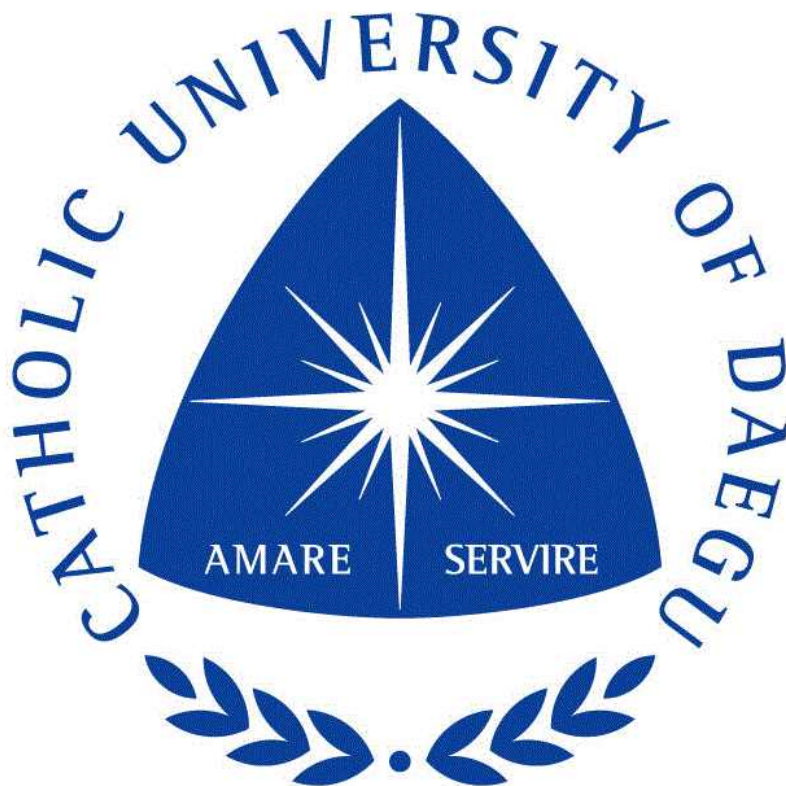




JAVA를 이용한 커피 자판기 시뮬레이터 프로그램의 구현

JAVA 프로그래밍(01)



학과	컴퓨터공학과
학번	20116742
이름	권양하
담당 교수	김미혜 교수님
제출일	2021. 06. 12

목 차

1. 서론	3
2. 관련 연구	3
3. 프로그램 설계	4
3.1. 프로그램 수행 계획	
3.2. 프로그램 기획	
3.3. 프로그램 구성	
4. 프로그램 구현	5
4.1. 클래스 선언 및 main() 메소드 선언	
4.2. 자바의 GUI	
4.3. 소스코드 설명	
5. 프로그램의 동작과 결과	8
6. 결과 분석 및 토의	9
6.1. 커피 선택	
6.2. 커피 추출 시 문구	
6.3. 선택한 커피 재료의 재고	
6.4. 커피 재료 재고 부족	
6.5 프로그램 종료	
7. 결론	11
7.1. 보고서의 결론	
7.2. 프로젝트를 하면서 느낀 점	
7.3. 향후 자신의 프로그래밍에 대한 짧은 소견	
8. 참고문헌	12
9. 별 첨(소스 코드와 주석)	12

1. 서론

커피자판기 프로그램을 JAVA 언어로 구현하여 커피자판기에서 커피가 나올 때 사용되는 재료의 양을 표시하는 실행 결과와 과정을 다루고 있다.

1. 관련 연구
2. 프로그램 설계
3. 프로그램 구현
4. 프로그램 동작과 결과
5. 결과 분석 및 토의
6. 결론
7. 참고 문헌
8. 별첨(소스코드와 주석)

