第１回レポート課題

学生証番号: 2600170522-8

名前: CAI Ming

1. 課題の説明

今回の課題を通して、JAVA言語に基づいて、グラフィックを描くために使用できるツールを完成させました。

ソフトウェアインターフェイスを以下に示します（図1）。

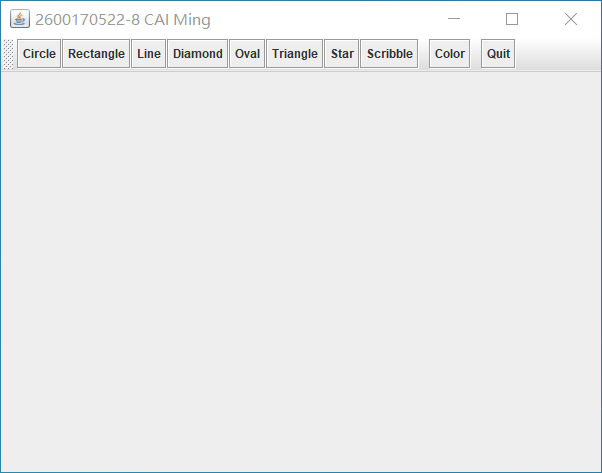


図1

ソフトウェアの主要なUML図を図2と図3に示している

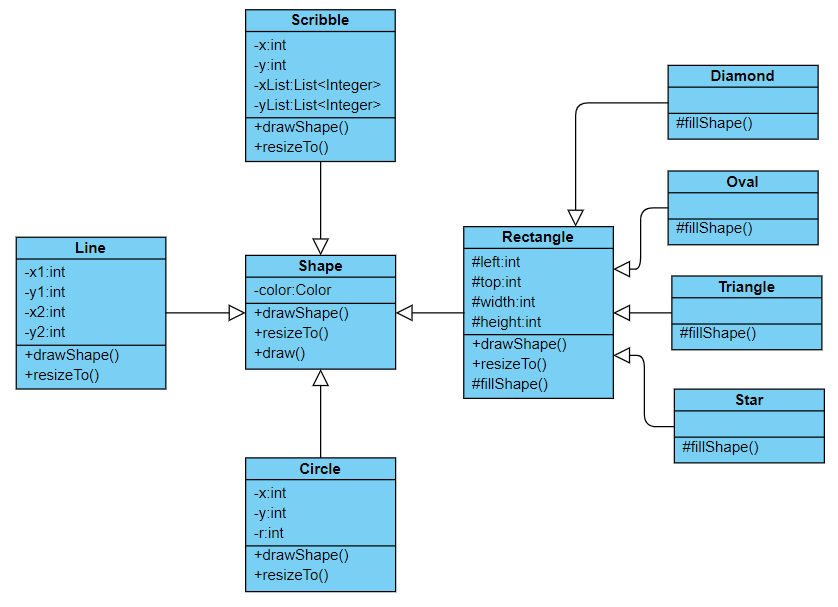


図2

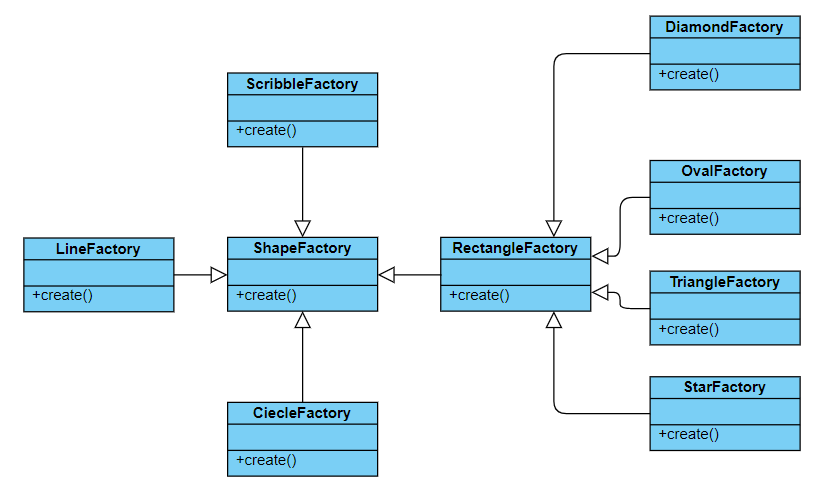


図3

1. 実装した機能
2. それぞれの図形を描画できること
3. 円、長方形、線などの描画をできる

これまでの講義によると、円、長方形、線などの描画機能が実現されているので、もう繰り返し説明をしない。

1. 楕円、菱形以などの描画をできる

これらは今回のレポートの基本要求であり、講義に従って実装した。

両方のクラスはRectangleクラスから継承された。次に、これら2つのクラスを具体的に紹介する。

OvalとDiamondクラス図は図4のように表す。

