CSED232 Object-Oriented Programming (Fall 2018)

Programming Assignment #5

Operator Overloading & QT

Due date: 2018년 12월 4일 화요일 오후 11:59

본 과제에서는 QT 환경에서 간단한 GUI를 활용해 "계산기 프로그램"을 구현한다. 이번 과제를 통해 Operator Overloading의 개념을 활용한 class를 디자인 하고 이를 실제 프로그래밍에 적용해본다.

- **계산기 프로그램**
  - 총 3가지 (**Matrix**, **Numeric**, **String**) 타입의 연산을 수행하는 프로그램이다.
  - 연산은 총 4가지 **ADD, SUB, MUL, DIV**(**+**, **-**, **x**, **/**)를 지원한다.
  - 연산은 [(피연산자1) (연산자) (피연산자2)] 형식으로 2가지 피연산자를 갖으며, **피연산자들의 타입 조합별**로 연산자가 수행하는 연산이 각각 다르다. (표 1 참조)

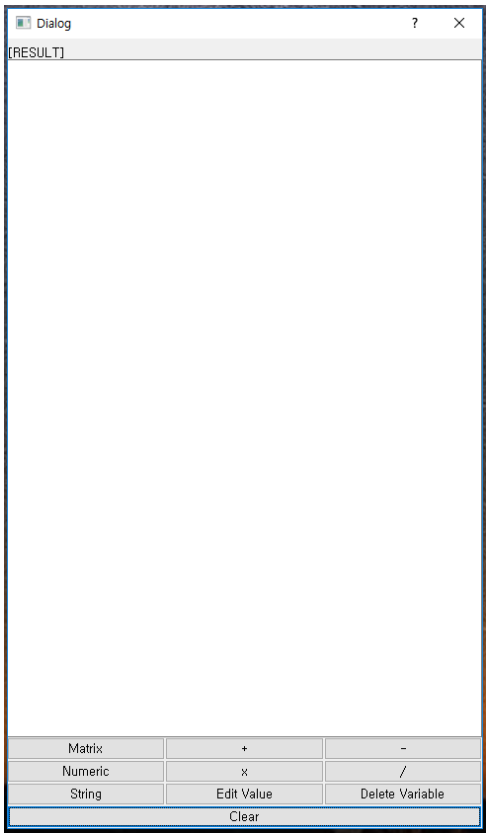


Figure 1. 프로그램 개요 (자세한 내용은 추후 설명)

1. 피연산자 설명

Matrix	2차원 (NxM) 행렬 (값은 float형) Example) $\begin{pmatrix} 3.14 & 4.23 \\ 2 & 1.34 \end{pmatrix}$ (2 × 2 *행렬*)
Numeric	실수 (값은 float형) Example) 12.4456
String	문자열 Example) "abcd"

2. 피연산자 타입 조합별 연산자 연산 방법

피연산자 1 \ 피연산자 2	**Matrix**	**Numeric**	**String**
Matrix	**ADD(+):** 2차원 행렬 + 연산 $\begin{pmatrix} 3.14 & 4.23 \\ 2 & 1.34 \end{pmatrix} + \begin{pmatrix} 1 & 2 \\ 3 & 4 \end{pmatrix} = \begin{pmatrix} 4.14 & 6.23 \\ 5 & 5.34 \end{pmatrix}$ **SUB(-):** 2차원 행렬 - 연산 $\begin{pmatrix} 3.14 & 4.23 \\ 2 & 1.34 \end{pmatrix} - \begin{pmatrix} 1 & 2 \\ 3 & 4 \end{pmatrix} = \begin{pmatrix} 2.14 & 2.23 \\ -1 & -2.66 \end{pmatrix}$ **MUL(x):** 2차원 행렬 x 연산 $\begin{pmatrix} 1 & 2 \\ 3 & 4 \end{pmatrix} \times \begin{pmatrix} 1 & 2 \\ 3 & 4 \end{pmatrix} = \begin{pmatrix} 7 & 10 \\ 15 & 22 \end{pmatrix}$ **DIV(/):** **불가능**	**ADD(+):** 행렬에 Numeric 값을 더함 $\begin{pmatrix} 3.14 & 4.23 \\ 2 & 1.34 \end{pmatrix} + 2 = \begin{pmatrix} 5.14 & 6.23 \\ 4 & 3.34 \end{pmatrix}$ **SUB(-):** 행렬에서 Numeric 값을 뺌 $\begin{pmatrix} 3.14 & 4.23 \\ 2 & 1.34 \end{pmatrix} - 2 = \begin{pmatrix} 1.14 & 2.23 \\ 0 & -0.66 \end{pmatrix}$ **MUL(x):** 행렬에 Numeric 값을 곱함 $\begin{pmatrix} 1 & 2 \\ 3 & 4 \end{pmatrix} \times 2 = \begin{pmatrix} 7 & 10 \\ 15 & 22 \end{pmatrix}$ **DIV(/):** 행렬에 Numeric 값을 나눔 $\begin{pmatrix} 1 & 2 \\ 3 & 4 \end{pmatrix} / 2 = \begin{pmatrix} 0.5 & 1 \\ 1.5 & 2 \end{pmatrix}$	**ADD(+):** **불가능** **SUB(-):** **불가능** **MUL(x):** **불가능** **DIV(/):** **불가능**

피연산자 2 / 피연산자 1	Matrix	Numeric	String
Numeric	**ADD(+):** 행렬에 Numeric 값을 더함 $2+\begin{pmatrix}3.14 & 4.23\\ 2 & 1.34\end{pmatrix}$ $=\begin{pmatrix}5.14 & 6.23\\ 4 & 3.34\end{pmatrix}$ **SUB(-):** 행렬에서 Numeric 값을 뺌 $2-\begin{pmatrix}3.14 & 4.23\\ 2 & 1.34\end{pmatrix}$ $=\begin{pmatrix}1.14 & 2.23\\ 0 & -0.66\end{pmatrix}$ **MUL(x):** 행렬에 Numeric 값을 곱함 $2\times\begin{pmatrix}1 & 2\\ 3 & 4\end{pmatrix}$ $=\begin{pmatrix}7 & 10\\ 15 & 22\end{pmatrix}$ **DIV(/):** 행렬에 Numeric 값을 나눔 $2/\begin{pmatrix}1 & 2\\ 3 & 4\end{pmatrix}$ $=\begin{pmatrix}0.5 & 1\\ 1.5 & 2\end{pmatrix}$	**ADD(+):** 실수 + 연산 $2+4.3=6.3$ **SUB(-):** 실수 - 연산 $2-4.3=-2.3$ **MUL(x):** 실수 x 연산 $2\times4.3=8.6$ **DIV(/):** 실수 / 연산 $4.3\div2=2.15$	**ADD(+):** String Concatenation $2.32+\text{"abc"}$ $=\text{"2.32abc"}$ **SUB(-):** 불가능 **MUL(x):** 소수점 아래는 버림하고, 정수부만 concatenation $2.32*\text{"abc"}$ $=\text{"abcabc"}$ **DIV(/):** 불가능
String	**ADD(+):** 불가능 **SUB(-):** 불가능 **MUL(x):** 불가능 **DIV(/):** 불가능	**ADD(+):** String Concatenation $2.32+\text{"abc"}$ $=\text{"2.32abc"}$ **SUB(-):** 불가능 **MUL(x):** 소수점 아래는 버림하고, 정수부만 concatenation $2.32*\text{"abc"}=\text{"abcabc"}$ **DIV(/):** 불가능	**ADD(+):** String Concatenation $\text{"}ef\text{"}+\text{"abc"}=\text{"efabc"}$ **SUB(-):** 불가능 **MUL(x):** 불가능 **DIV(/):** 불가능

