

# OOPT Stage 2030

<Analyze>

Version 3.0

**Project Team**

**T7 Team**

Date

**2017-05-20**

---

**Team Information**

201414134 오세욱

201414136 임현유

201211375 임동현

201211387 하헌규

## Table of Contents

1	Activity 2010. Revise Plan.....	3
2	Activity 2020. Synchronize Artifacts .....	6
3	Activity 2031. Define Essential Use Cases .....	6
4	Activity 2032. Refine Use Case Diagrams.....	11
5	Activity 2033. Define Domain Model.....	11
6	Activity 2034. Refine Glossary .....	12
7	Activity 2035. Define System Sequence Diagrams.....	12
8	Activity 2036. Define Operation Contracts .....	17
9	Activity 2038. Refine System Test Case.....	20
10	Activity 2039. Analyze Traceability Analysis.....	21

## 1. Activity 2010. Revise Plan

### 1.1 Project Objectives

기존의 Objective에서 대푯값을 선택하여 전체 결과값 내에서 검색하여 결과를 프로그램 내에서 출력하는 것으로 변경

<b>Ver1.0</b>	대푯값에 우선 순위를 부여하여 그 결과값을 먼저 출력하고, 확인할 동안 나머지 전체 결과를 출력하여 시간을 절약할 수 있는 Category Partition Tool 을 목표로 한다.
<b>Ver2.0</b>	사용자가 대푯값을 선택하여 그 결과값을 프로그램 상에서 출력하고, 전체 결과값은 파일로 출력하는 Category Partition Tool을 목표로 한다.

### 1.2 Functional Requirements

기존의 Functional Requirements는 implement를 생각하고 작성하였기 때문에 시스템 내부를 자세하게 설명함.

- **Ver1.0**

Ref	Function	Description
R1.1	Category 추가	Category 객체를 생성한다.
R1.2	Category 수정	Category를 수정한다.
R1.3	Category 삭제	Category 객체를 삭제한다.
R1.4	Value 추가	Value 객체를 생성한다.
R1.5	Value 수정	Value를 수정한다.
R1.6	Value 삭제	Value 객체를 삭제한다.
R1.7	Constraint 추가	Constraint를 추가한다.
R1.8	Constraint 수정	Constraint를 수정한다.
R1.9	Constraint 삭제	Constraint를 삭제한다.
R1.10	Value 우선 순위 선택	Value에 우선 순위를 부여한다.
R2	Test Case 계산	전체 Test Case를 계산한다.
R2.1	우선 순위 계산	우선 순위가 있는 Value에 관련된 Test Case를 먼저 계산하고 그 Test Case가 전체 Test Case중 가지는 비중을 계산한다.
R2.2	진행 상황 계산	전체 Test Case중 얼마큼 진행 했는지 계산한다.
R2.3	출력	결과값들을 출력한다.
R2.3.1	우선 순위 출력	우선 순위 Test Case를 화면에 출력한다.

R2.3.2	진행 상황 출력	진행 상황을 화면에 출력한다.
R2.3.3	File 출력	전체 Test Case를 File로 출력한다.
R3.1	초기화	모든 입력과 출력 화면을 초기화한다.
R4.1	프로그램 종료	프로그램을 종료한다.

- Ver2.0

Ref	Function	Description
R1.1	Category 추가	Category를 추가한다.
R1.2	Category 수정	Category를 수정한다.
R1.3	Category 삭제	Category를 삭제한다.
R1.4	대푯값 추가	대푯값을 추가한다.
R1.5	대푯값 수정	대푯값을 수정한다.
R1.6	대푯값 삭제	대푯값을 삭제한다.
R1.7	Constraint 추가	Constraint를 추가한다.
R1.8	Constraint 수정	Constraint를 수정한다.
R1.9	Constraint 삭제	Constraint를 삭제한다.
R1.10	대푯값 선택	검색하고 싶은 대푯값을 선택한다.
R2.1	Test Case 계산	Test Case를 계산한다.
R2.2	출력	결과를 출력한다.
R3.1	초기화	모든 입력과 출력 화면을 초기화한다.
R4.1	프로그램 종료	프로그램을 종료한다.

- Ver3.0

Ref	Function	Description
R1.1	Category 추가	Category를 생성한다.
R1.2	Category 수정	Category를 수정한다.
R1.3	Category 삭제	Category를 삭제한다.
R1.4	대푯값 추가	대푯값을 추가한다.
R1.5	대푯값 수정	대푯값을 수정한다.
R1.6	대푯값 삭제	대푯값을 삭제한다.
R1.7	Constraint 추가	Constraint를 추가한다.
R1.8	Constraint 삭제	Constraint를 삭제한다.
R1.9	대푯값 선택	검색하고 싶은 대푯값을 선택한다.
R2.1	Test Case 계산	Test Case를 계산한다.
R3.1	초기화	모든 입력과 출력 화면을 초기화한다.

### 1.3 Nonfunctional Requirements

Objective 변경으로 그에 따른 Nonfunctional Requirements도 변경

<b>Ver1.0</b>	우선순위 계산이 전체 계산보다 빠르게 출력
<b>Ver2.0</b>	선택한 대푯값의 Test Case가 화면에 빠르게 출력

