

## 1.1

<u>Draw</u>	
<ul style="list-style-type: none"><li>• Start point of the application</li><li>• Handle clear and quit events</li></ul>	DrawGUI

<u>DrawGUI</u>	
<ul style="list-style-type: none"><li>• Renders the GUI according to AWT.Frame standards</li><li>• Initializes action listener</li></ul>	Draw ShapeManager ColorItemListener DrawActionListener

<u>DrawActionListener</u>	
<ul style="list-style-type: none"><li>• Alerts Draw class about „clear“ and „quit“ events</li></ul>	Draw

<u>ColorItemListener</u>	
<ul style="list-style-type: none"><li>• Stores the chosen color</li></ul>	ItemListener DrawGUI

<u>ShapeManager</u>	
<ul style="list-style-type: none"><li>• Determines how mouse events are interpreted, depending on the shape mode selected</li></ul>	DrawGUI ScribbleDrawer RectangleDrawer OvalDrawer

<u>ScribbleDraw</u>	
<ul style="list-style-type: none"><li>• If active, mouse is interpreted as a pen</li></ul>	ShapeDrawer DrawGUI

## 1.1

2. User drückt auf Quit Button -> actionPerformed Methode wird am DrawActionListener aufgerufen, diese wiederum ruft die Methode „doCommand“ mit Parameter „quit“ an Draw auf -> Ruft Methode dispose() (schließt das Fenster) an DrawGUI auf und führt System.exit(0) aus.

3. User wählt Farbe aus Dropdown Menu aus -> Methode `itemStateChanged()` wird an `ColorItemListener` aufgerufen -> ausgewählte Farbe wird als Farbkonstante unter variable „color“ gespeichert

4. User drückt linke Maustaste -> `ScribbleDrawer` speichert x,y Koordinaten der Maus in `lastx`, `lasty` zwischen -> User zieht die Maus -> die `mouseDragged` Methode speichert die neuen x und y Koordinaten zwischen, setzt die Farbe nach der aktuell ausgewählten Farbe in `DrawGUI`, zeichnet eine Linie zwischen (`lastx`, `lasty`) und (x, y) in der gewählten Farbe, x und y werden als neue Werte für `lastx`, `lasty` eingesetzt

5. User drückt linke Maustaste -> die Mausposition wird in der Methode `RectangleDrawer` unter `pressx`, `pressy` gespeichert -> User zieht die gedrückte Maus -> `mouseDragged` löscht das vorige (wenn vorhanden) Rechteck und zeichnet ein neues zwischen dem Startpunkt und der aktuellen Mausposition -> User lässt die Maustaste los -> `mouseReleased` wird aufgerufen und zeichnet wie `mouseDragged` das Rechteck. Da `mouseDragged` nun bei Mausbewegungen nicht mehr aufgerufen wird, ist das Rechteck endgültig gezeichnet.

6. User wählt shape aus dropdown menu aus -> `itemStateChanged()` wird an `ShapeManager` aufgerufen -> `setCurrentDrawer()` wird mit dem ausgewählten shape als parameter aufgerufen -> `setCurrentDrawer` löscht den `MouseListener` für den vorigen Drawer und setzt neue `MouseListener` für den ausgewählten Drawer

\*der unterschied zwischen 4 und 5 ist, dass bei der Linie die Linie sofort unwiderruflich gezeichnet wird und beim Rechteck solange man die Maus Zieht, das vorige Rechteck immer wieder gelöscht und mit der neuen Mausposition neu gezeichnet wird. Erst wenn man die Maustaste loslässt wird das Rechteck unwiderruflich gezeichnet.