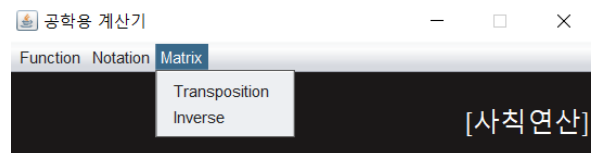


[공학용 계산기 Project]

멀티미디어공학과

2019112541 양유진

1. 프로그램 구조 및 기능 소개



(1) 프로그램 구조

이 프로그램은 자바 GUI를 이용하여 구현된 공학용 계산기이다. 사칙연산, Function(적분/미분, 삼각함수계산), Notation(진수변환), Matrix(행렬) 기능이 탑재 되어있다. 그림1의 *infoLabel*에서는 현재 무슨 기능으로 계산되는지를 보여주고, *inputTextField*는 현재 입력한 과정들을 보여준다. 버튼을 눌러 계산할 식을 입력한 후, Menu에서 기능을 선택하면 *resultLabel*에서 계산의 결과값이 나타난다.

(2) UI

공학용 계산기의 UI를 좀 더 깔끔하게 만들어 줄 수 있는 방안을 찾아보다가 Java 라이브러리에 내장되어 있는 Look And Feel 라이브러리를 알게 되어 라이브러리에 포함된 Nimbus 테마로 UI를 설정하였다.

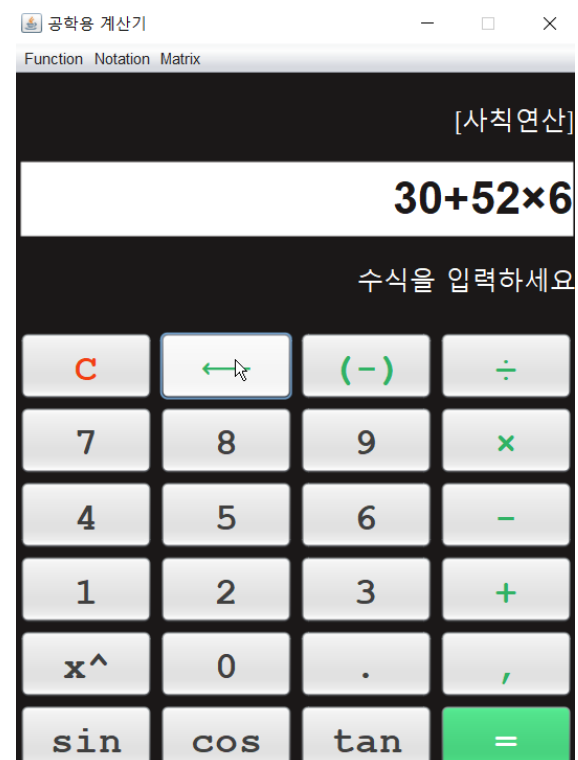
(3) 기능

① 버튼 별 기능

- C: inputTextField에 입력한 모든 값을 한 번에 지워준다.



- ←: 바로 전에 입력한 수식을 한 칸 지워준다.



- (-): 적분, 미분에서 음수인 계수와 차수를 표현할 때 사용된다.

Ex) $3x^2+(-)2x^1+3x^0$

- \div , \times , $-$, $+$: 기본 연산자

- $x^$: 적분, 미분에서 사용 Ex) $3x^2+(-)2x^1+3x^0$

- .: 소수점 표현 시 사용

- ,: 행렬 연산 시 요소들을 구별하여 입력할 때 사용

Ex) $2,15,3,24 == \begin{vmatrix} 2 & 15 \\ 3 & 24 \end{vmatrix} == \begin{vmatrix} 2 & 5 \\ 3 & 24 \end{vmatrix}$

- cos, sin, tan: 삼각함수 연산에서 삼각함수 표현 시 사용

- =: 사칙연산의 결과값을 도출할 때 사용

② 메뉴 별 기능

- 사칙연산 (default)

- 메뉴 선택을 하지 않아도 어떤 모드에서든지 사용할 수 있다.
- 계산 식 입력 후 "=" 버튼을 입력하면 결과가 도출된다.
- 연산자를 여러 개 사용하는 사칙 연산을 입력해도 우선순위를 이용하여 계산이 가능하다. Ex) $3+2\times 5-3$ (O)
- 단, 음수 계산은 할 수 없다. Ex) $-3+(-6)\div 2$ (X)

■ Function

- Integral (정적분)

- ✓ 입력된 적분식과 함께 입력한 두 값으로 정적분을 할 수 있다.

$$\text{Ex) } 3x^2 + (-)2x^1 + 3x^0, 3, 2 == \int_2^3 (3x^2 - 2x + 3) dx$$

- ✓ 차수와 계수의 개수의 제한이 없다
- ✓ 음수의 차수와 계수를 입력할 시 (-)3와 같이 (-)를 붙여 표현해 준다.
- ✓ 계산 식 입력 후 "Function-Integral" 메뉴를 클릭하면 적분식과 함께 적분 값이 나타난다.

