

[프로젝트 1 - 기본에 충실한 계산기 구현] 완료 보고서 및 코드 상세 설명서

2020.03.21 ~ 2020.03.23

Class CalController implements Initializable

필드	DATA TYPE	VARIABLE NAME	VALUE	용도 및 기능	비고
	Button	btn0~btn9	-	계산기 숫자 0~9	@FXML
	Button	btnPoint	-	계산기 소수점	
	Button	btnCencle	-	계산기 모두 취소 버튼	
	Button	btnDelete	-	지우기 버튼	◀
	Button	btnResult	-	=버튼	
	Button	btnPlus, btnMinus, btnx, btnDivide	-	더하기, 빼기, 곱하기, 나누기 연산 버튼	
			-		
	Label	result	-	입력값 및 결과값 텍스트 출력 라벨	
	String	outputVal	""	입력값 출력용	
	int	getNumbers	0	입력된 숫자들 나누어 저장	
	ArrayList<String>	getOper	new ArrayList	연산자 저장 리스트 배열	
	ArrayList<String>	getNumber	new ArrayList	outputVar에 저장된 숫자들을 나누어 저장.	

public void initialize(URL location, ResourceBundle resource) @Override

메소드	상세 기능	
	① 이벤트 핸들러 등록	

void clickBtn0~9(ActionEvent e)

메 소 드	지역 변수	DATA TYPE	VARIABLE NAME	VALUE	용도 및 기능
		String	inputVal	btn.getText()	버튼의 텍스트 값을 받아서 저장.
	상세 기능	① inputVar의 데이터를 saveInputVar() 메소드의 인자로 넘겨주는 역할.			

void clickBtnPlus(ActionEvent e), clickBtnMinus(ActionEvent e), clickBtnDivide(ActionEvent e), clickBtnMultiple(ActionEvent e)

메소드	지역 변수	DATA TYPE	VARIABLE NAME	VALUE	용도 및 기능
		String	inputVal	btn.getText()	버튼의 텍스트 값을 받아서 저장.
	상세 기능	① outputVar에 초기 숫자 입력값이 저장 되었는지 검사한다. ② 만약 ①의 결과가 true일 경우, operDuplicationCheck() 메소드를 호출하여 inputValue 값을 넘겨준다.			

void saveInputVal(String val)

메소드	지역 변수	DATA TYPE	VARIABLE NAME	VALUE	용도 및 기능
		String	val	-	매개 변수
	상세 기능	① outputVal 필드에 입력 값 순서대로 저장하는 기능. ② result 라벨에 텍스트 설정하는 기능(입력값 출력)			

void clickBtnDelete(ActionEvent e)

메소드	상세 기능	① outputVal의 맨 마지막 입력 데이터값을 지우는 기능
		② outputVal 데이터를 result 라벨의 텍스트로 설정하는 기능 (수정된 입력값 출력)

void clickBtnCencle(ActionEvent e)

메소드	상세 기능	① outputVar 변수를 "" 값으로 초기화.
		② result 라벨의 텍스트를 "0"으로 설정한다.

String sortNumber()

메소드	지역 변수	DATA TYPE	VARIABLE NAME	VALUE	용도 및 기능
		String	numStr	""	number 데이터를 합치는 기능.
		String	oper	-	파싱된 연산자를 저장한다.
		char	number	-	outputVar의 데이터를 문자 단위로 쪼개어 하나씩 차례로 대입한다.

드	상세 기능	① 숫자 데이터만 파싱하는 기능.
		② 0~9 사이의 숫자가 아니면 sortOperator() 메소드를 호출하여 oper값을 넘겨준다.
		③ numStr 변수를 리턴한다.

void sortOperator(String operVar)

메 소 드	지역 변수	DATA TYPE	VARIABLE NAME	VALUE	용도 및 기능
		String	operVal	-	매개 변수
	상세 기능	① 연산자 데이터만 파싱한다.			
		② operVar 매개변수를 들어온 순서대로 getOper() 배열 리스트에 저장한다.			

void sortOperator(String operVar)

메 소 드	지역 변수	DATA TYPE	VARIABLE NAME	VALUE	용도 및 기능
		String	operVal	-	매개 변수
	상세 기능	① 연산자 데이터만 파싱한다.			
		② operVal 매개변수를 들어온 순서대로 getOper() 배열 리스트에 저장한다.			

void operDuplicationCheck(String value)

메 소 드	지역 변수	DATA TYPE	VARIABLE NAME	VALUE	용도 및 기능
		String	value	-	매개 변수
		char	getEnd	-	value의 마지막 문자 저장 용도
	상세 기능	① outputVal값의 마지막 데이터를 getEnd에 저장			
		② getEnd에 저장된 데이터가 연산자 인지 검사.			
		③ 만약 ②의 결과가 true일 경우, 마지막 연산자는 삭제, 새로운 데이터인 value를 outputVal에 더한다.			
		④ 만약 ②의 결과가 false일 경우, saveInputVal() 메소드를 호출하여 value 값을 넘겨준다.			

void initializationAll()

메소드	상세 기능	① getOper 배열의 모든 값을 초기화 시킨다.(모든 데이터 삭제)
		② getNumver 배열의 모든 값을 초기화 시킨다. (모든 데이터 삭제)
		③ outputVal 필드에 ""를 저장한다.

void clickBtnEquals(ActionEvent e)

메 소 드	지역 변수	DATA TYPE	VARIABLE NAME	VALUE	용도 및 기능	
		double	num	0	getNumber의 인덱스 1부터의 데이터 저장	
		double	result	getNumber(0)	getNumber(0) 데이터를 얻어, double값으로 파싱 후 저장	
		int	operIndex	0	getOper 인덱스 카운터 용	
		int	a	(int)result	결과값이 정수인지 실수인지 검사하는 용도.	
		double	b	result - a		
	상세 기능	① getNumber 배열에 sortNumber의 리턴값을 추가한다.				
		② result에 getNumber(0)의 데이터를 double 값으로 파싱하여 저장한다.				
		③ getNumber의 사이즈 만큼 for을 돌린다.				
		④ getNumber의 데이터들을 하나씩 순서대로 double 타입으로 파싱 후, num 변수에 저장한다.				
		⑤ getOper의 데이터를 조회하여 연산자 종류를 검색 후 result에 num 값을 연산한다.				
		⑥ 계산을 마치면 operIndex를 1씩 증가 시킨다.				
		⑦ 결과값이 나오면 a에 result값을 int 타입으로 캐스팅 하여 저장한다.				
		⑧ b에 result -a 연산을 하여 저장한다.				
		⑨ b보다 0.1이 크면 int타입으로 캐스팅 후 outputVal에 저장한다. (Integer.toString() 이용)				
		⑩ 반대의 경우 result를 String 타입으로 파싱 후 outputVal에 저장한다.				
⑪ result 라벨의 텍스트에 outputVal 값을 설정한다.						
⑫ initialization() 메소드를 호출한다.						