

System Test

Structural Testing

Software Verification T2

1. Pairwise Testing

1.1 What is Pairwise Testing?

Black-Box Testing 을 하는 방법 중 한 종류가 Pairwise Testing 이다. Pairwise Testing 은 한 카테고리 내에서 테스트되어야 하는 각 Parameter 값들을 다른 Parameter 값과 최소한 한 번씩은 조합을 이루어 테스트가 되도록 한다. 하지만 각 Parameter 값들의 모든 조합을 Test Case 로 만드는 것은 사실상 불가능하다. 그래서 Pairwise Testing 은 두 Parameter 값들의 개별 조합만을 고려하여 Test Case 만 만든다. 두 Parameter 값들의 개별 조합만을 고려하여 Test Case 를 만들게 되면 각 Parameter 값들의 모든 조합을 Test Case 로 만드는 것보다 적은 수의 Test Case 를 만들면서 모든 조합을 테스트 할 수 있다.

1.2 Pairwise Testing Tool - AllPairs

AllPairs 는 Pairwise 조합을 생성해주는 도구이다. AllPairs 는 탐색적 테스트(Exploratory Testing)의 창시자인 제임스 바흐(James Bach)가 만든 도구로써 사용자가 각 파라미터의 값들을 입력하면 각 파라미터 값들의 조합을 출력으로 보여준다. 사용자가 두 Parameter 값들의 개별 조합을 비교해가면서 Pairwise 조합을 만들 수도 있지만 Parameter 의 수가 많아지고, Parameter 의 값들의 수도 많아지게 되면 Pairwise 조합을 생성해주는 도구를 사용하지 않고는 테스트 케이스를 만드는 것이 불가능해진다.

2. Pairwise Testing about Webtoon Paint.

2.1 Testable Features.

Activity 1003. Define Requirements

o Functional Requirements (categorized table)

Ref #	Function	Category
R1.1	파일 불러오기	Evident
R1.2	파일 저장하기	Evident
R2.1.1	브러시	Evident
R2.1.2	지우개	Evident
R2.2	굵기 조절	Evident
R2.3	색상 선택	Evident
R3.1	영역 선택	Evident
R3.2	영역 삭제	Evident
R3.3	영역 잘라내기	Evident
R3.4	영역 복사하기	Evident
R3.5	영역 붙여넣기	Evident
R3.6	영역 이동	Evident



4. System Sequence Diagrams

Use Case	Name of Action/Activation Event
1. 파일 불러오기	LoadImage()
2. 파일 저장하기	SaveImage()
3. 브러시	Brushing()
4. 지우개	Erasing()
5. 굵기 조절	SetLineStyle()
6. 색상 선택	SetColor()
7. 영역 선택	SelectArea()
8. 영역 삭제	DelArea()
9. 영역 크기 조절	SetAreaSize()
10. 영역 이동	MoveArea()
11. 영역 잘라내기	CutOffArea()
12. 영역 복사	CopyArea()
13. 영역 붙여넣기	PasteArea()
14. 문자 입력	Enter Text()
15. 영 전환하기	ChangeCase()
16. 영 크기 조절	AdjustCase()
17. 영 삭제하기	DelCase()
18. 영 크기 조절	SetCaseSize()
19. 영 합치기	MergeCase()

시스템 요구사항을 기반으로 테스트 가능한 기능 단위들이다.

파일 저장하기
파일 불러오기
브러시
지우개
굵기 조절
색상 선택
영역 선택하기
영역 복사하기
영역 삭제하기
영역 잘라내기
영역 붙여넣기
컷 추가하기
컷 삭제하기
컷 전환하기
컷 크기조절하기
컷 합치기

2.2 파일 저장하기

2.2.1 Parameter Values

file_name	file_type	image_width	image_height	total_cut_size
non-exist	bmp	600	600	0
	png	others	others	1
				others

Pairwise 조합을 생성해주는 AllPairs 도구를 사용하기 위해서 각 Parameter 들이 가질 수 있는 값들을 같은 Tab 을 간격으로 해서 입력해주어야 한다. 이때 Error Constraints 를 적용해서 Error 값을 제외하고 Pairwise 조합이 만들어지도록 한다.



file_save.txt 라는 파일을 통해서 AllPairs 도구에 입력하게 되는데 같은 Tab 을 간격으로 해야 하기 때문에 깔끔하게 정리되어 보이지는 않는다.

```
C:\Windows\system32\cmd.exe
c:\w\pairs>allpairs.exe file_save.txt

TEST CASES
case  file_type  image_width  image_height  total_cut_size  pairings
1      bmp        600         600          0              6
2      png        others      others        0              6
3      bmp        others      600          1              5
4      png        600        others        1              5
5      bmp        600        others      others        4
6      png        others      600          others        4

PAIRING DETAILS
var1  var2  value1  value2  appearances  cases
total_cut_size  file_type  0      bmp      1          1
total_cut_size  file_type  0      png      1          2
total_cut_size  file_type  1      bmp      1          3
total_cut_size  file_type  1      png      1          4
total_cut_size  file_type  others  bmp      1          5
total_cut_size  file_type  others  png      1          6
total_cut_size  image_width  0      600     1          1
total_cut_size  image_width  0      others  1          2
total_cut_size  image_width  1      600     1          4
total_cut_size  image_width  1      others  1          3
```

각 Parameter 들이 가질 수 있는 값들이 저장되어 있는 file_save.txt 파일을 입력으로 해서 AllPairs 도구를 실행시키면 Pairwise 조합을 만들어서 보여준다. 도스 창에 직접 출력하지 않고 file_save_result.txt 파일로 출력 받도록 하기 위해서 c:\wpairs>allpairs.exe file_save.txt > file_save_result.txt 명령어를 사용한다.

2.2.2 Pairwise Combination

case	file_type	image_width	image_height	total_cut_size	pairings
1	bmp	600	600	0	6
2	png	others	others	0	6
3	bmp	others	600	1	5
4	png	600	others	1	5
5	bmp	600	others	others	4
6	png	others	600	others	4

2.2.3 Summary of Pairwise Combination

Category-Partition 방법으로 Test Case 를 만들었을 때 243 개의 Test Case 가 만들어졌고, Error Constraints, Property Constraints, Single Constraints 등을 적용했을 때 24 개의 Test Case 로 줄었다. Pairwise Combination 으로 Test Case 를 만들어내면 Category-Partition 방법으로 Test Case 를 만들었을 때보다 훨씬 적은 6 개의 Test Case 가 만들어졌다.

Category-Partiton	Pairwise Combination
Step 1 : 243 Test Cases	6 Test Cases
Step 2 : 24 Test Cases	
Step 3 : 24 Test Cases	

2.3 파일 불러오기

2.3.1 Parameter Values

file_name	file_type	image_width	image_height	total_cut_size
exist	bmp	600	600	0
	png	others	others	1
				others

2.3.2 Pairwise Combination

case	file_type	image_width	image_height	total_cut_size	pairings
1	bmp	600	600	0	6
2	png	others	others	0	6
3	bmp	others	600	1	5
4	png	600	others	1	5
5	bmp	600	others	others	4
6	png	others	600	others	4

2.3.3 Summary of Pairwise Combination

Category-Partiton	Pairwise Combination
Step 1 : 243 Test Cases	6 Test Cases
Step 2 : 24 Test Cases	
Step 3 : 24 Test Cases	

2.4 브러시

2.4.1 Parameter Values

thickness	color	start_x_location	start_y_location	end_x_location	end_y_location	cur_cut_num
0	16 colors	0	0	0	0	1
1		1	1	1	1	others

2.4.2 Pairwise Combination

case	thickness	color	start_x_location	start_y_location	end_x_location	end_y_location	cur_cut_num	pairings
1	0	16 colors	0	0	0	0	1	21
2	1	16 colors	1	1	1	1	others	21
3	0	~16 colors	0	1	0	1	others	9
4	1	~16 colors	1	0	1	0	1	9
5	0	~16 colors	1	1	0	0	1	4
6	1	~16 colors	0	0	1	1	others	4
7	0	~16 colors	~1	~0	1	0	others	2
8	1	~16 colors	~0	~1	0	1	1	2

2.4.3 Summary of Pairwise Combination

Category-Partiton	Pairwise Combination
Step 1 : 1,458 Test Cases	8 Test Cases
Step 2 : 64 Test Cases	
Step 3 : 64 Test Cases	

2.5 지우개

2.5.1 Parameter Values

thickness	start_x_location	start_y_location	end_x_location	end_y_location	cur_cut_num
0	0	0	0	0	1
1	1	1	1	1	others

2.5.2 Pairwise Combination

case	thickness	start_x_location	start_y_location	end_x_location	end_y_location	cur_cut_num	pairings
1	0	0	0	0	0	1	15
2	0	1	1	1	1	others	15
3	1	0	1	0	1	1	11
4	1	1	0	1	0	others	11
5	~0	0	0	1	1	1	3
6	~0	1	1	0	0	others	3
7	~1	0	~1	~1	~0	others	1
8	~1	1	~0	~0	~1	1	1

2.5.3 Summary of Pairwise Combination

Category-Partiton	Pairwise Combination
Step 1 : 729 Test Cases	8 Test Cases
Step 2 : 64 Test Cases	
Step 3 : 64 Test Cases	

2.6 영역 선택하기

2.6.1 Parameter Values

start_x_location	start_y_location	end_x_location	end_y_location	cur_cut_num
0	0	0	0	1
50	50	100	100	others