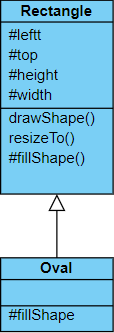
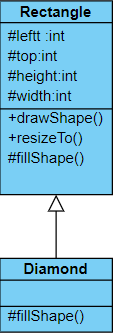
　 

図4

両方がfillShape()メソッドをoverrideした。OvalのfillShape()は簡単に「java.awt.Graphics」のfillOvalというメソッドを利用する（図5）

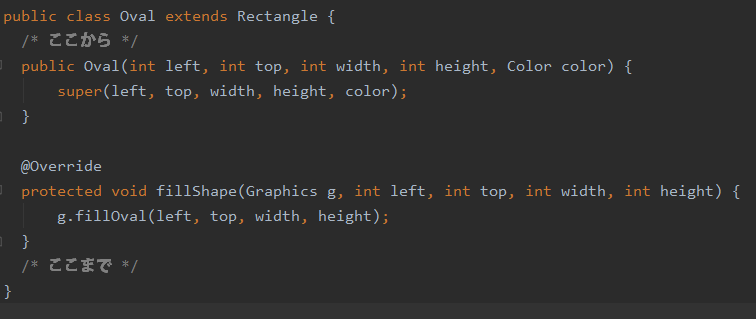


図5

DiamondのfillShape関数は「java.awt.Graphics」のfillOvalという関数を利用するため、全ての頂点座標の配列と座標の数が必要である。Diamondだから、4つの頂点があり、頂点は四角形の辺の中点である。そして、fillShapeメソッドを定義する（図6）。

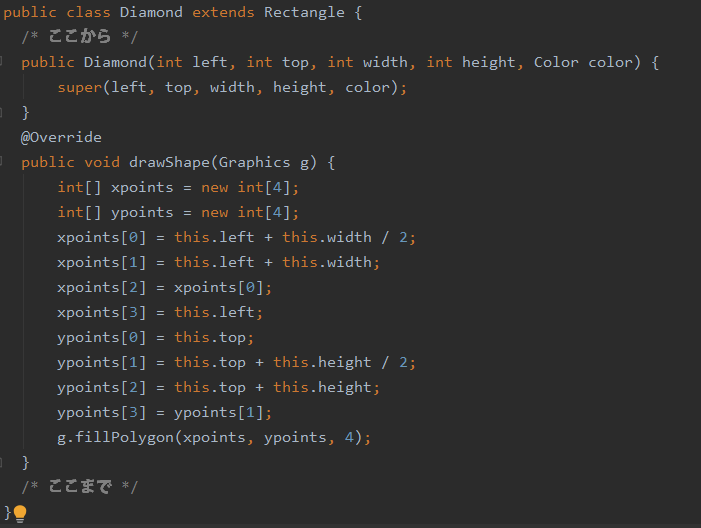


図6

1. 三角形、星、落書きなどの描画をできる

三角形も星もRectangleクラスから継承されたで、落書きはLineクラスから継承された。TriangleとStarとScribbleのクラス図は図7のように表す。