2. 관련 연구

[4장, 9장] 커피자판기 프로그램에서 JFrame 클래스를 상속받아 CoffeeMachine 클래스를 작성해 코드를 재사용해, 코드 작성에 드는 시간과 비용을 줄이며 CoffeeMachine의 객체를 생성하면 스윙 프레임이 탄생하게 된다. 다음의 커피자판기 프로그램에서는 스윙 프레임을 800*700으로 설정하였다. [9장] 커피자판기의 재료를 표시하기 위해 BorderLayout 배치관리자를 이용해 재료 양의 정도를 일정한 간격으로 표시해 두었다. [6장] 커피를 선택해 원하는 커피를 추출할 때 사용되는 재료를 문자열 String[]을 사용해 Cup, Coffee, Water, Sugar, Cream을 표시하였다. [10장] ActionListener 인터페이스를 상속받은 MyActionListener 클래스를 선언하고, EventListener를 이용해 사용자가 원하는 커피를 선택하였을 때 호출되는 메소드를 나타낸다. 클릭된 버튼을 알아내기 위한 코드를 추가하여 사용자가 어떤 커피를 클릭했는지 확인할 수 있다. [11장] 콘솔창에 다음과 같은 재료들이 표시되며 이와 동시에 커피를 추출하였을 때 커피가 나오면서 커피 이미지를 표시하는 ImageIcon을 이용하여 콘솔창에 커피 이미지가 표시될 수 있도록 하였다. 그리고 이미지 레이블 컴포넌트를 JLabel에 출력한다. [14장] 마지막으로 MessageDialog를 사용해 사용자가 주문한 커피가 나오거나 커피를 만들 때 필요한 재료가 부족한 경우, 해당 Dialog가 출력되도록 하여 사용자에게 해당하는 문자열 메시지를 알려줄 수 있다. 이외에도 커피자판기 프로그램에는 다양한 관련 연구가 있다.

3. 프로그램 설계

3.1 프로그램 수행 계획

커피자판기 프로그램을 수행하기 위해 해당 소스 코드에 필요한 함수나 변수 등을 공부하며 JAVA 프로그래밍 프로그램 Eclipse IDE for JAVA Developer-2020-12를 이용해 소스 코드를 입력하고 실행해 볼 것이다.

3.2 프로그램 기획

커피자판기 프로그램에서 코드의 가장 기본적이면서도 중요한 Class를 생성하고 그 안에 커피자판기에 필요한 변수, 상수, 함수(메소드) 등 모든 프로그램 요소를 작성한다. 이외에도 커피자판기 프로그램에 필요한 JAVA 프로그래밍의 개념들을 이용해 위의 프로그램을 기획하였다. 그 개념들에 대해서 아래에 자세히 설명하였다.

3.2 프로그램 구성

커피자판기 프로그램의 구성은 간단히 사용자가 Americano, VanillaLatte, CreamLatte 중에서 선택해 원하는 커피를 추출할 수 있도록 하였으며 이때, 커피마다 필요로 하는 Cup, Coffee, Water, Sugar, Cream 재료의 양을 콘솔창에 표시되도록 하여 커피를 추출할 때마다 해당하는 재료가 줄어들도록 나타내었다. 모든 재료를 소진했을 경우, 다시 재료를 채워 넣기 위해 Reset 버튼을 추가하여 클릭할 경우 모든 재료가 가득 채워지도록 구성하였다.

4. 프로그램 구현

4.1 클래스 선언 및 main() 메소드 선언

1) public class CoffeeMachine

- 지정한 이름 CoffeeMachine을 클래스를 선언한다. 모든 클래스는 '{'으로 시작하여 '}'으로 끝난다.

2) public static void main(String[] args)

- main() 메소드를 선언하는 코드이다. 자바에서는 함수를 메소드라 부르며, 메소드는 '{'으로 시작하여 '}'으로 끝난다. 자바 클래스는 여러 개의 메소드를 포함할 수 있으며 프로그램 실행은 반드시 main() 메소드부터 시작한다.

3) int maxBarSize / full / 재료=0

- 지역변수를 이용해 커피자판기 프로그램에서 maxBarSize, full, 재료를 선언하고, 재료는 0으로 초기화한다.

4.2 자바의 GUI

1) 스윙(Swing)

- 커피자판기 프로그램에서는 JButton, JForm, JProgressBar, JDialog, JLabel, JPanel 등의 컴포넌트를 사용한다.

2) BorderLayout 배치관리자

- 커피자판기 프로그램에서 BorderLayout 배치관리자를 이용하여 SOUTH, CENTER 영역을 이용하여 콘솔창에 아래와 같이 나타나도록 지정하였다.