2.6.2 Pairwise Combination

case	start_x_location	start_y_location	end_x_location	end_y_location	cur_cut_num	pairings
1	0	0	0	0	1	10
2	0	50	100	100	others	10
3	50	0	100	0	others	8
4	50	50	0	100	1	8
5	~0	0	0	100	others	2
6	~0	50	100	0	1	2

2.6.3 Summary of Pairwise Combination

Category-Partiton	Pairwise Combination
Step 1 : 243 Test Cases	6 Test Cases
Step 2 : 32 Test Cases	
Step 3 : 32 Test Cases	

2.7 영역 복사하기

2.7.1 Parameter Values

isSelected	start_x_location	start_y_location	end_x_location	end_y_location	cur_cut_num
TRUE	0	0	0	0	1
	50	50	100	100	others

2.7.2 Pairwise Combination

case	start_x_location	start_y_location	end_x_location	end_y_location	cur_cut_num	pairings
1	0	0	0	0	1	10
2	0	50	100	100	others	10
3	50	0	100	0	others	8
4	50	50	0	100	1	8
5	~0	0	0	100	others	2
6	~0	50	100	0	1	2

2.7.3 Summary of Pairwise Combination

Category-Partiton	Pairwise Combination
Step 1 : 486 Test Cases	6 Test Cases
Step 2 : 32 Test Cases	
Step 3 : 32 Test Cases	

2.8 영역 잘라내기

2.8.1 Parameter Values

isSelected	start_x_location	start_y_location	end_x_location	end_y_location	cur_cut_num
TRUE	0	0	0	0	1
	50	50	100	100	others

2.8.2 Pairwise Combination

case	start_x_location	start_y_location	end_x_location	end_y_location	cur_cut_num	pairings
1	0	0	0	0	1	10
2	0	50	100	100	others	10
3	50	0	100	0	others	8
4	50	50	0	100	1	8
5	~0	0	0	100	others	2
6	~0	50	100	0	1	2

2.8.3 Summary of Pairwise Combination

Category-Partiton	Pairwise Combination
Step 1 : 486 Test Cases	6 Test Cases
Step 2 : 32 Test Cases	
Step 3 : 32 Test Cases	

2.9 영역 붙여넣기

2.9.1 Parameter Values

isClipboard	area_width	area_height	image_width	image_height	start_x_location	start_y_location	cur_cut_num
TRUE	300	300	200	200	0	0	1
	others	others	400	400	50	50	others
			others	others			

2.9.2 Pairwise Combination

case	isClipboard	area_width	area_height	image_width	image_height	start_x_location	start_y_location	cur_cut_num	pairings
1	TRUE	300	300	200	200	0	0	1	28
2	TRUE	others	others	400	400	50	50	others	28
3	TRUE	300	others	others	others	0	50	1	19
4	~TRUE	others	300	400	200	50	0	others	12
5	~TRUE	300	300	200	400	50	50	1	9
6	~TRUE	others	300	others	others	0	0	others	10
7	~TRUE	others	others	200	200	~0	50	others	5
8	~TRUE	300	others	400	400	0	0	1	6
9	~TRUE	300	~others	200	others	50	~0	others	3
10	~TRUE	others	~300	others	200	50	~50	1	3
11	~TRUE	~300	~300	others	400	~0	~0	~others	1
12	~TRUE	~others	~300	400	others	~50	~50	~1	1

2.9.3 Summary of Pairwise Combination

Category-Partiton	Pairwise Combination
Step 1 : 7,776 Test Cases	12 Test Cases
Step 2 : 288 Test Cases	
Step 3 : 288 Test Cases	

2.10 컷 삭제하기

2.10.1 Parameter Values

total_cut_size	cur_cut_num
1	1
others	others

2.10.2 Pairwise Combination

case	total_cut_size	cur_cut_num	pairings
1	1	1	1
2	1	others	1
3	others	1	1
4	others	others	1

2.10.3 Summary of Pairwise Combination

Category-Partiton	Pairwise Combination
Step 1 : 9 Test Cases	4 Test Cases
Step 2 : 4 Test Cases	
Step 3 : 4 Test Cases	

2.11 컷 전환하기

2.11.1 Parameter Values

total_cut_size	cur_cut_num	dst_cut_num
1	1	1
others	others	others

2.11.2 Pairwise Combination

case	total_cut_size	cur_cut_num	dst_cut_num	pairings
1	1	1	1	3
2	1	others	others	3
3	others	1	others	3
4	others	others	1	3

2.11.3 Summary of Pairwise Combination

Category-Partiton	Pairwise Combination
Step 1 : 27 Test Cases	4 Test Cases
Step 2 : 8 Test Cases	
Step 3 : 8 Test Cases	

2.12 컷 크기조절하기

2.12.1 Parameter Values

cut_width	cut_height	dst_cut_width	dst_cut_height	cur_cut_num
300	300	0	0	1
others	others	200	200	others
		400	400	
		others	others	

2.12.2 Pairwise Combination

case	cut_width	cut_height	dst_cut_width	dst_cut_height	cur_cut_num	pairings
1	300	300	0	0	1	10
2	others	others	0	200	others	10
3	others	300	200	0	others	8
4	300	others	200	200	1	8
5	300	300	400	400	others	8
6	others	others	400	others	1	8
7	others	others	others	400	1	7
8	300	300	others	others	others	7
9	~300	others	0	0	~others	1
10	~others	300	0	200	~1	1
11	~300	~others	200	400	~1	1
12	~others	~300	200	others	~others	1
13	~others	~others	400	0	~1	1
14	~300	~300	400	200	~others	1
15	~300	~others	others	0	~others	1
16	~others	~300	others	200	~1	1
17	~others	~300	0	400	~others	1
18	~300	~others	0	others	~1	1

2.12.3 Summary of Pairwise Combination

Category-Partiton	Pairwise Combination
Step 1 : 432 Test Cases	18 Test Cases
Step 2 : 128 Test Cases	
Step 3 : 128 Test Cases	

2.13 컷 합치기

2.13.1 Parameter Values

total_cut_size	sum_image_width	sum_image_height
1	600	600
others	others	others

2.13.2 Pairwise Combination

case	total_cut_size	sum_image_width	sum_image_height	pairings
1	1	600	600	3
2	1	others	others	3
3	others	600	others	3
4	others	others	600	3

2.12.3 Summary of Pairwise Combination

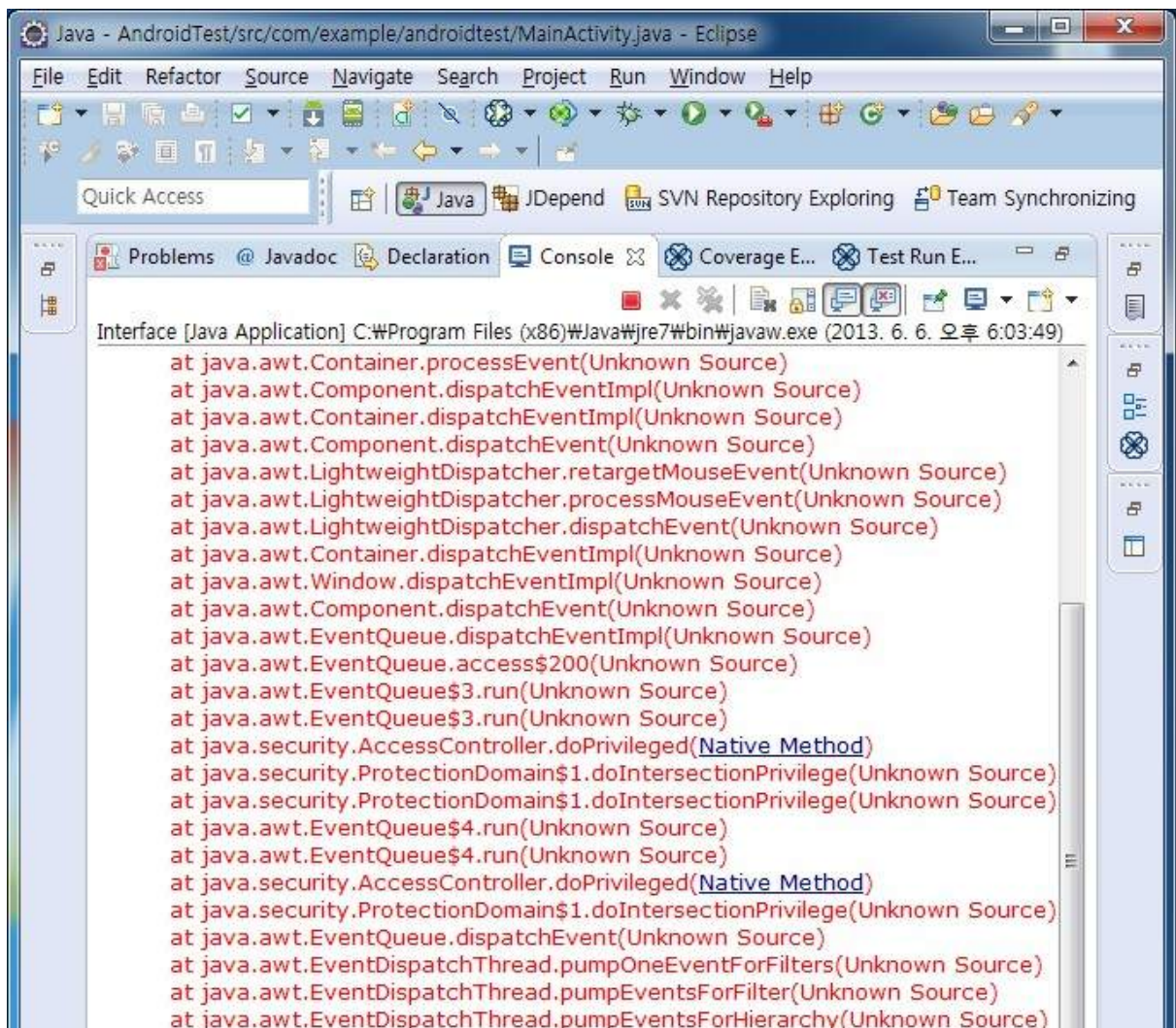
Category-Partiton	Pairwise Combination
Step 1 : 27 Test Cases	4 Test Cases
Step 2 : 8 Test Cases	
Step 3 : 8 Test Cases	