Table 1. 피연산자 타입 조합별 연산자

3. 프로그램 기능

1) 전체적인 UI 설명

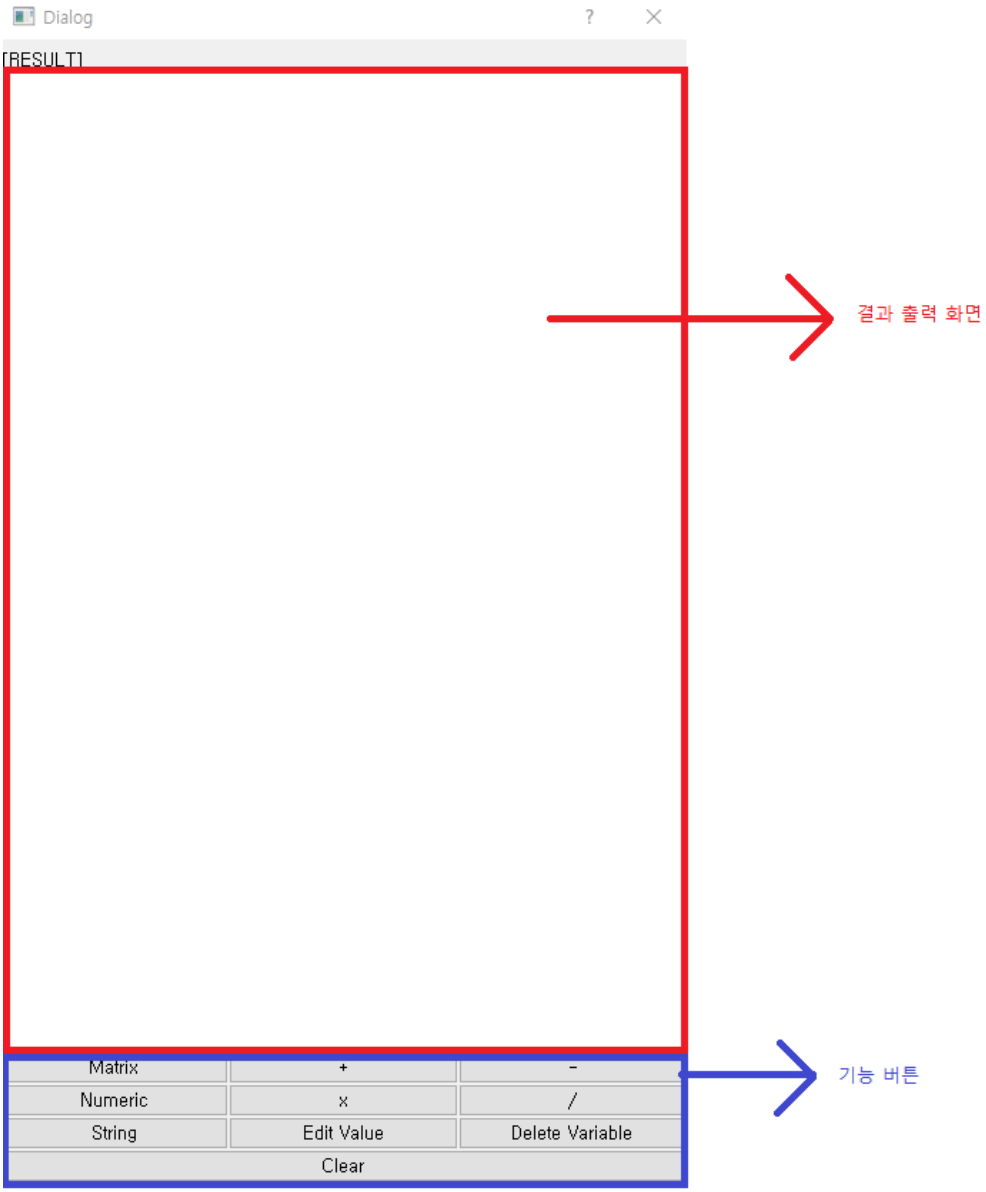
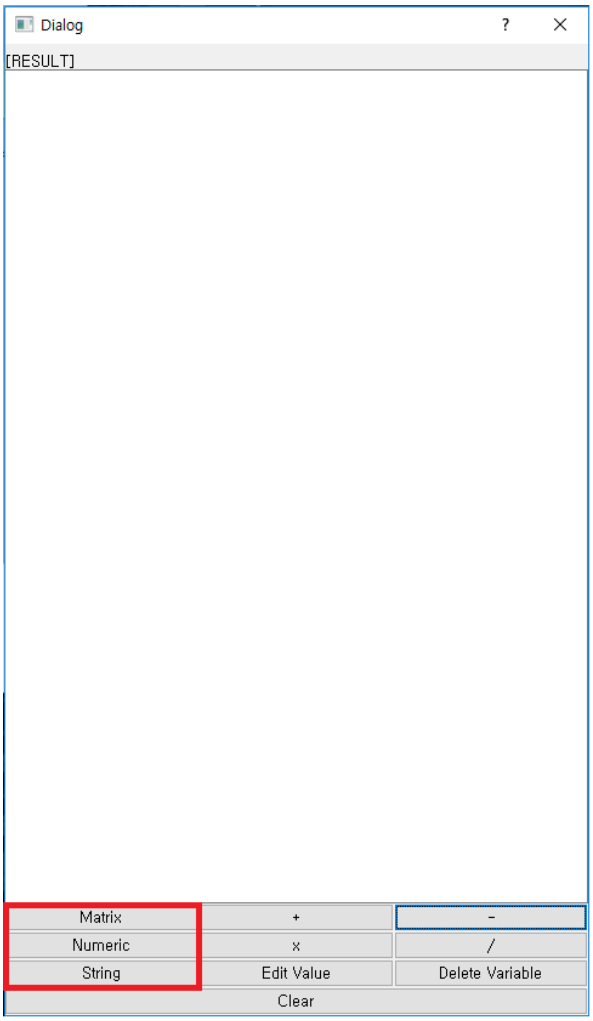


Figure 2. 전체 UI 형식

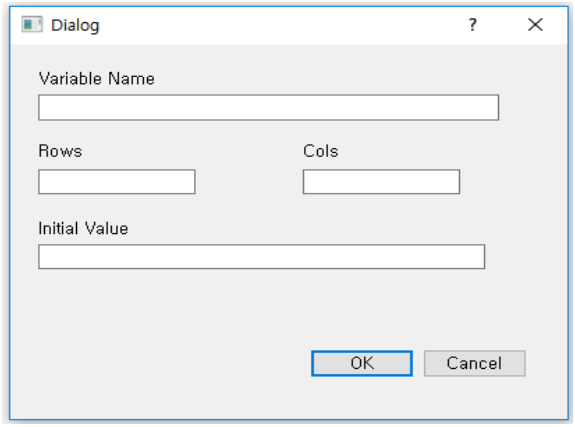
- **결과 출력 화면**에는 **기능 버튼**들로부터 수행된 기능에 대한 결과를 출력해준다.
- **전체 다이얼로그의 크기**는 **(480x800)** 으로 구성하며, 내부 컴포넌트들은 자유롭게 배치한다.