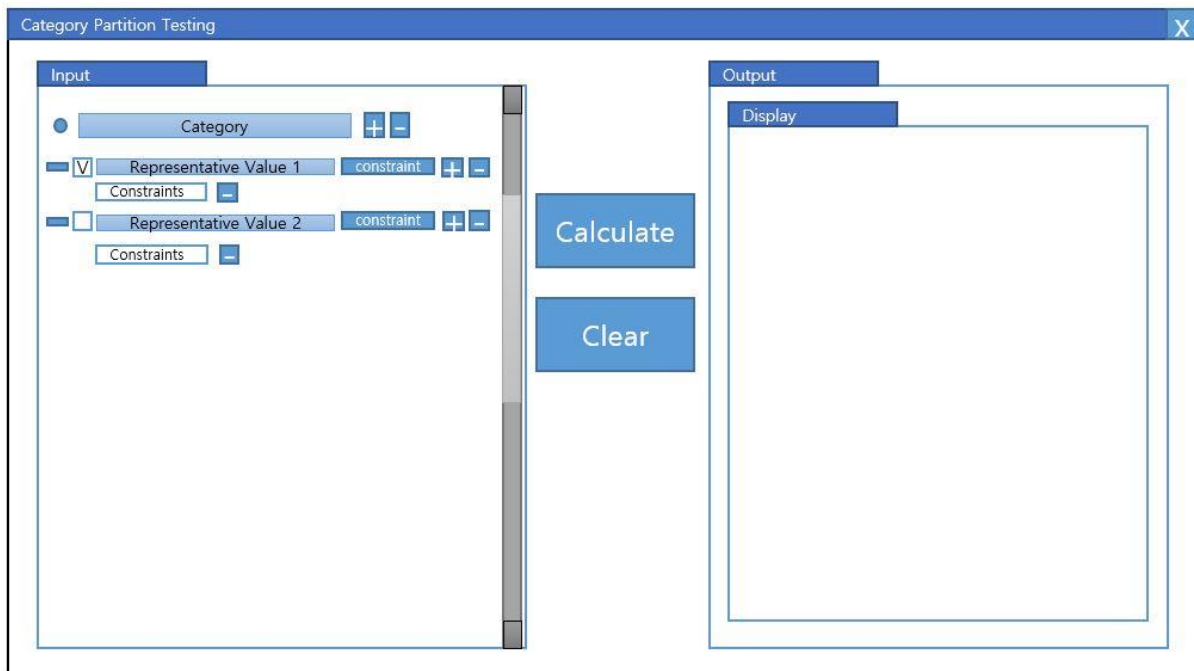
### 1.4 Terms in Glossary

Value의 의미가 불명확하여 대푯값(Representative Value)로 변경

<b>Ver1.0</b>	Value
<b>Ver2.0</b>	대푯값(Representative Value)

### 1.5 Implement Prototype

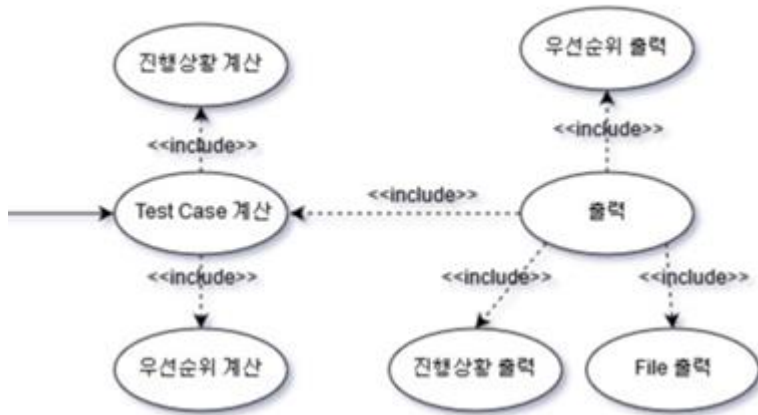
Objective의 변경으로 버튼, 화면 재배치



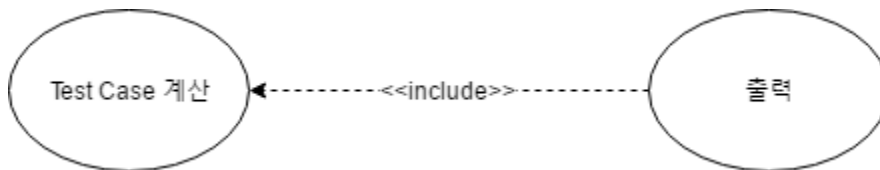
### 1.6 Use-Case Diagram

System 내부의 동작을 생략

• Ver1.0



• Ver2.0



2. Activity 2020. Synchronize Artifacts

Stage 1000 을 Ver3.0으로 수정

3. Activity 2031. Define Essential Use Cases

<b>Use Case</b>	Category 추가
<b>Actor</b>	Actor-based
<b>Purpose</b>	Category를 추가한다.
<b>Overview</b>	Category를 추가한다.
<b>Type</b>	Primary
<b>Cross Reference</b>	Functions : R1.1 Use Cases :
<b>Pre-Requisites</b>	N/A
<b>Typical Courses of Events</b>	(A) : Actor, (S) : System 1. (A) : Category 추가를 요청한다. 2. (S) : Category를 생성한다.

<b>Alternative Courses of Events</b>	N/A
<b>Exceptional Courses of Events</b>	Category 이름을 ‘Category #number’로 명명한다.

<b>Use Case</b>	Category 수정
<b>Actor</b>	Actor-based
<b>Purpose</b>	Category를 수정한다.
<b>Overview</b>	User에게 이름을 입력 받아서 기존의 Category를 수정한다.
<b>Type</b>	Primary
<b>Cross Reference</b>	Functions: R1.2 Use Cases:
<b>Pre-Requisites</b>	Category가 존재해야 한다.
<b>Typical Courses of Events</b>	(A) : Actor, (S) : System 1. (A) : Category를 선택하고 수정을 요청한다. 2. (A) : 이름을 입력한다. 2. (S) : 선택 된 Category의 이름을 입력 받은 이름으로 수정한다.
<b>Alternative Courses of Events</b>	N/A
<b>Exceptional Courses of Events</b>	이름을 입력하지 않을 경우 변경하지 않는다.

<b>Use Case</b>	Category 삭제
<b>Actor</b>	Actor-based
<b>Purpose</b>	Category를 삭제한다.
<b>Overview</b>	User가 요청하면 Category를 삭제한다.
<b>Type</b>	Primary
<b>Cross Reference</b>	Functions: R1.3 Use Cases:
<b>Pre-Requisites</b>	Category가 존재해야 한다.
<b>Typical Courses of Events</b>	(A) : Actor, (S) : System 1. (A) : Category를 선택하고 삭제를 요청한다. 2. (S) : Category 삭제 여부를 사용자에게 확인한다. 3. (A) : Category 삭제를 승인한다. 2. (S) : Category를 삭제한다.
<b>Alternative Courses of Events</b>	N/A
<b>Exceptional Courses of Events</b>	Category 삭제를 승인하지 않을 경우 삭제를 취소한다. Category에 Property Constraint가 존재하면 삭제하지 않는다.