- Differential (미분)

- ✓ 입력된 미분식과 함께 입력한 값으로 미분을 할 수 있다.

$$\text{Ex) } 3x^2 + (-)2x^1 + 3x^0, 3 == (3x^2 - 2x + 3)', f'(3)$$

- ✓ 차수와 계수의 제한이 없다.
- ✓ 음수의 차수와 계수를 입력할 시 (-)3와 같이 (-)를 붙여 표현해 준다.
- ✓ 계산 식 입력 후 "Function-Integral" 메뉴를 클릭하면 미분식과 함께 미분 값이 나타난다.

- Trigonometric (삼각함수 계산)

- ✓ 입력된 삼각함수 사칙연산 식을 계산한다.
- ✓ Degree 값을 숫자로 입력해도 자동으로 Radian으로 계산된다.

$$\text{Ex) } \sin 30 == \sin(30^\circ) == \sin\left(\frac{\pi}{6}\right)$$

- ✓ 덧셈, 뺄셈, 나눗셈, 곱셈 등 간단한 사칙연산만 가능하다.
- ✓ 계산 식 입력 후 "Function-Trigonometric" 메뉴를 클릭하면 삼각함수 계산 값이 도출된다.
- ✓ 결과 값이 소수 둘째자리까지 반올림되어 나타난다.

■ Notation (진수 변환)

- Binary (2진수 변환)

- ✓ 입력된 10진수 값을 2진수로 변환한다.
- ✓ 10진수 입력 후 "Notation-Binary" 메뉴를 클릭하면 변환 값이 도출된다.

- Hex (16진수 변환)

- ✓ 입력된 10진수 값을 16진수로 변환한다.
- ✓ 10진수 입력 후 "Notation-Hex" 메뉴를 클릭하면 변환 값이 도출된다.

■ Matrix (행렬)

- Transposition (전치 행렬)

- ✓ 음수 가능, 자릿수 제한 X
- ✓ 입력된 2X2 행렬 값을 전치 행렬로 바꿔준다.

$$\text{Ex) } 6,15,12,3 \Rightarrow \begin{vmatrix} 6 & 12 \\ 15 & 3 \end{vmatrix} == \begin{vmatrix} 6 & 12 \\ 15 & 3 \end{vmatrix}$$

- ✓ 행렬 요소 입력 후, "Matrix-Transposition" 메뉴를 클릭하면 전치 행렬이 나타난다.

- Inverse (역 행렬)

- ✓ 음수 가능, 자릿수 제한 X
- ✓ 입력된 2X2 행렬 값을 역행렬로 바꿔준다.

$$\text{Ex) } 6,15,12,3 \Rightarrow \begin{vmatrix} -0.02 & 0.09 \\ 0.07 & -0.04 \end{vmatrix} == \begin{vmatrix} -0.02 & 0.09 \\ 0.07 & -0.04 \end{vmatrix}$$

- ✓ 행렬 요소 입력 후, "Matrix-Transposition" 메뉴를 클릭하면 전치 행렬이 나타난다.

2. 기능 별 동작 분석 및 결과

(1) 사칙연산



그림 1 "=" 버튼 누를 시 결과 도출

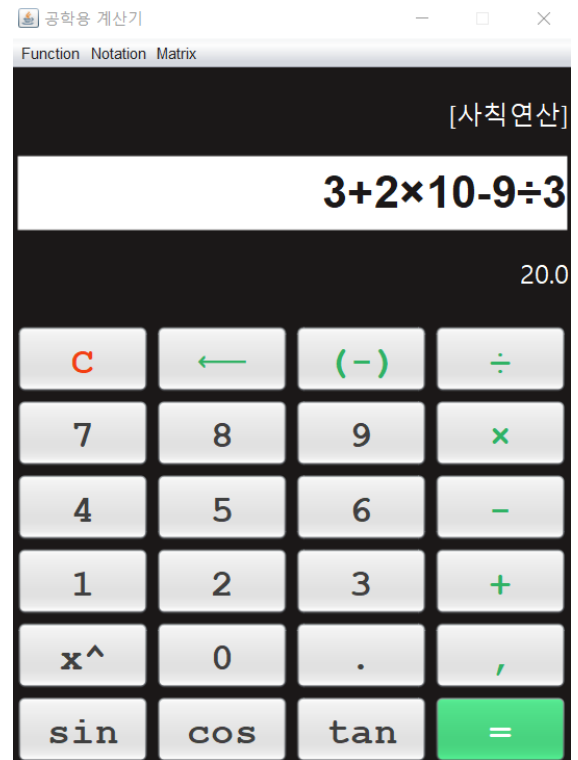


그림 2 복잡한 사칙 연산을 우선 순위를 이용하여 계산할 수 있다.



그림 3 소수점이 들어간 사칙 연산이 가능하다.

