

# "Priority" Category Partitioning Testing Tool

OOPT stage 2050 & 2060

Construct & Testing

---

Project Team

T4

201311265 김상원

201210194 김정환

201311269 김제헌

201311297 이상명

Date

2017-05-31

## 목차

2050 Construct.....	3
Activity 2051. Implement Class & Methods Definitions.....	3
Activity 2052. Implement Windows .....	22
Activity 2055. Write Unit Test Code .....	31
2060 Test .....	34
Activity 2061. Unit Testing.....	34
Activity 2063. System Testing.....	35

## 2050 Construct

## Activity 2051. Implement Class &amp; Methods Definitions

<b>Type</b>	Class
<b>Name</b>	MainSystem
<b>Purpose</b>	전반적인 프로그램의 수행을 제어하는 클래스
<b>Overview (Class)</b>	폴더선택, 파일분석, 화면출력 등을 제어한다.
<b>Cross Reference</b>	Functions: R1.1 R1.2 R1.3
<b>Input (Method)</b>	-
<b>Output (Method)</b>	-
<b>Abstract Operation(Method)</b>	-
<b>Exceptional Courses of Events</b>	N/A

<b>Type</b>	Class
<b>Name</b>	FileManager
<b>Purpose</b>	프로그램의 상황을 파일 형태로 저장하는 클래스
<b>Overview (Class)</b>	Specification 정보를 파일 형태로 저장, 불러오기 등을 한다. 최근 파일 목록을 업데이트 한다.
<b>Cross Reference</b>	Functions: R1.1 R1.2 R5
<b>Input (Method)</b>	-
<b>Output (Method)</b>	-
<b>Abstract Operation(Method)</b>	-
<b>Exceptional Courses of Events</b>	N/A

<b>Type</b>	Method
<b>Name</b>	newSpecification()
<b>Purpose</b>	새로운 .yzb 파일을 만든다.
<b>Overview (Class)</b>	전달 받은 이름으로 새로운 .yzb 파일을 만든다. 성공 시 0을, 실패 시 1을 반환한다.

<b>Cross Reference</b>	Functions: R1.1
<b>Input (Method)</b>	String newFileName
<b>Output (Method)</b>	Integer
<b>Abstract Operation(Method)</b>	-
<b>Exceptional Courses of Events</b>	N/A

<b>Type</b>	Method
<b>Name</b>	loadSpecification()
<b>Purpose</b>	저장된 .yzb 파일을 불러온다.
<b>Overview (Class)</b>	전달 받은 filePath의 .yzb 파일을 불러온다. 성공 시 0을, 실패 시 1을 반환한다.
<b>Cross Reference</b>	Functions: R1.2
<b>Input (Method)</b>	String filePath
<b>Output (Method)</b>	Integer
<b>Abstract Operation(Method)</b>	-
<b>Exceptional Courses of Events</b>	전달 받은 파일경로가 유효하지 않은 경우에는 예외 처리한다.

<b>Type</b>	Method
<b>Name</b>	saveSpecification()
<b>Purpose</b>	현재 Specification 의 정보를 .yzb 파일로 저장한다.
<b>Overview (Class)</b>	현재 열려있는 .yzb 파일에 Specification 정보를 저장한다. 성공 시 0을, 실패 시 1을 반환한다.
<b>Cross Reference</b>	Functions: R5
<b>Input (Method)</b>	void
<b>Output (Method)</b>	Integer
<b>Abstract Operation(Method)</b>	-
<b>Exceptional Courses of Events</b>	N/A

<b>Type</b>	Method
<b>Name</b>	saveSpecification()
<b>Purpose</b>	현재 Specification 의 정보를 다른 이름의 .yzb 파일로 저장한다.
<b>Overview (Class)</b>	전달받은 파일 경로의 새로운 .yzb 파일로 현재 Specification 정보를 저장한다. 성공 시 0을, 실패 시 1을 반환한다.
<b>Cross Reference</b>	Functions: R5
<b>Input (Method)</b>	String filePath
<b>Output (Method)</b>	Integer
<b>Abstract Operation(Method)</b>	-
<b>Exceptional Courses of Events</b>	N/A

<b>Type</b>	Method
<b>Name</b>	updateRecentList()
<b>Purpose</b>	최근 파일 목록을 업데이트 한다.
<b>Overview (Class)</b>	전달받은 파일 경로를 최근 파일 목록에 추가한다. 성공 시 0을, 실패 시 1을 반환한다.
<b>Cross Reference</b>	Functions: R1.2
<b>Input (Method)</b>	String filePath
<b>Output (Method)</b>	Integer
<b>Abstract Operation(Method)</b>	-
<b>Exceptional Courses of Events</b>	N/A

<b>Type</b>	Method
<b>Name</b>	getPath()
<b>Purpose</b>	현재 열려 있는 .yzb 파일의 경로를 반환한다.
<b>Overview (Class)</b>	현재 .yzb 파일의 경로를 String으로 반환한다.
<b>Cross Reference</b>	Functions: R1.1 R1.2

<b>Input (Method)</b>	void
<b>Output (Method)</b>	String
<b>Abstract Operation(Method)</b>	-
<b>Exceptional Courses of Events</b>	N/A

<b>Type</b>	Class
<b>Name</b>	Specification
<b>Purpose</b>	Category 객체 배열, 테스트 케이스 맵, property 테이블 정보를 소유한 클래스
<b>Overview (Class)</b>	
<b>Cross Reference</b>	Functions: R2.1 R2.2 R2.3 R2.4 R3 R4
<b>Input (Method)</b>	-
<b>Output (Method)</b>	-
<b>Abstract Operation(Method)</b>	-
<b>Exceptional Courses of Events</b>	N/A

<b>Type</b>	Method
<b>Name</b>	generateTestCases()
<b>Purpose</b>	테스트 케이스를 생성한다.
<b>Overview (Class)</b>	테스트 케이스를 생성하여 <String, Integer> 형태의 Map에 저장한다. 성공 시 0을 반환한다.
<b>Cross Reference</b>	Functions: R3 Use Cases:
<b>Input (Method)</b>	void
<b>Output (Method)</b>	Integer
<b>Abstract Operation(Method)</b>	-
<b>Exceptional Courses of Events</b>	N/A

<b>Type</b>	Method
<b>Name</b>	export2excel()
<b>Purpose</b>	테스트 케이스를 엑셀(.xls) 파일로 export 한다.
<b>Overview (Class)</b>	우선순위 점수 순으로 정렬하여 엑셀에 저장한다. 성공 시 0을 반환한다.
<b>Cross Reference</b>	Functions: R4
<b>Input (Method)</b>	void
<b>Output (Method)</b>	Integer
<b>Abstract Operation(Method)</b>	-
<b>Exceptional Courses of Events</b>	N/A

<b>Type</b>	Method
<b>Name</b>	getCategory()
<b>Purpose</b>	인덱스에 해당하는 Category를 반환한다.
<b>Overview (Class)</b>	Specification이 소유한 Category 배열에서 전달받은 categoryIndex 에 해당하는 Category 객체를 반환한다.
<b>Cross Reference</b>	Functions: R2.1
<b>Input (Method)</b>	Integer categoryIndex
<b>Output (Method)</b>	Category
<b>Abstract Operation(Method)</b>	-
<b>Exceptional Courses of Events</b>	N/A

<b>Type</b>	Method
<b>Name</b>	setCategory()
<b>Purpose</b>	인덱스에 해당하는 Category 객체의 이름을 변경한다.
<b>Overview (Class)</b>	Specification이 소유한 Category 객체 배열 중에서 전달받은 categoryIndex의 해당하는 Category 객체의 이름을 전달받은 categoryName으로 바꾼다. 성공 시 0을 반환한다.

<b>Cross Reference</b>	Functions: R2.1
<b>Input (Method)</b>	Integer categoryIndex, String categoryName
<b>Output (Method)</b>	Integer
<b>Abstract Operation(Method)</b>	-
<b>Exceptional Courses of Events</b>	전달받은 categoryName과 같은 이름의 Category 객체가 이미 있는 경우 예외처리 경고 메시지를 띄운다.

<b>Type</b>	Method
<b>Name</b>	addCategory()
<b>Purpose</b>	새로운 Category 객체를 추가한다.
<b>Overview (Class)</b>	Specification이 소유한 Category 객체 배열에 전달받은 categoryName 이름의 새로운 객체를 추가한다. 성공 시 0을 반환한다.
<b>Cross Reference</b>	Functions: R2.1
<b>Input (Method)</b>	String categoryName
<b>Output (Method)</b>	Integer
<b>Abstract Operation(Method)</b>	-
<b>Exceptional Courses of Events</b>	N/A

<b>Type</b>	Method
<b>Name</b>	delCategory()
<b>Purpose</b>	인덱스에 해당하는 Category 객체를 삭제한다.
<b>Overview (Class)</b>	Specification이 소유한 Category 객체 배열 중에서 전달받은 categoryIndex에 해당하는 Category 객체를 삭제한다. 성공 시 0을 반환한다.
<b>Cross Reference</b>	Functions: R2.1 R2.2 R2.3 R2.4
<b>Input (Method)</b>	Integer categoryIndex
<b>Output (Method)</b>	Integer
<b>Abstract Operation(Method)</b>	-



<b>Exceptional Courses of Events</b>	N/A
--------------------------------------	-----

<b>Type</b>	Method
<b>Name</b>	updateData()
<b>Purpose</b>	Property가 삭제되었을 때 삭제된 Property를 if-property로 설정해 놓았을 경우, 해당 if-property를 삭제한다.
<b>Overview (Class)</b>	
<b>Cross Reference</b>	Functions: R2.3 R2.4
<b>Input (Method)</b>	Void
<b>Output (Method)</b>	void
<b>Abstract Operation(Method)</b>	-
<b>Exceptional Courses of Events</b>	N/A

<b>Type</b>	Class
<b>Name</b>	Category
<b>Purpose</b>	Representative Value 배열을 저장하는 클래스
<b>Overview (Class)</b>	이름과 Representative Value들을 갖는다. Category 이름을 변경하기, Representative Value 추가, 삭제, 수정을 한다.
<b>Cross Reference</b>	Functions: R2.1 R2.2 R2.3 R2.4
<b>Input (Method)</b>	-
<b>Output (Method)</b>	-
<b>Abstract Operation(Method)</b>	-
<b>Exceptional Courses of Events</b>	N/A

<b>Type</b>	Method
<b>Name</b>	setCategoryName()
<b>Purpose</b>	Category 이름을 정한다.