<b>Use Case</b>	대폿값 추가
<b>Actor</b>	Actor-based
<b>Purpose</b>	대폿값을 추가한다.
<b>Overview</b>	대폿값을 추가한다.
<b>Type</b>	Primary
<b>Cross Reference</b>	Functions: R1.4 Use Cases:
<b>Pre-Requisites</b>	Category가 존재해야 한다.
<b>Typical Courses of Events</b>	(A) : Actor, (S) : System 1. (A) : Category를 선택하고 대폿값 추가를 요청한다. 2. (S) : 대폿값을 선택된 Category에 추가한다.
<b>Alternative Courses of Events</b>	N/A
<b>Exceptional Courses of Events</b>	대폿값의 이름을 RV #number로 명명한다.

<b>Use Case</b>	대폿값 수정
<b>Actor</b>	Actor-based
<b>Purpose</b>	대폿값을 수정한다.
<b>Overview</b>	User에게 대폿값의 이름을 입력 받아 대폿값을 수정한다.
<b>Type</b>	Primary
<b>Cross Reference</b>	Functions: R1.5 Use Cases:
<b>Pre-Requisites</b>	대폿값이 존재해야 한다.
<b>Typical Courses of Events</b>	(A) : Actor,(S) : System 1. (A) : 대폿값을 선택하고 수정을 요청한다. 2. (A) : 이름을 입력한다. 2. (S) : 선택 된 대폿값을 입력 받은 이름으로 수정한다
<b>Alternative Courses of Events</b>	N/A
<b>Exceptional Courses of Events</b>	이름을 입력하지 않을 경우 변경하지 않는다.

<b>Use Case</b>	대폿값 삭제
<b>Actor</b>	Actor-based
<b>Purpose</b>	대폿값을 삭제한다.
<b>Overview</b>	User가 대폿값 삭제를 요청하면 대폿값을 삭제한다.
<b>Type</b>	Primary
<b>Cross Reference</b>	Functions : R1.6 Use Cases :



<b>Pre-Requisites</b>	대폿값이 존재해야 한다.
<b>Typical Courses of Events</b>	(A) : Actor,(S) : System 1. (A) : 대폿값을 선택하고 삭제를 요청한다. 2. (S) : 대폿값 삭제 여부를 사용자에게 확인한다. 3. (A) : 대폿값 삭제를 승인한다. 2. (S) : 대폿값을 삭제한다
<b>Alternative Courses of Events</b>	N/A
<b>Exceptional Courses of Events</b>	대폿값 삭제를 승인하지 않을 경우 삭제를 취소한다.

<b>Use Case</b>	Constraint 추가
<b>Actor</b>	Actor-based
<b>Purpose</b>	Constraint를 추가한다.
<b>Overview</b>	User가 Constraint 추가를 요청하면 Constraint를 추가한다.
<b>Type</b>	Primary
<b>Cross Reference</b>	Functions : R1.7 Use Cases :
<b>Pre-Requisites</b>	대폿값이 존재해야 한다.
<b>Typical Courses of Events</b>	(A) : Actor, (S) : System 1. (A) : 대폿값을 선택하고 Constraint 추가를 요청한다. 2. (A) : Constraint의 이름과 속성을 입력한다. 3. (S) : 입력 받은 이름과 속성을 갖는 Constraint를 선택된 대폿값에 추가한다.
<b>Alternative Courses of Events</b>	N/A
<b>Exceptional Courses of Events</b>	Property 속성을 갖는 Constraint가 존재하지 않을 때, If Property 속성을 갖는 Constraint 추가요청이 들어올 경우 Constraint를 추가하지 않는다.

<b>Use Case</b>	Constraint 삭제
<b>Actor</b>	Actor-based
<b>Purpose</b>	Constraint를 삭제한다.
<b>Overview</b>	User가 Constraint 삭제를 요청하면 Constraint를 삭제한다.
<b>Type</b>	Primary
<b>Cross Reference</b>	Functions : R1.8 Use Cases :
<b>Pre-Requisites</b>	Constraint가 존재해야 한다.
<b>Typical Courses of Events</b>	(A) : Actor,(S) : System

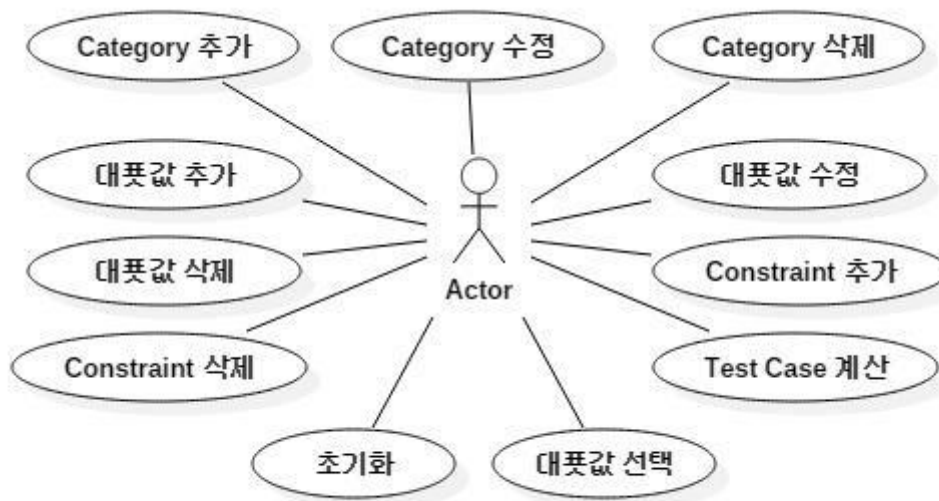
	1. (A) : Constraint을 선택하고 삭제를 요청한다. 2. (S) : Constraint를 삭제한다
<b>Alternative Courses of Events</b>	N/A
<b>Exceptional Courses of Events</b>	N/A

<b>Use Case</b>	대폿값 선택
<b>Actor</b>	Actor-based
<b>Purpose</b>	대폿값을 선택한다.
<b>Overview</b>	User가 대폿값을 선택, 해제하면 대폿값의 선택 여부를 수정한다,
<b>Type</b>	Primary
<b>Cross Reference</b>	Functions : R1.9 Use Cases :
<b>Pre-Requisites</b>	대폿값이 존재해야 한다.
<b>Typical Courses of Events</b>	(A) : Actor, (S) : System 1. (A) : 대폿값 선택/해제를 요청한다. 2. (S) : 대폿값의 선택 여부를 변경한다.
<b>Alternative Courses of Events</b>	N/A
<b>Exceptional Courses of Events</b>	N/A