(2) 정적분

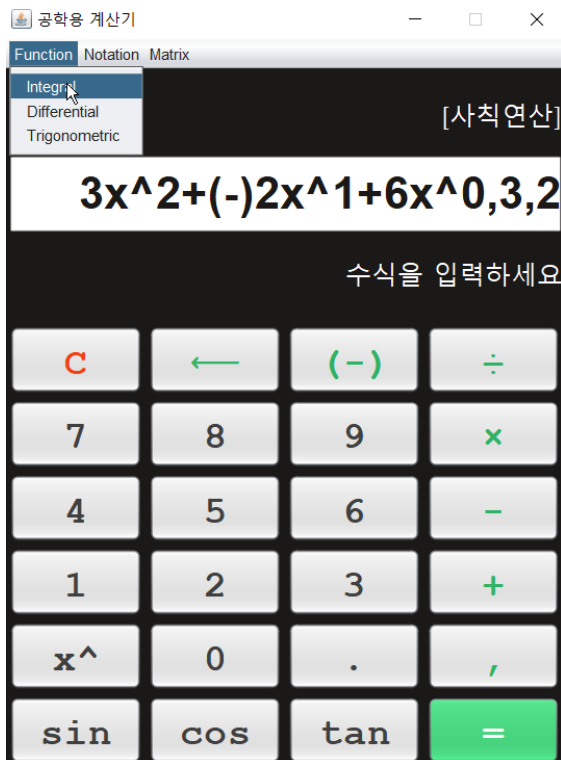


그림 1 Menu – Function – Integral 클릭시 [정적분]으로 전환 및 결과 도출

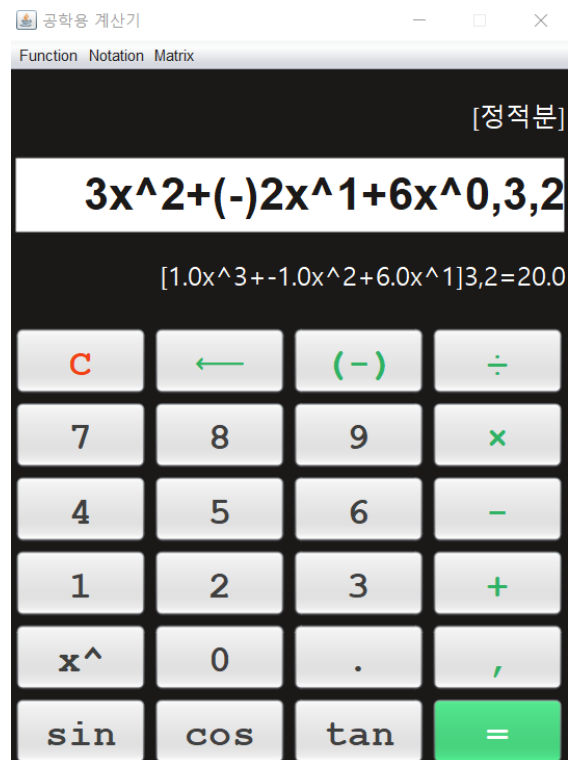


그림 2 $\int_2^3 3x^2 + (-2)x + 6dx = [x^3 - x^2 + 6x]_2^3 = 20.0$

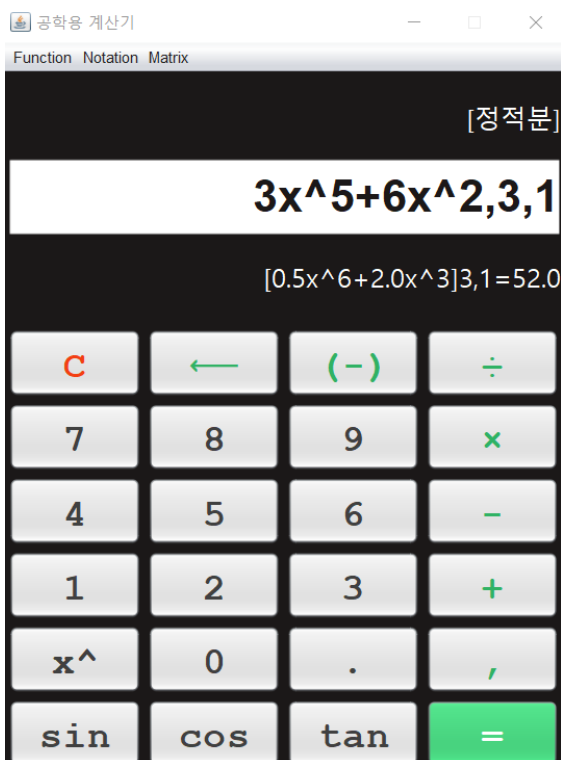


그림 3 $\int_1^3 3x^5 + 6x^2 dx = [0.5x^6 + 2x^3]_1^3 = 52.0$

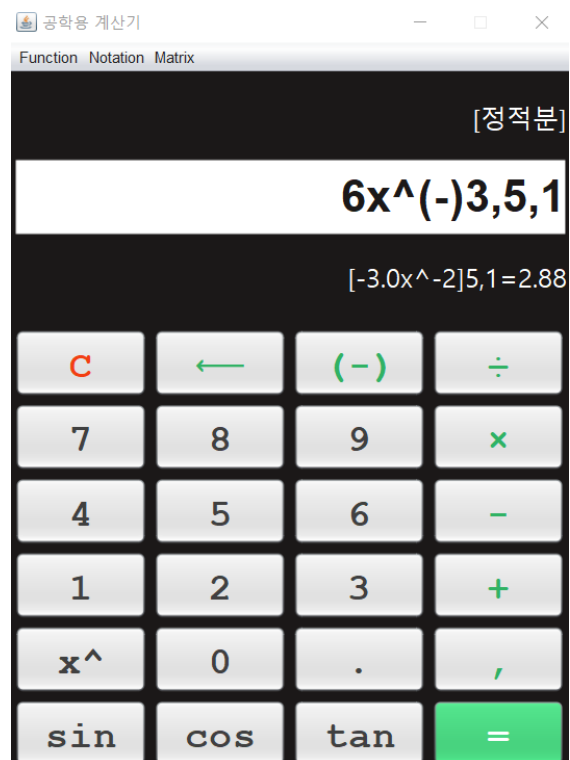