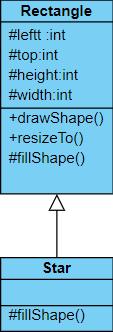
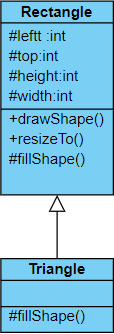
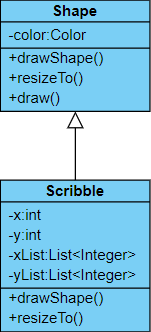
　  　

図7

TriangleとStarはDiamondとOvalと同じで、fillShapeメソッドだけをoverrideした。

Triangleの関数fillShapeも全ての頂点座標の配列と座標の数が必要である。三つの頂点があり、一番上の頂点は四角形の上の辺の中点で、他の二つの頂点は四角形の下の角の座標である（図8）。

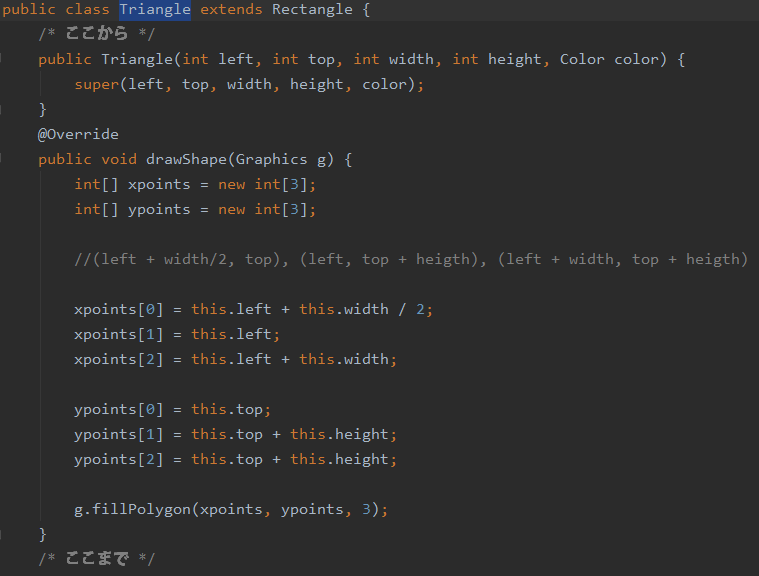


　　　　図8

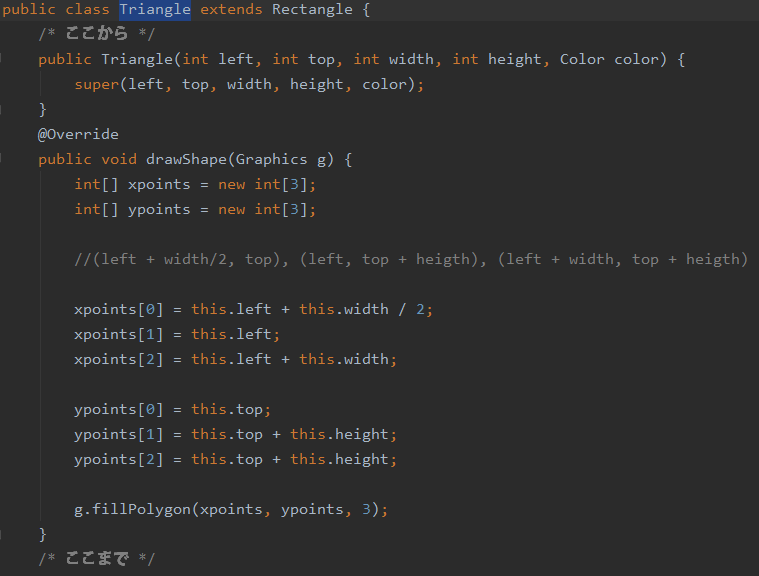
StarのfillShapeメソッド順番に五つの頂点を配列に作り、「java.awt.Graphics」のdrawPolygonの利用し、順に星の形を描く（図9）。