3) 이벤트 처리

- 버튼 클릭 시 호출되는 메소드 public void actionPerformed(ActionEvent e)와 클릭한 버튼을 알아낼 수 있는 JButton 컴포넌트를 사용하여 이벤트 클래스를 작성한다.

4.3 소스코드 설명

- 1) 커피자판기 프로그램에서 다양한 컴포넌트를 여러 군데에 클래스로 사용되기 때문에 패키지 이름이 중복되면 불편하다. 따라서 import 명령을 사용하면 클래스가 속한 패키지 명을 생략할 수 있다.

```
import java.awt.BorderLayout;
import java.awt.Color;
import java.awt.Container;
import java.awt.Graphics;
import java.awt.event.ActionEvent;
import java.awt.event.ActionListener;

import javax.swing.BorderFactory;
import javax.swing.ImageIcon;
import javax.swing.JButton;
import javax.swing.JFrame;
import javax.swing.JLabel;
import javax.swing.JOptionPane;
import javax.swing.JPanel;
import javax.swing.SwingConstants;
import javax.swing.border.Border;
```

- 2) JFrame을 상속받아 CoffeeMachine 클래스를 만들어 커피자판기 프로그램의 프레임을 생성시킨다. JLabel, JButton, JPanel 등의 컴포넌트를 이용해 커피자판기 프로그램이 원활하게 진행될 수 있도록 소스 코드를 작성하였다.

```
public class CoffeeMachine extends JFrame {
    Container contentPane;
    JLabel title = new JLabel("Welcome, YangHa CoffeeShop^____^", JLabel.CENTER);
    String [] menu_text = {"Americano", "VanillaLatte", "CreamLatte", "Reset"};
    JButton[] btn = new JButton[4];
    JPanel top_panel = new JPanel();
```

3) ImageIcon를 이용해 커피자판기 프로그램에서 사용자가 원하는 커피를 추출 하였을 때, “커피 나왔습니다.”라는 문구와 동시에 커피모양 아이콘이 등장하도록 설정하였다. 이때, 커피 아이콘의 저장 위치를 문자열에 작성하여 콘솔창에 설정 한 커피 아이콘이 등장할 수 있도록 한다.

```
ImageIcon image = new ImageIcon("C:\\Users\\pc\\OneDrive\\바탕 화면\\images\\coffee.png");
JLabel coffee_image = new JLabel(image);
```

4) 커피자판기 프로그램의 콘솔창에 menu와 coffee의 위치를 설정하기 위해 BorderLayout 배치관리자를 이용해 menu는 남(SOUTH), coffee는 중앙 (CENTER)로 코드를 작성하였다.

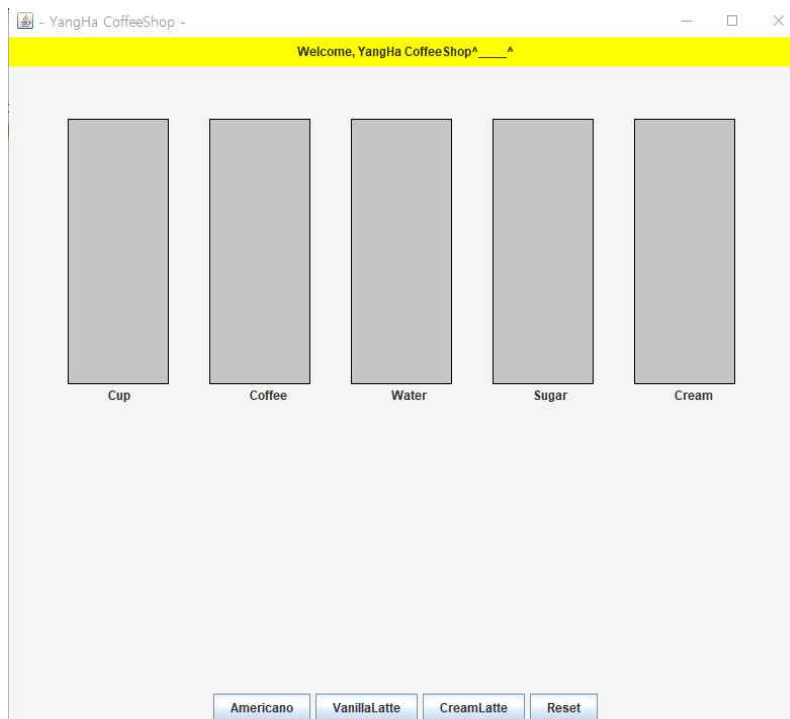
```
contentPane.add(menu, BorderLayout.SOUTH);
contentPane.add(coffee, BorderLayout.CENTER);
```

