

Webtoon Paint

Static Analysis Reaction Report
Final

Team 2

200911371 김민철

200911381 김진현

200911417 정명권

차례

1. About Name
 - 1. Variable Name
 - 2. Method Name
2. About Visibility Modifier
3. About If
 - 1. Simplify Boolean Expression
 - 2. If Stmts Must Use Braces
4. About Unused &
5. Comment
5. Others
 - 1. Anon Inner Length
 - 2. Hidden Field
 - 3. System Println & Avoid Print Stack Trace
 - 4. Cyclomatic Complexity
 - 5. NCSS Method Count

1. About Name

- 변수명을 소문자로 변경
- _가 들어가는 변수에서 _를 제거
 - button, button_ 로 시작하는 변수들을 각각의 기능에 맞는 변수 이름으로 변경
- 메소드명의 시작을 문자를 소문자로 변경

1. 1. Variable Name

Name	After Modifying
btnNewBUtton	loadbutton
btnNewButton_1	savebutton
btnNewButton_2	brushbutton
btnNewButton_3	eraserbuton
btnNewButton_4	areabutton
btnNewButton_5	delareabutton
btnNewButton_6	cutoffbutton
btnNewButton_7	copybutton
btnNewButton_8	pastebutton
btnNewButton_9	addcutbutton
btnNewButton_10	delcutbutton
btnNewButton_11	mergebutton
btnNewButton_13	sharpbutton
btnNewButton_14	thickbutton

1. 1. Variable Name (계속)

Name	After Modifying
Area	area
bef_x	befx
bef_y	befy
BrushConf	brushconf
C	c
CanvasList	canvaslist
CanvasPanel	canvaspanel
CB	cb
CI	ci
ColorGroup	colorgroup
CurrentGraphic	currentgraphic
CurrentImage	currentimage
Cut	cut
CutListGroup	cutlistgroup
CutListPanel	cutlistpanel
Draw	draw
EnabledCutNum	enabledcutnum

1. 1. Variable Name (계속)

Name	After Modifying
File	file
flowLayout_1	flowlayout1
flowLayout_2	flowlayout2
IF	inf
SelectedTool	selectedtool
SizeGroup	sizegroup
ToolGroup	toolgroup
ToolLeft	toolleft
ToolTop	tooltop

1. 1. Variable Name (계속)

Name	After Modifying
button	c1button
button_1	c2button
button_2	c3button
button_3	c4button
button_4	c5button
button_5	c6button
button_6	c7button
button_7	c8button
button_8	c9button
button_9	c10button
button_10	c11button
button_11	c12button
button_12	c13button
button_13	c14button
button_14	c15button
button_15	c16button
ButtonList	buttonlist

1. 2. Method Name

Name	After Modifying
LoadCut	loadCut
SaveCut	saveCut

2. About 'Visibility Modifier'

- 클래스를 캡슐화하기 위해 public 대신 private를 사용
 - Main 클래스의 필드들은 따로 캡슐화 하지 않음.
- 클래스 내부 변수에 접근하는 accessor 함수를 생성
 - 클래스 내부에서만 사용되는 필드들은 accessor 함수를 만들지 않음.
- 메소드 안에서 생성된 변수들은 accessor 함수를 만들지 않음
- List 등 자바에서 제공하는 자료구조는 accessor 함수를 만들지 않음