<b>Overview (Class)</b>	전달받은 categoryName으로 이름을 정한다. 성공 시 0을 반환한다.
<b>Cross Reference</b>	Functions: R2.1
<b>Input (Method)</b>	String categoryName
<b>Output (Method)</b>	Integer
<b>Abstract Operation(Method)</b>	-
<b>Exceptional Courses of Events</b>	N/A

<b>Type</b>	Method
<b>Name</b>	getCategoryName()
<b>Purpose</b>	Category 이름을 반환한다.
<b>Overview (Class)</b>	Category 이름을 String으로 반환한다.
<b>Cross Reference</b>	Functions: R2.1
<b>Input (Method)</b>	void
<b>Output (Method)</b>	String
<b>Abstract Operation(Method)</b>	-
<b>Exceptional Courses of Events</b>	N/A

<b>Type</b>	Method
<b>Name</b>	getRepresentativeValue()
<b>Purpose</b>	인덱스에 해당하는 Representative Value를 반환한다.
<b>Overview (Class)</b>	Category가 소유한 RepresentativeValue 객체 배열 중에서 전달받은 representativeIndex에 해당하는 RepresentativeValue 객체를 반환한다.
<b>Cross Reference</b>	Functions: R2.2
<b>Input (Method)</b>	Integer representativeIndex
<b>Output (Method)</b>	RepresentativeValue
<b>Abstract Operation(Method)</b>	-
<b>Exceptional Courses of Events</b>	N/A

<b>Events</b>	
---------------	--

<b>Type</b>	Method
<b>Name</b>	setRepresentativeValue()
<b>Purpose</b>	인덱스에 해당하는 Representative Value의 이름을 수정한다.
<b>Overview (Class)</b>	Category가 소유한 RepresentativeValue 객체 배열 중에서 전달받은 representativeIndex에 해당하는 RepresentativeValue 객체의 이름을 전달받은 representativeValueName으로 수정한다. 성공 시 0을 반환한다.
<b>Cross Reference</b>	Functions: R2.2
<b>Input (Method)</b>	Integer representativeValueIndex, String representativeValueName
<b>Output (Method)</b>	Integer
<b>Abstract Operation(Method)</b>	-
<b>Exceptional Courses of Events</b>	N/A

<b>Type</b>	Method
<b>Name</b>	addRepresentativeValue()
<b>Purpose</b>	Representative Value를 추가한다.
<b>Overview (Class)</b>	Category가 소유한 RepresentativeValue 객체 배열에 전달받은 representativeValueName을 이름으로 갖는 새로운 RepresentativeValue 객체를 추가한다. 성공 시 0을 반환한다.
<b>Cross Reference</b>	Functions: R2.2
<b>Input (Method)</b>	String representativeValueName
<b>Output (Method)</b>	Integer
<b>Abstract Operation(Method)</b>	-
<b>Exceptional Courses of Events</b>	N/A

<b>Type</b>	Method
<b>Name</b>	delRepresentativeValue()
<b>Purpose</b>	인덱스에 해당하는 Representative Value 를 삭제한다.
<b>Overview (Class)</b>	Category가 소유한 RepresentativeValue 객체 배열 중에서 전달받은 representativeIndex에 해당하는 RepresentativeValue 객체를 삭제한다. 성공 시 0을 반환한다.
<b>Cross Reference</b>	Functions: R2.2 R2.3 R2.4
<b>Input (Method)</b>	Integer representativeValueIndex
<b>Output (Method)</b>	Integer
<b>Abstract Operation(Method)</b>	-
<b>Exceptional Courses of Events</b>	N/A

<b>Type</b>	Class
<b>Name</b>	RepresentativeValue
<b>Purpose</b>	property 와 if-property, single, error constraints와 priority를 갖는 클래스
<b>Overview (Class)</b>	이름과 property 배열, if-property 배열, single 및 error 여부와 priority를 갖는다. Property와 if-property를 추가, 삭제, 수정하고 single 및 error constraint를 설정한다.
<b>Cross Reference</b>	Functions: R2.2 R2.3 R2.4 R2.5 R2.6
<b>Input (Method)</b>	-
<b>Output (Method)</b>	-
<b>Abstract Operation(Method)</b>	-
<b>Exceptional Courses of Events</b>	N/A

<b>Type</b>	Method
<b>Name</b>	getNumberOfPropertyConstraints()

<b>Purpose</b>	property 개수를 반환한다.
<b>Overview (Class)</b>	RepresentativeValue가 소유한 Property 배열의 크기를 반환한다.
<b>Cross Reference</b>	Functions: R2.3
<b>Input (Method)</b>	Void
<b>Output (Method)</b>	Integer
<b>Abstract Operation(Method)</b>	-
<b>Exceptional Courses of Events</b>	N/A

<b>Type</b>	Method
<b>Name</b>	getNumberOfPropertyConstraints()
<b>Purpose</b>	If-property 개수를 반환한다.
<b>Overview (Class)</b>	RepresentativeValue가 소유한 If-Property 배열의 크기를 반환한다.
<b>Cross Reference</b>	Functions: R2.4
<b>Input (Method)</b>	Void
<b>Output (Method)</b>	Integer
<b>Abstract Operation(Method)</b>	-
<b>Exceptional Courses of Events</b>	N/A

<b>Type</b>	Method
<b>Name</b>	setRepresentativeValueName()
<b>Purpose</b>	이름을 설정한다.
<b>Overview (Class)</b>	property가 있으면 property의 representative value 이름 배열에서 현재 이름을 삭제하고 전달받은 새로운 이름인 representativeValueName을 추가한다. 전달받은 representativeValueName으로 이름을 설정한다. 성공 시 0을 반환한다.
<b>Cross Reference</b>	Functions: R2.2
<b>Input (Method)</b>	String representativeValueName

<b>Output (Method)</b>	Integer
<b>Abstract Operation(Method)</b>	-
<b>Exceptional Courses of Events</b>	N/A

<b>Type</b>	Method
<b>Name</b>	getRepresentativeValueName()
<b>Purpose</b>	이름을 반환한다.
<b>Overview (Class)</b>	RepresentativeValue의 이름을 반환한다.
<b>Cross Reference</b>	Functions: R2.2
<b>Input (Method)</b>	Void
<b>Output (Method)</b>	String
<b>Abstract Operation(Method)</b>	-
<b>Exceptional Courses of Events</b>	N/A

<b>Type</b>	Method
<b>Name</b>	setProperty()
<b>Purpose</b>	인덱스에 해당하는 property의 이름을 변경한다.
<b>Overview (Class)</b>	RepresentativeValue가 소유한 Property 배열 중에서 전달받은 propertyIndex에 해당하는 Property 객체의 이름을 전달받은 propertyName으로 수정한다. 성공 시 0을 반환한다.
<b>Cross Reference</b>	Functions: R2.3
<b>Input (Method)</b>	Integer propertyIndex, String propertyName
<b>Output (Method)</b>	Integer
<b>Abstract Operation(Method)</b>	-
<b>Exceptional Courses of Events</b>	중복된 이름을 전달받은 경우 예외처리한다.

<b>Type</b>	Method
-------------	--------

<b>Name</b>	addProperty()
<b>Purpose</b>	Property를 추가한다.
<b>Overview (Class)</b>	Specification이 소유한 property table에 전달받은 propertyName을 이름으로 갖는 property 가 있으면 해당 property를 RepresentativeValue가 소유한 Property 배열에 전달받은 propertyName을 이름으로 갖는 새로운 Property 객체를 추가한다. 없다면 property table에 전달받은 propertyName을 이름으로 갖는 새로운 Property 객체를 생성하여 추가하고 이를 RepresentativeValue가 소유한 Property 배열에 추가한다. 성공 시 0을 반환한다.
<b>Cross Reference</b>	Functions: R2.3
<b>Input (Method)</b>	String propertyName
<b>Output (Method)</b>	Integer
<b>Abstract Operation(Method)</b>	-
<b>Exceptional Courses of Events</b>	N/A

<b>Type</b>	Method
<b>Name</b>	delProperty()
<b>Purpose</b>	Property를 삭제한다.
<b>Overview (Class)</b>	RepresentativeValue가 소유한 Property 배열 중에서 전달받은 propertyIndex에 해당하는 Property 객체를 삭제한다. 성공 시 0을 반환한다.
<b>Cross Reference</b>	Functions: R2.3 R2.4
<b>Input (Method)</b>	Integer propertyIndex
<b>Output (Method)</b>	Integer
<b>Abstract Operation(Method)</b>	-
<b>Exceptional Courses of Events</b>	N/A

<b>Type</b>	Method
<b>Name</b>	setIfProperty()
<b>Purpose</b>	If-Property의 이름을 수정한다.
<b>Overview (Class)</b>	Specification이 소유한 property table에서, 전달받은 ifPropertyName을 이름으로 갖는 Property 객체를 RepresentativeValue가 소유한 If-Property 배열 중에서 전달받은 propertyIndex에 넣는다. 성공 시 0을 반환한다.
<b>Cross Reference</b>	Functions: R2.4
<b>Input (Method)</b>	Integer ifPropertyIndex, String ifPropertyName
<b>Output (Method)</b>	Integer
<b>Abstract Operation(Method)</b>	-
<b>Exceptional Courses of Events</b>	ifPropertyName이 property table에 없을 경우 예외 처리한다.