5) ActionListener 인터페이스를 상속받은 MyActionListener 클래스를 선언하 고, 추상 메소드 actionPerformed(ActionEvent e)를 작성한다. 또한 클릭된 버 튼을 알아내기 위해 e.getSource()를 이용해 알아낼 수 있다..

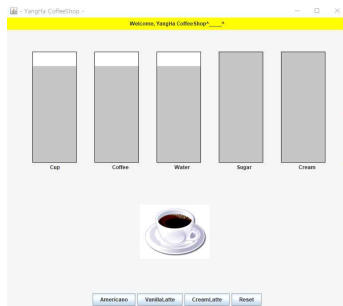
```
class MyActionListener implements ActionListener{
    public void actionPerformed(ActionEvent e) {
        JButton jb = (JButton)e.getSource();
```

5. 프로그램 동작과 결과

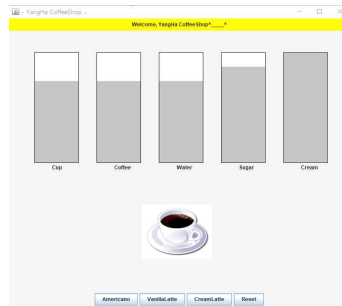
1) 아래의 그림과 같이 커피자판기 프로그램 시작화면



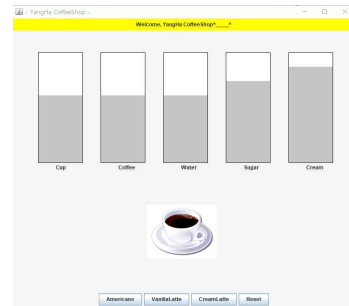
2) 커피자판기 프로그램에서 사용자가 원하는 커피 버튼(Americano, VanillaLatte, CreamLatte)을 클릭할 경우, 선택한 커피가 추출되며 추출 시 필요한 커피 재료가 줄어들고, 동시에 커피 아이콘이 나타나면서 커피가 추출되었다는 사실 확인