그림 4 $\int_1^5 6x^{-3} dx = [-3x^{-2}]_1^5 = 2.88$

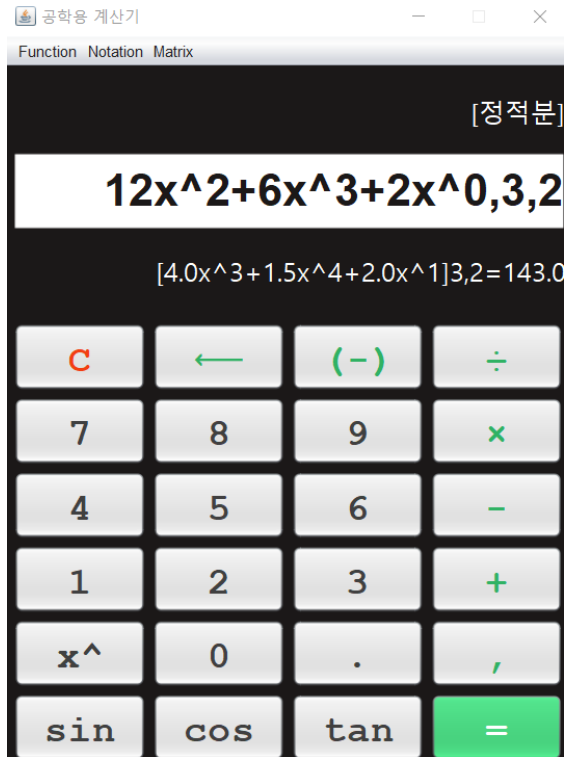


그림 5 $\int_2^3 12x^2 + 6x^3 + 2dx = [4x^3 + 1.5x^4 + 2x]_2^3 = 143.0$

(3) 미분

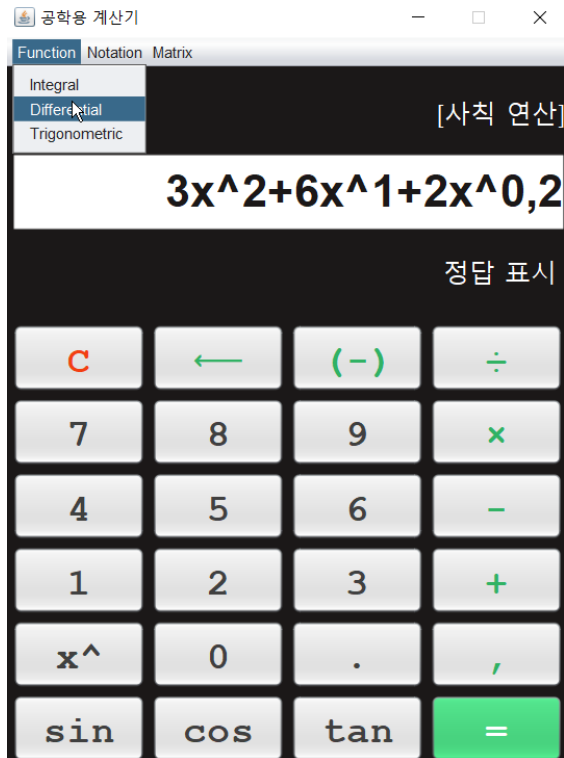


그림 1 Menu – Function – Differential 클릭 시 [미분]으로 전환 및 결과 도출

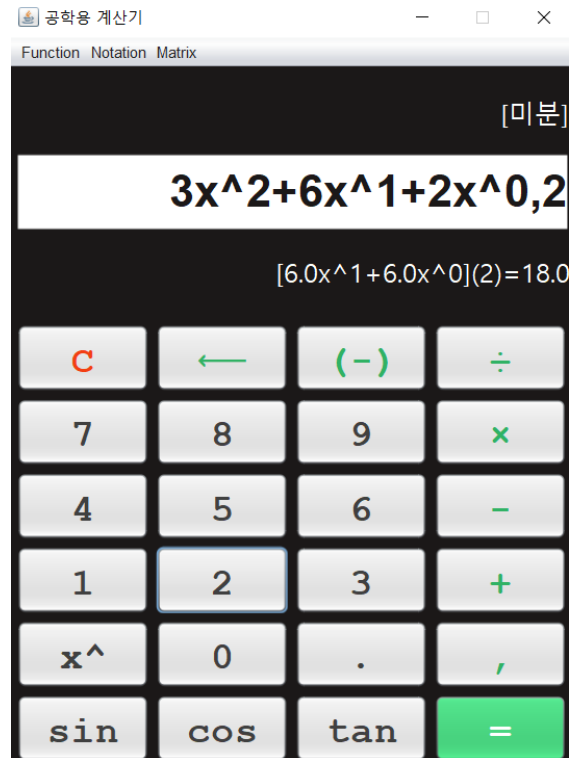


그림 2 $(3x^2 + 6x + 2)' = 6x + 6, f'(2) = 18$

