panel2.add(click); // 버튼을 추가한다
        click.setPreferredSize(new Dimension(150, 100)); // 크기를 설정한다
        Panel2.setBackground(new Color(216, 141, 186)); // 배경을 설정한다
        contentPanel.add(Panel2, BorderLayout.SOUTH); // 판넬을 추가한다

        // 관리자 로그인 클릭시
        click.addActionListener(new ActionListener() {
            public void actionPerformed(ActionEvent e) {
                new Vending_Admin_Login(pppw); // 관리자 로그인 클래스 호출
                dispose(); // 화면 종료
            }
        });

        setVisible(true); // 화면 출력
        pack();
    }
}
```

Vending_Drink 클래스

```
// 음료수 클래스 선언
public class Vending_Drink {
    // 필요한 변수 선언
    private String Drink_name; // 음료수 이름
    private int price; // 음료 가격
    private int number; // 음료 개수

    // 음료 생성자 선언
    public Vending_Drink(String name, int price, int number) {
        this.Drink_name = name; // 음료 이름 정한다
        this.price = price; // 음료 가격 정한다
        this.number = number; // 음료 개수 정한다
    }

    // 음료 이름 반환 메소드
    public String getDrinkName() {
        return this.Drink_name;
    }

    // 음료 가격 반환 메소드
    public int getDrinkPrice() {
        return this.price;
    }

    // 음료 개수 반환 메소드
    public int getDrinkNumber() {
        return this.number;
    }

    // 음료 개수 빼는 메소드
    public void Drinkextraction(int cnt) {
        this.number = this.number - cnt;
    }

    // 음료 개수 추가 메소드
    public void Drinkinsert(int cnt) {
        this.number = this.number + cnt;
    }

    // 음료 개수가 있는지 확인
    public boolean check_drink() {
        if (this.getDrinkNumber() < 1)
            return false;
        else
            return true;
    }

    // 음료 이름 변경 메소드
    public void setName(String name) {
        this.Drink_name = name;
    }

    // 음료 가격 변경 메소드
    public void setPrice(int p) {
        this.price = p;
    }
}
```

Vending_Money 클래스

```
public class Vending_Money {
    // 품목 관리 선언
    private String money_name; // 돈 이름
    private int money_price; // 돈의 가격
    private int money_number; // 돈의 개수
    protected static int total; // 돈의 총 합
    private int money; // 돈

    // 돈의 생성자 선언
    public Vending_Money(String name, int price, int number) {
        this.money_name = name; // 돈 이름
        this.money_price = price; // 돈의 가격을 저장함
        this.money_number = number; // 돈의 개수를 저장함
    }

    // 금액으로 교환한 초기화
    public Vending_Money() {
        money = 0;
        total = 0;
    }

    // 돈의 이름 반환 메소드
    public String getMoneyName() {
        return this.money_name;
    }

    // 돈의 가격 반환 메소드
    public int getMoneyPrice() {
        return this.money_price;
    }

    // 돈의 개수 반환 메소드
    public int getMoneyNumber() {
        return this.money_number;
    }

    // 돈 개수 줄이기 메소드
    public void moneyInsert(int cnt) {
        this.money_number += this.money_number - cnt;
    }

    // 돈 개수 늘리기 메소드
    public void moneyInsert(int cnt) {
        this.money_number += this.money_number + cnt;
    }

    // 돈 추가 메소드
    public void moneyInsert(int cnt) {
        total += cnt;
        return total;
    }

    // 돈 개수 설정 메소드
    void setMoney(int n) {
        this.money_price = n;
    }

    // 100원 돈이 들어갔을때 1000원이 넘는지 확인하는 메소드
    boolean insert_Money_100(int money_100) {
        int n = 0;
        n += money_100 * 10;
        total += n;
        // 만약 1000원이 넘지않는다면 false 반환
        if (total > 1000) {
            total -= n;
            return false;
        }
        return true;
    }

    // 100원 돈이 들어갔을때 1000원이 넘는지 확인하는 메소드
    boolean insert_Money_1000(int money_1000) {
        int n = 0;
        n += money_1000 * 1000;
        total += n;
        // 만약 10000원이 넘지않는다면 false 반환
        if (total > 10000) {
            total -= n;
            return false;
        }
        return true;
    }