Class			
Interface	public int size;	클래스 내부에서만 사용되는 변수라 accessor 함수를 만들지 않음.	
Interface	public int color;		
Interface	public int SelectedTool;		
Interface	public int EnabledCutNum;		
Interface	public Project pj;		
Interface	public DrawingTool dt;		
Interface	public AreaTool at;		
Interface	public JPanel CanvasPanel;		
Interface	public JPanel CutListPanel;		
Interface	public List<Canvas> CanvasList;		
Interface	public List<JToggleButton> ButtonList;		
Interface	public ButtonGroup CutListGroup = new ButtonGroup();		
Interface	public ButtonGroup ToolGroup = new ButtonGroup();		
Interface	public ButtonGroup SizeGroup = new ButtonGroup();		
Interface	public ButtonGroup ColorGroup = new ButtonGroup();		
Interface	public Toolkit tk;		
Interface	public Cursor brush;		함수 내부에서만 사용되는 변수라 accessor 함수를 만들지 않음.
Interface	public Cursor eraser;		
Interface	int x;	자바에서 제공하는 자료구조라 accessor 함수를 만들지 않음.	
Interface	int y;		
Project	public List<Cut> cuts;		
Project	public int bef_x;	accessor 함수 생성.	
Project	public int bef_y;		
Project	public int thick;		
AreaTool	CopyImagetoClipboard CB;	클래스 내부에서만 사용 되는 변수라 accessor 함수를 만들지 않음.	
AreaTool	Image i;		
Cut	public int cutnum;	accessor 함수 생성	
Cut	public int width;		
Cut	public int height;		
Cut	public boolean cutstat		
Cut	public BufferedImage img;		
DrawingTool	public Brush brush;		
DrawingTool	public Brush eraser;		
DrawingTool	public int linesize;		
Brush	public int color;		

3. About 'IF'

- 조건식에 Boolean 변수가 들어갈 경우 불필요한 비교를 제거.
- IF문을 중괄호로 묶음.
- Else 문을 중괄호로 묶음.

3.1. Simplify Boolean Expression

- 조건식에 Boolean 변수가 들어갈 경우 불필요한 비교를 제거.

Violations	Simplify Boolean Expression (14)		
Line Num	Class		
211	Interface	if (ButtonList.get(i).isSelected() == true) {	
282	Interface	if (SelectedTool == 3 && at.checkAreaOn(e.getX(), e.get	
436	Interface	if (pj.cuts.get(i).cutstat == true) {	(조건식 == true) -> (조건식)
451	Interface	if (pj.cuts.get(i).cutstat == true) {	(조건식 == false) -> (!조건식) 으로 수정
471	Interface	if (pj.cuts.get(i).cutstat == true) {	
507	Interface	if (CanvasList.get(i).isVisible() == true)	
568	Interface	if (c.cutstat == true) {	
573	Interface	if (at.getAreaStat() == true) {	
1059	Interface	if (CanvasList.isEmpty() == false) {	
1061	Interface	if (CanvasList.get(i).isVisible() == true) {	
132	Project	if (cuts.get(i).cutstat == true) {	
151	Project	if (cuts.get(i).cutstat == true) {	
194	InterfaceTest	if (IF.pj.cuts.get(i).cutstat == true) {	
220	InterfaceTest	if (IF.pj.cuts.get(i).cutstat == true) {	

3.2. If Stmts Must Use Braces

- If 문을 중괄호로 묶음.
- Else 문을 중괄호로 묶음.

Violations	If Stmts Must Use Braces (4)		
Line Num	Class		
507	Interface	if (CanvasList.get(i).isVisible() == true) CanvasList.get(i).se	중괄호로 묶음.
532	Interface	if (pj.cuts.get(i).width > w) w = pj.cuts.get(i).width;	
801	Interface	if (pj.cuts.size() > 1) requestMergeCut();	
1071	Interface	if (e.getButton() == 1)	
Violations	If Else Stmts Must Use Braces (9)		
Line Num	Class		
442	Interface	else CanvasList.get(i).setVisible(false);	중괄호로 묶음.
475	Interface	if (i == ButtonList.size()) ButtonList.get(i-1).setSelected(true);	
477	Interface	else ButtonList.get(i).setSelected(true);	
1077	Interface	else CanvasPanel.setCursor(new Cursor(Cursor.DEFAULT));	
85	Project	if (size > 1) thick = 2;	
87	Project	else thick = 0;	
135	Project	if (i == cuts.size()) cuts.get(i-1).cutstat = true;	
137	Project	else cuts.get(i).cutstat = true;	
141	Project	else cuts.clear();	

4. About 'Unused & Comment'

- 사용하지 않는 메소드 제거.
 - 자동으로 생성된 메소드는 제거하지 않음.
- 사용하지 않는 필드 제거.
- 사용하지 않는 Import문과 주석을 제거.