<b>Type</b>	Method
<b>Name</b>	addIfProperty()
<b>Purpose</b>	If-Property를 추가한다.
<b>Overview (Class)</b>	Specification이 소유한 property table에서, 전달받은 ifPropertyName을 이름으로 갖는 Property 객체를 RepresentativeValue가 소유한 If-Property 배열에 추가한다. 성공 시 0을 반환한다.
<b>Cross Reference</b>	Functions: R2.4
<b>Input (Method)</b>	String ifPropertyName
<b>Output (Method)</b>	Integer
<b>Abstract Operation(Method)</b>	-
<b>Exceptional Courses of Events</b>	ifPropertyName이 property table에 없을 경우 예외 처리한다.

<b>Type</b>	Method
<b>Name</b>	delfProperty()
<b>Purpose</b>	If-Property를 삭제한다.



<b>Overview (Class)</b>	RepresentativeValue가 소유한 If-Property 배열 중에서 전달받은 propertyIndex에 해당하는 Property 객체를 삭제한다. 성공 시 0을 반환한다.
<b>Cross Reference</b>	Functions: R2.4
<b>Input (Method)</b>	Integer ifPropertyIndex
<b>Output (Method)</b>	Integer
<b>Abstract Operation(Method)</b>	-
<b>Exceptional Courses of Events</b>	N/A

<b>Type</b>	Method
<b>Name</b>	setSingleError()
<b>Purpose</b>	single constraint 또는 error constraint를 설정한다.
<b>Overview (Class)</b>	singleErrorConstraint 에 전달받은 1 또는 2를 설정한다. (1=single 2=error) 성공 시 0을 반환한다.
<b>Cross Reference</b>	Functions: R2.5
<b>Input (Method)</b>	Integer singleError
<b>Output (Method)</b>	Integer
<b>Abstract Operation(Method)</b>	-
<b>Exceptional Courses of Events</b>	N/A

<b>Type</b>	Method
<b>Name</b>	setPriorityRank()
<b>Purpose</b>	Priority를 설정한다.
<b>Overview (Class)</b>	전달받은 값으로 priority를 설정한다. 성공 시 0을 반환한다.
<b>Cross Reference</b>	Functions: R2.6
<b>Input (Method)</b>	Integer priorityRank
<b>Output (Method)</b>	Integer

<b>Abstract Operation(Method)</b>	-
<b>Exceptional Courses of Events</b>	N/A

<b>Type</b>	Method
<b>Name</b>	getProperty()
<b>Purpose</b>	인덱스에 해당하는 Property를 반환한다.
<b>Overview (Class)</b>	Property 배열에서 propertyIndex에 해당하는 Property 객체를 반환한다.
<b>Cross Reference</b>	Functions: R2.3
<b>Input (Method)</b>	Integer propertyIndex
<b>Output (Method)</b>	Property
<b>Abstract Operation(Method)</b>	-
<b>Exceptional Courses of Events</b>	N/A

<b>Type</b>	Method
<b>Name</b>	getIfProperty()
<b>Purpose</b>	인덱스에 해당하는 If-Property를 반환한다.
<b>Overview (Class)</b>	If-property 배열에서 ifPropertyIndex에 해당하는 Property 객체를 반환한다.
<b>Cross Reference</b>	Functions: R2.4
<b>Input (Method)</b>	Integer ifPropertyIndex
<b>Output (Method)</b>	Property
<b>Abstract Operation(Method)</b>	-
<b>Exceptional Courses of Events</b>	N/A

<b>Type</b>	Method
<b>Name</b>	getSingleError()

<b>Purpose</b>	single 또는 error 여부를 반환한다.
<b>Overview (Class)</b>	singleErrorConstraint을 반환한다. (single=1, error=2)
<b>Cross Reference</b>	Functions: R2.5
<b>Input (Method)</b>	Void
<b>Output (Method)</b>	Integer
<b>Abstract Operation(Method)</b>	-
<b>Exceptional Courses of Events</b>	N/A

<b>Type</b>	Method
<b>Name</b>	getPriorityRank()
<b>Purpose</b>	우선순위 점수를 반환한다.
<b>Overview (Class)</b>	Priority를 반환한다.
<b>Cross Reference</b>	Functions: R2.6
<b>Input (Method)</b>	Void
<b>Output (Method)</b>	Integer
<b>Abstract Operation(Method)</b>	-
<b>Exceptional Courses of Events</b>	N/A

<b>Type</b>	Class
<b>Name</b>	Property
<b>Purpose</b>	자신의 이름과 자신을 property로 설정한 representative value들 이름의 배열을 갖는 클래스
<b>Overview (Class)</b>	자신의 이름을 수정하거나 반환하고, 자신을 property로 설정한 representative value들의 이름을 추가, 삭제한다.
<b>Cross Reference</b>	Functions: R2.3 R2.4
<b>Input (Method)</b>	-
<b>Output (Method)</b>	-
<b>Abstract Operation(Method)</b>	-
<b>Exceptional Courses of Events</b>	N/A

<b>Events</b>	
---------------	--

<b>Type</b>	Method
<b>Name</b>	getName()
<b>Purpose</b>	이름을 반환한다.
<b>Overview (Class)</b>	name을 반환한다.
<b>Cross Reference</b>	Functions: R2.3 R2.4
<b>Input (Method)</b>	void
<b>Output (Method)</b>	String
<b>Abstract Operation(Method)</b>	-
<b>Exceptional Courses of Events</b>	N/A

<b>Type</b>	Method
<b>Name</b>	setName()
<b>Purpose</b>	이름을 설정한다.
<b>Overview (Class)</b>	전달받은 name으로 이름을 설정한다. 성공 시 0을 반환한다.
<b>Cross Reference</b>	Functions: R2.3 R2.4
<b>Input (Method)</b>	String name
<b>Output (Method)</b>	Integer
<b>Abstract Operation(Method)</b>	-
<b>Exceptional Courses of Events</b>	N/A

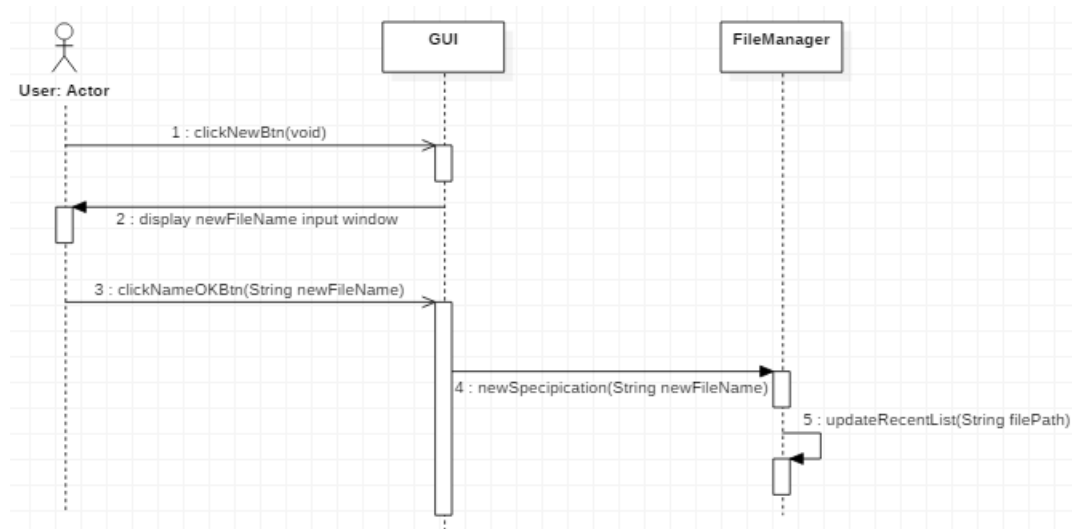
<b>Type</b>	Method
<b>Name</b>	addName()
<b>Purpose</b>	representative value 이름을 배열에 추가한다.
<b>Overview (Class)</b>	자신을 property로 설정한 representative value 이름 배열에 전달받은 valueName을 추가한다.
<b>Cross Reference</b>	Functions: R2.3 R2.4

<b>Input (Method)</b>	String valueName
<b>Output (Method)</b>	void
<b>Abstract Operation(Method)</b>	-
<b>Exceptional Courses of Events</b>	N/A