3. Pairwise Testing Result

3.1 파일 저장하기

num	category	pass/fail	case	file_type	image_width	image_height	total_cut_size	pairings
1	파일 저장하기 (file_save)	middle	1	bmp	600	600	0	6
2		middle	2	png	others	others	0	6
3		fail	3	bmp	others	600	1	5
4		pass	4	png	600	others	1	5
5		fail	5	bmp	600	others	others	4
6		pass	6	png	others	600	others	4

[Middle] Test Case 파일 저장하기의 1, 2 번 Case 의 경우 생성된 컷이 하나도 없을 때 파일 저장하기 기능을 수행하게 되면 Eclipse 상에서 Error 가 발생한다. 이 현상에 대한 Error 처리가 필요하다고 본다.

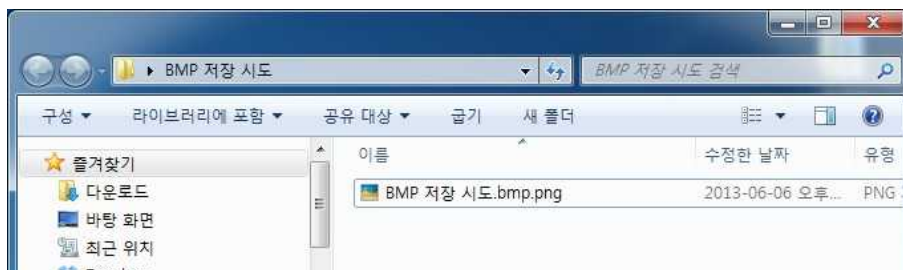
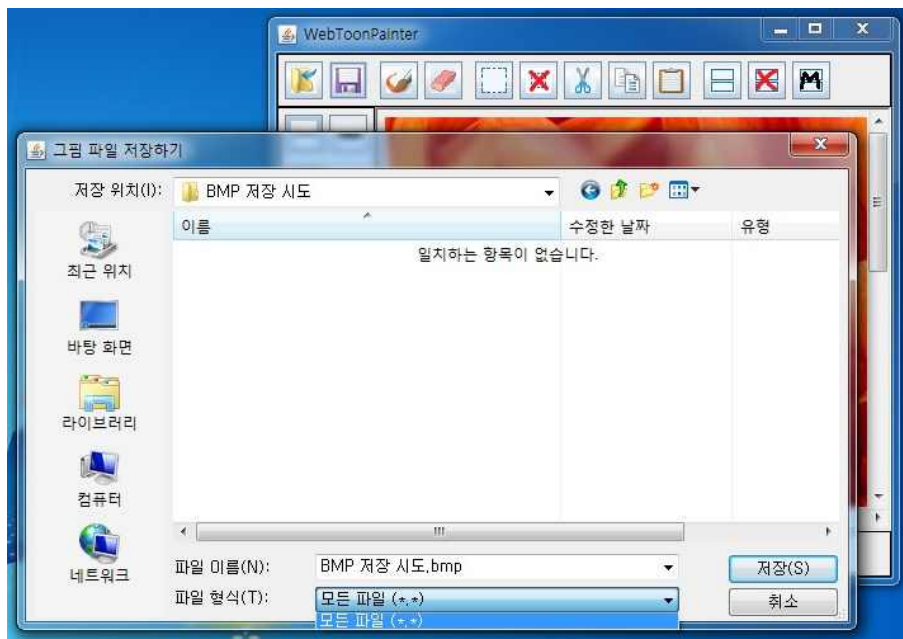


[Error] OSP Stage 1000 보고서 v4 에서 Activity 1003. Define Requirements, Functional Requirements 파일 처리 부분에서 BMP, PNG 등의 이미지 파일을 불러오고 저장한다고 명시하고 있지만 프로그램 상에서는 BMP, PNG 등의 이미지 파일을 불러오는 것은 정상적으로 작동하지만 이미지 파일을 저장하는 것은 PNG 이미지 파일 형식으로 고정되어 있어서 BMP 이미지 파일 형식으로 저장할 수 없다.

Activity 1003. Define Requirements

- Functional Requirements

- 파일 처리
 - BMP, PNG 등의 이미지 파일을 불러오고 저장한다
- 기본적인 이미지 편집
 - 브러시 도구로 선을 그린다
 - 새사 선택 기능은 통해 브러시 색의 색을 바꾸다



위 그림을 보면 이미지 파일 저장 시 BMP 이미지 파일 형식으로 저장을 하였지만 실제 저장된 파일을 보면 PNG 이미지 파일 형식으로 고정되어 있는 것을 볼 수 있다.

3.2 파일 불러오기

	category	pass/fail	case	file_type	image_width	image_height	total_cut_size	pairings
7	파일 불러오기 (file_load)	pass	1	bmp	600	600	0	6
8		pass	2	png	others	others	0	6
9		pass	3	bmp	others	600	1	5
10		pass	4	png	600	others	1	5
11		pass	5	bmp	600	others	others	4
12		pass	6	png	others	600	others	4

3.3 브러시

	category	pass/fail	case	thickness	color	start_x_location	start_y_location	end_x_location	end_y_location	cur_cut_num	pairings
13	브러시	pass	1	0	16 colors	0	0	0	0	1	21
14	(brush_use)	pass	2	1	16 colors	1	1	1	1	others	21

3.4 지우개

	category	pass/fail	case	thickness	start_x_location	start_y_location	end_x_location	end_y_location	cur_cut_num	pairings
15	지우개 (eraser_use)	pass	1	0	0	0	0	0	1	15
16		pass	2	0	1	1	1	1	others	15
17		pass	3	1	0	1	0	1	1	11
18		pass	4	1	1	0	1	0	others	11

3.5 영역 선택하기

	category	pass/fail	case	start_x_location	start_y_location	end_x_location	end_y_location	cur_cut_num	pairings
19	영역 선택하기 (area_select)	pass	1	0	0	0	0	1	10
20		pass	2	0	50	100	100	others	10
21		pass	3	50	0	100	0	others	8
22		pass	4	50	50	0	100	1	8

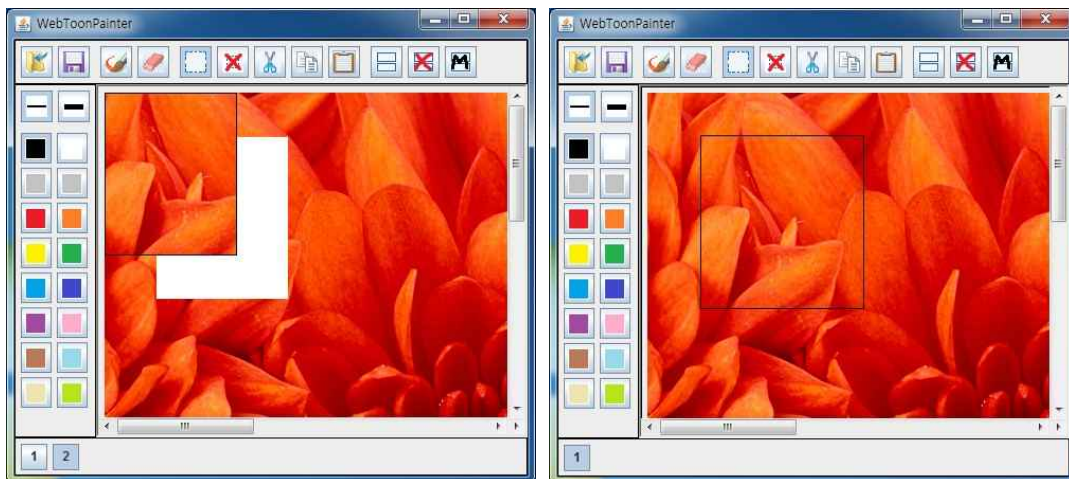
3.6 영역 복사하기

	category	pass/fail	case	start_x_location	start_y_location	end_x_location	end_y_location	cur_cut_num	pairings
23		middle	1	0	0	0	0	1	10
24	영역 복사하기 (area_copy)	middle	2	0	50	100	100	others	10
25		middle	3	50	0	100	0	others	8
26		middle	4	50	50	0	100	1	8

[Middle] OSP Stage 2030 보고서 v4 및 2040, 2050 보고서 v3 에서 Activity 2041. Design Real Use Case R3.4 영역 복사하기 부분을 보면 선택된 영역을 클립보드에 저장한다고 되어 있다. 프로그램 수행 시 영역을 복사하기 위해서 영역을 선택하게 되는데 선택된 영역을 해제하지 않고 영역 붙여넣기 기능을 수행 시 선택된 영역이 삭제된다. 영역이 선택된 상태로 영역 붙여넣기를 수행하면 영역이 잘라내기 수행 후 붙여넣기가 되고, 영역이 선택되지 않은 상태로 영역 붙여넣기를 수행하면 정상적으로 영역 복사하기 기능이 수행된다. 영역이 선택되어 있더라도 영역 붙여넣기 수행 시 선택된 영역을 삭제하지 않고, 영역 선택만 해제되도록 하고, 정상적으로 영역 붙여넣기 수행되어야 하는 것으로 보인다.