4. About 'Unused & Comment'(계속)



Violations	Unused Private Field (1)		
Line Num	Class		
30	AreaTool	private int statonarea;	사용하지 않는 필드 제거
Violations	Unused Private Method (1)		
Line Num	Class		
1104	Interface	private static void addPopup(Component component, fir	GUI 관련 자동으로 만들어진 메소드여서 삭제하지 않음
Violations	Unused Imports (4)		
Line Num	Class		
	Interface	import java.awt.Image;	사용하지 않는 import 제거
	Interface	import javax.imageio.ImageIO;	
	Interface	import javax.swing.JScrollPane;	
	Interface	import javax.swing.JScrollBar;	
Violations	Avoid Commented-Out Lines Of Code (10)		
Line Num	Class		
133	Interface	//CanvasList.get(EnabledCutNum).setCursor(Cursor.getPr	사용하지 않는 code 제거
145	Interface	//CanvasList.get(EnabledCutNum).setCursor(Cursor.getPr	
240	Interface	//System.out.println("(" + e.getX0 + "," + e.getY0 + ")");	
321	Interface	//System.out.println("released st " + at.getStartX0 + " " +	
342	Interface	//System.out.println("(" + e.getX0 + "," + e.getY0 + ")");	
372	Interface	//System.out.println("dragg st " + at.getStartX0 + " " + a	
390	Interface	//System.out.println("(" + e.getX0 + "," + e.getY0 + ")");	
438	Interface	//System.out.println("EnabledCut:" + EnabledCutNum);	
596	Interface	/* switch(this.SelectedTool) {	
32	Project	//System.out.println(color);	

5. Others

- 익명 클래스의 라인 수가 지나치게 김.
→ 주로 이벤트에 관련된 부분인데, 새로운 함수를 만들어서 라인 수를 줄임.
- Image i 와 for 문의 i 가 겹침.
→ Image img로 변경
- 불필요한 System Println
→ 동작 확인을 위해 작성했던 코드를 삭제
- Cyclomatic Complexity가 높은 함수
→ switch case로 color선택 하던 부분을 상수로 만들어 사용.
→ 새로운 함수를 만들어서 Cyclomatic Complexity를 줄임.
- NCSS Method Count
→ 새로운 함수를 만들어서 라인 수를 줄임.

5.1. Anon Inner Length

- 익명 클래스의 라인 수가 지나치게 김.

```
public void mousePressed(MouseEvent e) {
    // TODO Auto-generated method
    int x, y;
    x = e.getX();
    y = e.getY();
    // set start x, y
    if(at.getAreaStat()){
        if(at.checkAreaOn(x, y)){
            startMoveArea(x, y);
        }
        else{ // cancel area
            cancelArea();
        }
    }
    if(selectedtool == 3 && !at.ch
        startSelectArea(x, y);
    }
}
```

```
//start move area
private void startMoveArea(int x, int y){
    requestMoveArea();
    at.setCurrentXY(x, y);
}
//start select area
private void startSelectArea(int x, int y){
    at.clearArea();
    at.setStartArea(x, y);
    at.setEndArea(x, y);
}
//during select area
private void duringSelectArea(int x, int y){
    Cut c = pj.cuts.get(enabledcutnum);
    int w = c.getWidth();
    int h = c.getHeight();
    x = setRange(x, w);
    y = setRange(y, h);
    at.setEndArea(x, y);
}
//finish select area
private void finishSelectArea(int x, int y){
    Cut c = pj.cuts.get(enabledcutnum);
    int w = c.getWidth();
    int h = c.getHeight();
    x = setRange(x, w);
    y = setRange(y, h);

    at.setEndArea(x, y);
    if(at.getWidth() != 0 && at.getHeight() != 0){
        // copy cut image to area image
        at.setArea();
        at.selectArea(pj.cuts.get(enabledcutnum));
    }
}
```