    // 1000원 이하로 남은 10000 메소드
    void over_10000() {
        total -= 10000;
    }

    int returnMoney() {
        return total;
    }
}

total = total + n;
// 잔돈 반환 메소드
int return_1000() {
    int r = 1000 * 0;
    // 만약 총 금액이 1000원 이하이면
    if (total > 0) {
        // 잔돈 1000원 돈이 1개 이하일 때 반환
        if (this.getMoneyNumber() < 1) {
            return 0;
        }
        // 총 금액이 1000원 이상일 경우
        if (total > 1000) {
            // 잔돈 1000원 돈이 1개 이하일 때 반환
            for (int i = 0; i <= this.getMoneyNumber(); i++) {
                int money = total - 1000;
                this.money_extraction(i);
                total = money;
            }
            // 잔돈 1000원 이하로 남은
            if (total < 1000) {
                r = total + 1 * 1;
                break;
            }
        }
        return r_1000;
    }
    // 잔돈 반환 메소드
    int return_500() {
        int r = 500 * 0;
        // 만약 총 금액이 500원 이하이면
        if (total > 0) {
            // 잔돈 500원 돈이 1개 이하일 때 반환
            if (this.getMoneyNumber() < 1) {
                return 0;
            }
            // 총 금액이 500원 이상일 경우
            if (total > 500) {
                // 잔돈 500원 돈이 1개 이하일 때 반환
                for (int i = 0; i <= this.getMoneyNumber(); i++) {
                    int money = total - 500;
                    this.money_extraction(i);
                    total = money;
                }
                // 잔돈 500원 이하로 남은
                if (total < 500) {
                    r = total + 1 * 1;
                    break;
                }
            }
            return r_500;
        }
    }
}

int r_100 = 0;
// 만약 총 금액이 100원 이하이면
if (total > 0) {
    // 잔돈 100원 돈이 1개 이하일 때 반환
    if (this.getMoneyNumber() < 1) {
        return 0;
    }
    // 만약 1000원 돈이 1개 이하일 때
    if (total > 1000) {
        for (int i = 0; i <= this.getMoneyNumber(); i++) {
            int money = total - 100;
            this.money_extraction(i);
            total = money;
        }
        // 만약 100원 이하로 남은
        if (total < 100) {
            r_100 = 1 * 1;
            break;
        }
    }
    return r_100;
}
// 잔돈 반환 메소드
int return_50() {
    int r = 50 * 0;
    // 만약 총 금액이 50원 이하이면
    if (total > 0) {
        // 잔돈 50원 돈이 1개 이하일 때 반환
        if (this.getMoneyNumber() < 1) {
            return 0;
        }
        // 총 금액이 50원 이상일 경우
        if (total > 50) {
            // 잔돈 50원 돈이 1개 이하일 때 반환
            for (int i = 0; i <= this.getMoneyNumber(); i++) {
                int money = total - 50;
                this.money_extraction(i);
                total = money;
            }
            // 잔돈 50원 이하로 남은
            if (total < 50) {
                r = total + 1 * 1;
                break;
            }
        }
        return r_50;
    }
}
```

Admin_Login 클래스

```
// 로그인 클래스
public class Vending_Admin_Login extends JFrame {

    private JPanel loginPanel = new JPanel(new GridLayout(4, 2)); // 로그인 패널을 4*2로 초기화
    private JLabel pwLabel = new JLabel("비밀번호 "); // 비밀번호 라벨 추가
    private JPasswordField pwText = new JPasswordField(); // 비밀번호 입력하는 칸을 추가
    private JButton loginBtn = new JButton("로그인"); // 로그인 버튼 추가
    JButton end = new JButton("종료"); // 종료 버튼 추가

    // 생성자 선언
    public Vending_Admin_Login(String pppww) {
        this.setContentPane(loginPanel); // 컴포넌트를 붙인다
        // 패널에 추가한다
        loginPanel.add(pwLabel);
        loginPanel.add(pwText);
        loginPanel.add(loginBtn);
        loginPanel.add(end);
        // 라벨을 가운데로 설정한다
        pwLabel.setHorizontalAlignment(NORMAL);

        setSize(350, 150); // 크기 설정
        this.setLocationRelativeTo(null); // 위치 설정
        this.setVisible(true); // 화면에 표시
        setDefaultCloseOperation(JFrame.DISPOSE_ON_CLOSE); // x버튼 누르면 꺼지게 한다

        // 로그인 버튼을 누르면
        loginBtn.addActionListener(new ActionListener() {
            public void actionPerformed(ActionEvent e) {

                // 비번 확인 테스트 코드
                String pw = pwText.getText().trim();

                // 만약 아무것도 입력이 안되면 리턴
                if (pw.length() == 0) {
                    JOptionPane.showMessageDialog(null, "비밀번호를 입력 하셔야 합니다.", "비번을 입력!", JOptionPane.DEFAULT_OPTION);
                    return;
                }

                // 같으면 어드민 패널 호출
                if (pw.equals(pppww)) {
                    JOptionPane.showMessageDialog(null, "로그인 성공", "로그인 확인!", JOptionPane.DEFAULT_OPTION);
                    new Vending_Admin_Setting(pppww);
                    dispose();
                    return;
                }
                // 아닐 경우는 실패
                JOptionPane.showMessageDialog(null, "로그인 실패", "로그인 확인!", JOptionPane.DEFAULT_OPTION);
            }
        });