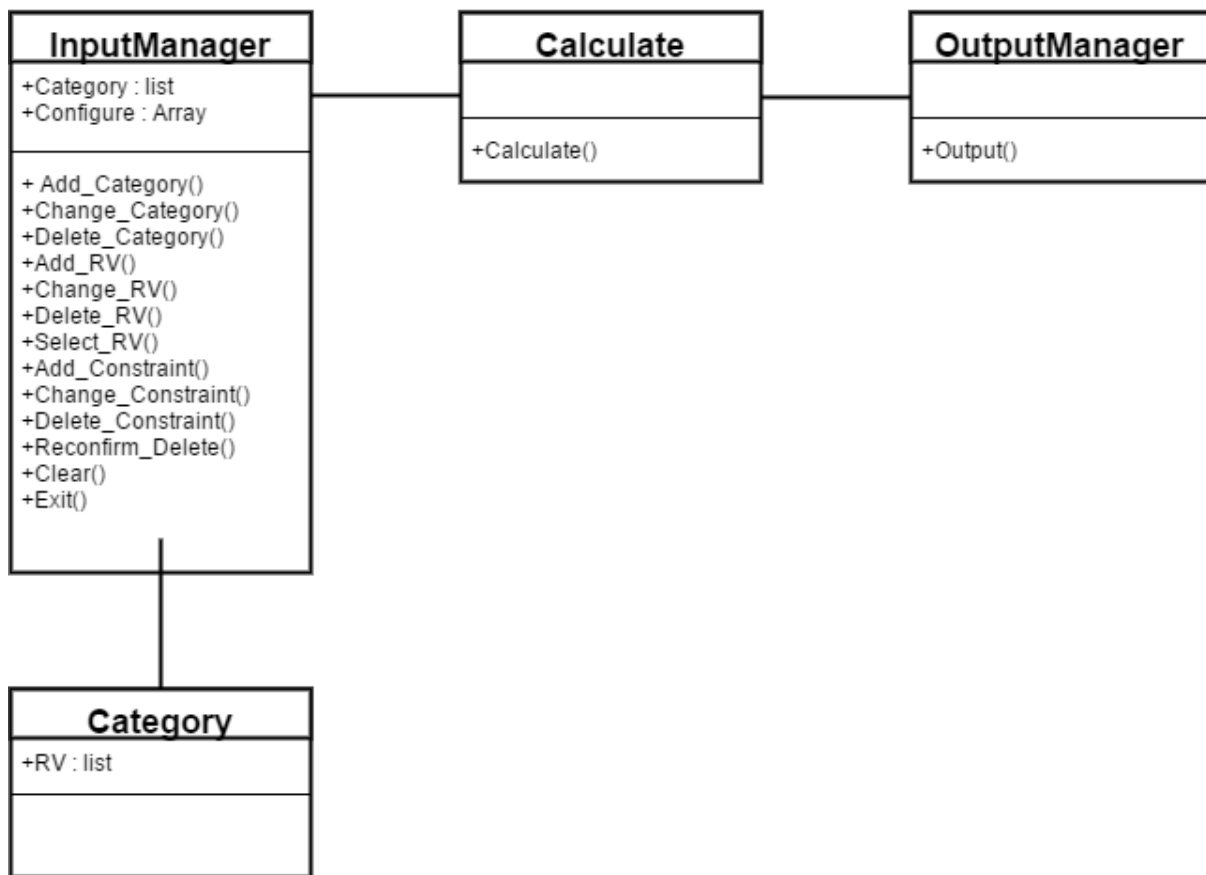
<b>Use Case</b>	Test Case 계산
<b>Actor</b>	Actor-based
<b>Purpose</b>	Test Case를 계산한다.
<b>Overview</b>	User가 계산을 요청하면 Test Case를 계산한다.
<b>Type</b>	Primary
<b>Cross Reference</b>	Functions : R2.1 Use Cases :
<b>Pre-Requisites</b>	Category, 대폿값, Constraint가 입력되어야 한다.
<b>Typical Courses of Events</b>	(A) : Actor, (S) : System 1. (A) : Test Case 계산을 요청한다. 2. (S) : Test Case를 계산한다. 3. (S) : 계산된 결과를 출력한다.
<b>Alternative Courses of Events</b>	N/A
<b>Exceptional Courses of Events</b>	선택된 대폿값이 있으면 선택된 대폿값에 대한 Test Case를 계산 후 출력한다.

<b>Use Case</b>	초기화
<b>Actor</b>	User
<b>Purpose</b>	모든 입력과 출력 화면을 초기화한다.
<b>Overview</b>	User가 초기화를 요청하면 입력과 출력화면을 초기화 한다.
<b>Type</b>	Primary
<b>Cross Reference</b>	Functions : R3.1 Use Cases :
<b>Pre-Requisites</b>	N/A
<b>Typical Courses of Events</b>	(A) : Actor, (S) : System 1. (A) : 초기화를 요청한다. 2. (S) : InputManager을 초기화한다. 3. (S) : 출력화면을 초기화한다.
<b>Alternative Courses of Events</b>	N/A
<b>Exceptional Courses of Events</b>	N/A