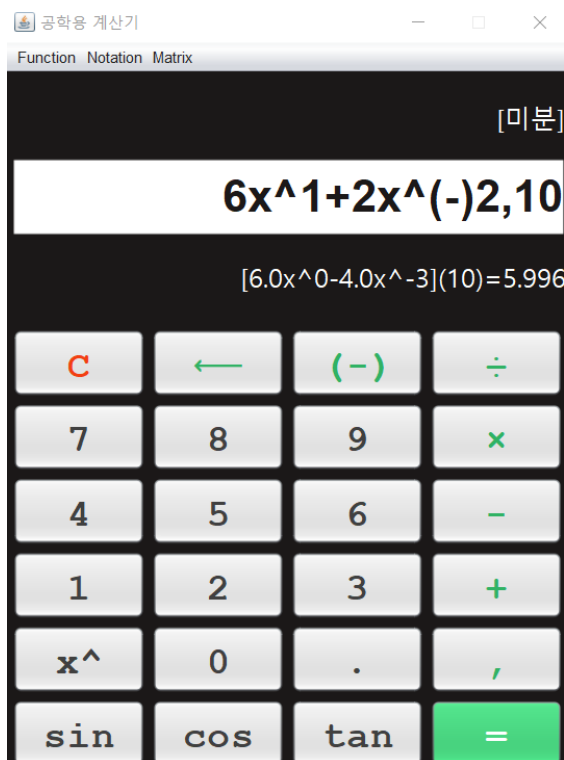


그림 3 $(6x + 6x^{-2})' = 6 - 6x^{-3}, f'(10) = 5.996$

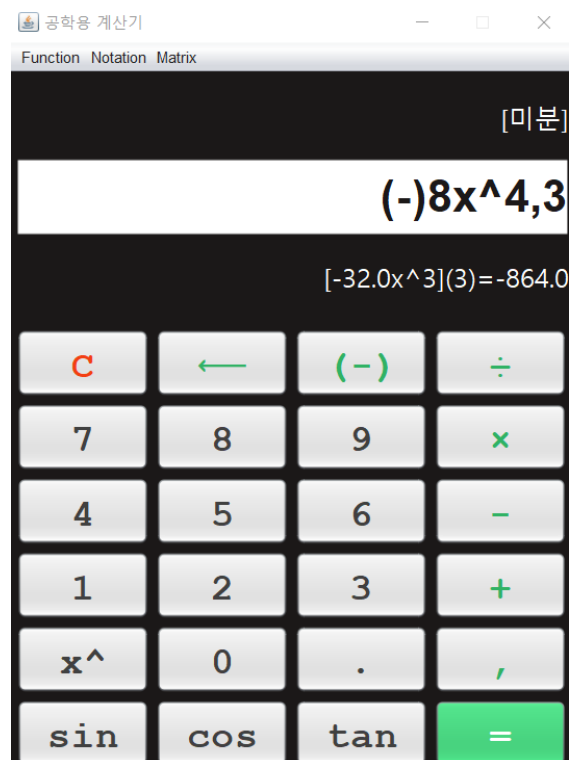


그림 4 $(-8x^4)' = -32x^3, f'(3) = -864$

(4) 삼각함수 계산

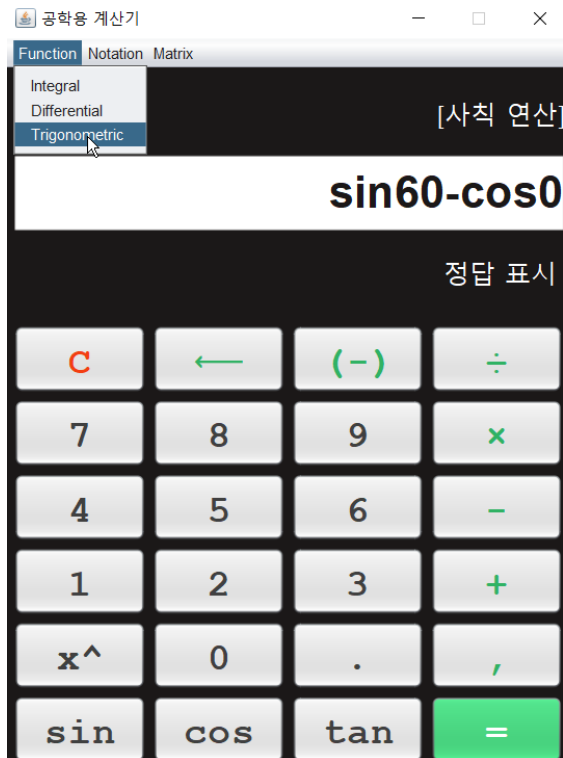


그림 1 Menu-Function-Trigonometric을 클릭 시 [삼각함수 계산]으로 전환 및 결과 도출

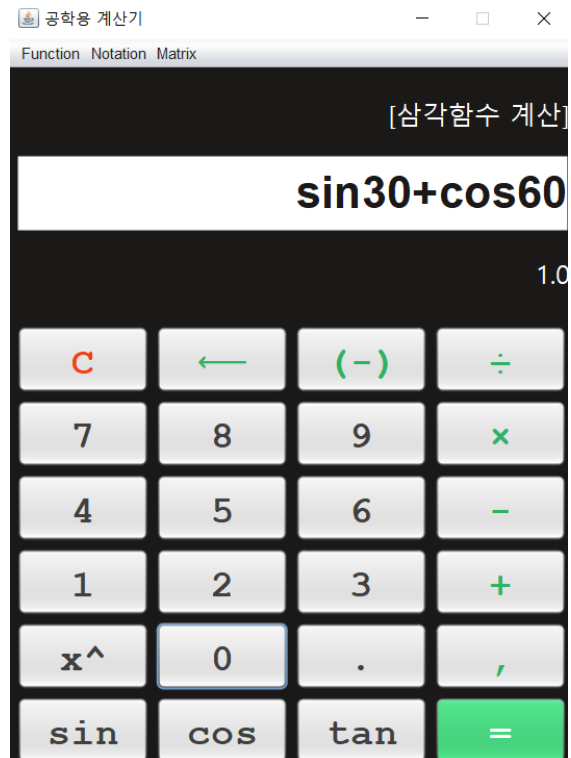


그림 2 $\sin 30 + \cos 60 = 1.0$