        // 종료 버튼은 그냥 끝
        end.addActionListener(new ActionListener() {
            public void actionPerformed(ActionEvent e) {
                new Vending_Frame(pppww);
                dispose();
            }
        });
    }
}
```

Vending_Admin_Setting 클래스

[illegible]

Admin_Drink_Change 클래스

```
// 이름 변경 클래스
public class Admin_Drink_Change extends JFrame {

    private JPanel Change = new JPanel(new GridLayout(3, 2)); // 3*2로 설정
    private JLabel ChangeL = new JLabel("바꿀 이름"); // 라벨 설정
    private JTextField ChangeN = new JTextField(); // 입력칸 설정
    private JButton ChangeB = new JButton("변경");// 변경 버튼 설정

    // 생성자 선언
    public Admin_Drink_Change(int n) {
        // 추가
        this.setContentPane(Change);
        Change.add(ChangeL);
        Change.add(ChangeN);
        Change.add(ChangeB);

        // 라벨 가운데 정렬
        ChangeL.setHorizontalAlignment(NORMAL);

        setSize(350, 150); // 크기 설정
        this.setLocationRelativeTo(null); // 위치 설정
        this.setVisible(true); // 화면에 보이게
        setDefaultCloseOperation(JFrame.DISPOSE_ON_CLOSE); // x버튼 누를시 종료

        // 버튼 누르면 입력한 이름으로 변경
        ChangeB.addActionListener(new ActionListener() {
            public void actionPerformed(ActionEvent e) {
                String b = ChangeN.getText().trim();
                Drink_Array.DrinkL.get(n).setName(b);
                JOptionPane.showMessageDialog(null, "이름이 " + b + " 로 변경되었습니다");
                dispose();
            }
        });
    }
}
```

Admin_Drink_Change_Price 클래스

```
// 음료 이름 변경 클래스
public class Admin_Drink_Change_Price extends JFrame {

    // 필요한 변수 선언
    private JPanel Change = new JPanel(new GridLayout(3, 2)); // 패널 설정
    private JLabel ChangeL = new JLabel("바꿀 금액"); // 라벨 설정
    private JTextField ChangeN = new JTextField(); // 표시 부분 설정
    private JButton ChangeB = new JButton("변경"); // 버튼 설정

    // 생성자 선언
    public Admin_Drink_Change_Price(int n) {
        this.setContentPane(Change);
        Change.add(ChangeL);
        Change.add(ChangeN);
        Change.add(ChangeB);

        // 라벨 가운데 정렬
        ChangeL.setHorizontalAlignment(NORMAL);

        // 현재 프레임 창 가운데 정렬 setSize를 먼저 해주어야 정상적으로 프레임이 가운데 정렬이 된다
        setSize(350, 150);
        this.setLocationRelativeTo(null);
        this.setVisible(true);
        setDefaultCloseOperation(JFrame.DISPOSE_ON_CLOSE);

        // 버튼을 누를경우 변경
        ChangeB.addActionListener(new ActionListener() {
            public void actionPerformed(ActionEvent e) {
                String b = ChangeN.getText().trim();
                int num = Integer.parseInt(b);
                Drink_Array.DrinkL.get(n).setPrice(num);
                JOptionPane.showMessageDialog(null, "음료의 가격이 " + num + "원으로 변경되었습니다");
                dispose();
            }
        });
    }
}
```

Admin_Change_Pw 클래스

```

// 비밀번호 변경 설정 JFrame 상속
public class Admin_Change_pw extends JFrame {

    private JPanel loginPanel = new JPanel(new GridLayout(4, 2)); // 패널 4*2로 설정
    private JLabel pwLabel = new JLabel("바꿀 비밀번호 "); // 바꿀 비밀번호 라벨 설정
    private JPasswordField pwText = new JPasswordField(); // 비밀번호 입력 설정
    private JButton loginBtn = new JButton("변경"); // 변경 버튼 설정
    JButton end = new JButton("종료"); // 종료 버튼 설정

    // 생성자 선언
    public Admin_Change_pw(String ppww) {
        // 패널에 추가
        this.setContentPane(loginPanel);
        loginPanel.add(pwLabel);
        loginPanel.add(pwText);
        loginPanel.add(loginBtn);
        loginPanel.add(end);
        pwLabel.setHorizontalAlignment(NORMAL);

        setSize(350, 150); // 사이즈 설정
        this.setLocationRelativeTo(null); // 위치 설정
        this.setVisible(true); // 화면에 표시
        setDefaultCloseOperation(JFrame.DISPOSE_ON_CLOSE); // x버튼 누르면 종료