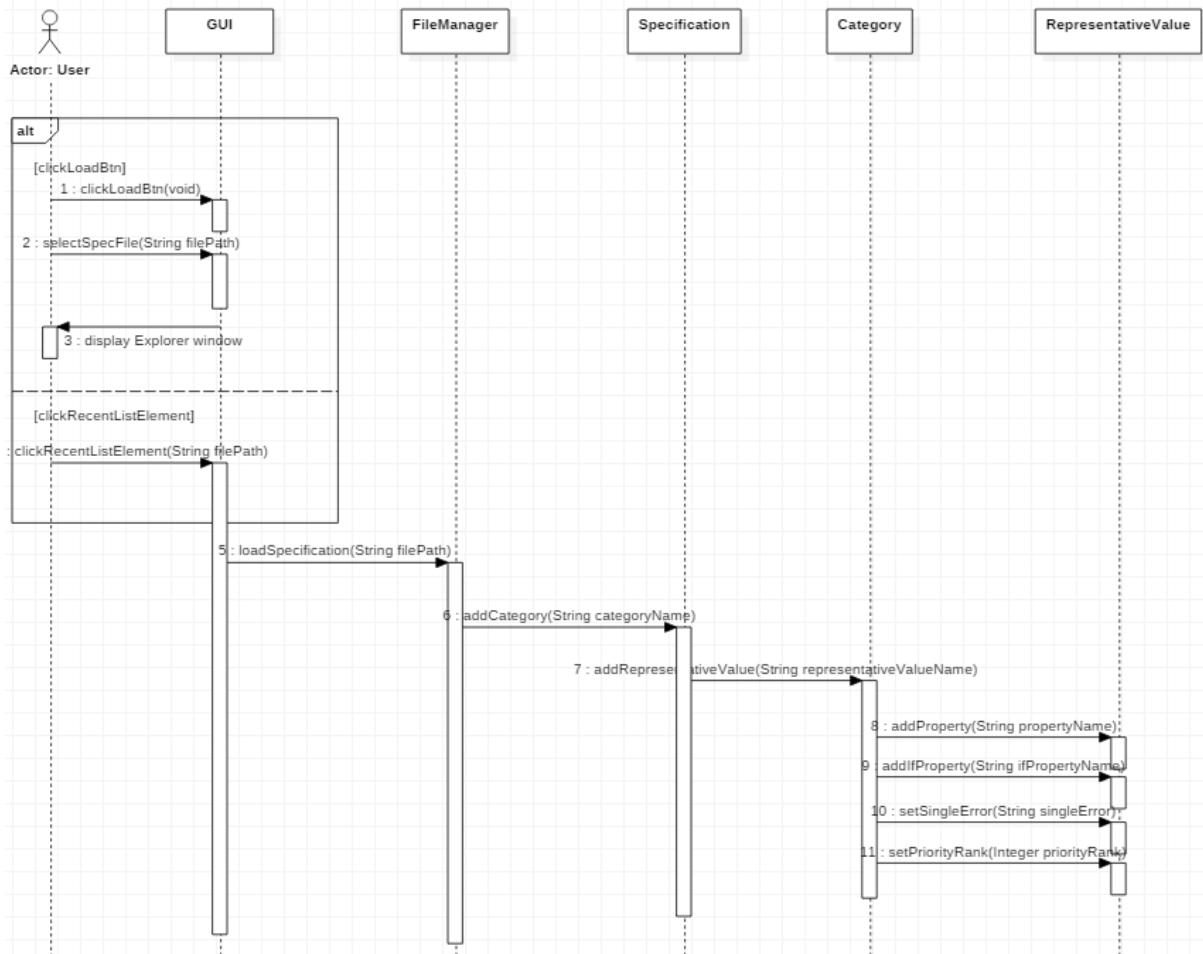
<b>Type</b>	Method
<b>Name</b>	isOnly()
<b>Purpose</b>	자신을 property로 설정한 representative value가 한 개인지 여부를 반환한다.
<b>Overview (Class)</b>	자신을 property로 설정한 representative value 이름 배열의 크기가 1이면 true, 1이 아니면 false를 반환한다.
<b>Cross Reference</b>	Functions: R2.3 R2.4
<b>Input (Method)</b>	void
<b>Output (Method)</b>	Boolean
<b>Abstract Operation(Method)</b>	-
<b>Exceptional Courses of Events</b>	N/A

<b>Type</b>	Method
<b>Name</b>	delName()
<b>Purpose</b>	배열에서 representative value 이름을 삭제한다.
<b>Overview (Class)</b>	자신을 property로 설정한 representative value 이름 배열에 전달받은 valueName을 삭제한다.
<b>Cross Reference</b>	Functions: R2.3 R2.4
<b>Input (Method)</b>	String valueName
<b>Output (Method)</b>	void
<b>Abstract Operation(Method)</b>	-
<b>Exceptional Courses of Events</b>	N/A

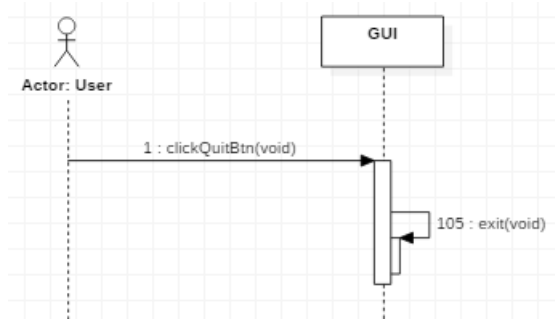
### Activity 2052. Implement Windows



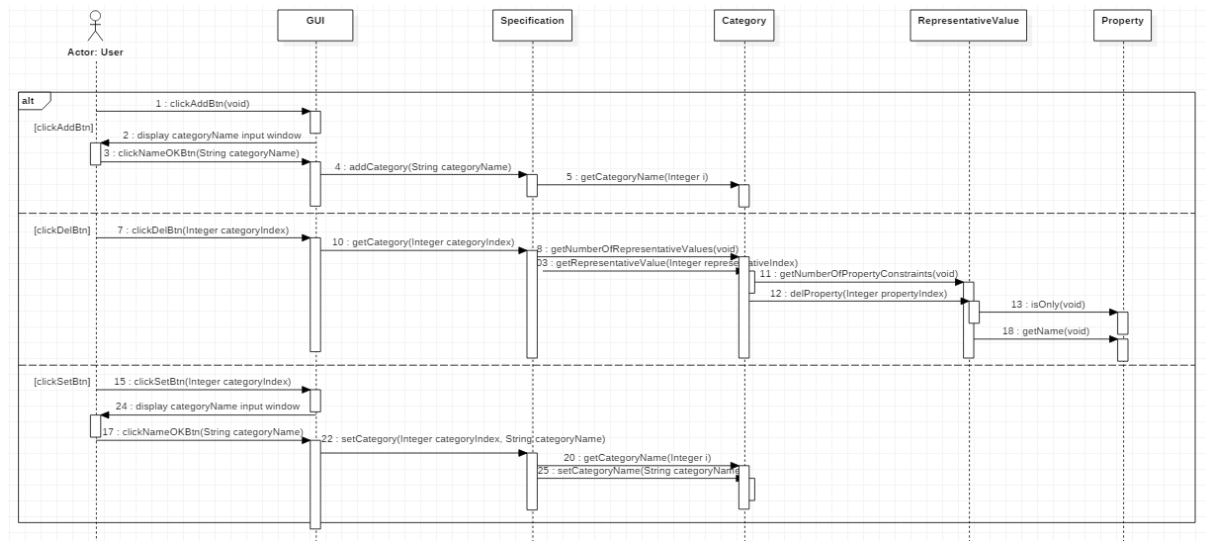
<b>Name</b>	newSpecification
<b>Responsibilities</b>	GUI에서 '새로 만들기'버튼을 클릭하고, 새로 만들 Specification파일 이름을 입력 받는다.
<b>Type</b>	GUI
<b>Cross Reference</b>	R1.1
<b>Notes</b>	저장할 파일 이름을 입력 받는 화면이 출력된다. 추후 사용자로부터 입력 받은 이름으로 새로운 specification파일을 생성한다. 최근 파일 항목에 반영한다.
<b>Pre-Conditions</b>	N/A
<b>Post-Conditions</b>	Specification파일을 작성하기 위한 화면으로 넘어간다.



<b>Name</b>	loadSpecification
<b>Responsibilities</b>	GUI에서 '불러오기' 버튼을 클릭하거나, 최근 파일 목록 하나를 클릭한다.
<b>Type</b>	GUI
<b>Cross Reference</b>	R1.2
<b>Notes</b>	기존에 작성한 Specification 파일을 불러온다. 최근 파일 항목에 반영한다.
<b>Pre-Conditions</b>	불러올 Specification 파일 또는 recent file list에 specification 파일이 있어야 한다.
<b>Post-Conditions</b>	Specification 파일을 작성하기 위한 화면으로 넘어간다.