2) 변수 선언



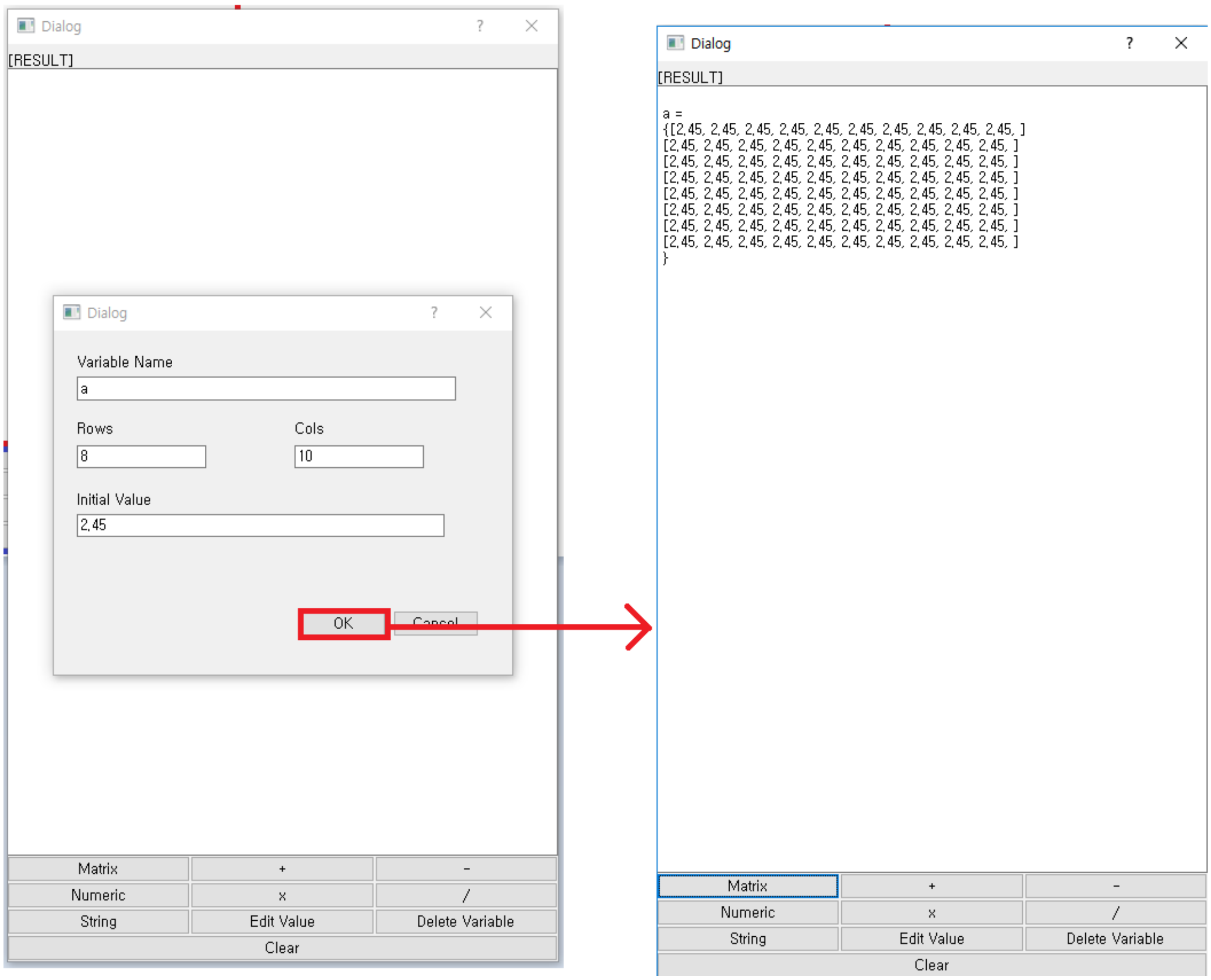
- 변수 선언을 위한 버튼이다.
- 각각의 버튼 (Matrix, Numeric, String)을 클릭 시 변수 선언을 위한 다이얼로그가 나타난다.
- 각각의 다이얼로그에 대해서는 추후 설명을 참고한다.

2-1) Matrix 다이얼로그



- Variable Name에는 변수명을 입력한다.
- Rows, Cols는 각각 행의 수와 열의 수를 입력한다.
- Initial Value는 행렬의 초기값을 입력한다.

- 실행 예시



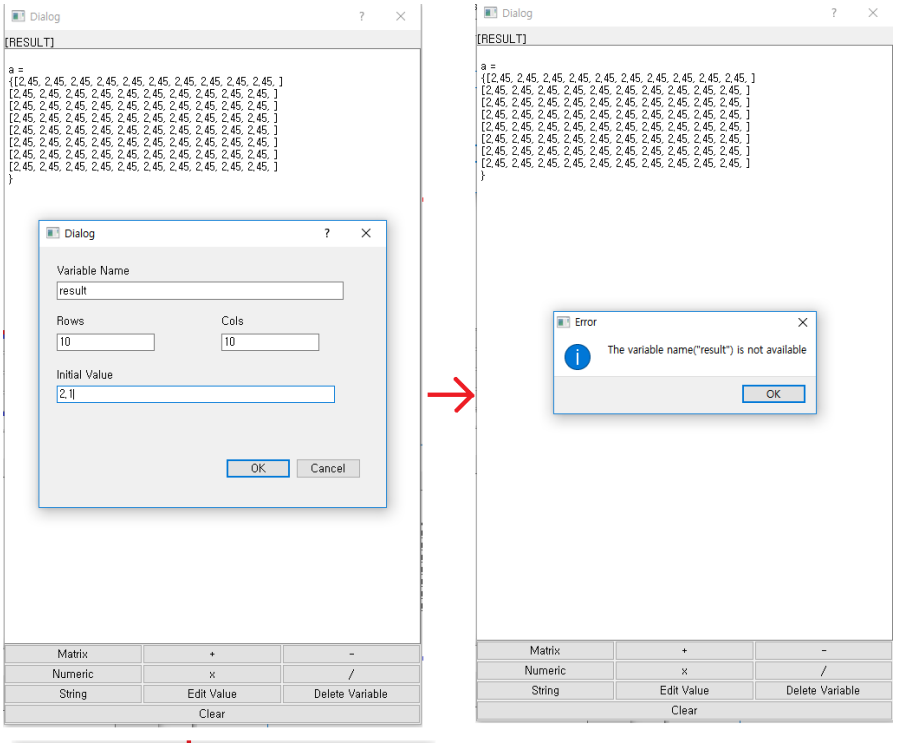
- 입력 후 OK를 누르면 다음과 같이 결과 값을 **결과 출력 화면**에 출력해준다.
- 결과 값은 아래와 같은 형식을 따른다.

```
변수명 =
{[값 1, 값 2, ……, 값 M, ]
[값 M+1, 값 M+2, ……, 값 2M, ]
……
[값 (N-1)M, 값 (N-1)M+1, ……, 값 NM, ]
}
```