Americano 버튼 클릭

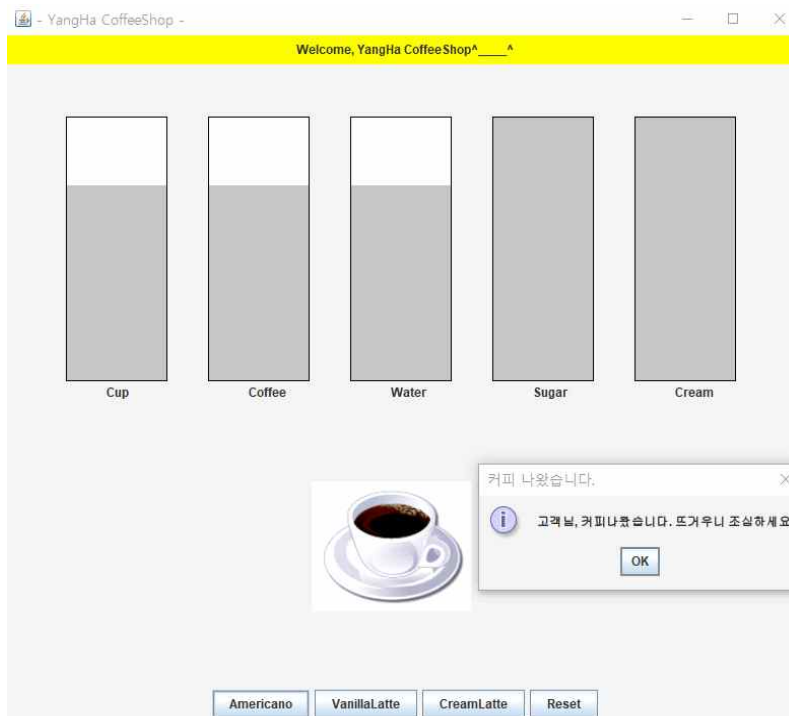


VanillaLatte 버튼 클릭

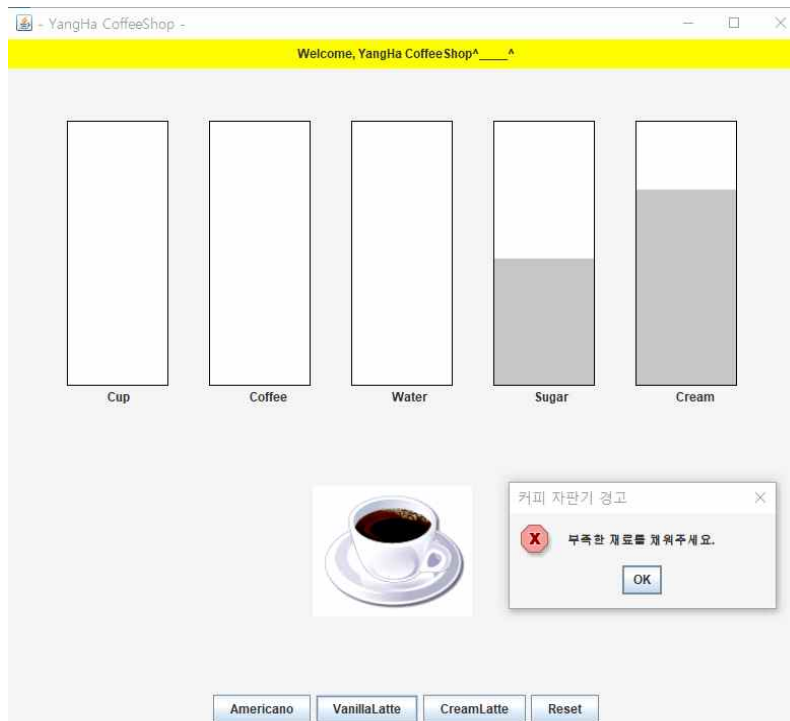


CreamLatte 버튼 클릭

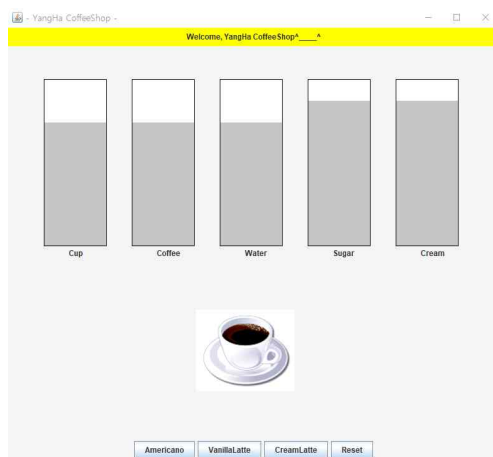
3) 사용자가 원하는 커피 버튼 클릭 시 새 창에 “고객님 커피 나왔습니다. 뜨거우니 조심하세요.”라는 문구가 뜨며 커피 추출 완료



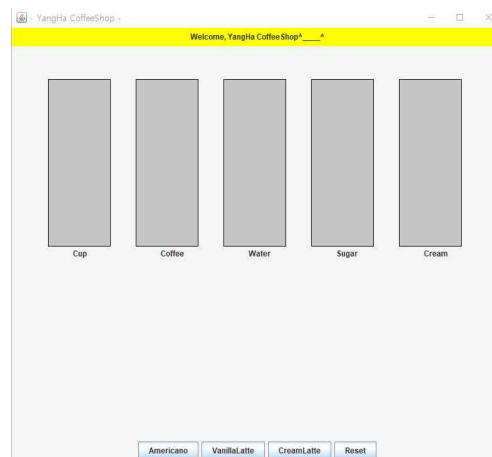
4) 커피 자판기 프로그램에 커피 추출 시 필요한 재료가 소진되었을 경우, 사용자가 커피를 추출하려고 할 때, 새 창에 경고 문자 생성



