12181818 이주훈 실습과제5 – 계산기 프로그램

**주안점**

먼저 요구사항에 따른 기능을 레이아웃으로 구현하고 이후 각 기능을 할당했습니다. 기능을 구현할 때는 계산기를 만들면서 생기는 몇 가지 Exception을 처리하면서 구현하였습니다. 사칙연산을 제외한 1/x, x제곱, 루트x연산은 숫자 후 연산을 누르면 바로 출력되도록 하였고 사칙연산은 숫자->연산기호->숫자->’=’을 누름으로써 결과 값이 나오도록 하였습니다.

또한 5x4의 그리드 레이아웃을 사용하였고 숫자는 노란색, 연산은 하늘색, =연산만 빨간색으로 구분하여 인터페이스를 만들었습니다. 연산에서는 소수점 값이 나올 확률이 높음으로 결과는 double형으로 출력했습니다.

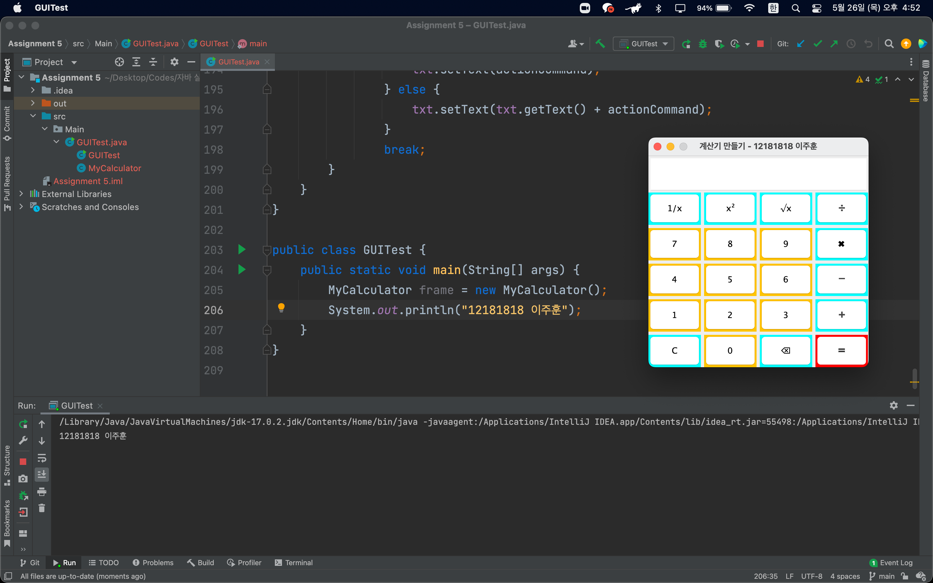
**소스코드**

package Main;  
  
import javax.swing.\*;  
import java.awt.event.\*;  
import java.awt.\*;  
import java.util.Objects;  
  
class MyCalculator extends JFrame implements ActionListener {  
 private JTextField txt;  
 private JPanel panel;  
 double arg;  
 String operator="";  
  
 public MyCalculator() {  
 setTitle("계산기 만들기 - 12181818 이주훈");  
 setDefaultCloseOperation(*EXIT\_ON\_CLOSE*);  
 txt = new JTextField(10);  
 txt.setFont(new Font("serif", Font.*BOLD*, 40));  
 txt.setEditable(false);   
 txt.setHorizontalAlignment(JTextField.*RIGHT*); // 오른쪽 부터 정렬되도록  
 add(txt, BorderLayout.*NORTH*);  
  
 panel = new JPanel();  
 panel.setLayout(new GridLayout(5, 4,5,5));  
 add(panel, BorderLayout.*CENTER*);  
  