Activity 2041. Design Real Use Case

Use Case	R3.4 영역 복사하기
Actors	사용자
Purpose	선택된 영역을 클립보드에 저장한다.
Overview	사용자가 영역 복사하기를 하면 선택된 영역이 클립보드에 복사된다.
Type	Primary
Cross Reference	N/A
Pre-Requisites	영역이 선택되어 있어야 한다.
UI Widgets	Window-1
Typical Courses of Events	(A) : 사용자, (S) : 시스템 1. (A) 영역 복사 버튼(H)를 누름 2. (S) 선택된 영역을 클립보드에 복사
Alternative Courses of Events	N/A
Exceptional Courses of Events	N/A



3.7 영역 잘라내기

	category	pass/fail	case	start_x_location	start_y_location	end_x_location	end_y_location	cur_cut_num	pairings
27	영역 잘라내기 (area_crop)	pass	1	0	0	0	0	1	10
28		pass	2	0	50	100	100	others	10
29		pass	3	50	0	100	0	others	8
30		pass	4	50	50	0	100	1	8

3.8 영역 붙여넣기

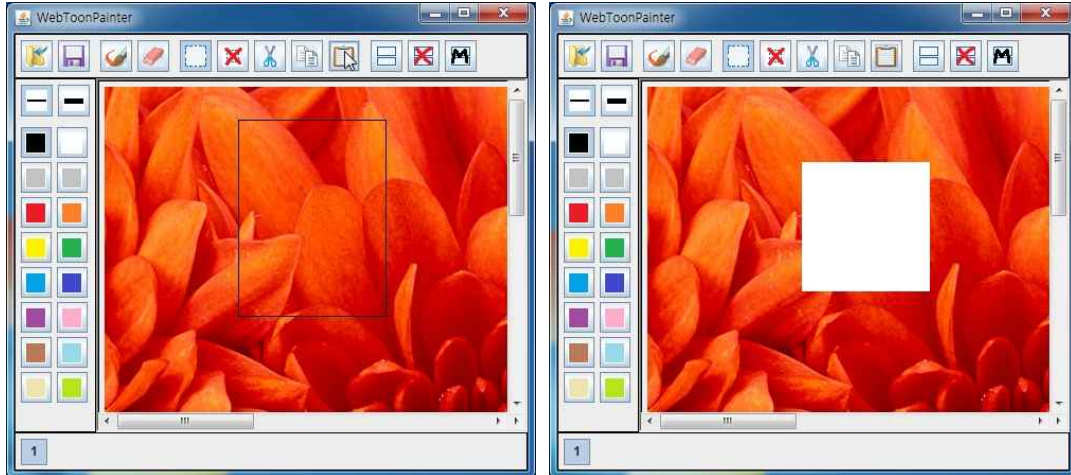
	category	pass/fail	case	isClipboard	area_width	area_height	image_width	image_height	start_x_location	start_y_location	cur_cut_num	pairings
31	영역 붙여넣기 (area_paste)	middle	1	TRUE	300	300	200	200	0	0	1	28
32		middle	2	TRUE	others	others	400	400	50	50	others	28
33		middle	3	TRUE	300	others	others	others	0	50	1	19

[Middle] OSP Stage 2030 보고서 v4 및 2040, 2050 보고서 v3 에서 Activity 2041. Design Real Use Case R3.5 영역 붙여넣기 부분을 보면 클립보드에 저장된 이미지가 화면에 복사된다고 되어 있다. 프로그램 수행 시 영역을 선택한 상태로 영역 붙여넣기 기능을 수행하면 선택된 영역의 이미지는 삭제되고 클립보드에 있는 이미지가 붙여넣기 된다. 클립보드에 이미지가 없어도 선택된 영역의 이미지는 삭제되고 클립보드에 이미지가 없기 때문에 붙여넣기 되는 이미지는 없게 된다. 영역이 선택되어 있더라도 영역 붙여넣기 수행 시 선택된 영역을 삭제하지 않고, 영역 선택만 해제되도록 하고, 정상적으로 영역 붙여넣기 수행되어야 하는 것으로 보인다.

이 내용이 영역 복사하기 기능 수행 시 발생하는 Error 와 같다.

Activity 2041. Design Real Use Case

Use Case	R3.5 영역 붙여넣기
Actors	사용자
Purpose	클립보드에 저장된 이미지를 붙여 넣음.
Overview	사용자가 붙여 넣기를 누르면 클립보드에 저장된 이미지가 화면에 복사됨.
Type	Primary
Cross Reference	N/A
Pre-Requisites	클립보드에 저장된 이미지가 있어야 함.
UI Widgets	Window-1
Typical Courses of Events	(A) : 사용자, (S) : 시스템 1. (A) 영역 붙여넣기 버튼(I) 를 누름 2. (S) 기존의 영역선택 해제 3. (S)(Q)의 최좌측상단에 클립보드의 이미지 크기만큼 영역(R)을 생성 4. (S)(R)의 picture에 클립보드의 이미지를 복사함.
Alternative Courses of Events	N/A
Exceptional Courses of Events	Line 2. 저장된 이미지가 컷보다 크면 영역크기조절을 통해 컷 최대크기로 조절



3.9 영역 삭제하기

[Error] OSP Stage 2030 보고서 v4 및 2040, 2050 보고서 v3 에서 Activity 2041. Design Real Use Case R3.2 영역 삭제하기 부분을 보면 사용자가 del 키를 누르면 선택된 영역이 지워진다고 되어 있다. 하지만 영역이 선택된 상태에서 del 키를 눌러도 영역이 삭제되지 않는다.

Activity 2041. Design Real Use Case	
Use Case	R3.2 영역 삭제
Actors	사용자
Purpose	선택된 영역을 삭제한다.
Overview	사용자가 del키를 누르면 선택된 영역이 지워진다.
Type	Primary
Cross Reference	N/A
Pre-Requisites	영역이 선택되어 있어야 함.
UI Widgets	Window-1
Typical Courses of Events	(A) : 사용자, (S) : 시스템 1. (A) 영역 선택 후 버튼(F)를 누름 2. (S) 선택영역 (R)이 지워지고 하얀 공백이 생김. 3. (S) 선택영역이 해제됨.
Alternative Courses of Events	N/A
Exceptional Courses of Events	N/A

3.10 컷 삭제하기

	category	pass/fail	case	total_cut_size	cur_cut_num	pairings
34	컷 삭제하기 (cut_delete)	pass	1	1	1	1
35		pass	2	1	others	1
36		pass	3	others	1	1
37		pass	4	others	others	1