5) 커피 자판기에 커피 재료를 충전시키고 싶을 시 Reset 버튼을 클릭하게 되면 모든 커피 재료 충전되어 처음 화면과 동일



커피 추출 후 화면



Reset 버튼 클릭 후 화면

6) Close 버튼 클릭할 경우, 커피 자판기 프로그램이 종료



6. 결과 분석 및 토의

6.1 커피 선택

String [] 문자열을 이용해 커피 메뉴를 차례대로 입력하고 커피의 재고가 부족할 경우, 클릭하는 Reset 버튼도 함께 입력한다. 커피 메뉴와 Reset을 버튼, 프로그램을 종료할 Close 버튼을 설정하기 위해 String에 입력해둔 4개의 문자를 JButton (스윙 컴포넌트)을 이용해 버튼이 출력되도록 한다. 다음 버튼을 클릭하였을 때 사용자가 원하는 커피를 추출됨과 동시에 택한 커피에 필요한 재료의 재고가 줄어들도록 한다.

6.2 커피 추출 시 문구

사용자가 선택한 커피가 추출될 경우, 택한 커피에 필요한 재료의 재고가 줄어든다. 동시에 “고객님 커피나왔습니다. 뜨거우니 조심하세요”라는 문구가 새로운 콘솔창에 출력하기 위해 showMessageDialog() 메소드를 이용해 사용자에게 문자열 메시지를 출력한다. 또한 문구와 함께 커피 이미지를 출력하기 위해 ImageIcon을 이용해 커피 이미지도 함께 출력한다.

6.3 선택한 커피 재료의 재고

사용자가 선택한 커피를 추출하게 되면 택한 커피에 필요한 재료가 사용되면서 커피 자판기에는 재고가 줄어들게 된다. 이때 줄어드는 재고를 표시하기 위해 각 커피의 버튼을 숫자(0, 1, 2)로 설정해 재고가 줄어들도록 작성하였다.

6.4 커피 재고 부족

사용자가 선택한 커피 재료의 재고가 부족하면 `showMessageDialog()` 메소드를 이용해 새로운 콘솔창에 “부족한 재료를 채워주세요.”라는 문구가 출력되며 커피 자판기에 경고 메시지가 발생한다. 부족한 재고를 채우기 위해 커피 메뉴와 함께 Reset 버튼을 만들어 버튼을 클릭하면 커피 자판기 프로그램의 첫 화면과 같이 모든 재고가 채워져 출력되도록 하였다.

6.5 프로그램 종료

커피 자판기 프로그램을 종료하고 싶을 경우, 하단에 커피 메뉴와 Reset 버튼과 같이 있는 Close 버튼을 클릭하면 프로그램이 종료된다.

7. 결론

7.1 보고서의 결론

JAVA언어로 구현한 커피 자판기 프로그램을 통해 커피 선택, 커피 재고 감소, 커피 추출 문구, 재고 부족 문구, 커피 재고 추가, 커피 자판기 종료 할 수 있는 소스 코드를 구현하여 실행할 수 있었다.

7.2 프로젝트를 하면서 느낀 점

어떤 프로젝트를 만들지 고민하던 중에 처음에는 GUI 응용프로그램을 사용하지 않고 원하는 JAVA 텀 프로젝트를 완성하려 하였다. 그런데 자바 책을 살펴보니, 책의 절반이 GUI 응용프로그램을 이용하여서 나도 높은 완성도를 위해 다음의 응용프로그램을 사용하여 작성하게 되었다. 프로젝트를 하면서 ‘나만의 앱 만들기’의 심화 단계라는 생각이 많이 들었다. 앱은 소스 코드도 간단하다 생각하고 이를 핸드폰에 실행시키는 반면 JAVA를 이용해 하나의 프로그램을 만들어 실행하는 것은 심화된 소스 코드와 다양한 옵션을 추가할 수 있다고 생각하였기 때문이다. 개인적으로 이론 수업도 이해하는 데 있어 많은 도움이 되지만 나는 직접 코드도 작성해보고 코드를 이해하였을 때 비로소 JAVA 프로그래밍과 친해졌다고 할 수 있다는 것을 깨닫게 되었다. 그리고 프로젝트가 성공적으로 완성되었을 때 느끼는 희열감을 알게 되었기 때문에 뭔가의 문제가 발생하였을 때 이를 해결해서 빨리 희열과 성취감을 느끼고 싶다는 욕심이 더 생기게 되었다. 앞으로도 어떠한 프로그래밍에 대해 이해하고 공부하고 싶다면 이론을 읽어보고 이론보다 직접 코드를 실행시켜 보고 문제점도 해결해보고 싶다. 또한 JAVA 프로그래밍 소스 코드를 공부하면서 어떻게 좀 더 쉽게 사용할 수 있는지에 대해 공부해 볼 것이다. 앞으로도 대학교를 진학하면서 다양한 프로그램을 제작해보고 싶다는 욕심이 생겼다..