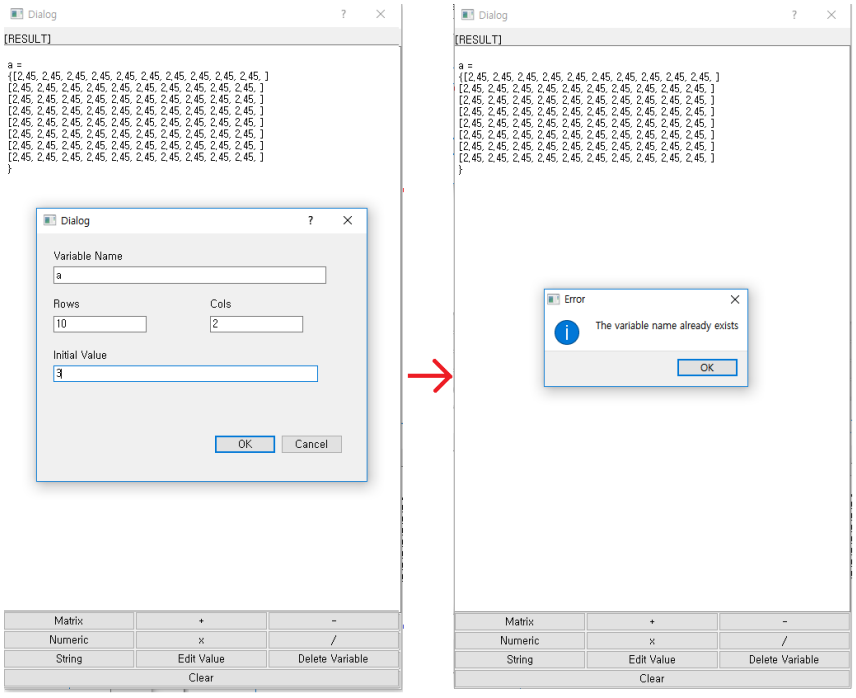
- **변수명은 "result" 예약어 빼고는 무엇이든 가능하다.**

- 오류 출력 예시

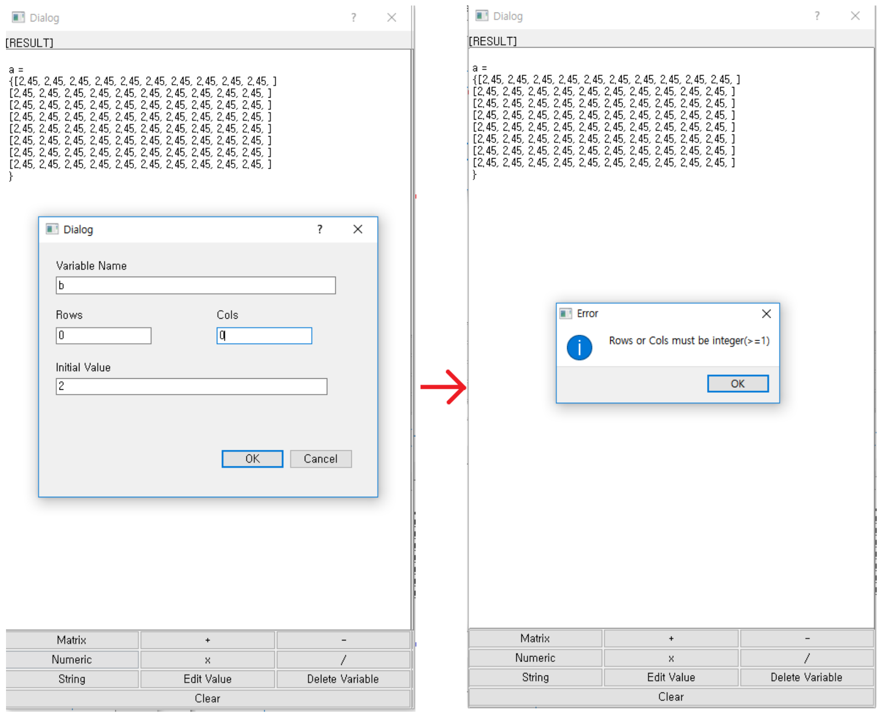
① 예약어(“result”)를 변수명으로 입력했을 경우



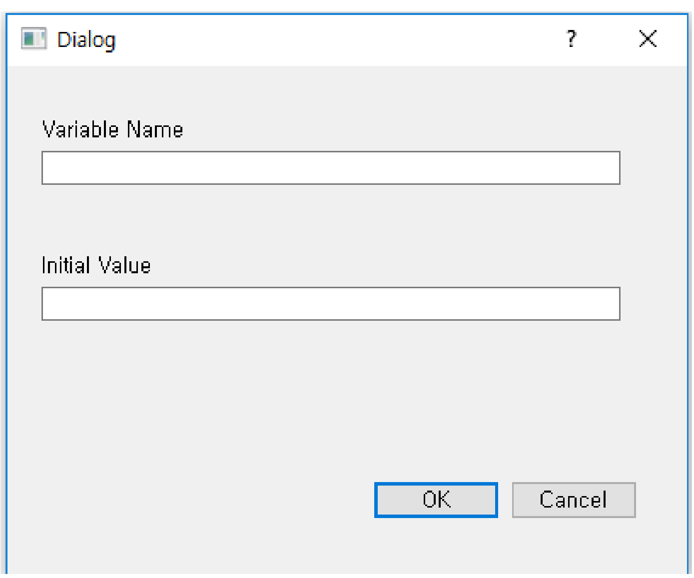
② Variable Name이 이미 존재하는 경우



③ Rows 값이나 Cols 값이 잘못된 경우

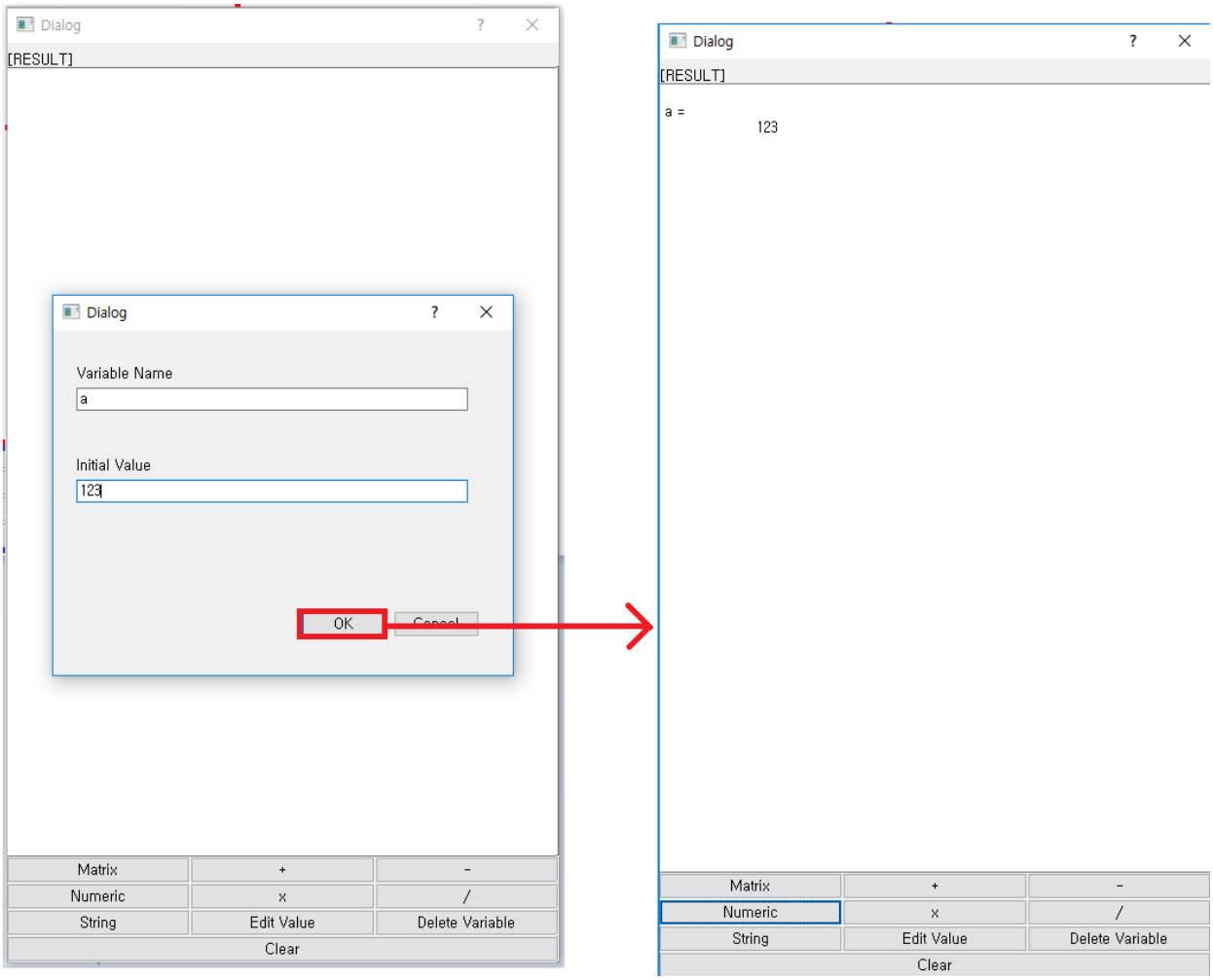


2-2) **Numeric 다이얼로그 / String 다이얼로그**