#### Activity 2032. Refine Use Case Diagrams



#### 4. Activity 2033. Define Domain Model



5. Activity 2034. Refine Glossary

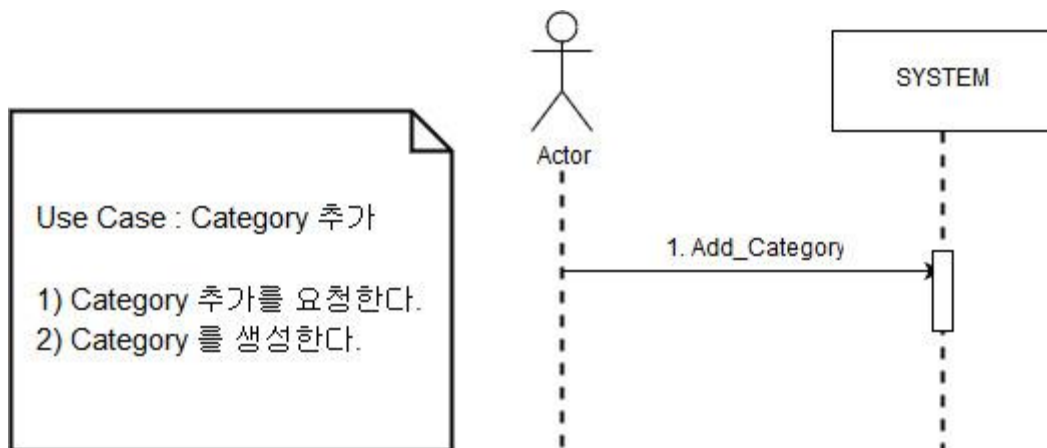
Glossary	Category	Description
InputManager	Class	사용자의 Input을 관리
Category	Class	User가 추가한 Category 객체
Calculate	Class	Test Case를 계산
OutputManager	Class	결과 출력
Category.RV:List	Attribute	Category 대푯값 리스트
Input.Manager.Category:List	Attribute	Category의 전체 리스트
InputManager.Configure:Array	Attribute	User가 선택한 대푯값의 배열
InputManager.Add_Category()	Operation	Category를 추가한다.
InputManager.Change_Category()	Operation	Category를 수정한다.
InputManager.Delete_Category()	Operation	Category를 삭제한다.
InputManager.Add_RV()	Operation	대푯값을 추가한다.
InputManager.Change_RV()	Operation	대푯값을 수정한다.
InputManager.Delete_RV()	Operation	대푯값을 삭제한다.
InputManager.Add_Constraint()	Operation	Constraint를 추가한다.
InputManager.Delete_Constraint()	Operation	Constraint를 삭제한다.

InputManager.Clear()	Operation	입력과 출력을 초기화한다.
Calculate.Calculate()	Operation	Test Case를 계산한다.
OutputManager.Output()	Operation	화면과 파일을 출력한다.

## 6. Activity 2035. Define System Sequence Diagrams

Use Case	Name of Actor-based Event	System Operations
Category 추가	Add_Category	Add_Category()
Category 수정	Change_Category	Change_Category()
Category 삭제	Delete_Category	Delete_Category()
대푯값 추가	Add_RV	Add_RV()
대푯값 수정	Change_RV	Change_RV()
대푯값 삭제	Delete_RV	Delete_RV()
Constraint 추가	Add_Constraint	Add_Constraint()
Constraint 삭제	Delete_Constraint	Delete_Constraint()
대푯값 선택	Select_Constraint	Select_Constraint()
Test Case 계산	Calculate	Calculate()
초기화	Clear	Clear()

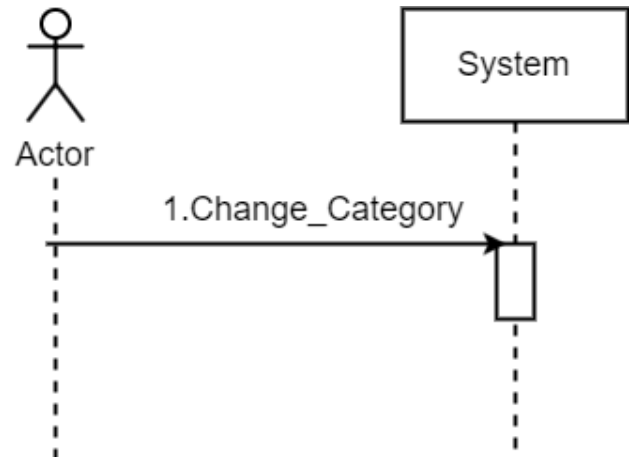
### 6.1 Category 추가



### 6.2 Category 수정

Use Case : Category 수정

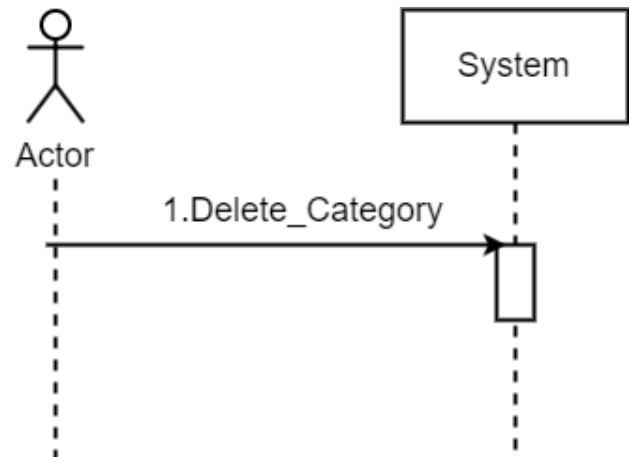
- 1) Category를 선택하고 수정을 요청한다.
- 2) 이름을 입력한다.
- 3) 선택된 Category의 이름을 입력 받은 이름으로 수정한다.



### 6.3 Category 삭제

Use Case : Category 삭제

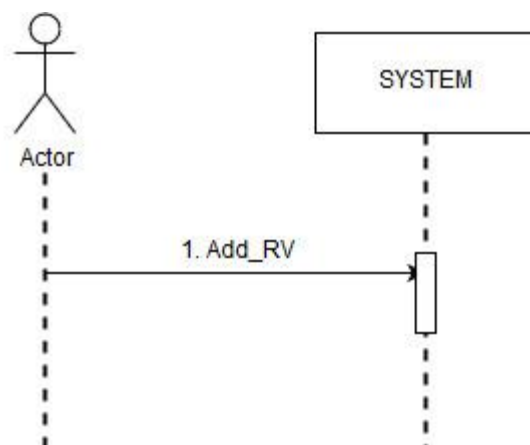
- 1) Category를 선택하고 삭제를 요청한다.
- 2) 카테고리 삭제 여부를 사용자에게 확인한다.
- 3) Category의 삭제를 승인한다.
- 4) Category를 삭제한다.