7.3 향후 자신의 프로그래밍에 대한 짧은 소견

아직 나는 처음부터 끝까지 스스로 소스 코드를 작성하기에는 힘든 실력을 가지고 있다. 그렇기에 2021년 남은 시간 동안 JAVA 언어에 대해 더 공부해 언젠가 누군가 작성해둔 소스 코드를 참고하지 않고도 원하는 프로그램을 만들어 볼 수 있었으면 한다. 그래서 지금부터 매일 꾸준히 공부해 남에게 알려줄 수 있는 실력으로 향상되길 희망한다.

8. 참고문헌

[1] 황기태, 김효수 “명품 JAVA Programming”, 생능출판사

[2] 명품 JAVA Programming p794쪽 참고

[3] JAVA 언어 프로그램 종료 소스

<https://coding-factory.tistory.com/526>

[4] 커피 자판기 프로그램 소스 코드 예시

<http://blog.naver.com/PostView.nhn?blogId=roundboy1&logNo=220429880009&redirect=Dlog&widgetTypeCall=true>

9. 별첨(소스 코드와 주석)

/* 다음과 같은 색으로 내가 직접 수정, 보완한 부분을 명시해두었다.*/

```
import java.awt.BorderLayout;
import java.awt.Color;
import java.awt.Container;
import java.awt.Graphics;
import java.awt.event.ActionEvent;
import java.awt.event.ActionListener;
```

```
import javax.swing.BorderFactory;
import javax.swing.ImageIcon;
import javax.swing.JButton;
import javax.swing.JFrame;
import javax.swing.JLabel;
import javax.swing.JOptionPane;
import javax.swing.JPanel;
import javax.swing.SwingConstants;
import javax.swing.border.Border;
class MyLabel extends JLabel {
    int maxBarSize;
    int full;
    int 재료;
```

```
MyLabel(int maxBarSize){
    this.maxBarSize = maxBarSize; //초기화
    재료=0;
```

```

    }
    @Override
    protected void paintComponent(Graphics g) {
        // TODO Auto-generated method stub
        super.paintComponent(g);
        g.setColor(Color.white);
        int height = (int)((double)(this.getHeight())/maxBarSize*full);

        g.fillRect(0, 0, this.getWidth(), height);
        System.out.println(height);
        System.out.println(this.getHeight());
    }

}

public class CoffeeMachine extends JFrame { //JFrame을 상속받은 CoffeeMachine을 작성
    Container contentPane;
    JLabel title = new JLabel("Welcome, YangHa CoffeeShop^____^", JLabel.CENTER);
    String [] menu_text = {"Americano","VanillaLatte","CreamLatte", "Reset","Close"};
    JButton[] btn = new JButton[5];

    JPanel top_panel = new JPanel();
    JPanel menu = new JPanel();
    JPanel coffee = new JPanel();
    MyLabel[] bar = new MyLabel[5];
    JLabel[] 재료 = new JLabel[5]; //각 스윙 컴포넌트 생성

    String [] bar_name = {"Cup","Coffee","Water","Sugar", "Cream"};
    ImageIcon image = new ImageIcon("C:\\Users\\pc\\OneDrive\\바탕 화면\\images\\coffee.png");
    JLabel coffee_image = new JLabel(image);
    public CoffeeMachine() {
        // TODO Auto-generated constructor stub

        setTitle("- YangHa CoffeeShop -");
        setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE); //프레임이 닫힐 때 프로그램도 함께 종료.
        contentPane = getContentPane();
        contentPane.setLayout(new BorderLayout());

        top_panel.setBackground(Color.YELLOW);
        top_panel.add(title);
        contentPane.add(top_panel, BorderLayout.NORTH);
        for(int i=0;i<btn.length;i++){
            btn[i] = new JButton(menu_text[i]);
            btn[i].setSize(100,50);
            btn[i].addActionListener(new MyActionListener());
            menu.add(btn[i]);
        }
        /*센터 커피 버튼 그리기 */
        int width=58;
        Border border = BorderFactory.createLineBorder(Color.black, 1);