図9

ScribbleクラスはLineクラスから継承された。マウスの軌跡の座標の並びを保存するインスタンス変数を定義．resizeToメソッドによる、その時点のマウスの座標を上記の変数に追加する． drawShapeメソッドによる、上記の変数に保存された座標の並びにしたがって、「java.awt.Graphics」のdrawPolygonの利用し、線を描画していく。図10はScribbleの追加されたメソッド定義である。

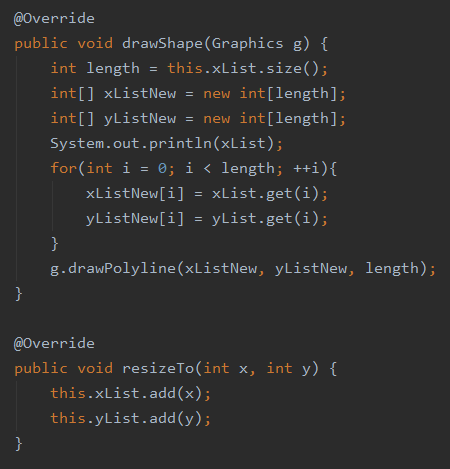


図10

1. Colorを選べること

最初に設計するColorというボタンを設計した。JButtonのaddActionListenerメソッドを利用して、マウスがこのボタン上にあるかどうかを確認。そして、showDialogメソッドによる、色を選ぶ画面を切り替える。図形の色を選べる。