그림 3 $\tan 30 \times \cos 45 = 0.41$

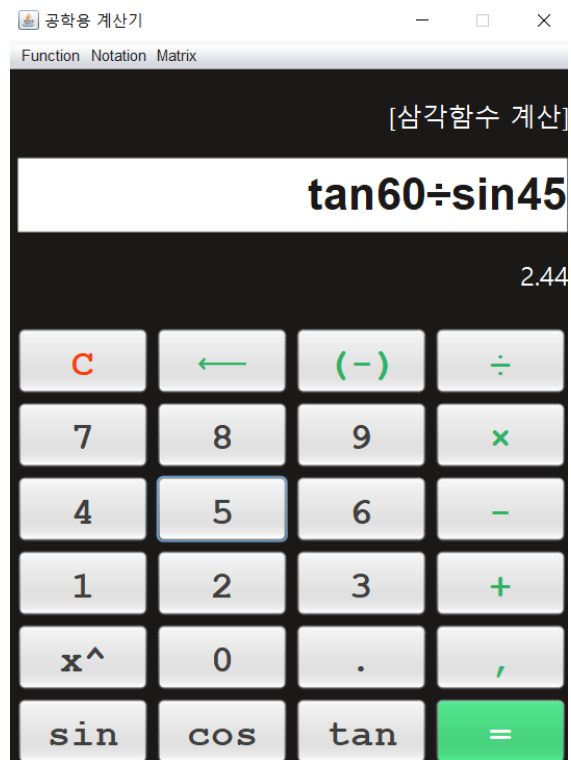


그림 4 $\tan 60 \div \sin 45 = 2.44$

(5) 2진수 변환



그림 1 Menu – Notation – Binary 클릭 시 [10진수에서 2진수로 변환]으로 전환 및 결과 도출



그림 2 $8 = 100(2)$



그림 3 $100 = 1100100(2)$

(6) 16진수 변환



그림 1 Menu - Notation - Hex 클릭 시 [2진수에서 16진수로 변환]으로 전환 및 결과 도출



그림 2 $24 = 18(16)$



그림 3 $63 = 3F(16)$

(7) 전치행렬