- Variable Name에는 변수명을 입력한다.
- Initial Value는 행렬의 초기값을 입력한다.

- 실행 예시



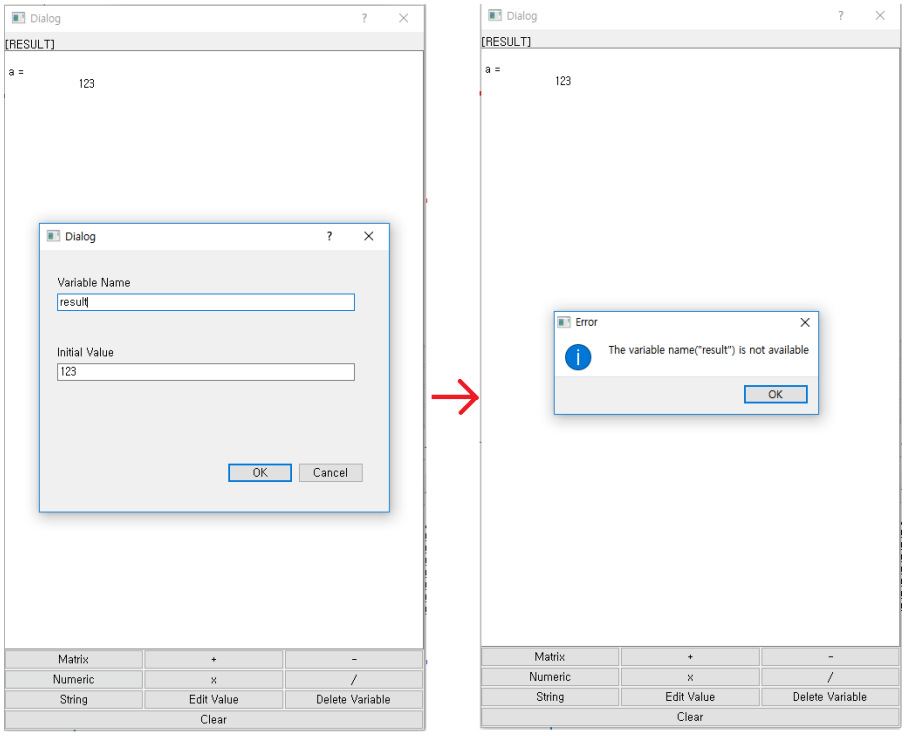
- 입력 후 OK를 누르면 다음과 같이 결과 값을 **결과 출력 화면**에 출력해준다.
- 결과 값은 아래와 같은 형식을 따른다.

변수명 = **(₩t)값**

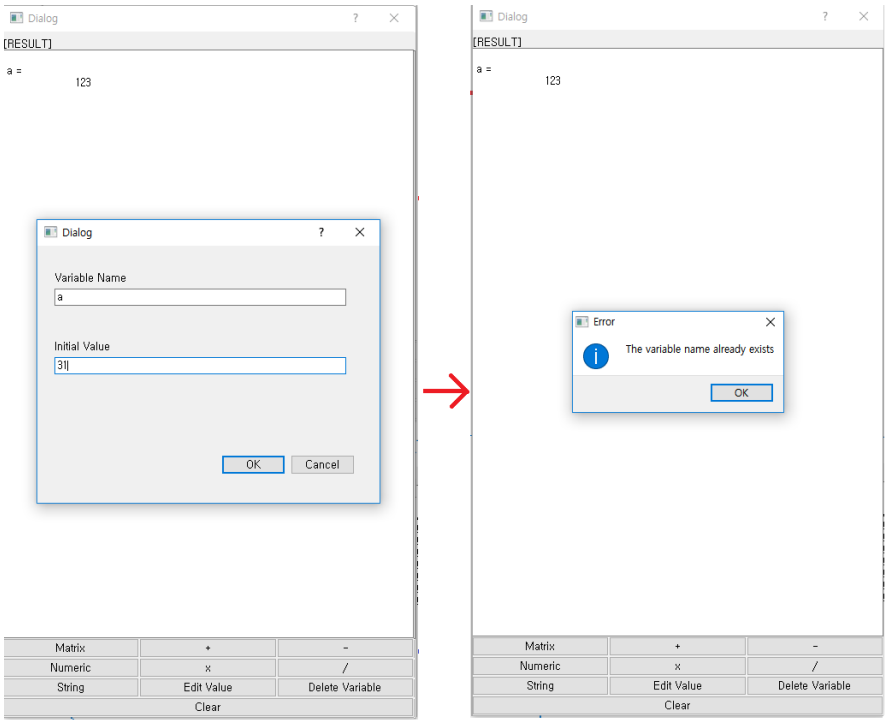
- **변수명은 "result" 예약어 빼고는 무엇이든 가능하다.**
- **String 변수 선언 또한 똑같은 과정으로 진행된다.**