### 6.4 대푯값 추가

Use Case : 대푯값 추가

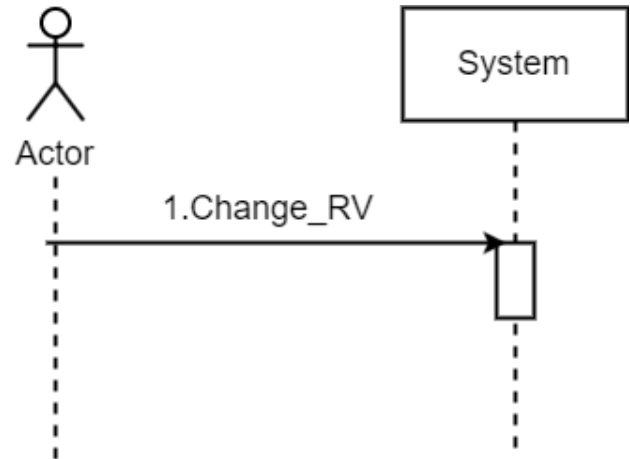
- 1) Category를 선택하고 대푯값 추가를 요청한다.
- 2) 대푯값을 선택된 Category에 추가한다.



6.5 대푯값 수정

Use Case : 대푯값 수정

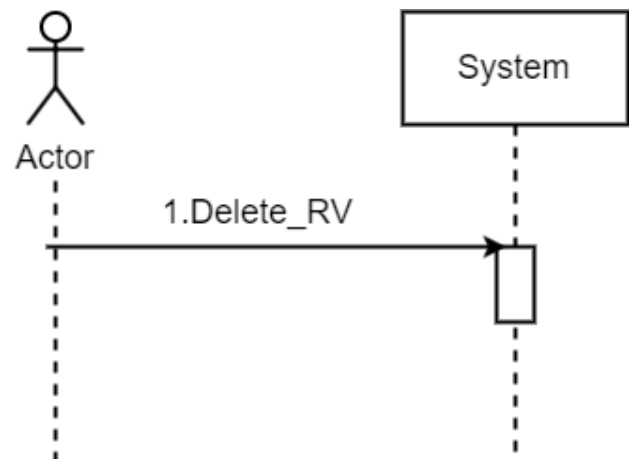
- 1) 대푯값을 선택하고 수정을 요청한다.
- 2) 이름을 입력한다.
- 3) 선택된 대푯값을 입력 받은 이름으로 수정한다.



6.6 대푯값 삭제

Use Case : 대푯값 삭제

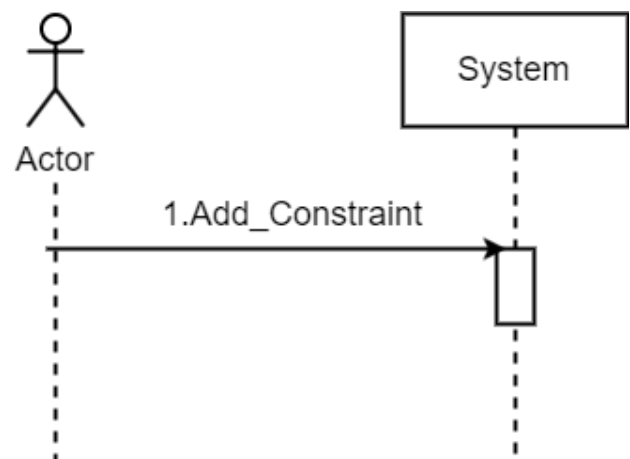
- 1) 대푯값을 선택하고 삭제를 요청한다.
- 2) 대푯값 삭제 여부를 사용자에게 확인한다.
- 3) 대푯값 삭제를 승인한다.
- 4) 대푯값을 삭제한다.



6.7 Constraint 추가

Use Case : Constraint 추가

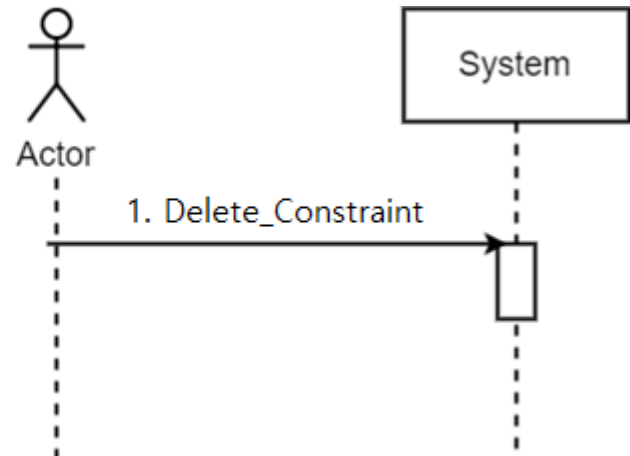
- 1) 대푯값 선택하고 Constraint 추가를 요청한다.
- 2) Constraint의 이름과 속성을 입력한다.
- 3) 입력 받은 이름과 속성을 갖는 Constraint를 선택된 대푯값에 추가한다.



6.8 Constraint 삭제

Use Case : Constraint 삭제

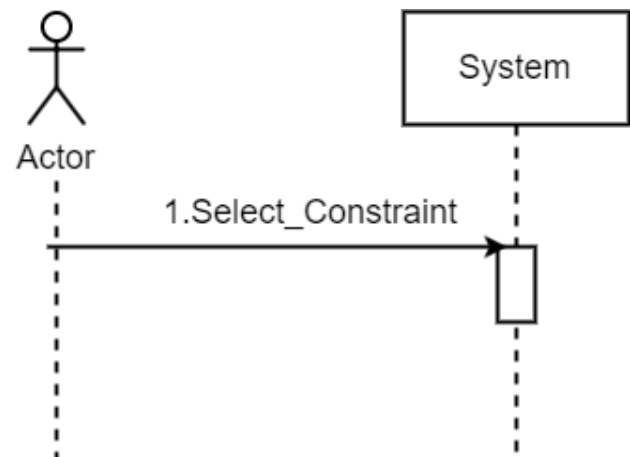
- 1) Constraint을 선택하고 삭제를 요청한다.
- 2) Constraint를 삭제한다



6.9 대푯값 선택

Use Case : 대푯값 선택

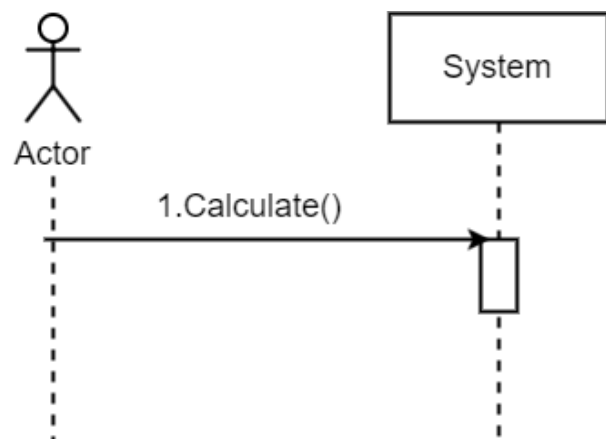
- 1) 대푯값의 선택/해제를 요청한다.
- 2) 대푯값의 선택 여부를 변경한다.