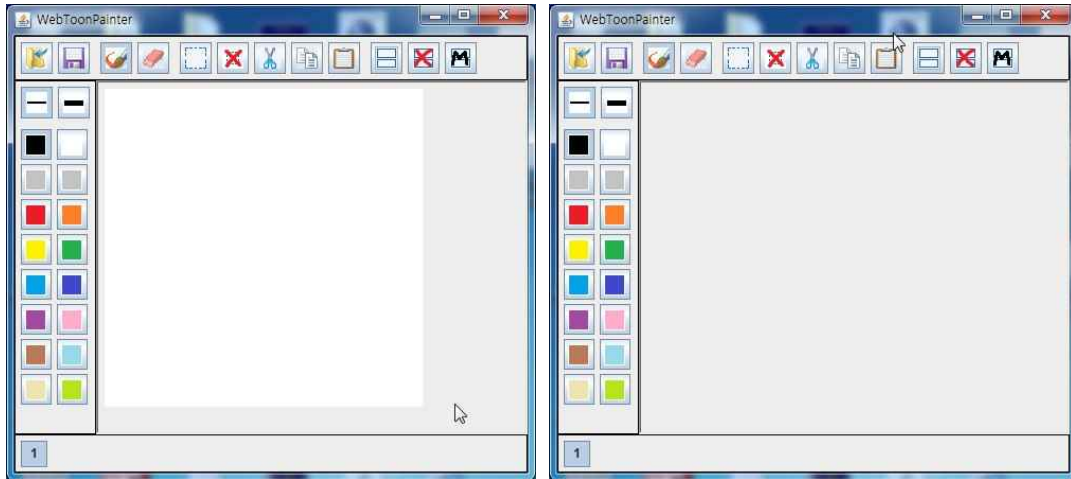
3.11 컷 전환하기

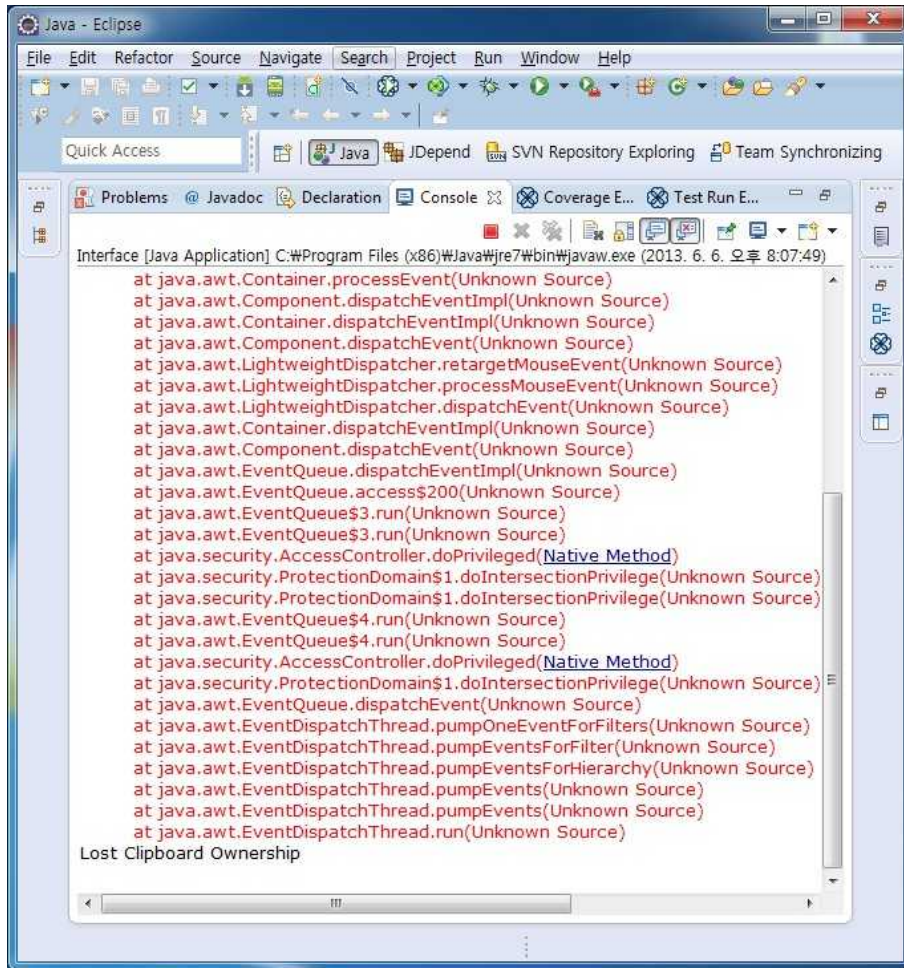
	category	pass/fail	case	total_cut_size	cur_cut_num	dst_cut_num	pairings
38	컷 전환하기 (cut_switch)	pass	1	1	1	1	3
39		pass	2	1	others	others	3
40		pass	3	others	1	others	3
41		pass	4	others	others	1	3

3.12 컷 크기조절하기

	category	pass/fail	case	cut_width	cut_height	dst_cut_width	dst_cut_height	cur_cut_num	pairings
42	컷 크기조절하기 (cut_resize)	fail	1	300	300	0	0	1	10
43		fail	2	others	others	0	200	others	10
44		fail	3	others	300	200	0	others	8
45		pass	4	300	others	200	200	1	8
46		pass	5	300	300	400	400	others	8
47		pass	6	others	others	400	others	1	8
48		pass	7	others	others	others	400	1	7
49		pass	8	300	300	others	others	others	7

[error] 컷 크기 조절 시 Width 또는 Height 값을 하나라도 0 값으로 조절하게 되면 Eclipse 상에서 Error 가 발생합니다.





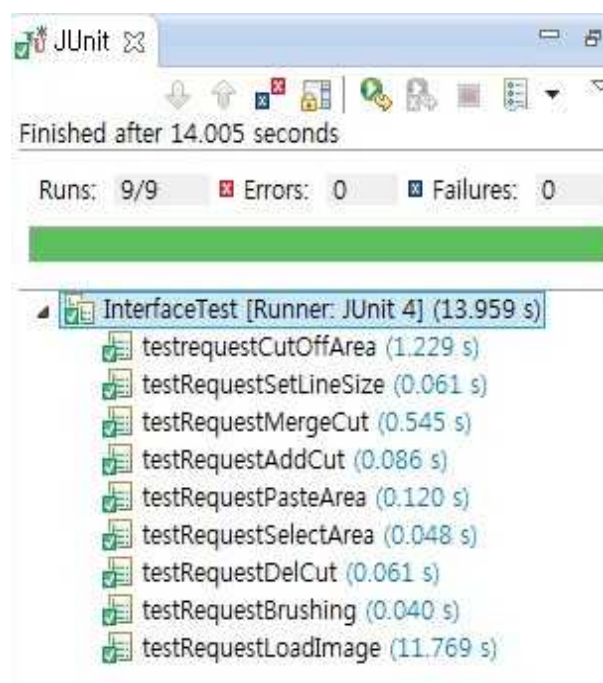
3.13 컷 합치기

category	pass/fail	case	total_cut_size	sum_image_width	sum_image_height	pairings
50	pass	1	1	600	600	3
51	pass	2	1	others	others	3
52	pass	3	others	600	others	3
53	pass	4	others	others	600	3

4. Clover

Code Coverage 를 측정하는 도구인 'Clover'를 통해 'SMA Team 2'가 수행한 Test Case 의 결과에 대한 Coverage 를 분석해보았다.

4.1 Clover Result



SMA Team 2 에서 자체적으로 수행한 JUnit Test 모두 성공하였다. 그리고 해당 Test Case 에 대하여 Clover 를 수행한 결과는 다음과 같았다.

Elem	Cov%	Av Me Cpx	Cpx
Painter			
(default package)	57.5%	1.9	275.0
Area.java	45.0%	1.2	22.0
AreaTool.java	43.9%	1.3	41.0
Brush.java	33.3%	1.0	3.0
Cut.java	88.2%	1.0	4.0
DrawingTool.java	60.0%	1.0	4.0
Interface.java	52.8%	2.0	145.0
InterfaceTest.java	100.0%	2.3	23.0
Project.java	33.6%	6.6	33.0

수행 결과, 전체 Code Coverage 비율은 57.5%로 프로젝트 전체에 대한 Test 가 수행되었다고 보기엔 부족했다. 만들어진 기능을 호출하는 클래스인 Cut, DrawingTool, Interface 에 대한 Code Coverage 은 대체적으로 높았던 것에 반해, 실질적인 세부기능은 Test 되지 않았다고 판단했다.

Coverage 비율이 가장 높은 것은 잘라내기 기능을 수행하는 Cut 클래스였고, Coverage 비율이 가장 낮은 것은 그림판 프로젝트 자체를 수행하는 메인 클래스인 Project 클래스였다.

4.2 Code Coverage Analysis

4.2.1. Area.java

Method	Code Coverage	Line Count	Branch Count
Area	45.0%	1.2	22.0
Area()	100.0%	-	1.0
copyImg()	0.0%	-	1.0
deepCopy(BufferedImage)	0.0%	-	1.0
getEndX()	0.0%	-	1.0
getEndY()	0.0%	-	1.0
getHeight()	66.7%	-	2.0
getImg()	100.0%	-	1.0
getStartX()	100.0%	-	1.0
getStartY()	100.0%	-	1.0
getWidth()	66.7%	-	2.0
pasteImg(BufferedImage)	100.0%	-	1.0
setArea()	0.0%	-	3.0
setArea(int, int, int, int)	100.0%	-	1.0
setEndX(int)	0.0%	-	1.0
setEndY(int)	0.0%	-	1.0
setImg(BufferedImage)	0.0%	-	1.0
setStartX(int)	0.0%	-	1.0
setStartY(int)	0.0%	-	1.0

Area Class 의 경우, 전체 Method 18 개 중 Coverage 비율이 60% 이상인 Method 는 8 개였고, Statement Coverage 는 전체 54 개 중 26 개였다. 그 중 분기문에 해당하는 Branch 는 8 개였는데 Cover 된 분기문은 getHeight()함수와 getWidth()함수에서 단 2 개에 불과했다. 따라서, Statement Coverage 는 48%, Branch Coverage 는 25%였다.

Cover 가 되지 않은 대부분의 Statements 는 Setter, Getter Method 에 속한 Statement 였으므로, 해당 프로젝트에 Critical 하지 않다고 판단하였으며, 이후에 2 차로 진행할 Clover Process 에서는 해당 Statements 를 예외처리하기로 판단했다.

4.2.2 AreaTool.java

File/Method	Coverage	Count	Score
AreaTool.java	43.9%	1.3	41.0
AreaTool	39.5%	1.2	31.0
CopyImagetoClipboard	100.0%	1.0	2.0
AreaTool()	100.0%	-	1.0
checkAreaOn(int, int)	0.0%	-	5.0
clearArea()	100.0%	-	1.0
copyArea(Cut, int, int)	100.0%	-	1.0
delArea(Cut)	0.0%	-	1.0
getAreaStat()	100.0%	-	1.0
getClipboard()	100.0%	-	3.0
getEndX()	0.0%	-	1.0
getEndY()	0.0%	-	1.0
getHeight()	0.0%	-	1.0
getImg()	0.0%	-	1.0
getMovable()	0.0%	-	1.0
getStartX()	0.0%	-	1.0
getStartY()	0.0%	-	1.0
getVisible()	0.0%	-	1.0
getWidth()	0.0%	-	1.0
moveArea(int, int)	0.0%	-	1.0
selectArea(Cut)	100.0%	-	1.0
setArea(int, int, int, int)	100.0%	-	1.0
setArea()	0.0%	-	1.0
setClipboard()	100.0%	-	1.0
setCurrentXY(int, int)	0.0%	-	1.0
setEndArea(int, int)	0.0%	-	1.0

4.2.2.1. AreaTool Class

AreaTool Class 의 경우, 전체 Method 25 개 중 Cover 된 Method 는 8 개였고, Statement Coverage 는 전체 57 개 중 26 개였다, 그 중 분기문에 해당하는 Branch 는 6 개였고 모두 Test 되지 않았다. 따라서, Statement Coverage 는 45.6%였다.