        // 로그인 버튼 설정
        loginBtn.addActionListener(new ActionListener() {
            public void actionPerformed(ActionEvent e) {

                // 비밀번호 변수 설정
                String pw = pwText.getText().trim();

                // 입력 변수가 없으면 오류
                if (pw.length() == 0) {
                    JOptionPane.showMessageDialog(null, "바꿀 비밀번호를 입력 하셔야 합니다.", "비번을 입력!", JOptionPane.DEFAULT_OPTION);
                    return;
                }

                // 아닐경우는 변경
                else {
                    JOptionPane.showMessageDialog(null, "비밀번호 변경이 완료되었습니다", "변경 성공!", JOptionPane.DEFAULT_OPTION);
                    new Vending_Frame(pw);
                    dispose();
                    return;
                }
            }
        });

        // 종료 버튼 누르시 종료
        end.addActionListener(new ActionListener() {
            public void actionPerformed(ActionEvent e) {
                new Vending_Frame(ppww);
                dispose();
            }
        });
    }
}

```

Admin Drink 클래스

[illegible]

Admin_File 클래스

```
// 어드민 파일 클래스
public class Admin_FILE {

    // 일별 매출 메소드
    String Admin_day(String a, String b) {
        String line = "";
        // 예외 처리 함수
        try {
            // 파일 객체 생성
            File file = new File("일별매출\\" + a + " " + b + ".txt");
            // 입력 스트림 생성
            FileReader filereader = new FileReader(file);
            // 입력 버퍼 생성
            BufferedReader bufReader = new BufferedReader(filereader);

            // 라인을 읽고 반환
            while ((line = bufReader.readLine()) != null) {
                return (line);
            }
            // 파일 닫는다
            bufReader.close();
            // 파일이 없으면 오류
        } catch (FileNotFoundException e) {
            // TODO: handle exception
            // 오류 발생시 종료
        } catch (IOException e) {
            System.out.println(e);
        }
        // 파일에서 읽은 값을 반환
        return (line);
    }

    // 월별 매출 메소드
    String Admin_Month(String a, String b) {
        String line = "";
        // 예외 처리 함수
        try {
            // 파일 객체 생성
            File file = new File("월별매출\\" + a + " " + b + ".txt");
            // 입력 스트림 생성
            FileReader filereader = new FileReader(file);
            // 입력 버퍼 생성
            BufferedReader bufReader = new BufferedReader(filereader);

            // 라인을 읽고 반환
            while ((line = bufReader.readLine()) != null) {
                return (line);
            }
            // 파일 닫는다
            bufReader.close();
            // 파일이 없으면 오류
        } catch (FileNotFoundException e) {
            // TODO: handle exception
            // 오류 발생시 종료
        } catch (IOException e) {
            System.out.println(e);
        }
        // 파일에서 읽은 값을 반환
        return (line);
    }
}
```


Admin_Money_get 클래스

```
// 만약 1000원 동전이 5개 이하라면 0 표시
if (Money_Array.MoneyL.get(4).getMoneyNumber() <= 5) {
    get1000 = new JLabel("1000원 : 0개");
    // 5개를 뺀 나머지를 표시
} else {
    num_1000 = Money_Array.MoneyL.get(4).getMoneyNumber() - 5;
    get1000 = new JLabel("1000원 : " + num_1000 + "개");
    Money_Array.MoneyL.get(4).setnum(Money_Array.MoneyL.get(4).getMoneyNumber() - num_1000);
}

Cok.setForeground(Color.WHITE);
Cok.setBackground(Color.BLACK);

getList.setBounds(30, 10, 140, 30);
get10.setBounds(40, 50, 80, 20);
get50.setBounds(40, 80, 80, 20);
get100.setBounds(40, 110, 80, 20);
get500.setBounds(40, 140, 80, 20);
get1000.setBounds(40, 170, 80, 20);
Cok.setBounds(100, 200, 60, 30);

add(getList);
add(get10);
add(get50);
add(get100);
add(get500);
add(get1000);
add(Cok);

setBounds(500, 200, 180, 270);

Cok.addActionListener(new ActionListener() {
    public void actionPerformed(ActionEvent e) {
        Money_Array.MoneyL.get(4).Moneyextraction(Money_Array.MoneyL.get(4).getMoneyNumber() - 5);
        Money_Array.MoneyL.get(3).Moneyextraction(Money_Array.MoneyL.get(3).getMoneyNumber() - 5);
        Money_Array.MoneyL.get(2).Moneyextraction(Money_Array.MoneyL.get(2).getMoneyNumber() - 5);
        Money_Array.MoneyL.get(1).Moneyextraction(Money_Array.MoneyL.get(1).getMoneyNumber() - 5);
        Money_Array.MoneyL.get(0).Moneyextraction(Money_Array.MoneyL.get(0).getMoneyNumber() - 5);
        setVisible(false);
    }
});
}
```