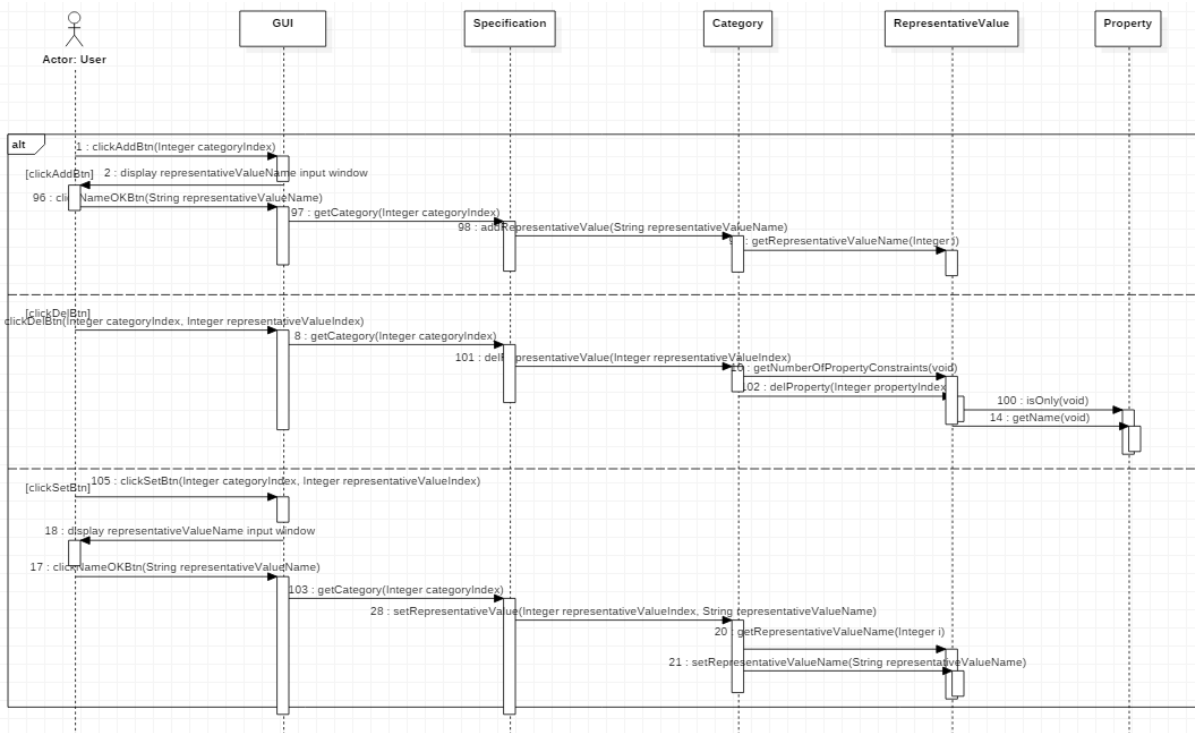
<b>Name</b>	shutdownProgram
<b>Responsibilities</b>	GUI에서 '종료하기' 버튼을 클릭한다.
<b>Type</b>	GUI
<b>Cross Reference</b>	R1.3
<b>Notes</b>	프로그램을 종료한다.
<b>Pre-Conditions</b>	N/A
<b>Post-Conditions</b>	N/A



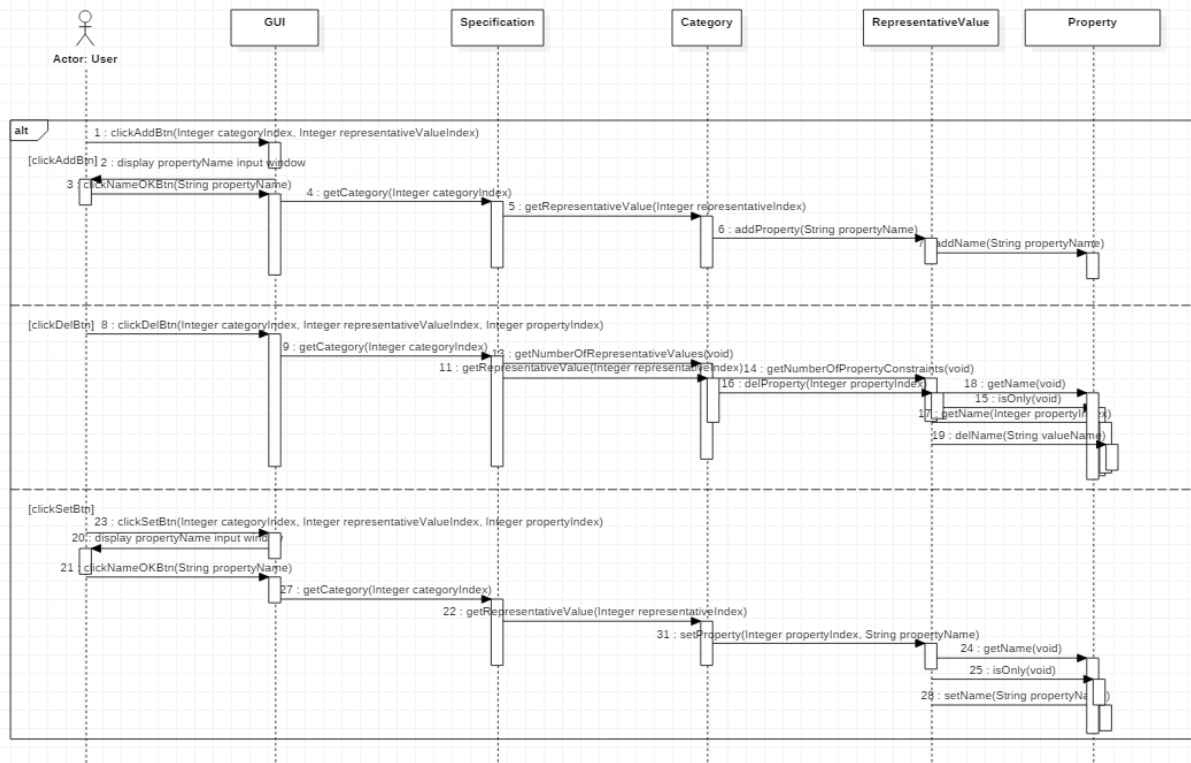
<b>Name</b>	setCategory
<b>Responsibilities</b>	GUI에서 category에 대해 '추가', 또는 '수정', '삭제' 버튼을 클릭한다.
<b>Type</b>	GUI
<b>Cross Reference</b>	R2.1 R2.2 R2.3 R2.4
<b>Notes</b>	새로운 category를 추가 또는 이미 작성된 category를 수정하거나 지운다.
<b>Pre-Conditions</b>	'수정', '삭제' 버튼이 작동하기 위해서는 이미 추가된



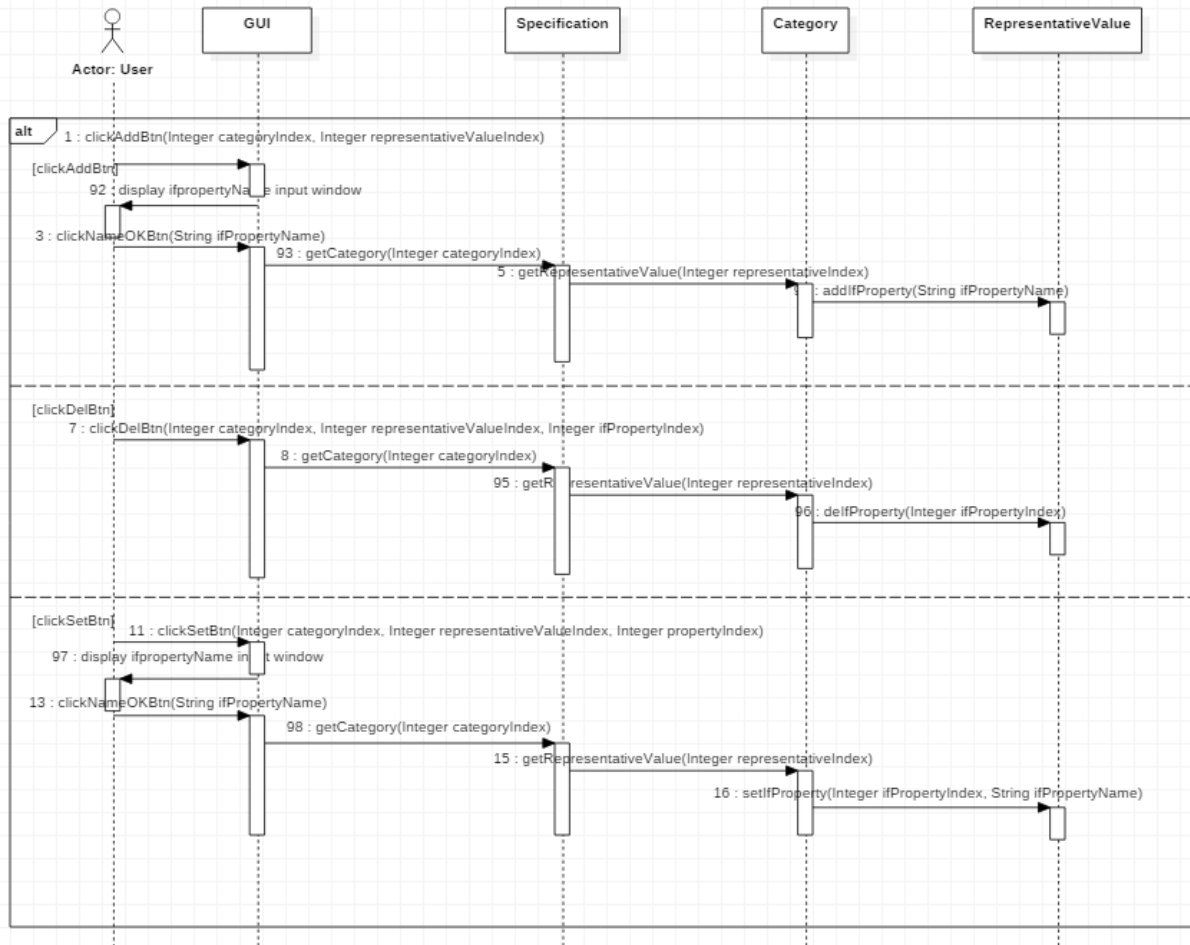
	category가 있어야 한다.
<b>Post-Conditions</b>	클릭한 버튼의 결과가 Specification 및 화면에 반영된다.