6.10 Test Case 계산

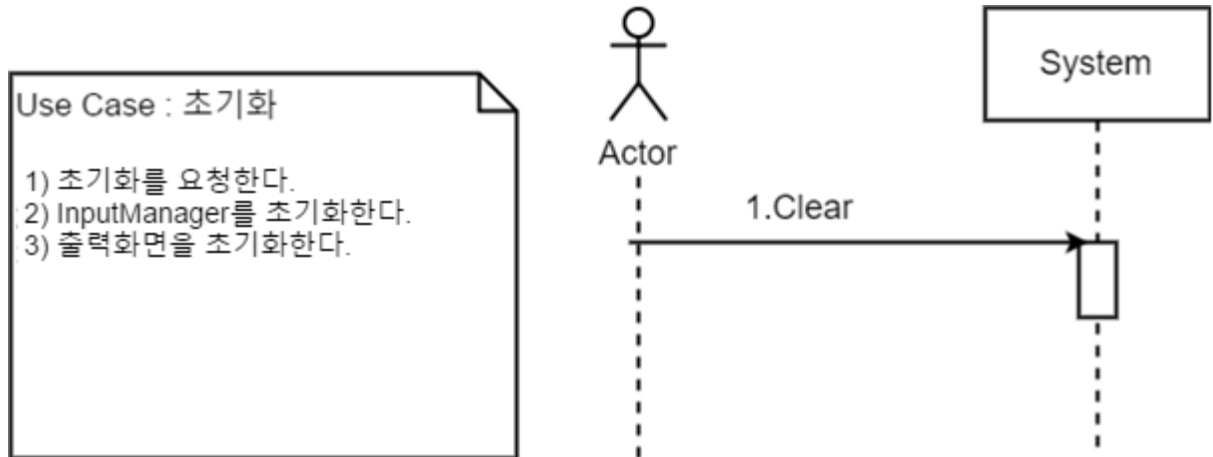
Use Case : Test Case 계산

- 1) Test Case 계산을 요청한다.
- 2) Test Case 를 계산한다.
- 3) 계산된 결과를 출력한다.





6.11 초기화



7. Activity 2036. Define Operation Contracts

Use Case	Name of Actor-based Event	System Operation
Category 추가	Add Category()	Add_Category()
Category 수정	Change Category()	Change_Category()
Category 삭제	Delete Category()	Delete_Category()
대푯값 추가	Add RV()	Add_RV()
대푯값 수정	Change RV()	Change_RV()
대푯값 삭제	Delete RV()	Delete_RV()
Constraint 추가	Add Constraint()	Add_Constraint()
Constraint 삭제	Delete Constraint()	Delete_Constraint()
대푯값 선택	Select Constraint()	Select_Constraint()
Test Case 계산	Calculate()	Calculate()
초기화	Clear()	Clear()

<b>Name</b>	Add_Category()
<b>Responsibility</b>	Category를 추가한다.
<b>Type</b>	System
<b>Cross Reference</b>	R1.1
<b>Exception</b>	Category 이름을 'Category #number'로 명명한다.

<b>Output</b>	Category 객체가 추가된 List
<b>Pre-Conditions</b>	N/A
<b>Post-Conditions</b>	Category가 생성되고 List에 업데이트된다.

<b>Name</b>	Change_Category()
<b>Responsibility</b>	Category의 이름을 수정한다.
<b>Type</b>	System
<b>Cross Reference</b>	R1.2
<b>Exception</b>	이름을 입력하지 않으면 변경하지 않는다.
<b>Output</b>	수정된 Category
<b>Pre-Conditions</b>	Category가 존재해야 한다.
<b>Post-Conditions</b>	Category의 이름이 수정된다.

<b>Name</b>	Delete_Category()
<b>Responsibility</b>	Category을 삭제한다.
<b>Type</b>	System
<b>Cross Reference</b>	R1.3
<b>Exception</b>	삭제가 승인되지 않을 경우 삭제를 취소한다.
<b>Output</b>	Category 객체가 삭제된 List
<b>Pre-Conditions</b>	Category가 존재해야 한다.
<b>Post-Conditions</b>	Category가 삭제된다.

<b>Name</b>	Add_RV()
<b>Responsibility</b>	대폿값을 추가한다.
<b>Type</b>	System
<b>Cross Reference</b>	R1.4
<b>Exception</b>	대폿값의 이름을 RV #number로 명명한다.
<b>Output</b>	대폿값이 추가된 Category
<b>Pre-Conditions</b>	Category가 존재해야 한다.
<b>Post-Conditions</b>	Category에 대폿값이 추가된다.

<b>Name</b>	Change_RV()
<b>Responsibility</b>	대폿값의 이름을 변경한다.
<b>Type</b>	System
<b>Cross Reference</b>	R1.5

<b>Exception</b>	이름을 입력하지 않으면 변경하지 않는다.
<b>Output</b>	이름이 변경된 대푯값
<b>Pre-Conditions</b>	대푯값이 존재해야 한다.
<b>Post-Conditions</b>	대푯값의 이름이 변경된다.

<b>Name</b>	Delete_RV()
<b>Responsibility</b>	대푯값을 삭제한다.
<b>Type</b>	System
<b>Cross Reference</b>	R1.6
<b>Exception</b>	삭제가 승인되지 않을 경우 삭제를 취소한다.
<b>Output</b>	대푯값이 삭제된 Category
<b>Pre-Conditions</b>	대푯값이 존재해야 한다.
<b>Post-Conditions</b>	Category에서 대푯값이 삭제된다.