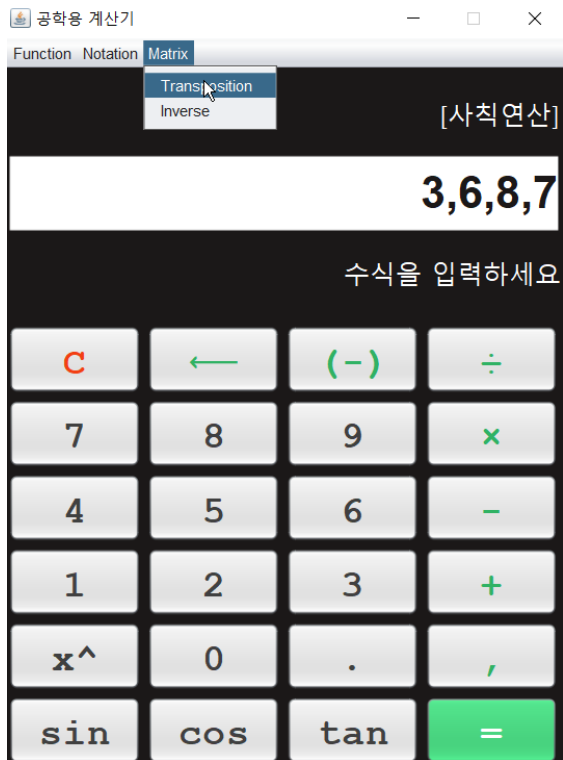


그림 1 Menu-Matrix-Transposition 클릭 시 [전치행렬]로 변환 및 결과 행렬 도출

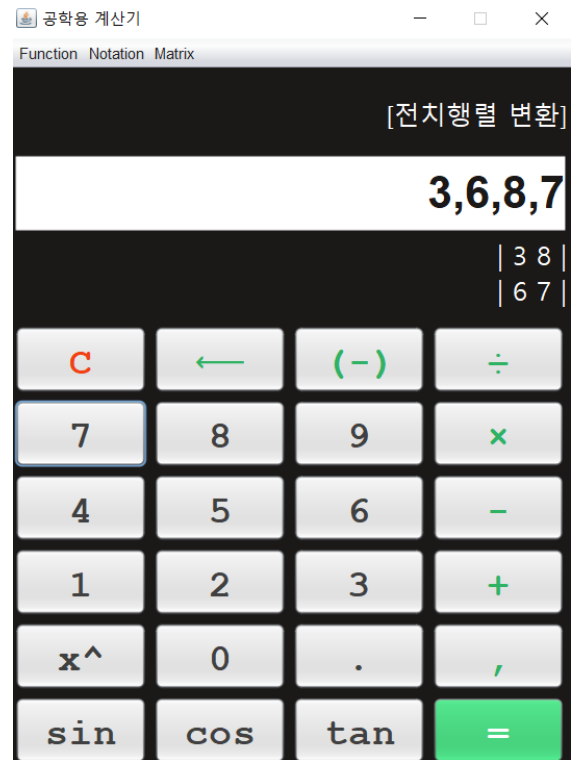


그림 2 $\begin{vmatrix} 3 & 8 \\ 6 & 7 \end{vmatrix} = \begin{vmatrix} 3 & 8 \\ 6 & 7 \end{vmatrix}$

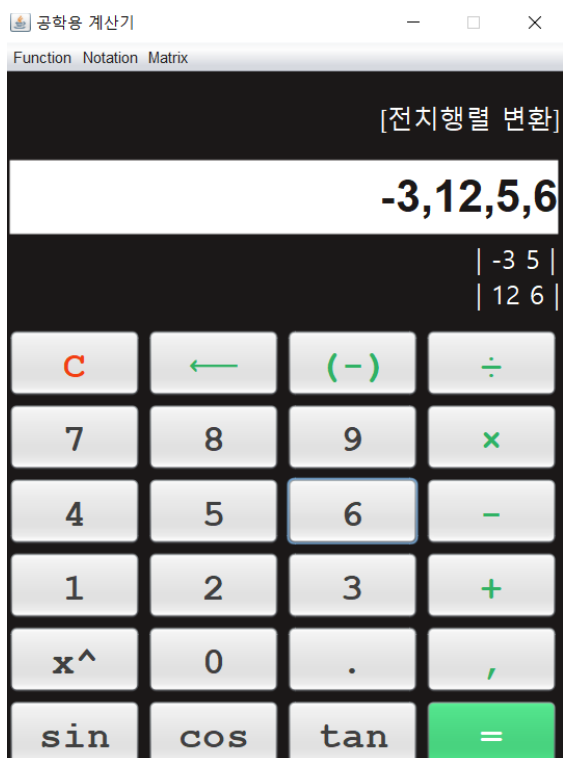


그림 3 $\begin{vmatrix} -3 & 5 \\ 12 & 6 \end{vmatrix} = \begin{vmatrix} -3 & 5 \\ 12 & 6 \end{vmatrix}$