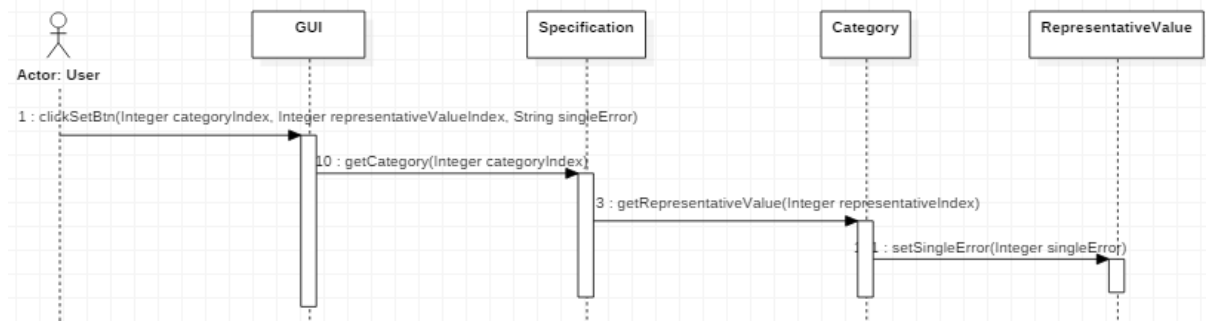
<b>Name</b>	setRepresentativeValue
<b>Responsibilities</b>	GUI에서 representativeValue에 대해 '추가', 또는 '수정', '삭제'버튼을 클릭한다.
<b>Type</b>	GUI
<b>Cross Reference</b>	R2.2 R2.3 R2.4
<b>Notes</b>	새로운 representativeValue를 추가 또는 이미 작성된 representativeValue를 수정하거나 지운다.
<b>Pre-Conditions</b>	'수정', '삭제'버튼이 작동하기 위해서는 이미 추가된 representativeValue가 있어야 한다.
<b>Post-Conditions</b>	클릭한 버튼의 결과가 Specification 및 화면에 반영된다.



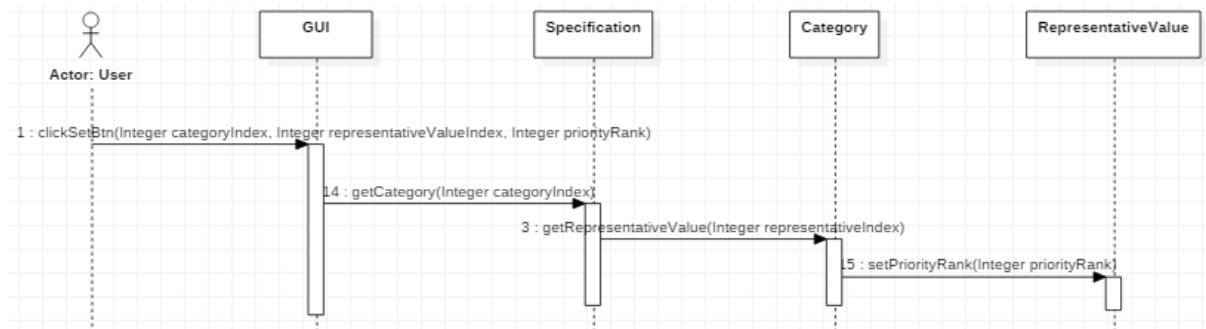
<b>Name</b>	setProperty
<b>Responsibilities</b>	GUI에서 property에 대해 '추가', 또는 '수정', '삭제'버튼을 클릭한다.
<b>Type</b>	GUI
<b>Cross Reference</b>	R2.3 R2.4
<b>Notes</b>	새로운 property를 추가 또는 이미 작성된 property를 수정하거나 지운다.
<b>Pre-Conditions</b>	'수정', '삭제'버튼이 작동하기 위해서는 이미 추가된 property가 있어야 한다.
<b>Post-Conditions</b>	클릭한 버튼의 결과가 Specification 및 화면에 반영된다.



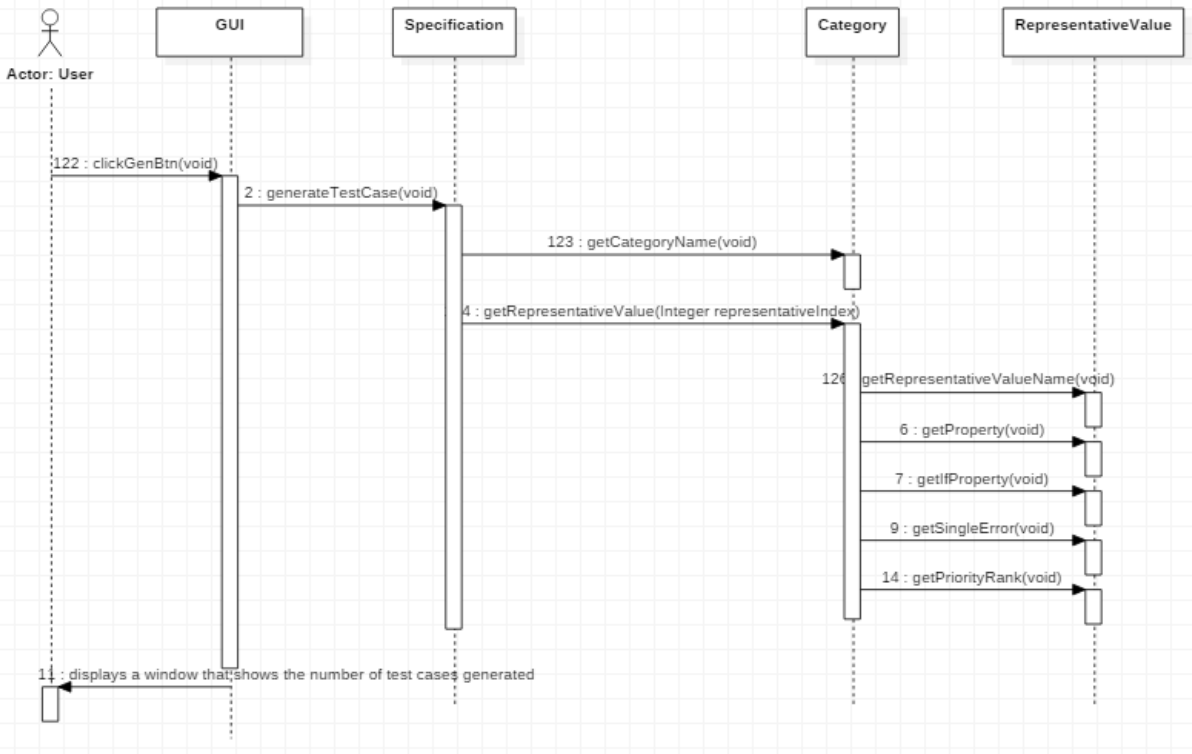
<b>Name</b>	setIfProperty
<b>Responsibilities</b>	GUI에서 if-property에 대해 '추가', 또는 '수정', '삭제'버튼을 클릭한다.
<b>Type</b>	GUI
<b>Cross Reference</b>	R2.4
<b>Notes</b>	새로운 if-property를 추가 또는 이미 작성된 if-property를 수정하거나 지운다.
<b>Pre-Conditions</b>	'수정', '삭제'버튼이 작동하기 위해서는 이미 추가된 if-property가 있어야 한다.
<b>Post-Conditions</b>	클릭한 버튼의 결과가 Specification 및 화면에 반영된다.



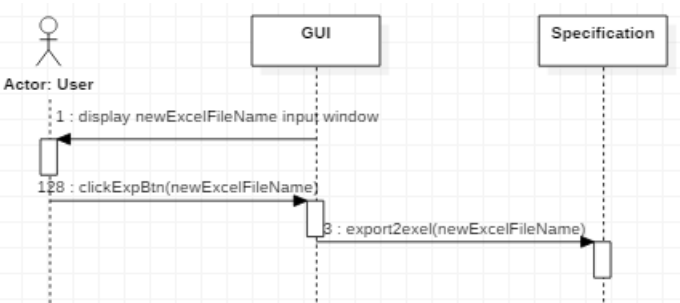
<b>Name</b>	setSingleError
<b>Responsibilities</b>	GUI에서 single 또는 error constraint에 대해서 클릭한다.
<b>Type</b>	GUI
<b>Cross Reference</b>	R2.5
<b>Notes</b>	선택 받은 single 또는 error constraint 중 하나를 부여한다. [선택을 안 할 수도 있음]
<b>Pre-Conditions</b>	N/A
<b>Post-Conditions</b>	클릭한 버튼의 결과가 Specification 및 화면에 반영된다.