```
// 클래스 선언
public class Admin_Money_get extends JDialog {
    private JLabel getList = new JLabel("수금"); // 수금 이름
    // 각 동전의 라벨을 선언
    private JLabel get10;
    private JLabel get50;
    private JLabel get100;
    private JLabel get500;
    private JLabel get1000;
    private JButton Cok = new JButton("Go"); // 적밀노 버튼

    // 생성자 선언
    public Admin_Money_get(Frame frame, String title) {
        // 기본 설정
        super(frame, title);
        setLayout(null);
        int num_10;
        int num_50;
        int num_100;
        int num_500;
        int num_1000;

        // 만약 10원 동전이 5개 이하라면 0 표시
        if (Money_Array.MoneyL.get(0).getMoneyNumber() <= 5) {
            get10 = new JLabel(" 10원 : 0개");
            // 5개를 뺀 나머지를 표시
        } else {
            num_10 = Money_Array.MoneyL.get(0).getMoneyNumber() - 5;
            get10 = new JLabel(" 10원 : " + num_10 + "개");
            Money_Array.MoneyL.get(0).setnum(Money_Array.MoneyL.get(0).getMoneyNumber() - num_10);
        }

        // 만약 50원 동전이 5개 이하라면 0 표시
        if (Money_Array.MoneyL.get(1).getMoneyNumber() <= 5) {
            get50 = new JLabel(" 50원 : 0개");
            // 5개를 뺀 나머지를 표시
        } else {
            num_50 = Money_Array.MoneyL.get(1).getMoneyNumber() - 5;
            get50 = new JLabel(" 50원 : " + num_50 + "개");
            Money_Array.MoneyL.get(1).setnum(Money_Array.MoneyL.get(1).getMoneyNumber() - num_50);
        }

        // 만약 100원 동전이 5개 이하라면 0 표시
        if (Money_Array.MoneyL.get(2).getMoneyNumber() <= 5) {
            get100 = new JLabel(" 100원 : 0개");
            // 5개를 뺀 나머지를 표시
        } else {
            num_100 = Money_Array.MoneyL.get(2).getMoneyNumber() - 5;
            get100 = new JLabel(" 100원 : " + num_100 + "개");
            Money_Array.MoneyL.get(2).setnum(Money_Array.MoneyL.get(2).getMoneyNumber() - num_100);
        }

        // 만약 500원 동전이 5개 이하라면 0 표시
        if (Money_Array.MoneyL.get(3).getMoneyNumber() <= 5) {
            get500 = new JLabel(" 500원 : 0개");
            // 5개를 뺀 나머지를 표시
        } else {
            num_500 = Money_Array.MoneyL.get(3).getMoneyNumber() - 5;
            get500 = new JLabel(" 500원 : " + num_500 + "개");
            Money_Array.MoneyL.get(3).setnum(Money_Array.MoneyL.get(3).getMoneyNumber() - num_500);
        }
    }
}
```

Admin_Money 클래스

```
// 클래스 선언
public class Admin_Money extends JFrame {

    // 생성자 선언
    public Admin_Money(String ppww) {
        JPanel M = new JPanel(new GridLayout(2, 3)); // 2*3 설정
        // 사이즈, 색상 설정
        M.setPreferredSize(new Dimension(450, 400));
        M.setForeground(new Color(0, 0, 0));
        M.setBackground(new Color(255, 255, 255));

        // =====
        // 동전에 관한 gui 설정 부분
        JPanel Money_Number_1 = new JPanel();
        JLabel Money_Label_1 = new JLabel(Money_Array.MoneyL.get(0).getMoneyname());
        JButton MoneyButton_1 = new JButton(Money_Array.MoneyL.get(0).getMoneyname() + " 추가");
        Money_Number_1.add(new JLabel(new ImageIcon(7 + ".png")));
        MoneyButton_1.setForeground(new Color(0, 0, 255));
        MoneyButton_1.setBackground(new Color(255, 255, 255));
        Money_Number_1.add(Money_Label_1);
        Money_Number_1.add(MoneyButton_1);
        M.add(Money_Number_1);
        Money_Number_1.setBackground(new Color(236, 251, 255));

        JPanel Money_Number_2 = new JPanel();
        JLabel Money_Label_2 = new JLabel(Money_Array.MoneyL.get(1).getMoneyname());
        JButton MoneyButton_2 = new JButton(Money_Array.MoneyL.get(1).getMoneyname() + " 추가");
        Money_Number_2.add(new JLabel(new ImageIcon(8 + ".png")));
        MoneyButton_2.setForeground(new Color(0, 0, 255));
        MoneyButton_2.setBackground(new Color(255, 255, 255));
        Money_Number_2.add(Money_Label_2);
        Money_Number_2.add(MoneyButton_2);
        M.add(Money_Number_2);
        Money_Number_2.setBackground(new Color(236, 251, 255));