해당 클래스도 Area Class 와 마찬가지로 Test 되지 않은 대부분의 Code 가 Setter, Getter Method 였지만, checkAreaOn()과 moveArea() method 의 경우, 주요 기능의 핵심 method 이므로 해당 method 에 대한 Unit Test Case 가 필요할 것으로 보인다.

4.2.2.2 AreaTool.CopyImagetoClipboard

해당 클래스는 Code Coverage 는 100%였다.

4.2.2.3 AreaTool.CopyImagetoClipboard.TransferableImage

해당 클래스에서 Test 가 되지 않은 것은 4 개의 Branch 문이었는데, 그 중 2 개는 Exception 에 대한 분기문이므로 2 차 Process 에 제외하고 isDataFlavorSupported() 내부의 분기문에 대한 Test Case 가 필요할 것으로 보인다.

4.2.3 Brush.java

AreaTool.java		43.9%	1.3	41.0
Brush.java		33.3%	1.0	3.0
Brush		33.3%	1.0	3.0
Brush()		100.0%	-	1.0
getColor()		0.0%	-	1.0
setColor(int)		0.0%	-	1.0

Brush Class 의 경우, 전체 Method 3 개 중 1 개, 전체 Statements 3 개 중 1 개에만 Test Case 가 수행되었지만, 나머지는 Setter, Getter 함수이므로 2 차 Process 에서 제외했다.

4.2.4 Cut.java

Cut.java		88.2%	1.0	4.0
Cut		88.2%	1.0	4.0
Cut(int)		100.0%	-	1.0
LoadCut(String)		100.0%	-	1.0
SaveCut(String)		0.0%	-	1.0
setXY(int, int)		100.0%	-	1.0

Cut Class 의 경우, 전체 Method 4 개 중 3 개, 전체 Statements 13 개 중 12 개에 대하여 Unit Test 가 수행되었다. Cover 되지 않은 SaveCut()의 경우, LoadCut() 함수와 연계되는 함수이므로, Coverage 에 문제가 되지 않는다고 판단하였다.

4.2.5 DrawingTool.java

DrawingTool.java		60.0%	1.0	4.0
DrawingTool		60.0%	1.0	4.0
DrawingTool()		100.0%	-	1.0
getLineSize()		100.0%	-	1.0
setColor(int)		0.0%	-	1.0
setLineSize(int)		0.0%	-	1.0

DrawingTool Class 의 경우, 전체 Method 4 개 중 2 개, 전체 Statements 6 개 중 4 개에 대하여 Unit Test 가 수행되었다. 이 클래스도 마찬가지로 Cover 되지 않은 Statements 는 Setter, Getter Method 이므로 2 차 프로세스에서 생략할 것이다.

4.2.6 Interface.java

File/Method	Coverage	Branch	Statements
Interface.java	52.8%	2.0	145.0
Interface	52.8%	2.0	145.0
addPopup(Component, JPopupMenu)	0.0%	-	1.0
deepCopy(BufferedImage)	0.0%	-	1.0
Interface()	100.0%	-	1.0
main(String[])	0.0%	-	1.0
paint(Graphics)	0.0%	-	5.0
requestAddCut()	100.0%	-	3.0
requestBrushing()	57.1%	-	2.0
requestChangeCut()	0.0%	-	3.0
requestCopyArea()	75.0%	-	2.0
requestCutOffArea()	83.3%	-	2.0
requestDelArea()	0.0%	-	2.0
requestDelCut()	88.4%	-	10.0
requestErasing()	0.0%	-	2.0
requestLoadImage()	93.8%	-	2.0
requestMergeCut()	93.3%	-	4.0
requestMoveArea()	0.0%	-	1.0
requestPasteArea()	0.0%	-	1.0
requestSaveImage()	0.0%	-	2.0
requestSelectArea()	100.0%	-	1.0
requestSetColor(int)	0.0%	-	1.0
requestSetCutSize()	0.0%	-	1.0
requestSetLineSize(int)	100.0%	-	3.0
showWindow1()	0.0%	-	1.0

Interface Class 에서는 전체 73 개 중 Coverage 가 60%가 넘어가는 Method 는 9 개이고, 전체 Statements 602 개 중 349 개가 Cover 되었고, 그 중 Branch 는 전체 104 개 중 54 개가 Cover 되었다. 따라서 Statements Coverage 는 58%, Branch Coverage 는 52%가 되었다.

이 중 View 의 동작에 해당되는 actionPerformed(ActionEvent) : void Method 들과 Mouse Action 들은 Unit Test 에 의미가 없으므로 2 차 Process 에서 제외할 것이다. Interface Class 의 경우 일부 핵심기능에 대한 Unit Test 가 없고(e.g) requestSaveImage(), requestErasing(), requestSetColor(), requestDelArea()...) paint()에 대한 Unit Test 가 아예 이루어지지 않았다.

4.2.7 Project.java











File/Method	Coverage	Branch	Statements
Project.java	33.6%	6.6	33.0
Project	33.6%	6.6	33.0
addCut()	100.0%	-	2.0
brushing(int, int, int, int, int)	0.0%	-	21.0
delCut()	90.0%	-	6.0
Project()	100.0%	-	1.0
setCutSize(int, int)	100.0%	-	3.0

마지막으로, Project Class 는 전체 Method 5 개 중 4 개, 전체 Statements 101 개 중 25 개에 대하여 Unit Test 가 수행되었고, 그 중 Branch 는 전체 22 개 중 2 개가 Cover 되었다. 따라서 Statements Coverage 는 24.7%, Branch Coverage 는 9%가 되었다.



이 중 Color 선정에 대한 Case 를 하나로 본다면 Branch 는 7 개로 줄어들고 줄어든 Branch 에 대한 Unit Test 가 필요할 것으로 판단된다.

4.3 Conclusion

4.3.1 Before

Elem	Cov%	Av Me Cpx	Cpx
Painter		1.9	275.0
(default package)		1.9	275.0
Area.java		1.2	22.0
AreaTool.java		1.3	41.0
Brush.java		1.0	3.0
Cut.java		1.0	4.0
DrawingTool.java		1.0	4.0
Interface.java		2.0	145.0
InterfaceTest.java		2.3	23.0
Project.java		6.6	33.0

4.3.2 After

Elem	Cov%	Av Me Cpx	Cpx
Painter		1.7	179.0
(default package)		1.7	179.0
Area.java		1.0	4.0
AreaTool.java		1.4	14.0
Brush.java		1.0	1.0
Cut.java		1.0	3.0
DrawingTool.java		1.0	1.0
Interface.java		1.6	120.0
InterfaceTest.java		2.4	22.0
Project.java		3.5	14.0

4.3.3 Conclusion

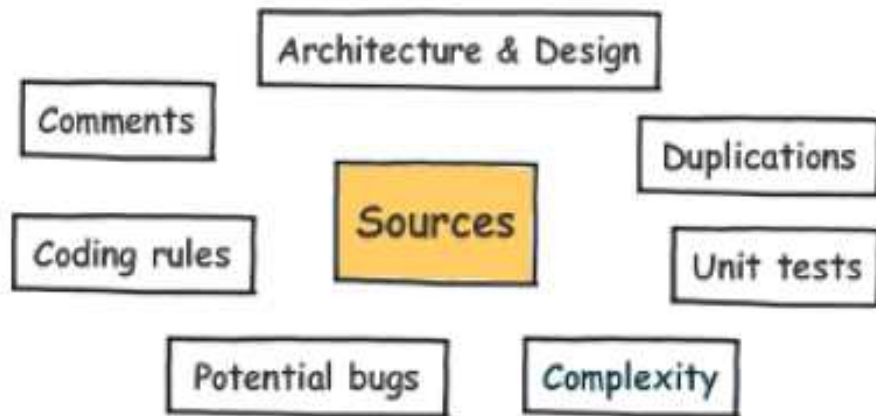
앞서 Code Coverage Analysis 에서 언급한 Setter, Getter Method 와 Unit Test 가 의미가 없는 Method 들을 Context Filter 를 통해 제외하고 Clover Process 를 진행한 결과, 위와 같은 결과가 나왔다. 분석 결과, 영역 선택과 관련 클래스들에 대한 기능들에 대한 Unit Test 와 Interface Class 에 있는 주요 기능에 대한 Unit Test 가 부족한 것으로 판단이 되었다.

5. Static Analysis - Sonar

5.1 What is Sonar?

Sonar 정적분석 도구는 소스 코드에 대한 전반적인 품질을 확보할 수 있도록 제공하는 통합플랫폼이다.