<b>Name</b>	setPriorityRank
<b>Responsibilities</b>	GUI에서 priority에 대해 총 1~5점 중 하나를 클릭한다.
<b>Type</b>	GUI
<b>Cross Reference</b>	R2.6
<b>Notes</b>	priority(중요도)를 선택 받은 1~5점 중 하나로 부여한다.
<b>Pre-Conditions</b>	N/A
<b>Post-Conditions</b>	클릭한 버튼의 결과가 Specification 및 화면에 반영된다.

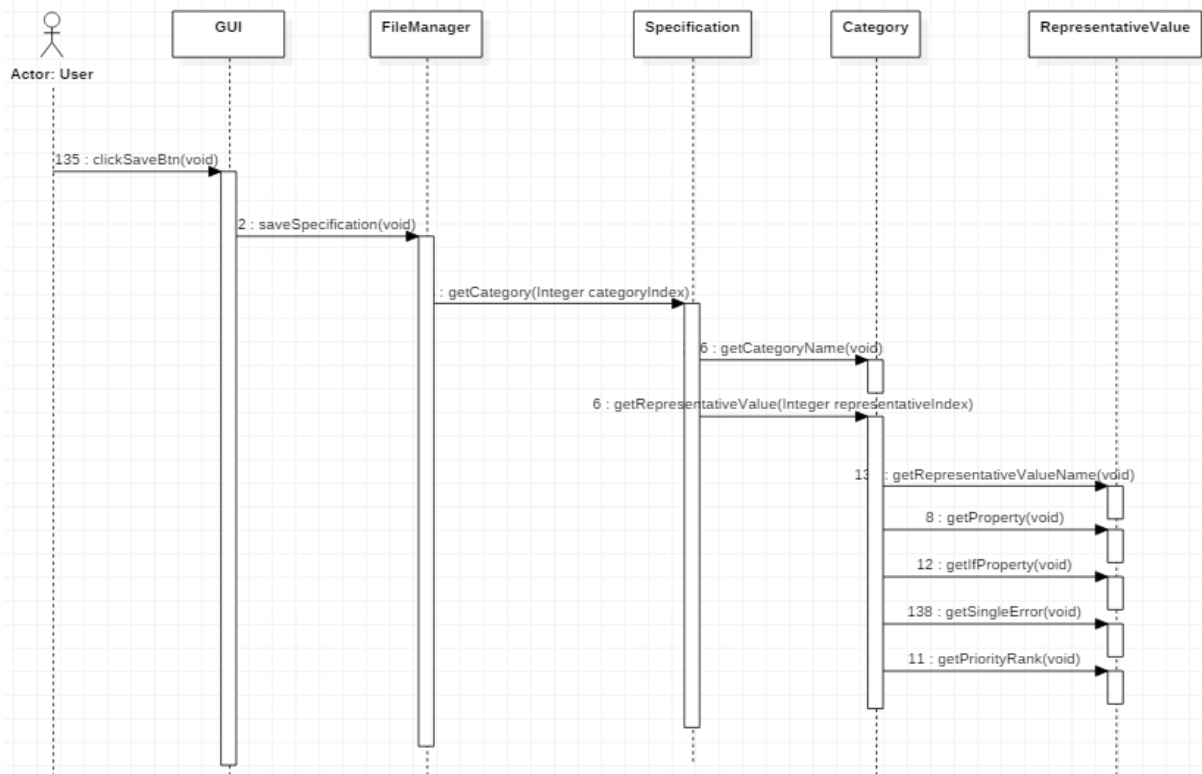


<b>Name</b>	generateTestCases
<b>Responsibilities</b>	GUI에서 'Test case 생성 후 엑셀로 저장하기' 버튼을 클릭한다.
<b>Type</b>	GUI
<b>Cross Reference</b>	R3
<b>Notes</b>	입력된 Specification을 통해 test case를 만들고 그 총 개수를 화면을 통해 사용자에게 알려준다.
<b>Pre-Conditions</b>	입력된 Specification이 test case를 만들기 위한 충분한 값이 입력되어야 한다.
<b>Post-Conditions</b>	test case를 만들고 그 총 개수를 화면을 통해 사용자에게 알려준다.



<b>Name</b>	export2excel
-------------	--------------

<b>Responsibilities</b>	GUI에서 생성할 Test case 엑셀파일의 이름을 입력 받는다.
<b>Type</b>	GUI
<b>Cross Reference</b>	R4
<b>Notes</b>	Test case generate이 끝난 후, 생성할 엑셀파일의 이름을 받는다. 이후 생성된 test cases를 Priority순으로 정렬한 후, 입력 받은 이름의 엑셀 파일에 저장한다.
<b>Pre-Conditions</b>	입력된 Specification이 test case를 만들기 위한 충분한 값이 입력되어서, Generate test cases가 실행되어야 한다.
<b>Post-Conditions</b>	생성된 Test case를 입력 받은 이름의 엑셀 파일에 저장한다.



<b>Name</b>	saveSpecification
<b>Responsibilities</b>	GUI에서 '저장하기' 버튼을 클릭한다.
<b>Type</b>	GUI
<b>Cross Reference</b>	R5
<b>Notes</b>	specification입력 중간저장 버튼을 누르면, 현재까지의 작업 상태를 저장한다.
<b>Pre-Conditions</b>	N/A
<b>Post-Conditions</b>	현재까지의 작업 상태를 specification 파일로 저장한다.

## Activity 2055. Write Unit Test Code

1. Identify all methods in design

Class Name	Method Name
MainSystem	void main()
FileManager	int newSpecification (String fileName)
	int loadSpecification (String filePath)
	int saveSpecification()
	int saveSpecification (String filePath)
	int updateRecentList (String filePath)
Specification	int getNumberOfCategories()
	Category getCategory(int categoryIndex)
	int addCategory(String categoryName)
	int delCategory(int categoryIndex)
	void updateData()
	int generateTestCase()
	void updatePropertyList(ArrayList<Property> propertyList, int categoryIndex, int rvIndex)
	ArrayList<String> refreshTestCaseList(ArrayList<RepresentativeValue> testCaseList, int categoryIndex, int rvIndex)
	int prioritySum(ArrayList<RepresentativeValue> testCaseList)
	boolean ifConstraintsExistInPropertyList(ArrayList<Property> propertyList, int categoryIndex, int rvIndex)
	void f(int categoryIndex, ArrayList<Property> propertyList, ArrayList<RepresentativeValue> testCaseList, HashMap<ArrayList<String>, Integer> testCase, int prioritySum)
	List<ArrayList<String>> sortByValue(HashMap<ArrayList<String>, Integer> testCase)
	int export2excel(String fileName)
Category	setCategoryName(String)
	getCategoryName()
	addRepresentativeValue(String)
	delRepresentativeValue(int)

	setRepresentativeValue(int, String)
	getRepresentativeValue(int)
	getNumberOfRepresentativeValues()
RepresentativeValue	setRepresentativeValueName(String)
	getRepresentativeValueName()
	addProperty(String)
	setProperty(int, String)
	delProperty(int)
	getProperty(int)
	getNumberOfPropertyConstraints()
	addIfProperty(String)
	setIfProperty(int, String)
	dellfProperty(int)
	getIfProperty(int)
	getNumberOfIfPropertyConstraints()
	setSingleError(int)
	getSingleError()
	setPriorityRank(int)
getPriorityRank()	
Property	setName(String)
	getName()
	addName(String)
	delName(String)
	isOnly()

## 2. Write Test Code

```

@Test //정상적으로 갯수 반환하는지
public void testGetNumberOfCategories() {
    Specification s = new Specification();
    s.addCategory("a");
    s.addCategory("b");
    s.addCategory("c");

    assertEquals(3, s.getNumberOfCategories());
}

```



```

@Test //테스트 케이스 생성이 정확하게 되는지.
public void testGenerateTestCase() {
    Specification s = new Specification();
    s.addCategory("a");
    s.addCategory("b");
    s.getCategory(0).addRepresentativeValue("a1");
    s.getCategory(0).addRepresentativeValue("a2");
    s.getCategory(0).addRepresentativeValue("a3");
    s.getCategory(0).getRepresentativeValue(0).addProperty("a1");
    s.getCategory(1).addRepresentativeValue("b1");
    s.getCategory(1).addRepresentativeValue("b2");
    s.getCategory(1).addRepresentativeValue("b3");
    s.getCategory(1).addRepresentativeValue("b4");
    s.getCategory(1).getRepresentativeValue(1).addIfProperty("a1");
    s.getCategory(1).getRepresentativeValue(2).setSingleError(1);

    assertEquals(8, s.generateTestCase());
}

@Test //엑셀로 정확히 짜지는지
public void testExport2excel() {
    Specification s = new Specification();
    s.addCategory("a");
    s.addCategory("b");
    s.getCategory(0).addRepresentativeValue("a1");
    s.getCategory(0).addRepresentativeValue("a2");
    s.getCategory(0).addRepresentativeValue("a3");
    s.getCategory(0).getRepresentativeValue(0).addProperty("a1");
    s.getCategory(1).addRepresentativeValue("b1");
    s.getCategory(1).addRepresentativeValue("b2");
    s.getCategory(1).addRepresentativeValue("b3");
    s.getCategory(1).addRepresentativeValue("b4");
    s.getCategory(1).getRepresentativeValue(1).addIfProperty("a1");
    s.getCategory(1).getRepresentativeValue(2).setSingleError(1);
    s.generateTestCase();
    assertEquals(0, s.export2excel("dd.xls"));
    File f = new File("dd.xls");
    assertTrue(f.exists());
}

@Test
public void testSetCategory() {
    Specification s = new Specification();
    s.addCategory("a");
    s.addCategory("b");
    s.addCategory("c");

    assertEquals(0, s.setCategory(2, "d"));
    assertEquals(1, s.setCategory(2, "b"));
}