```

```

for(int i=0;i<bar.length;i++){

    bar[i] = new MyLabel(100);
    bar[i].setBorder(border);
    bar[i].setBackground(Color.LIGHT_GRAY);
    bar[i].setOpaque(true);
    bar[i].setLocation(width,50);
    재료[i] = new JLabel(bar_name[i]);
    재료[i].setSize(100,70);
    재료[i].setLocation(width+40, 280);
    bar[i].setSize(100,254);
    width+=140;
    coffee.add(bar[i]);
    coffee.add(재료[i]);
}

coffee_image.setLocation(300, 400);
coffee_image.setSize(157, 125); //이미지 폭157, 높이 125 크기로 설정
coffee.add(coffee_image);
coffee_image.setVisible(false); //false의 경우 프레임이 숨겨짐.

coffee.setLayout(null);
contentPane.add(menu, BorderLayout.SOUTH);
contentPane.add(coffee, BorderLayout.CENTER);
//커피 뽑으면 커피 출력

setVisible(true);
setSize(800, 700);
}

class MyActionListener implements ActionListener{

    void message() {
        JOptionPane.showMessageDialog(null, "고객님. 커피 나왔습니다. 뜨거운니 조심하세요","커피 나왔습니다.",
JOptionPane.INFORMATION_MESSAGE);
    }
    @Override
    public void actionPerformed(ActionEvent e) {
        // TODO Auto-generated method stub
        JButton jb = (JButton)e.getSource();
        int message=0;
        //자판기 경고문구 때문에 리셋 버튼을 상단으로 올려서 자판기 경고를 피한다.
        if(jb.equals(btn[3])){
            bar[0].full = 0; bar[0].재료 = 0;
            bar[1].full = 0; bar[0].재료 = 0;
            bar[2].full = 0; bar[0].재료 = 0;
            bar[3].full = 0; bar[0].재료 = 0;
            bar[4].full = 0; bar[0].재료 = 0;
            coffee_image.setVisible(false);
            repaint();
            System.out.println("리셋");
        }
    }
}

```

```

    }

    for(int i=0;i<bar.length;i++){
        if(bar[i].재료 == 8) {
            JOptionPane.showMessageDialog(null, "부족한 재료를 채워주세요.", "커피 자판기 경고",
JOptionPane.ERROR_MESSAGE);
            coffee_image.setVisible(false);
            message=1;
        }
        break;
    }
    if(message==0){

        if(jb.equals(btn[0])){

            bar[0].full += 13; bar[0].재료++;
            bar[1].full += 13; bar[1].재료++;
            bar[2].full += 13; bar[2].재료++;
            repaint();
            System.out.println("Americano 입니다.");
            coffee_image.setVisible(true);
            message();
        }
        else if(jb.equals(btn[1])){
            bar[0].full += 13; bar[0].재료++;
            bar[1].full += 13; bar[1].재료++;
            bar[2].full += 13; bar[2].재료++;
            bar[3].full += 13; bar[3].재료++;
            repaint();
            System.out.println("VanillaLatte 입니다.");
            coffee_image.setVisible(true);
            message();
        }
        else if(jb.equals(btn[2])){
            bar[0].full += 13; bar[0].재료++;
            bar[1].full += 13; bar[1].재료++;
            bar[2].full += 13; bar[2].재료++;
            bar[3].full += 13; bar[3].재료++;
            bar[4].full += 13; bar[4].재료++;
            repaint();
            System.out.println("CreamLatte 입니다.");
            coffee_image.setVisible(true);
            message();
        }
        else if(jb.equals(btn[4])) {
            System.exit(0); //자바 프로그램 종료.
        }
    }
}

```

```
}  
}  
public static void main(String[] args) {  
  
    // TODO Auto-generated method stub  
    new CoffeeMachine();  
}  
}
```