- Client/Server 구조
- C/C++/JAVA(기본지원) 등 다양한 프로그램 언어(+15) 지원
- 확장 모듈(Plugins) 설치로 다양한 도구와 유연한 통합
- 웹기반 애플리케이션으로 다양한 결과를 서버에 통합 관리 용이



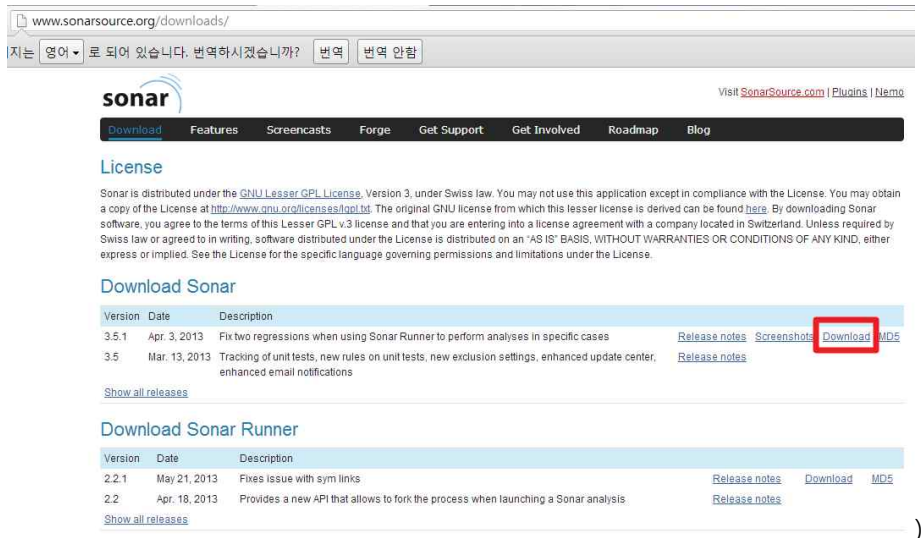
5.2 Install Sonar

5.2.1 Install Sonar Server

아래의 설치 과정은 JDK 가 설치되어 JAVA 환경이 구축되어 있다는 전제로 시작한다.

5.2.1.1 Sonar 사이트로 들어가서 Sonar Server 를 다운로드 한다.

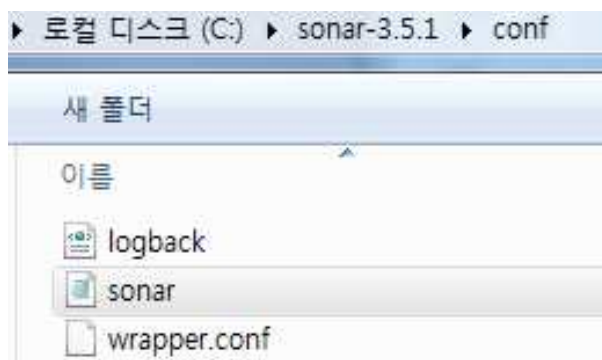
(<http://www.sonarsource.org/downloads/>)



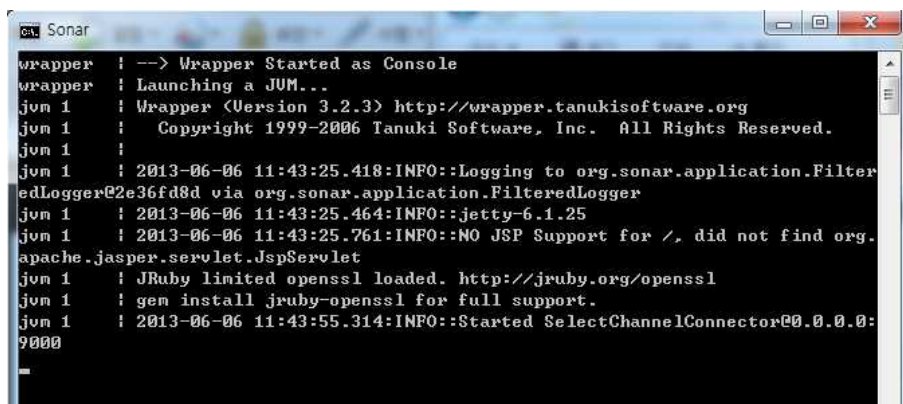
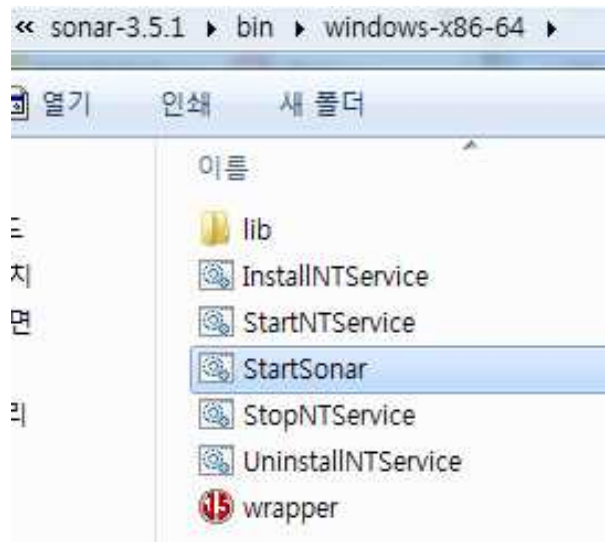
5.2.1.2 압출을 풀어 놓는다. (한글 경로가 들어가지 않도록 주의)



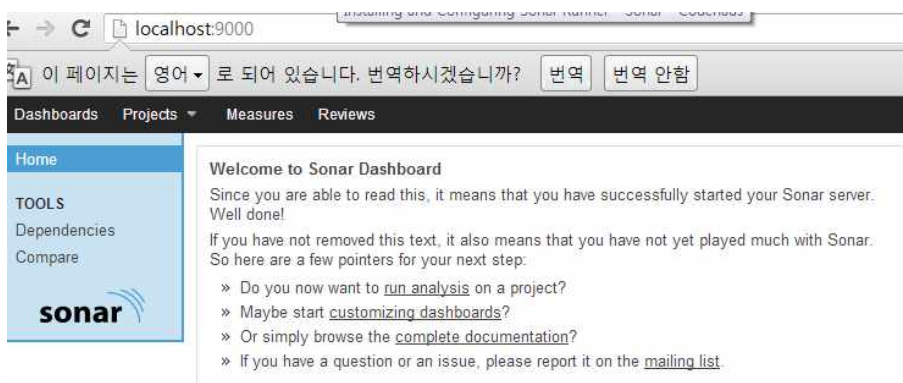
5.2.1.3 필요 시 설치경로(/conf/sonar.properties)를 수정한다.
(WAS, Database 설정 가능)



5.2.1.4 설치경로에 /bin/사용자플랫폼/StartSonar.bat 을 실행한다.



http://localhost:9000/ 접속하여 확인한다. <관리자 기본계정 admin/admin>



5.2.1 Install Sonar Client

5.2.1.4 <http://www.sonarsource.org/downloads/> 에서 Sonar Runner 다운로드.

Download Sonar Runner

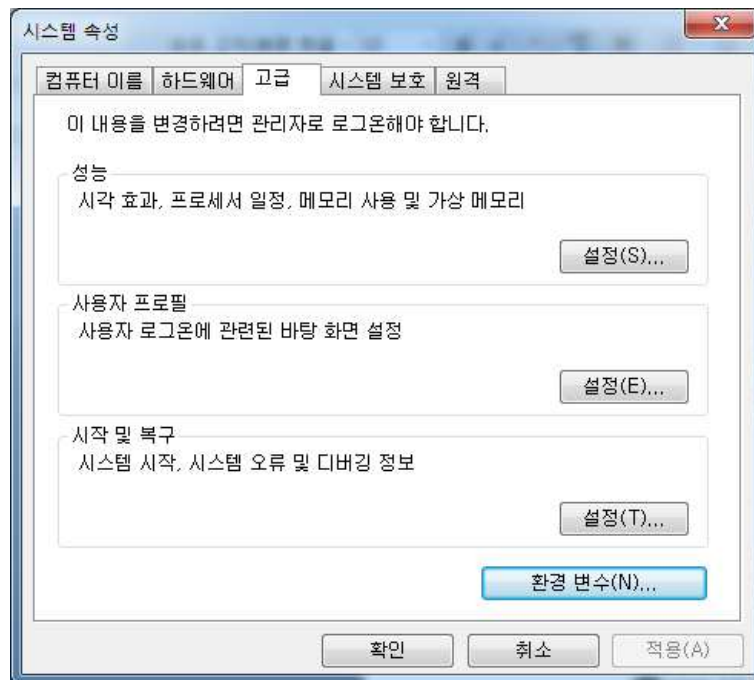
Version	Date	Description	Release notes	Download	MD5
2.2.1	May 21, 2013	Fixes issue with sym links	Release notes	Download	MD5
2.2	Apr. 18, 2013	Provides a new API that allows to fork the process when launching a Sonar analysis	Release notes		

[Show all releases](#)

5.2.1.4 설치와 프로젝트 설정한다.



압축을 푼다.



환경변수를 설정한다.