- **오류 출력 예시**

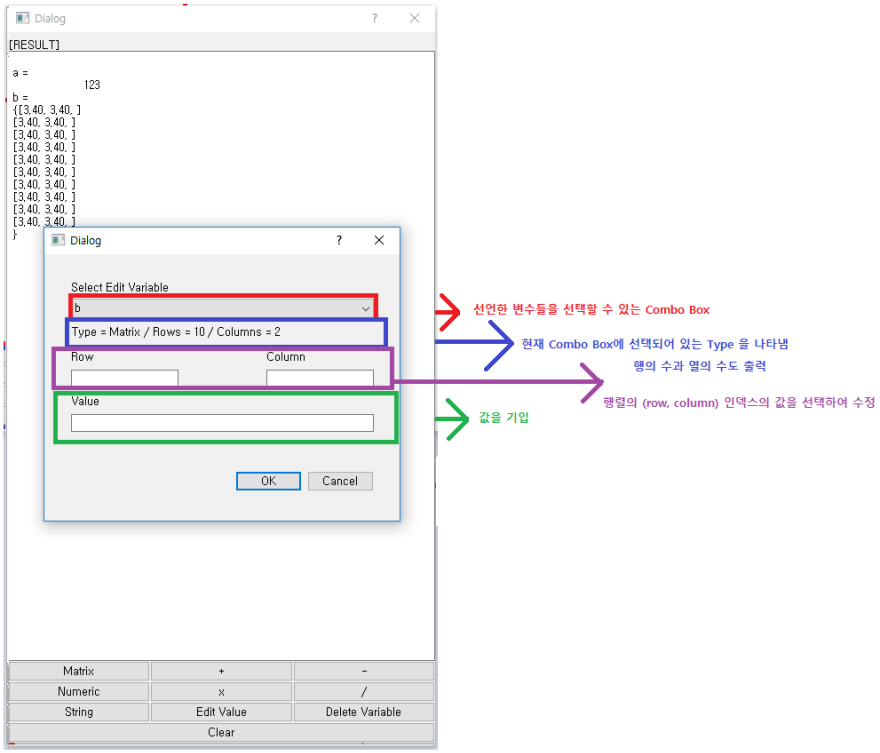
① 예약어(“result”)를 변수명으로 입력했을 경우



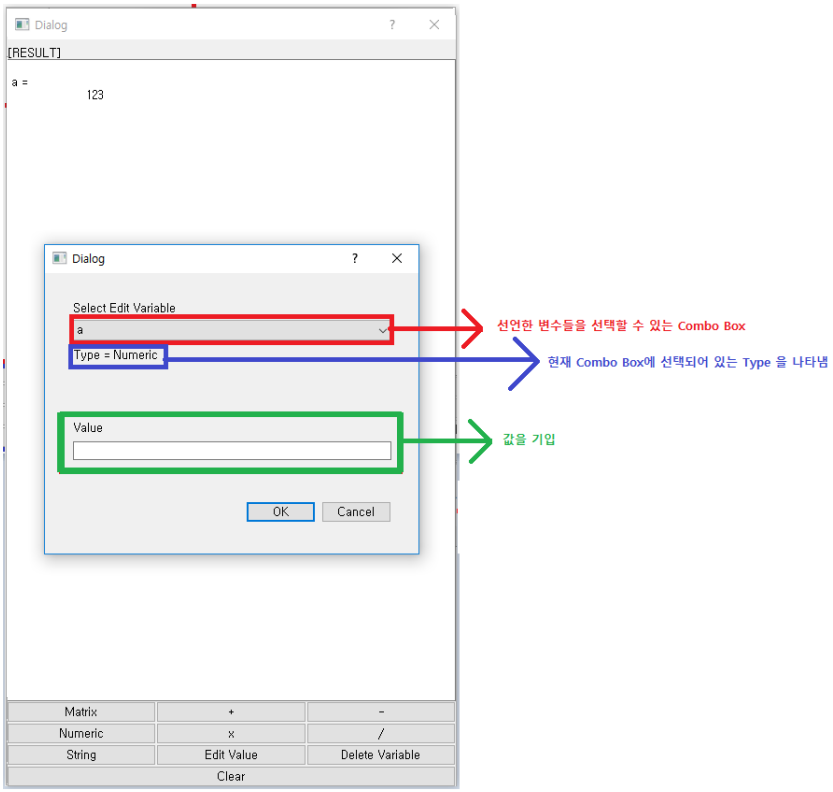
② Variable Name이 이미 존재하는 경우



3) 변수의 값 수정 다이얼로그 (Edit Value)