<b>Name</b>	Add_Constraint()
<b>Responsibility</b>	Constraint를 추가한다.
<b>Type</b>	System
<b>Cross Reference</b>	R1.7
<b>Exception</b>	N/A
<b>Output</b>	Constraint가 추가된 대푯값
<b>Pre-Conditions</b>	대푯값이 존재해야 한다.
<b>Post-Conditions</b>	대푯값에 Constraint가 추가된다.

<b>Name</b>	Delete_Constraint()
<b>Responsibility</b>	Constraint를 삭제한다.
<b>Type</b>	System
<b>Cross Reference</b>	R1.8
<b>Exception</b>	삭제가 승인되지 않을 경우 삭제를 취소한다.
<b>Output</b>	Constraint가 삭제된 대푯값
<b>Pre-Conditions</b>	Constraint가 존재해야 한다.
<b>Post-Conditions</b>	대푯값에서 Constraint가 삭제된다.

<b>Name</b>	Select_RV()
<b>Responsibility</b>	RV의 선택 여부를 변경한다.
<b>Type</b>	System

<b>Cross Reference</b>	R1.9
<b>Exception</b>	N/A
<b>Output</b>	선택 여부가 변경된 Value
<b>Pre-Conditions</b>	대폿값이 존재해야 한다.
<b>Post-Conditions</b>	대폿값의 선택 여부가 변경된다.

<b>Name</b>	Calculate()
<b>Responsibility</b>	Test Case를 계산한다.
<b>Type</b>	System
<b>Cross Reference</b>	R2.1
<b>Exception</b>	N/A
<b>Output</b>	계산된 Test Case
<b>Pre-Conditions</b>	입력한 대폿값이 존재해야 한다.
<b>Post-Conditions</b>	화면에 선택된 대폿값의 Test Case가 출력되고, 전체 결과가 파일로 출력된다.

<b>Name</b>	Clear()
<b>Responsibility</b>	입력과 출력을 초기화한다.
<b>Type</b>	System
<b>Cross Reference</b>	R3.1
<b>Exception</b>	N/A
<b>Output</b>	N/A
<b>Pre-Conditions</b>	N/A.
<b>Post-Conditions</b>	입력과 화면 출력이 모두 삭제된다..

## 8. Activity 2038. Refine System Test Case

### 8.1 Functional Requirements System Test Case

Test Number	Test 항목	Description	Use Case	System Function
1-1	Category 추가 Test	Category를 추가 할 때 Category #number로 추가되는지 확인한다.	Category 추가	R1.1
2-1	Category 수정 Test	Category의 이름이 수정되는지 확인한다.	Category 수정	R1.2
2-2	Category 수정 Test	Category의 수정을 요청하고 이름을 입력하지 않았을 때 기존의 이름인지 확인한다.	Category 수정	R1.2
3-1	Category 삭제 Test	Category의 삭제를 요청하고 승인하면	Category 삭제	R1.3

		Category가 삭제되는지 확인한다.		
3-2	Category 삭제 Test	Category의 삭제를 요청하고 승인하지 않으면 Category가 유지되는지 확인한다.	Category 삭제	R1.3
4-1	대푯값 추가 Test	대푯값을 추가 할 때 RV #number로 추가되는지 확인한다.	대푯값 추가	R1.4
5-1	대푯값 수정 Test	대푯값의 이름이 수정되는지 확인한다.	대푯값 수정	R1.5
5-2	대푯값 수정 Test	대푯값의 수정을 요청하고 이름을 입력하지 않았을 때 기존의 이름인지 확인한다.	대푯값 수정	R1.5
6-1	대푯값 삭제 Test	대푯값의 삭제를 요청하고 승인하면 대푯값이 삭제되는지 확인한다.	대푯값 삭제	R1.6
6-1	대푯값 삭제 Test	대푯값의 삭제를 요청하고 승인하지 않으면 대푯값 유지되는지 확인한다.	대푯값 삭제	R1.6
7-1	Constraint 추가 Test	Constraint의 이름과 속성을 입력하고 추가 할 때 입력된 이름과 속성대로 추가되는지 확인한다.	Constraint 추가	R1.7
7-2	Constraint 추가 Test	Property 속성을 갖는 Constraint가 존재하지 않을 때, If Property 속성을 갖는 Constraint 추가요청이 들어올 경우 Constraint를 추가하지 않는지 확인한다.	Constraint 추가	R1.7
8-1	Constraint 삭제 Test	Constraint의 삭제를 요청하고 승인하면 Constraint가 삭제되는지 확인한다.	Constraint 삭제	R1.8
9-1	대푯값 선택 Test	대푯값을 선택하고 Test Case를 계산했을 때 화면에 선택한 대푯값의 Test Case가 출력되는지 확인한다.	대푯값 선택	R1.9
10-1	Test Case 계산 Test	Test Case를 예상 결과값과 비교한다.	Test Case 계산	R2.1
11-1	초기화 Test	초기화 요청 시 모든 입력과 출력이 삭제되는지 확인한다.	초기화	R3.1

**8.2 Nonfunctional Requirements Test Case**

Category	Test Case
A. Interface	본 프로그램을 처음 접하는 사람들을 대상으로 각각 3개의 대푯값과 3개의 카테고리리를 작성하고 Test Case를 출력하는 데까지 10이내로 소요되는지 확인한다.
B. Display	선택한 대푯값의 Test Case를 1초 이내로 출력되는지 확인한다.

**9. Activity 2039. Analyze Traceability Analysis**

System Function		Use Case	Operation
R1.1	Category 추가	1.Category 추가	Add_Category()
R1.2	Category 수정	2.Category 수정	Change_Category()
R1.3	Category 삭제	3.Category 삭제	Delete_Category()
R1.4	대푯값 추가	4.대푯값 추가	Add_RV()
R1.5	대푯값 수정	5.대푯값 수정	Change_RV()
R1.6	대푯값 삭제	6.대푯값 삭제	Delete_RV()
R1.7	Constraint 추가	7.Constraint 추가	Add_Constraint()
R1.8	Constraint 삭제	8.Constraint 삭제	Delete_Constraint()
R1.9	대푯값 선택	9.대푯값 선택	Select_Constraint()
R2.1	Test Case 계산	10.Test Case 계산	Calculate()
R3.1	초기화	11.초기화	Clear()