(8) 역행렬



그림 1 Menu-Matrix-Inverse 클릭 시 [역행렬]로 변환 및 결과 행렬 도출

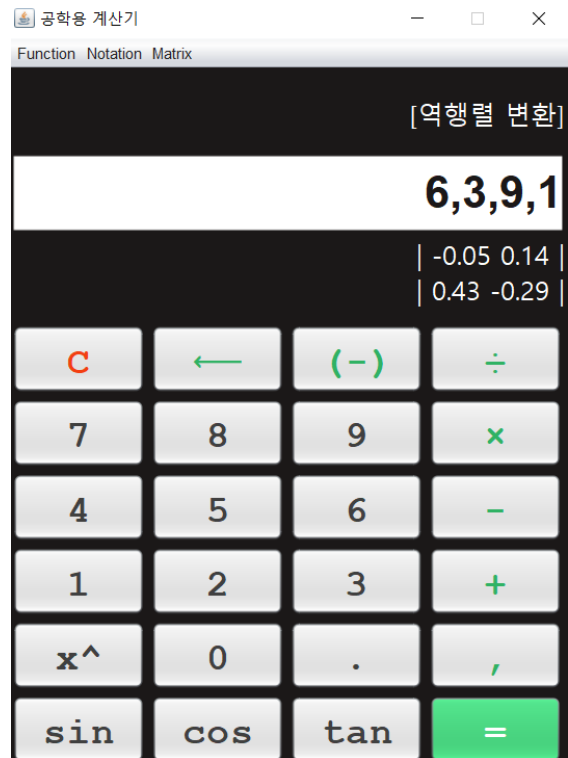


그림 2 $\begin{bmatrix} -0.05 & 0.14 \\ 0.43 & -0.29 \end{bmatrix}$



그림 3 $\begin{bmatrix} 0.13 & 0.1 \\ -0.2 & -0.4 \end{bmatrix}$

3. 예외처리

(1) 연산자, 버튼 예외처리



```
// 비어있을 경우 숫자만 입력할 수 있음
if (originalText.isEmpty())
    status = NUMBER_MODE;
switch (status) {
    case NUMBER_MODE: // 숫자를 입력 중일 때, 숫자/연산자 상관없이 입력 가능
        inputTextField.setText(originalText + button.getValue());
        break;
    case OPERATOR_MODE: // 직전 입력이 연산자 일 때, 숫자는 추가, 연산자 중 "="는 계산, 나머지는 치환
        if (button instanceof NumberButton) {
            inputTextField.setText(originalText + button.getValue());
        } else {
            inputTextField.setText(originalText.substring(0, originalText.length() - 1) + button.getValue());
        }
        isAbleToAddPoint = true;
        break;
    case DONE: // 계산이 끝났을 때 ("=" 버튼을 누른 이후)
        // 다시 점을 찍을 수 있게 바꿔줌
        isAbleToAddPoint = true;
        // 숫자를 바로 입력하면 다 지워진 상태에서 숫자만 입력됨
        if (button instanceof NumberButton) {
            inputTextField.setText(button.getValue());
        }
        // 연산자를 입력하면 뒤에 추가됨
        else if (button instanceof OperatorButton) {
            inputTextField.setText(inputTextField.getText() + button.getValue());
        }
        break;
    default:
        inputTextField.setText(originalText + button.getValue());
}
```

연산자, 버튼 예외처리로 사칙연산 중 mode를 나누어 연산자 뒤에는 연산자가 입력되지 않게 구현하였다. 또한 소수점을 찍고 나서 소수점을 바로 찍을 수 없게 flag을 이용하여 구현하였다.

(2) 잘못된 입력 값

잘못된 입력 값을 넣었을 경우나 올바른 계산 방법이 아닌 경우 경고 문구를 표시하고 다시 입력할 수 있게 구현하였다.



그림 1 미분식을 사칙연산으로 계산하려 한 경우

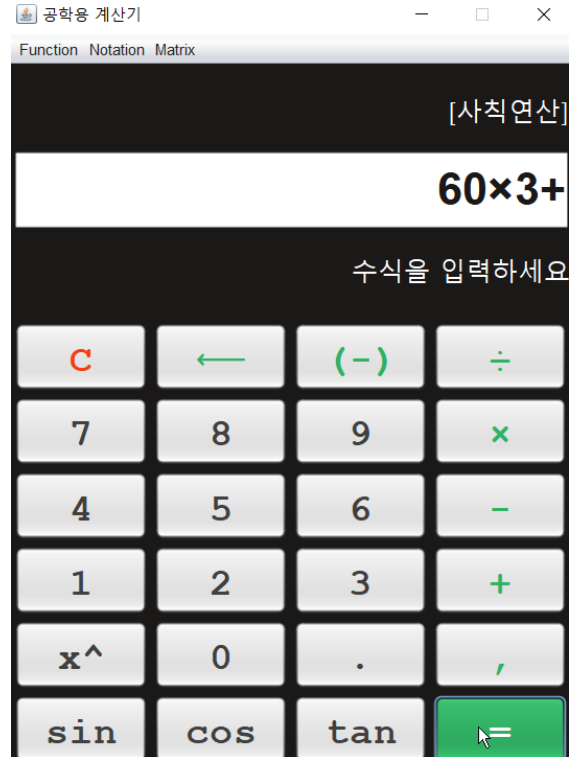


그림 2 계산식 입력을 끝내지 않은 상태(연산자로 끝난 상태)로 사칙연산을 시도한 경우



그림 3 행렬 계산 식인데 진수변환을 시도한 경우