```

```

@Test
public void testAddCategory() {
    Specification s = new Specification();
    s.addCategory("a");
    s.addCategory("b");
    s.addCategory("c");

    assertEquals(0, s.addCategory("d"));
    assertEquals(1, s.addCategory("a"));
}

```

## 2060 Test

### Activity 2061. Unit Testing

- CategoryTest.java

Finished after 4.029 seconds

Runs: 3/3 Errors: 0 Failures: 0

- test.CategoryTest [Runner: JUnit 4] (3.978 s)
  - testDelRepresentativeValue (0.001 s)
  - testSetRepresentativeValue (3.234 s)
  - testAddRepresentativeValue (0.743 s)

```

1 package test;
2
3 import static org.junit.Assert.*;
13
14 public class CategoryTest
15 {
16     static Hashtable<String, Property> pt;
17     static Category c1, c2;
18     static RepresentativeValue r1, r2;
19     @BeforeClass
20     public static void beforeClass()
21     {
22         pt = new Hashtable<String, Property>();
23         c1 = new Category("c1", pt);
24         c2 = new Category("c2", pt);
25         r1 = new RepresentativeValue("r1", pt);

```

- FileManagerTest.java

Finished after 5.777 seconds

Runs: 9/9 Errors: 0 Failures: 0

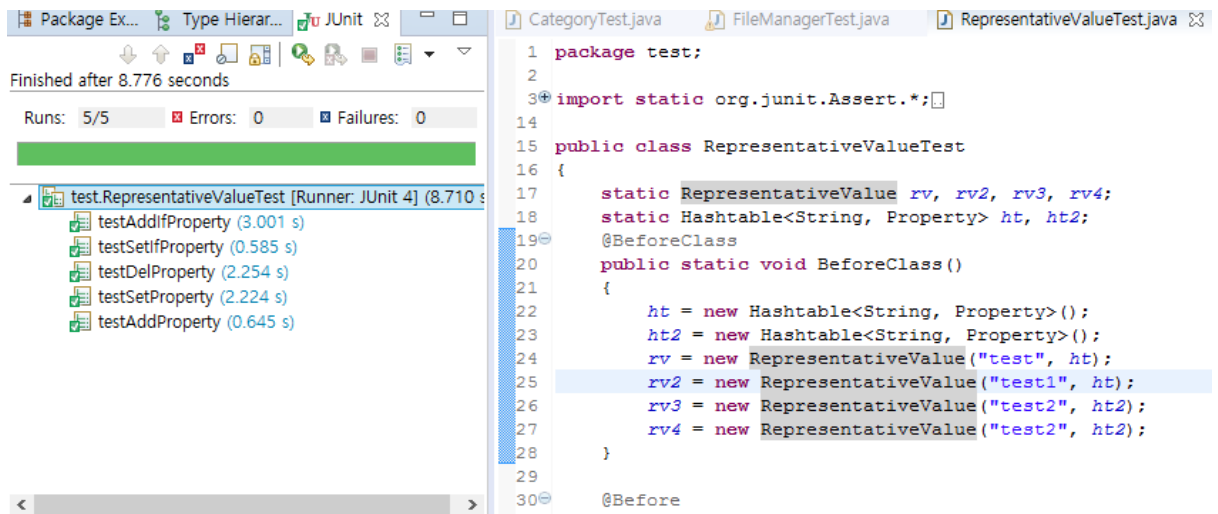
- test.FileManagerTest [Runner: JUnit 4] (5.710 s)
  - testUpdateRecentList (0.005 s)
  - testSaveSpecificationString1 (0.011 s)
  - testSaveSpecificationString2 (2.947 s)
  - testGetPath (0.000 s)
  - testNewSpecification1 (0.015 s)
  - testNewSpecification2 (1.768 s)
  - testSaveSpecification (0.002 s)
  - testLoadSpecification1 (0.010 s)
  - testLoadSpecification2 (0.948 s)

```

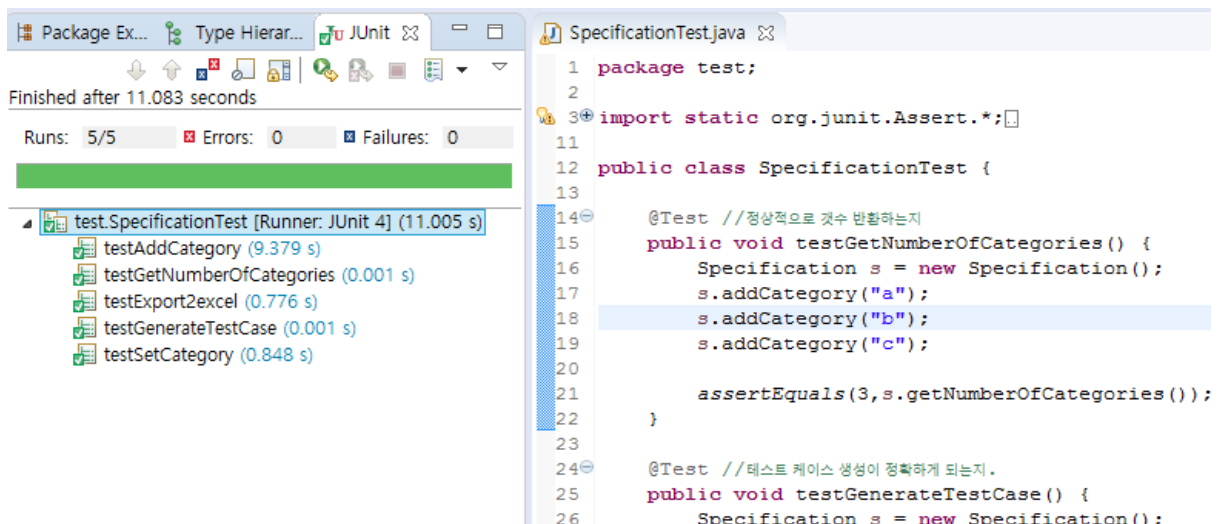
1 package test;
2
3 import static org.junit.Assert.*;
16
17 public class FileManagerTest {
18     private static Specification s;
19     private static FileManager f;
20     private static File file;
21
22     @BeforeClass
23     public static void beforeSetup() {
24         s = new Specification();
25         f = new FileManager(s);
26     }
27
28     @Test //생성할 것인지 테스트
29     public void testNewSpecification1() {
30         file = new File("newTest.yzb");
31         if(file.exists()) {
32             file.delete();
33         }
34         assertEquals(0, f.newSpecification("newTest"));
35         assertTrue(file.exists());
36     }

```

- RepresentativeValueTest.java



- SpecificationTest.java



Activity 2063. System Testing

No	Test 항목	Description	Use Case	System Function
1	새로 만들기 버튼	새로운 .yzb가 생성되는지 확인한다.	Make new specification file	R1.1
2	불러오기 버튼	기존의 .yzb를 불러오는지 확인한다.	Load specification file	R1.2
3	최근 파일 목록 선택	최근 파일 목록의 .yzb를 불러오는지 확인한다.	Load specification file	R1.2

4	종료 버튼	프로그램이 종료되는지 확인한다.	Shut down program	R1.3
5	Category 추가 버튼	Category가 추가 되는지 확인한다.	Set category	R2.1
6	Category 제거 버튼	Category가 제거 되는지 확인한다.	Set category	R2.1
7	Category 편집 버튼	Category가 편집 되는지 확인한다.	Set category	R2.1
8	Representative Value 추가 버튼	Representative Value가 추가 되는지 확인한다.	Set representative value	R2.2
9	Representative Value 제거 버튼	Representative Value가 제거 되는지 확인한다.	Set representative value	R2.2
10	Representative Value 편집 버튼	Representative Value가 편집 되는지 확인한다.	Set representative value	R2.2
11	Property 추가 버튼	Property가 추가 되는지 확인한다.	Set property	R2.3
12	Property 제거 버튼	Property가 제거 되는지 확인한다.	Set property	R2.3
13	Property 편집 버튼	Property가 편집 되는지 확인한다.	Set property	R2.3
14	If-Property 추가 버튼	If-Property가 추가 되는지 확인한다.	Set if-property	R2.4
15	If-Property 제거 버튼	If-Property가 제거 되는지 확인한다.	Set if-property	R2.4
16	If-Property 편집 버튼	If-Property가 편집 되는지 확인한다.	Set if-property	R2.4
17	Error & Single 설정 버튼	Single 또는 Error가 설정 되는지 확인한다.	Set single and error	R2.5
18	Prioirty 설정 버튼	Priority가 설정되는지 확인한다.	Set priority rank	R2.6
19	파일 - 저장 버튼	현재 .yzb 파일에 저장되는지 확인한다.	Save contemporary specification file	R5
20	파일 - 다른이름으로 저장 버튼	새로운 이름의 .yzb 파일에 저장되는지 확인한다.	Save contemporary specification file	R5
21	파일 - TestCase 생성 및 엑셀 저장	엑셀 파일이 생성되는지 확인한다.	Generate test cases, Export test case to excel file	R3, R4
22	파일 - 종료 버튼	프로그램이 종료되는지 확인한다.	Shut down program	R1.3