        JPanel Money_Number_3 = new JPanel();
        JLabel Money_Label_3 = new JLabel(Money_Array.MoneyL.get(2).getMoneyname());
        JButton MoneyButton_3 = new JButton(Money_Array.MoneyL.get(2).getMoneyname() + " 추가");
        Money_Number_3.add(new JLabel(new ImageIcon(9 + ".png")));
        MoneyButton_3.setForeground(new Color(0, 0, 255));
        MoneyButton_3.setBackground(new Color(255, 255, 255));
        Money_Number_3.add(Money_Label_3);
        Money_Number_3.add(MoneyButton_3);
        M.add(Money_Number_3);
        Money_Number_3.setBackground(new Color(236, 251, 255));

        JPanel Money_Number_4 = new JPanel();
        JLabel Money_Label_4 = new JLabel(Money_Array.MoneyL.get(3).getMoneyname());
        JButton MoneyButton_4 = new JButton(Money_Array.MoneyL.get(3).getMoneyname() + " 추가");
        Money_Number_4.add(new JLabel(new ImageIcon(10 + ".png")));
        MoneyButton_4.setForeground(new Color(0, 0, 255));
        MoneyButton_4.setBackground(new Color(255, 255, 255));
        Money_Number_4.add(Money_Label_4);
        Money_Number_4.add(MoneyButton_4);
        M.add(Money_Number_4);
        Money_Number_4.setBackground(new Color(236, 251, 255));

        JPanel Money_Number_5 = new JPanel();
        JLabel Money_Label_5 = new JLabel(Money_Array.MoneyL.get(4).getMoneyname());
        JButton MoneyButton_5 = new JButton(Money_Array.MoneyL.get(4).getMoneyname() + " 추가");
        Money_Number_5.add(new JLabel(new ImageIcon(7 + ".png")));
        MoneyButton_5.setForeground(new Color(0, 0, 255));
        MoneyButton_5.setBackground(new Color(255, 255, 255));
        Money_Number_5.add(Money_Label_5);
        Money_Number_5.add(MoneyButton_5);
        M.add(Money_Number_5);
        Money_Number_5.setBackground(new Color(236, 251, 255));

        // =====

        // 종료 버튼 설정
        JButton ENDButton = new JButton("종료");
        ENDButton.setBounds(1, 1, 1, 1);
        ENDButton.setForeground(new Color(0, 0, 255));
        ENDButton.setBackground(new Color(255, 255, 255));
        M.add(ENDButton);

        // 색상 설정과 기본 gui 설정
        M.setBackground(new Color(236, 251, 255));
        this.setContentPane(M);
        setSize(400, 450);
        this.setLocationRelativeTo(null);
        this.setVisible(true);
        setDefaultCloseOperation(JFrame.DISPOSE_ON_CLOSE);

        // =====
        // 동전 추가 버튼 설정

        MoneyButton_1.addActionListener(new ActionListener() {
            public void actionPerformed(ActionEvent e) {
                Money_Array.MoneyL.get(0).Moneyinsert(1);
                JOptionPane.showMessageDialog(null, "10원 1개를 추가했습니다");
            }
        });

        MoneyButton_2.addActionListener(new ActionListener() {
            public void actionPerformed(ActionEvent e) {
                Money_Array.MoneyL.get(1).Moneyinsert(1);
                JOptionPane.showMessageDialog(null, "50원 1개를 추가했습니다");
            }
        });

        MoneyButton_3.addActionListener(new ActionListener() {
            public void actionPerformed(ActionEvent e) {
                Money_Array.MoneyL.get(2).Moneyinsert(1);
                JOptionPane.showMessageDialog(null, "100원 1개를 추가했습니다");
            }
        });

        MoneyButton_4.addActionListener(new ActionListener() {
            public void actionPerformed(ActionEvent e) {
                Money_Array.MoneyL.get(3).Moneyinsert(1);
                JOptionPane.showMessageDialog(null, "500원 1개를 추가했습니다");
            }
        });

        MoneyButton_5.addActionListener(new ActionListener() {
            public void actionPerformed(ActionEvent e) {
                Money_Array.MoneyL.get(4).Moneyinsert(1);
                JOptionPane.showMessageDialog(null, "1000원 1개를 추가했습니다");
            }
        });