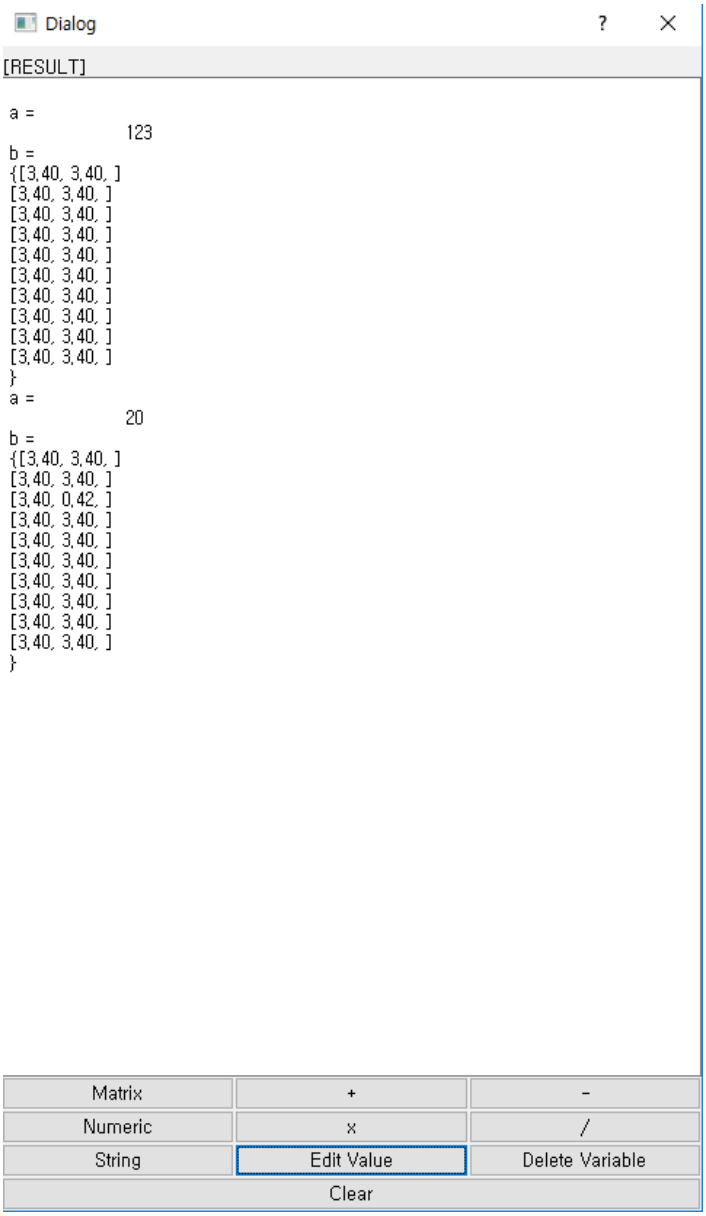


Matrix 변수를 선택했을 때 출력되는 다이얼로그



Numeric/String 변수를 선택했을 때 출력되는 다이얼로그

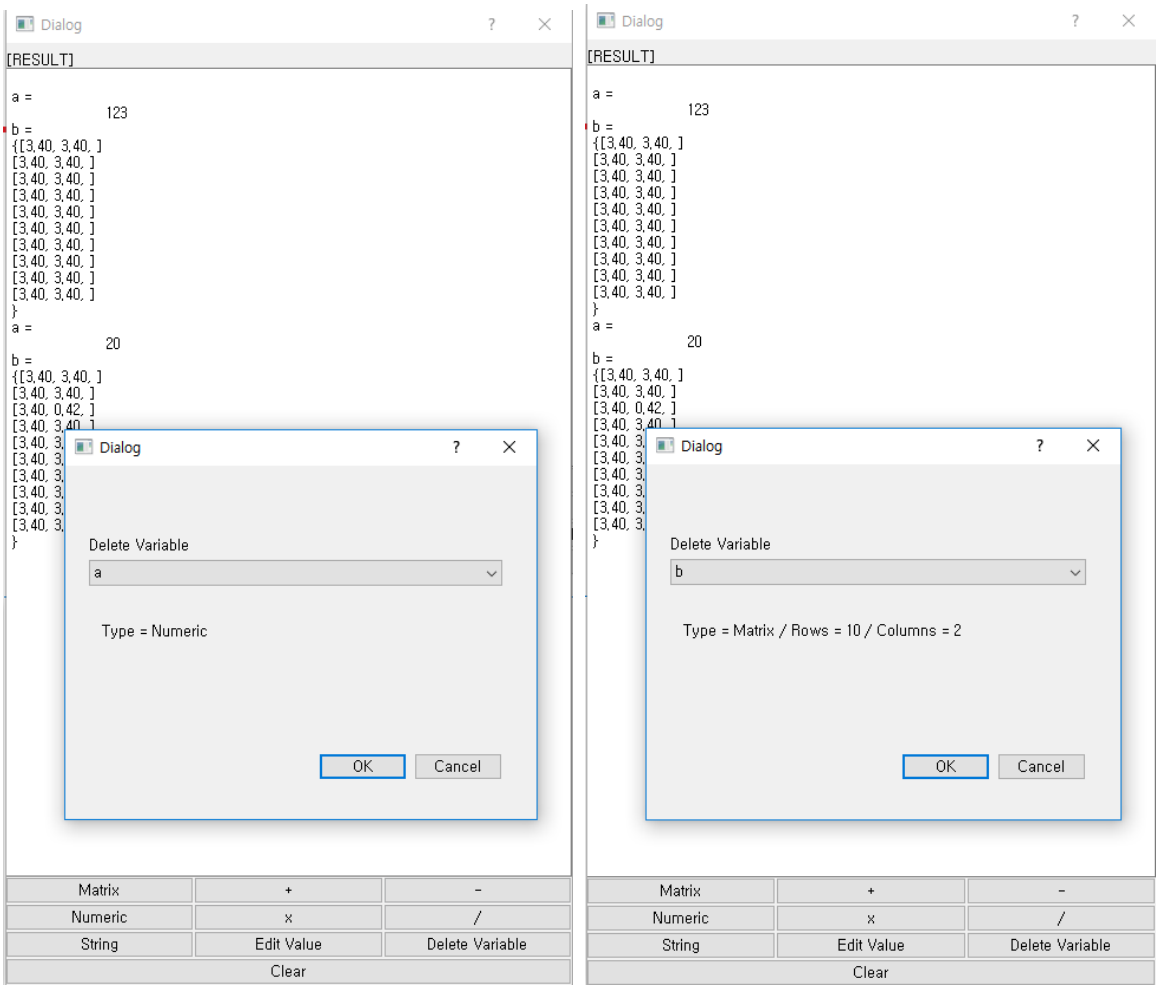
- 결과 출력



- 변경된 값을 **변수 선언과 마찬가지로 결과 출력 화면에 출력해준다.**

- 예시는 Numeric 타입(a)과 Matrix 타입(b)을 변경했을 때의 출력화면이다.

4) 변수 삭제 다이얼로그 (Delete Variable)

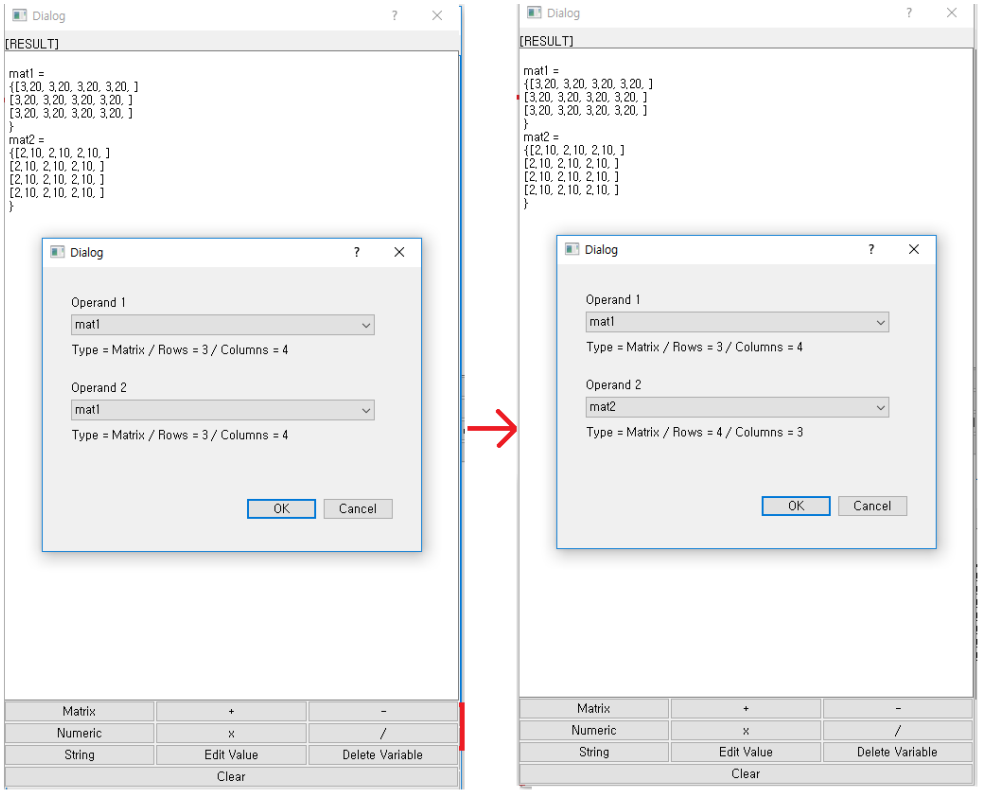


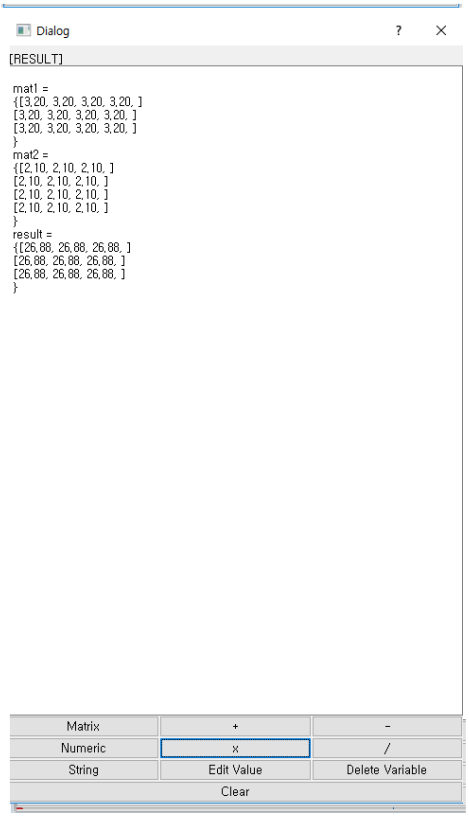
- Combo Box 에서 변수를 선택하여 삭제한다.
- **결과 출력 화면에는 결과에 대해 따로 출력하지 않는다.**