5.2. Hidden Field

- Image *i* 와 for 문의 *i* 가 겹침.
→ Image *img*로 변경

```
private class TransferableImage implements Transferable {

    private Image img;

    public TransferableImage( Image img ) {
        this.img = img;
    }

    public Object getTransferData( DataFlavor flavor )
    throws UnsupportedOperationException, IOException {
        if ( flavor.equals( DataFlavor.imageFlavor ) && img != null ) {
            return img;
        }
        else {
            throw new UnsupportedOperationException( flavor );
        }
    }

    public DataFlavor[] getTransferDataFlavors() {
        DataFlavor[] flavors = new DataFlavor[ 1 ];
        flavors[ 0 ] = DataFlavor.imageFlavor;
        return flavors;
    }

    public boolean isDataFlavorSupported( DataFlavor flavor ) {
        DataFlavor[] flavors = getTransferDataFlavors();
        for ( int i = 0; i < flavors.length; i++ ) {
            if ( flavor.equals( flavors[ i ] ) ) {
                return true;
            }
        }
        return false;
    }
}
```

5.3. System Println & Avoid Print Stack Trace

동작 확인을 위해 작성했던 코드를 삭제

Violations		
Line Num	Class	
78	AreaTool	System.out.println("Copy Area to ClipBoard");
84	AreaTool	System.out.println("Load ClipBoard to Area");
172	AreaTool	System.out.println("Lost Clipboard Ownership");
Violations		
Avoid Print Stack Trace (2)		
Line Num	Class	
657	Interface	e.printStackTrace();
673	Interface	e.printStackTrace();

동작을 위해 작성했던 코드 제거

5.4. Cyclomatic Complexity

- Cyclomatic Complexity가 높은 함수
→ switch case로 color선택 하던 부분을 상수로 만들어 사용.
→ 새로운 함수를 만들어서 Cyclomatic Complexity를 줄임.

```
//colors
private Color[] cl = new Color[16];

public void brushing(int x, int y, int size, int color, int stat)
{
    g = cuts.get(stat).getImg().getGraphics();
    g.setColor(cl[color]);
}
```

```
//color set
cl[0] = new Color(0,0,0);
cl[1] = new Color(255,255,255);
cl[2] = new Color(168, 168, 168);
cl[3] = new Color(92, 92, 92);
cl[4] = new Color(255, 0, 0);
cl[5] = new Color(255,128,0);
cl[6] = new Color(255,255,0);
cl[7] = new Color(0,255,0);
cl[8] = new Color(0,0,255);
cl[9] = new Color(0,0,160);
cl[10] = new Color(255,0,255);
cl[11] = new Color(255,196,196);
cl[12] = new Color(128,64,64);
cl[13] = new Color(0,128,192);
cl[14] = new Color(255,255,196);
cl[15] = new Color(168,255,0);
```

5.4. Cyclomatic Complexity(계속)

- Cyclomatic Complexity가 높은 함수
→ 새로운 함수를 만들어서 Cyclomatic Complexity를 줄임.

```
private void finishSelectArea(int x, int y){
    Cut c = pj.cuts.get(enabledcutnum);
    int w = c.getWidth();
    int h = c.getHeight();
    x = setRange(x, w);
    y = setRange(y, h);

    at.setEndArea(x, y);
    if(at.getWidth() != 0 && at.getHeight() != 0){
        // copy cut image to area image
        at.setArea();
        at.selectArea(pj.cuts.get(enabledcutnum));
        at.delArea(pj.cuts.get(enabledcutnum));
    }
    else{
        at.clearArea();
    }
}
```

```
//set range
static int setRange(int n, int range){
    if(n>0 || n<range){
        return n;
    }
    if(n<0){
        return 0;
    }
    return range;
}
```

5.4. Cyclomatic Complexity

- Cyclomatic Complexity가 높은 함수
 - switch case로 color선택 하던 부분을 상수로 만들어 사용.
 - 새로운 함수를 만들어서 Cyclomatic Complexity를 줄임.