        // 종료 버튼 설정
        ENDButton.addActionListener(new ActionListener() {
            public void actionPerformed(ActionEvent e) {
                new Vending_Admin_Setting(ppww);
                dispose();
            }
        });
    }
}
```

5. 일정 계획

5.1 간트 차트

번호	내 용	8	9	10	11	12
1	프로젝트 설정					
2	요구사항 분석					
3	상위 설계					
4	상세 설계					
5	코딩					
6	오류 수정 및 보안					
7	최종 테스트					
8	최종보고서 작성					

6. 실행 결과



1050

반환

10원

50원

100원

500원

1000원

450원 물



500원 커피



550원 이온음료



700원 고급커피



750원 탄산음료



1000원 비한음료



관리자 로그인

600

반환

10원

50원

100원

500원

1000원

450원

물



500원

커피



550원

이온음료



700원

고급커피



750원

탄산음료



Message



물이 나갔습니다

OK

관리자 로그인

600

반환

10원

50원

100원

500원

1000원

450원

물



500원

커피



550원

이온음료



700원

고급커피



750원

탄산음료



1000원

비산음료



관리자 로그인

0

반환

10원

50원

100원

500원

1000원

450원 물



500원 커피



550원 이온음료



700원 고급커피



750원 탄산음료



Message



거스름돈이 반환되었습니다

OK

관리자 로그인

0

바환

10원

50원

100원

500원

1000원

450원 물

500원 커피

550원 이온음료

700원 고급커피

750원 탄산음료













관리자 로그인

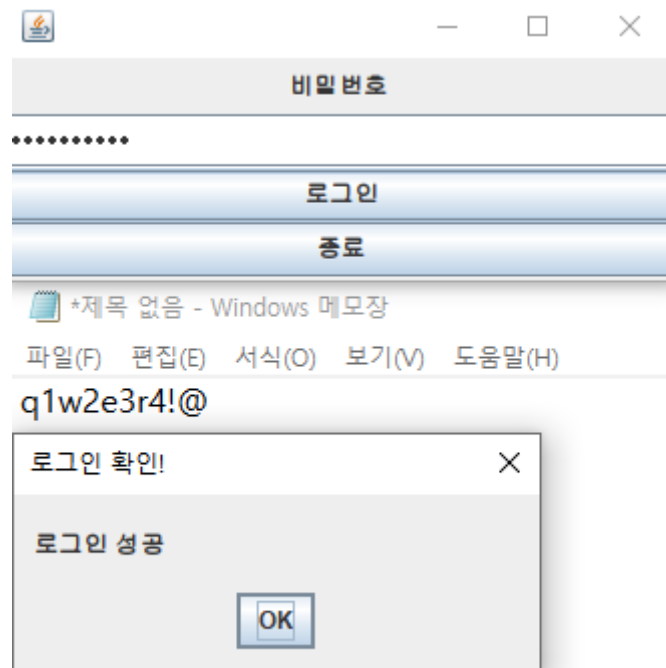
Message

i

거스름돈 1000원 : 0
거스름돈 500원 : 1
거스름돈 100원 : 1
거스름돈 50원 : 0
거스름돈 10원 : 0

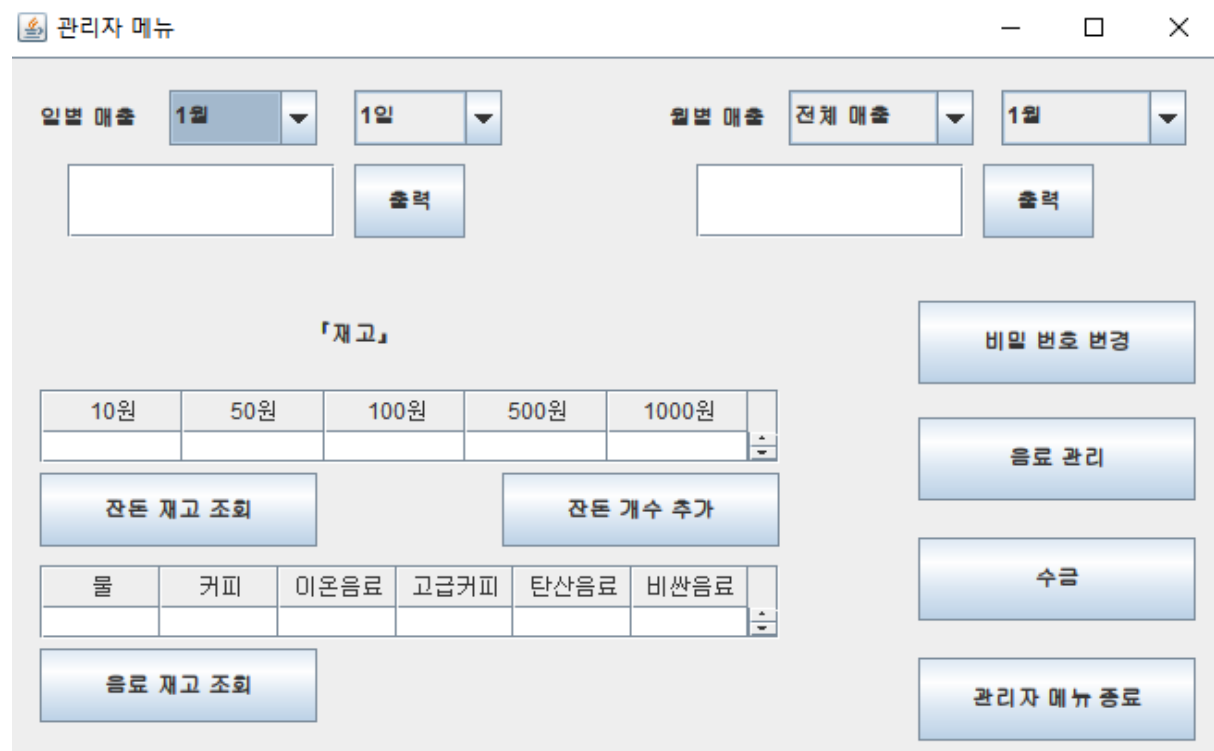
OK

로그인 부분



The screenshot shows a login window titled '비밀 번호' (Secret Number). It has a password input field with masked characters '*****', a '로그인' (Login) button, and a '종료' (End) button. Below the login window, a Windows Notepad window is open with the text 'q1w2e3r4!@'. A small dialog box titled '로그인 확인!' (Login Confirmation!) is overlaid on the Notepad, displaying '로그인 성공' (Login Success) and an 'OK' button.

관리자 메뉴