SONAR_RUNNER_HOME 환경변수 설정 후

시스템 변수 - Path 에 %SONAR_RUNNER_HOME%\bin; 추가한다.


```

# required metadata
sonar.projectKey=Painter
sonar.projectName=Painter
sonar.projectVersion=1.0

# optional description
sonar.projectDescription=Fake description

# path to source directories (required)
sonar.sources=src

# path to test source directories (optional)
sonar.tests=test

# path to project binaries (optional), for example directory of Java bytecode
sonar.binaries=bin

# optional comma-separated list of paths to libraries. Only path to JAR file is supported.
# sonar.libraries=path/to/library/*.jar,path/to/specific/library/myLibrary.jar,parent/*/*.jar

# The value of the property must be the key of the language.
sonar.language=java

# Additional parameters
sonar.my.property=value

```

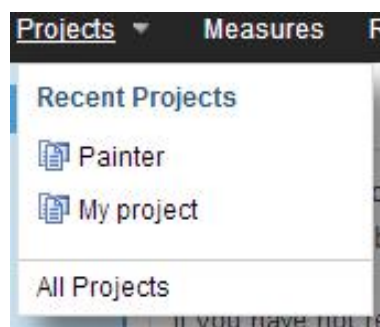
설치경로 /con/sonar-runner.properties 파일을 수정한다.

```

12:49:47.562 INFO - Persist graphs of components
12:49:47.654 INFO - ANALYSIS SUCCESSFUL, you can browse http://localhost:9000
12:49:47.656 INFO - Executing post-job class org.sonar.plugins.jmeter.JMeterPostJob
12:49:47.657 INFO - No JTL files found in target/jmeter-reportsnull
12:49:47.658 INFO - Executing post-job class org.sonar.plugins.core.batch.IndexProjectPostJob
12:49:47.836 INFO - Executing post-job class org.sonar.plugins.dbcleaner.ProjectPurgePostJob
12:49:47.866 INFO - -> Keep one snapshot per day between 2013-05-09 and 2013-06-05
12:49:47.869 INFO - -> Keep one snapshot per week between 2012-06-07 and 2013-05-09
12:49:47.869 INFO - -> Keep one snapshot per month between 2008-06-12 and 2012-06-07
12:49:47.870 INFO - -> Delete data prior to: 2008-06-12
12:49:47.884 INFO - -> Clean Painter [id=41]
INFO: -----
INFO: EXECUTION SUCCESS
INFO: -----
Total time: 25.280s
Final Memory: 20M/333M
INFO: -----
C:\Users\June\workspace\Painter>

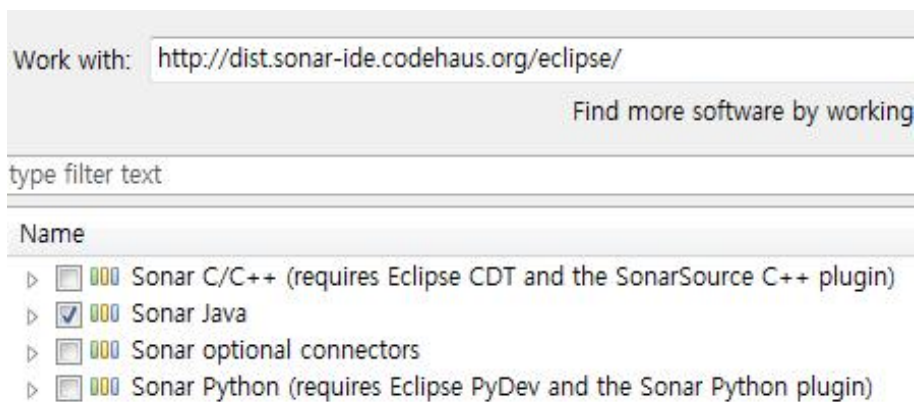
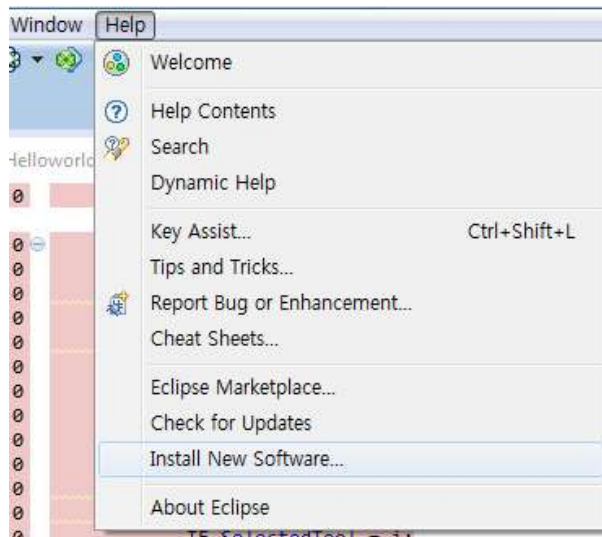
```

Command 창에서 sonar-runner 실행한다.

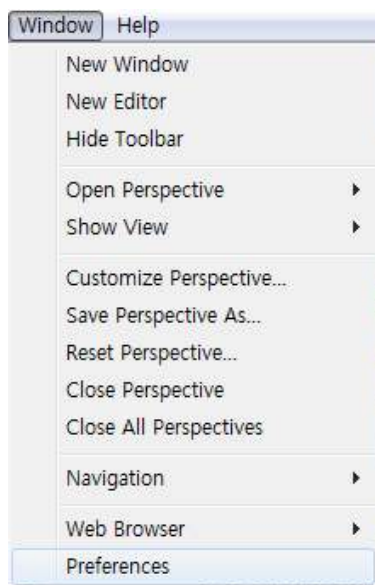


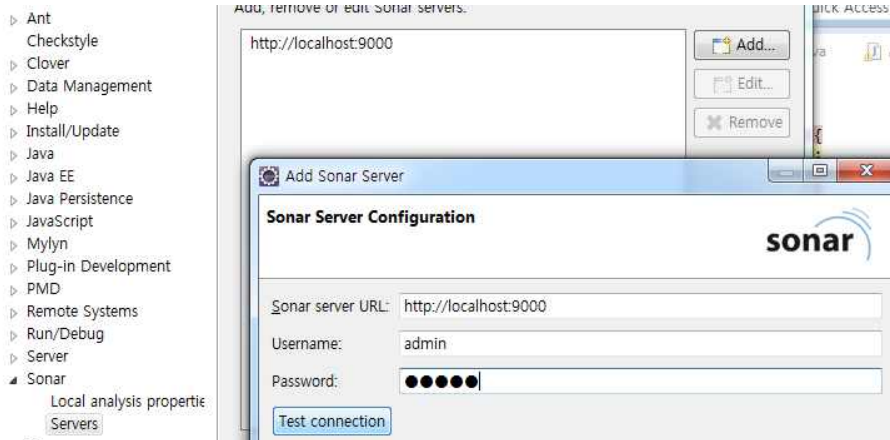
실행하면 프로젝트 conf 에 따라 서버에 프로젝트가 생성된다.

5.3 Eclipse Plugin 설치



Plugin 을 설치한다.



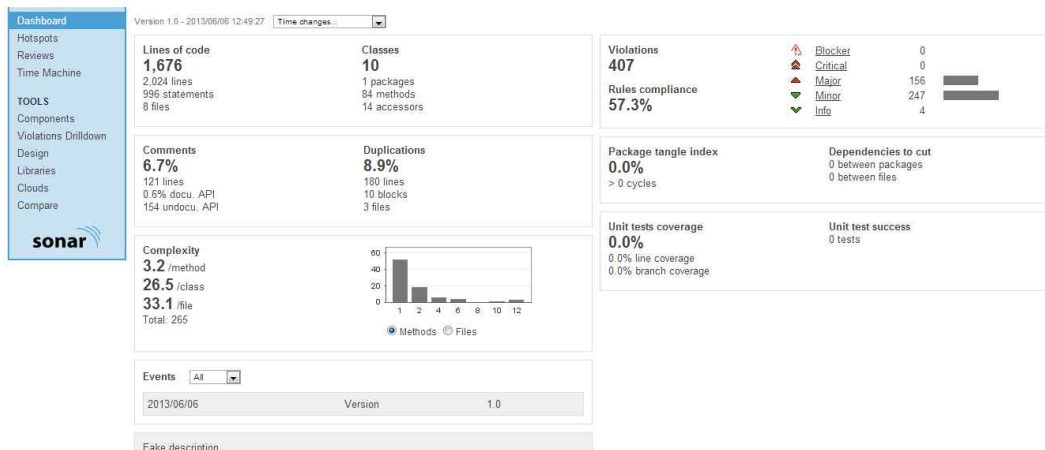


서버와의 연결을 체크한다.

5.4 Start Sonar



프로젝트 위에서 우 클릭 후 configure-associate with sonar 를 선택한다.



결과를 확인한다.

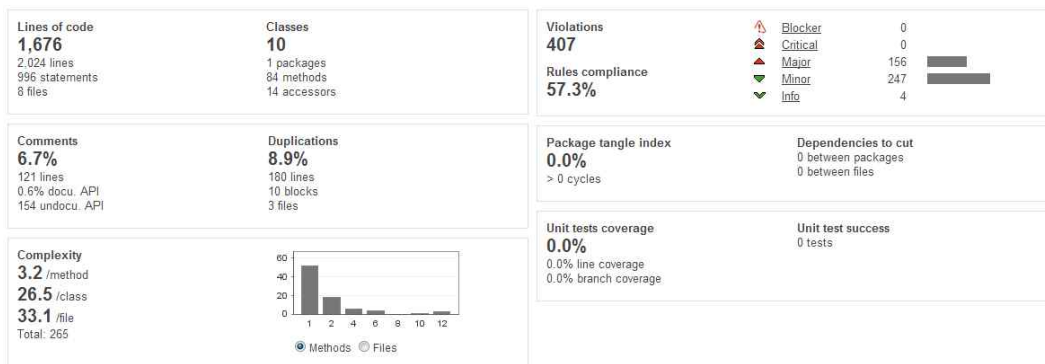


Eclipse IDE 에서도 결과를 확인할 수 있다.



계정을 추가한다.

5.5 Sonar Result



- Lines of code
코드의 라인, 주요 문구, 파일 등을 보여준다.
- Classes
패키지, 메소드, 메소드 유형을 보여준다.
- Comments
주석의 라인 수, API 수를 보여준다.
- Duplication
반복 라인, 블록, 파일의 수를 보여준다.
- Complexity
메소드별, 클래스별, 파일별 Complexity 를 보여준다.
- Violations& Rules compliance
룰에 걸리는 수치와 그에 따른 레벨별로 보여준다.
- Package tangle index
패키지 의존성 수치와 사이클 수를 보여준다.
- Unit test coverage
커버리지를 나타낸다.