 String[] syntaxs = new String[] {"1/x", "x²","√x","➗"};  
 for (String s : syntaxs) {  
 JButton btn = new JButton(s);  
 btn.addActionListener(this);  
 btn.setPreferredSize(new Dimension(60, 50));  
 btn.setBackground(Color.*cyan*);  
 btn.setOpaque(true);  
 panel.add(btn);  
 }  
  
 for (int i = 7; i <= 9; i++) {  
 JButton btn = new JButton("" + i);  
 btn.addActionListener(this);  
 btn.setPreferredSize(new Dimension(60, 50));  
 btn.setBackground(Color.*ORANGE*);  
 btn.setOpaque(true);  
 panel.add(btn);  
 }  
 {  
 JButton btn = new JButton("✖");  
 btn.addActionListener(this);  
 btn.setPreferredSize(new Dimension(60, 50));  
 btn.setBackground(Color.*cyan*);  
 btn.setOpaque(true);  
 panel.add(btn);  
 }  
 for (int i = 4; i <= 6; i++) {  
 JButton btn = new JButton("" + i);  
 btn.addActionListener(this);  
 btn.setPreferredSize(new Dimension(60, 50));  
 btn.setBackground(Color.*ORANGE*);  
 btn.setOpaque(true);  
 panel.add(btn);  
 }  
 {  
 JButton btn = new JButton("➖");  
 btn.addActionListener(this);  
 btn.setPreferredSize(new Dimension(60, 50));  
 btn.setBackground(Color.*cyan*);  
 btn.setOpaque(true);  
 panel.add(btn);  
 }  
 for (int i = 1; i <= 3; i++) {  
 JButton btn = new JButton("" + i);  
 btn.addActionListener(this);  
 btn.setPreferredSize(new Dimension(60, 50));  
 btn.setBackground(Color.*ORANGE*);  
 btn.setOpaque(true);  
 panel.add(btn);  
 }  
 {  
 JButton btn = new JButton("➕");  
 btn.addActionListener(this);  
 btn.setPreferredSize(new Dimension(60, 50));  
 btn.setBackground(Color.*cyan*);  
 btn.setOpaque(true);  
 panel.add(btn);  
 }  
 JButton clearButton = new JButton("C");  
 clearButton.addActionListener(this);  
 clearButton.setPreferredSize(new Dimension(60, 50));  
 clearButton.setBackground(Color.*cyan*);  
 clearButton.setOpaque(true);  
 panel.add(clearButton);  
  
 JButton btnZero = new JButton("" + 0);  
 btnZero.addActionListener(this);  
 btnZero.setPreferredSize(new Dimension(60, 50));  
 btnZero.setBackground(Color.*ORANGE*);  
 btnZero.setOpaque(true);  
 panel.add(btnZero);  
  
 JButton deleteButton = new JButton("⌫");  
 deleteButton.addActionListener(this);  
 deleteButton.setPreferredSize(new Dimension(60, 50));  
 deleteButton.setBackground(Color.*cyan*);  
 deleteButton.setOpaque(true);  
 panel.add(deleteButton);  
 {  
 JButton btn = new JButton("🟰");  
 btn.addActionListener(this);  
 btn.setPreferredSize(new Dimension(60, 50));  
 btn.setBackground(Color.*red*);  
 btn.setOpaque(true);  
 panel.add(btn);  
 }  
 pack();  
 setResizable(false);  
 setVisible(true);  
 }  
  
 public void actionPerformed(ActionEvent e) {  
 String actionCommand = e.getActionCommand(); // 컴포넌트의 텍스트 값  
  
 switch (actionCommand) {  
 case "C":  
 txt.setText("");  
 arg = 0.0;  
 operator = "";  
 break;  
 case "⌫":  
 if (txt.getText().length() > 0) {  
 txt.setText(txt.getText().substring(0, txt.getText().length() - 1));  
 }  
 break;  
 case "1/x":  
 txt.setText(String.*format*("%15.3f", 1.0 / Integer.*parseInt*(txt.getText())));  
 break;  
 case "x²":  
 long powNumber = Integer.*parseInt*(txt.getText());  
 txt.setText(String.*valueOf*(powNumber \* powNumber));  
 break;  
 case "√x":  
 double rootNum = Double.*parseDouble*(txt.getText());  
 txt.setText(String.*format*("%15.3f", Math.*sqrt*(rootNum)));  
  