The screenshot shows an administrator menu window titled '관리자 메뉴'. It features two sections for '일별 매출' (Daily Sales) and '월별 매출' (Monthly Sales). Each section has a dropdown menu for the month (currently '1월'), a text input field, and an '출력' (Output) button. The '월별 매출' section also has a dropdown for '전체 매출' (Total Sales). In the center, there is a section titled '「재고」' (Inventory) with a table of inventory levels for various items and two buttons: '잔존 재고 조회' (Check Remaining Inventory) and '잔존 개수 추가' (Add Remaining Count). Below this is another table for '음료 재고 조회' (Check Beverage Inventory) with columns for different beverage types. On the right side, there are four buttons: '비밀 번호 변경' (Change Secret Number), '음료 관리' (Beverage Management), '수금' (Payment), and '관리자 메뉴 종료' (End Administrator Menu).

10원	50원	100원	500원	1000원

물	커피	이온음료	고급커피	탄산음료	비싼음료

일별 매출

1월

1일

월별 매출

전체 매출

1월

출력

출력

「재고」

10원	50원	100원	500원	1000원	
5개	5개	5개	5개	5개	±

잔존 재고 조회

잔존 개수 추가

물	커피	이온음료	고급커피	탄산음료	비싼음료	
3개	3개	3개	3개	3개	3개	±

음료 재고 조회

비밀 번호 변경

음료 관리

수금

관리자 메뉴 종료

음료 관리



450원

물 추가

물 가격 변경

물 이름 변경



500원

커피 추가

커피 가격 변경

커피 이름 변경



550원

이온음료 추가

이온음료 가격 변경

이온음료 이름 변경



700원

고급커피 추가

고급커피 가격 변경

고급커피 이름 변경



750원

탄산음료 추가

탄산음료 가격 변경

탄산음료 이름 변경



1000원

비산음료 추가

비산음료 가격 변경

비산음료 이름 변경

종료

비밀번호 변경

바꿀 비밀번호

변경

취소

잔돈 관리

 — □ ×



10원

10원 추가



50원

50원 추가



100원

100원 추가



500원

500원 추가



1000원

1000원 추가

종료

수금

 수금 ×

수금

10원 : 0개

50원 : 0개

100원 : 0개

500원 : 0개

1000원 : 1개

Go

7. 느낀점

어디서나 볼 수 있는 자판기를 프로그램으로 구현하면서 정말 많은 코드를 처음으로 작성해보았습니다. 거의 3000줄이 가까이 만들면서 중간중간 코드가 꼬이는 부분을 해결하기 위해 많은 시간을 들였고 또한 GUI 부분을 만들면서 너무 어렵다는 생각을 많이 했었습니다. 그래도 이번 프로젝트를 진행하면서 다른 큰 프로젝트를 진행할 때 많은 도움이 될 수 있다는 것을 느꼈습니다. 이후 방학동안은 자바를 통해 다른 기계나 혹은 동작할 수 있는 무언가를 다시 한번 만들어 보고 싶다는 생각을 했습니다