5) 결과 창 / 변수 클리어 (Clear)

- 프로그램 초기 실행 상태로 회귀한다.
- 결과 창의 내용을 모두 지우고, 저장된 변수들 또한 삭제한다.

6) 연산 Dialog

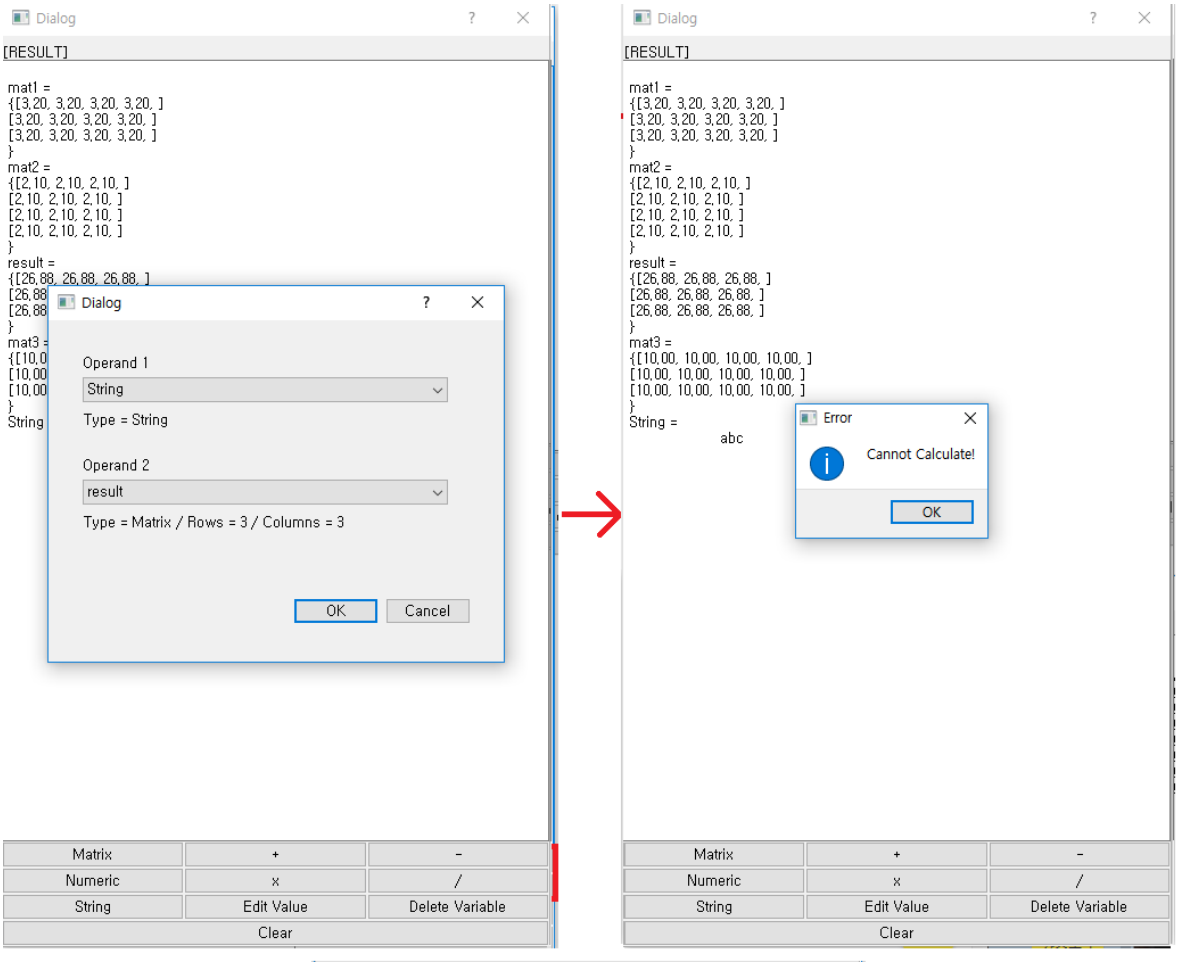

Dialog
[RESULT]
mat1 =
{[3.20, 3.20, 3.20, 3.20,]
[3.20, 3.20, 3.20, 3.20,]
[3.20, 3.20, 3.20, 3.20,]
}
mat2 =
{[2.10, 2.10, 2.10,]
[2.10, 2.10, 2.10,]
[2.10, 2.10, 2.10,]
[2.10, 2.10, 2.10,]
}
Operand 1
mat1
Type = Matrix / Rows = 3 / Columns = 4
Operand 2
mat1
Type = Matrix / Rows = 3 / Columns = 4
OK
Cancel
Operand 1
mat1
Type = Matrix / Rows = 3 / Columns = 4
Operand 2
mat2
Type = Matrix / Rows = 4 / Columns = 3
OK
Cancel
Matrix
+
-
Numeric
x
/
String
Edit Value
Delete Variable
Clear


Dialog
[RESULT]
mat1 =
{[3.20, 3.20, 3.20, 3.20,]
[3.20, 3.20, 3.20, 3.20,]
[3.20, 3.20, 3.20, 3.20,]
}
mat2 =
{[2.10, 2.10, 2.10,]
[2.10, 2.10, 2.10,]
[2.10, 2.10, 2.10,]
[2.10, 2.10, 2.10,]
}
result =
{[26.88, 26.88, 26.88,]
[26.88, 26.88, 26.88,]
[26.88, 26.88, 26.88,]
}
Matrix
+
-
Numeric
x
/
String
Edit Value
Delete Variable
Clear

- 위의 그림은 Matrix 간의 곱하기 연산을 수행하는 예시이다.
- Operand1과 Operand2를 선택한 후 [OK 버튼]을 클릭하면 result라는 변수에 해당 값이 할당되고 **result값은 다른 계산에 피연산자**로써 사용될 수 있다
- 위의 예시에서는 result는 Matrix 타입을 갖는다.
- **피연산자 타입 조합별 연산자 연산 방법**에서 설명했듯이, 타입 별로 연산 방법이 다르고 result에 할당되는 타입 또한 다르다.
- **Result 값은 연산을 할 때마다 덮어써진다.**

- **오류 출력 예시**

① 피연산자 타입 조합이 불가능한 경우 (예시, String 타입과 Matrix 타입의 합 연산 결과)



② Matrix 타입 간의 곱이 불가능한 경우