コードは図11のようにある。

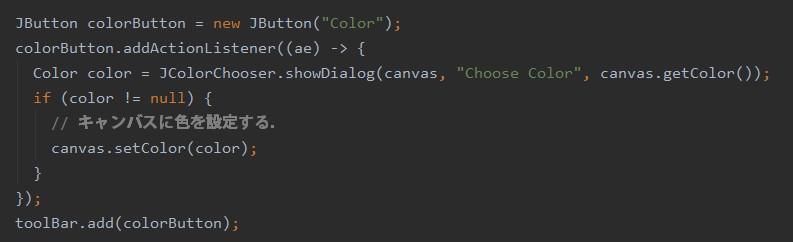


図11

1. 大きさを変えること

マウスの移動によると、図形の大きさと位置を変化することができる。Rectangleの実現に基づいて、説明をあげる。

まずは、RectangleのdrawShapeメソッドである（図12）。

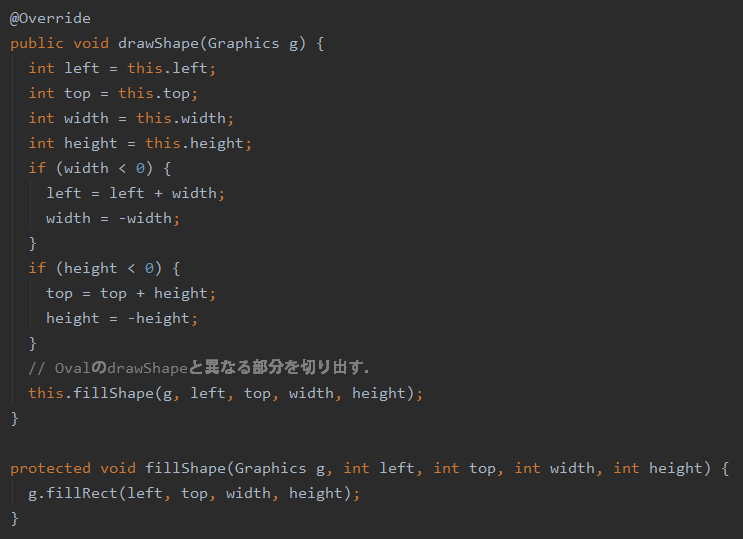


図12

四角形の大きさと位置は、left、top、width、heightによって決まる。

マウスをクリックするたびに新しい図形の位置が変化。Resizeメソッドがwidth、heightに影響し、大きさを決まる（図13）。

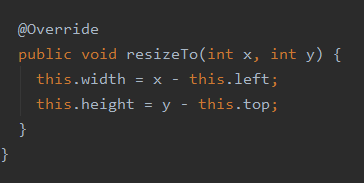


図13

ｘとｙはマウスの最後の座標である。

1. ToolBarを使用できる

まずはそれぞれのボタン（ToolButton）を作り、ToolBarの中に追加する。ToolButtonのクラス図は図14である。

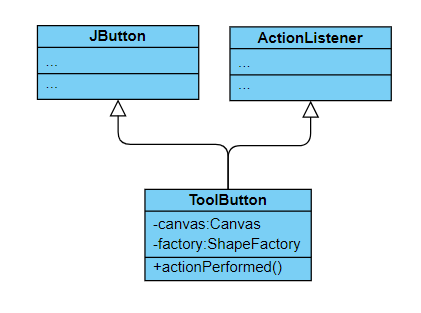


図14

図12で、Canvasのクラス図は図15である。

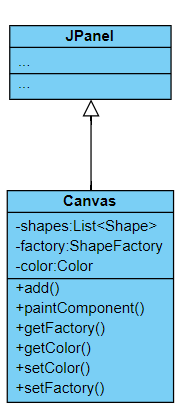


図15

1. 実行結果（図16、図17）

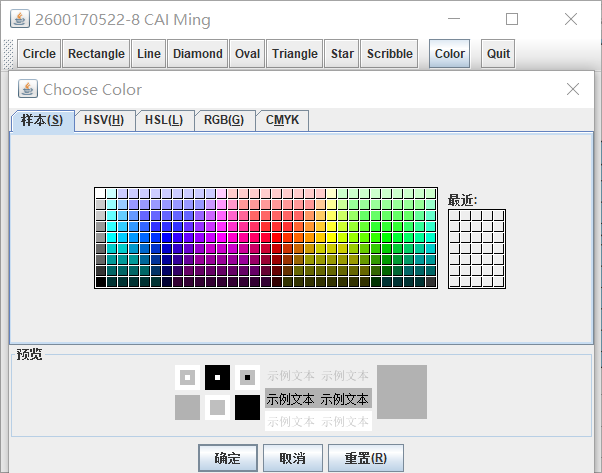


図16

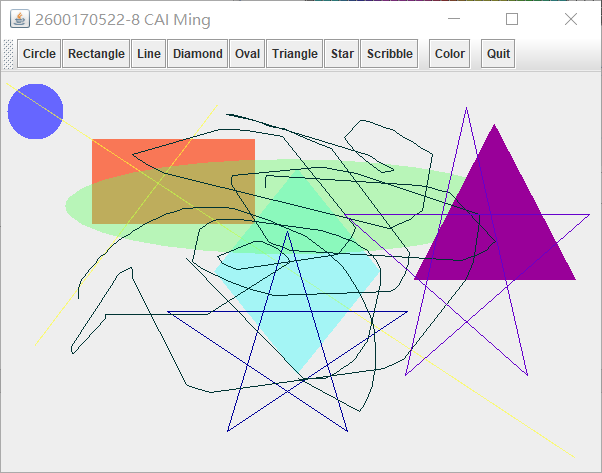


図17

1. 感想

今回の課題を通して、javaの使用を学びました。自分でこのソフトウェアをつくるのがうれしいです。

でも、ちょっと難しいと思います。理由はJavaに精通していないでしょう。だから、この授業は講義によると教えるじゃないで、デモンストレーションしながら生徒を教えた方がいいと思います。ほかに、JAVAの原理も知りたいです。例えば、JAVAのVMとか、マルチスレッドです。使用方法しかを教えないで、Javaを深く理解していないだろうと思います。