 break;  
 case "➗":  
 try {  
 arg = Double.*parseDouble*(txt.getText());  
 } catch (NumberFormatException ignored) {  
 }  
 operator = "/";  
 txt.setText("/");  
 break;  
 case "✖":  
 try {  
 arg = Double.*parseDouble*(txt.getText());  
 } catch (NumberFormatException ignored) {  
 }  
 operator = "\*";  
 txt.setText("\*");  
 break;  
 case "➖":  
 try {  
 arg = Double.*parseDouble*(txt.getText());  
 } catch (NumberFormatException ignored) {  
 }  
 operator = "-";  
 txt.setText("-");  
 break;  
 case "➕":  
 try {  
 arg = Double.*parseDouble*(txt.getText());  
 } catch (NumberFormatException ignored) {  
 }  
 operator = "+";  
 txt.setText("+");  
 break;  
 case "🟰":  
 if (Objects.*equals*(operator, "\*")) {  
 txt.setText(String.*format*("%15.3f",arg \* Double.*parseDouble*(txt.getText())));  
 operator = "";  
 } else if (Objects.*equals*(operator, "/")) {  
 txt.setText(String.*format*("%15.3f",arg / Double.*parseDouble*(txt.getText())));  
 operator = "";  
 } else if (Objects.*equals*(operator, "+")) {  
 txt.setText(String.*format*("%15.3f",arg + Double.*parseDouble*(txt.getText())));  
 operator = "";  
 } else if (Objects.*equals*(operator, "-")) {  
 txt.setText(String.*format*("%15.3f",arg - Double.*parseDouble*(txt.getText())));  
 operator = "";  
 }  
 break;  
 default:  
 String text = txt.getText();  
 if (Objects.*equals*(text, "\*") || Objects.*equals*(text, "/") || Objects.*equals*(text, "+") || Objects.*equals*(text, "-")) {  
 txt.setText(actionCommand);  
 } else {  
 txt.setText(txt.getText() + actionCommand);  
 }  
 break;  
 }  
 }  
}  
  
public class GUITest {  
 public static void main(String[] args) {  
 MyCalculator frame = new MyCalculator();

System.*out*.println("12181818 이주훈");

}  
}

**실행 결과**

 텍스트, 전자기기이(가) 표시된 사진

자동 생성된 설명텍스트, 전자기기이(가) 표시된 사진

자동 생성된 설명

* 실행화면 / 연산 예제 두개

**결론 및 느낀점**

\*참고\* 버튼에 setBackground()를 하면 강의에서는 버튼 전체의 색이 변했지만 Mac OS의 환경 때문인지 적용이 되지 않고 setOpaque()를 하면 테두리에만 적용되는 현상이 일어났습니다.먼저 레이아웃을 구성한 후에 버튼에 Action을 부여했습니다. 먼저 여러 케이스를 switch문으로 나누고 각 상황에 필요한 연산이나 값을 출력하도록 하였고 연산 기호만 두 번 연속으로 누르는 등에 대한 Exception을 처리함으로써 정상적으로 출력이 되도록 만들었습니다.(ex. 10, +, x, 10을 순서대로 누르고 =을 누르면 100이 나오도록) 생각보다 간단한 예제이면서도 여러가지를 고려해야 하는 과제였습니다.  
계산기를 완성하고 돌아보니 추후에는 코드를 함수화 하는 등 훨씬 간결하게 만들 수 있을 것 같았습니다. 다음에 구현을 하게 된다면 미리 구상을 하고 UI 코드의 중복을 줄여보도록 노력 하겠다는 생각을 했습니다. 또한 이는 다른 GUI 프로그래밍에서도 똑같이 적용할 수 있다는 생각이 들었습니다.  
마지막으로 다시 한 번 정리하며 UI를 짜는 것과 기능을 코딩하는 것의 적절한 조화가 이루어져야 좋은 프로그램이 나올 수 있다고 느꼈습니다.