

第六回 演習問題 (GUI2)

諸注意

- 今回、ファイルの提出は行わない。
- 課題ができた段階で教員もしくは TA を呼びその場でチェックを受ける。

課題 1

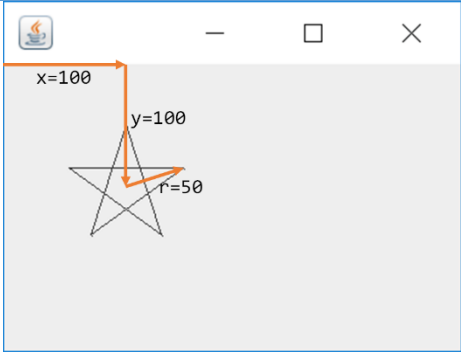
1	問題設定	座標 (x, y) を中心とし半径 r の五角形を描く際の頂点間を直線で結ぶことで星を描画することができる。これを <code>drawStar(Graphics g, int x, int y, int r);</code> で定義してほしい。図 2 にテスト例を示す。なお橙色矢印とそれに付随する文字列は説明用でありアプリ用で表示する必要はない。	
	テスト例	<pre>@Override public void paintComponent(Graphics g){ this.drawStar(g, 100, 100, 50); }</pre>	
	動作画面例		
	ヒント	練習問題 2 を参考にすると良い。もはや高校数学の話。 <code>drawPolygon()</code> を応用して開発できるとかっこよい。	

図 2. 動作画面例

課題 2

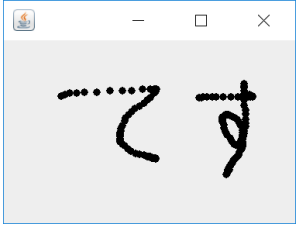
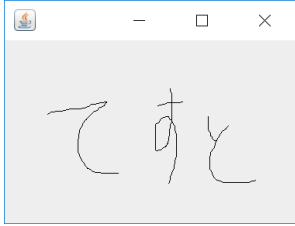
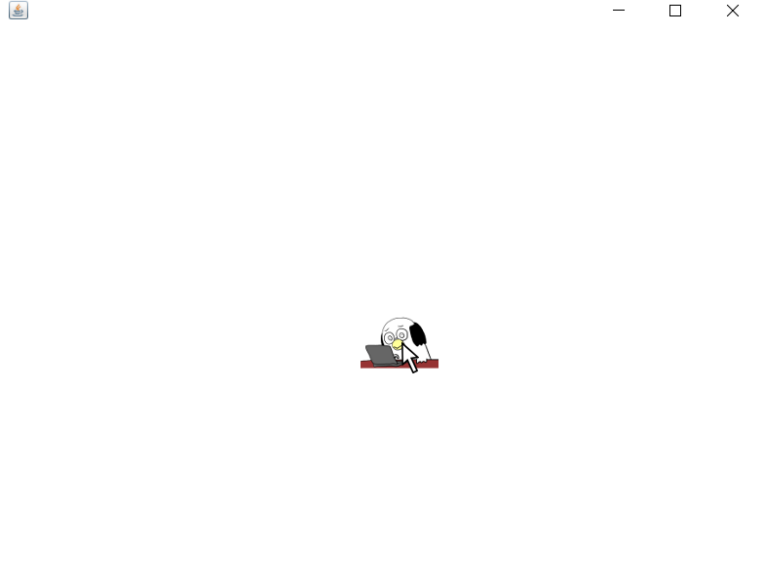
2	問題設定	<p>お絵かきアプリを開発したい。マウスイベントをうまく用いることで、クリック時に黒線でお絵かきできるアプリを開発してほしい。ただし、授業中の例をシンプルに用いると、図 3 の様に途切れ途切れになってしまうことが懸念される。そこで、線の太さは 1px で構わないので、図 4 のように途切れないように改造して実装してほしい。</p> <p>本課題ではマウスイベント内の <code>this.getGraphics();</code> で取得した <code>Graphics</code> インスタンスに対して直接描画処理を実装してくれて良い（ダブルバッファリングしない）。</p> <p>ドラッグ中にマウスポインタがフレーム外に移動したケースは想定しなくて良い（発生しないものとする）。</p>	
	動作画面例		
	ヒント	点と点を線でつなぐと。...	

図 3. シンプルな例

図 4. 正解画面例

課題 3

3-1	問題設定	<p>一昔前（もっと前かもしれない）に流行った「マウスカーソルを変更する」に近い動作を実現したい。図3のように JPanel 上にマウスカーソルがある時に画像が追跡してくるような動作を実装せよ。ただし、マウスカーソルが画像の中心に来るように注意せよ。</p> <p>本課題ではマウスイベント内の <code>this.getGraphics();</code> で取得した <code>Graphics</code> インスタンスに対して直接描画処理を実装してくれて良い（ダブルバッファリングしない）。</p> <p>ドラッグ中にマウスポインタがフレーム外に移動したケースは想定しなくて良い（発生しないものとする）。</p> <p>画像は演習問題同様に Web から DL し使用すること。</p>
	動作画面例	 <p>図3. 動作画面例</p>
	ヒント	<p>参考までに、画像に関連するメソッドの使用例を下記に示す。</p> <pre>// Imageインスタンスの取得 Image img = (new ImageIcon("img.png")).getImage(); // Imageインスタンスのサイズの取得 int width = img.getWidth(this); int height = img.getHeight(this); // 座標(0, 0)にImageインスタンスを描画 g.drawImage(img, 0, 0, this);</pre>

3-2	問題設定	<p>課題3-1をmouseMoved()内にて実装した場合、画像がたまにチラつく（点滅する）ことがある。これは白背景で塗りつぶし→画像の描画のタイムラグによって発生する現象である。ダブルバッファリングを用いることで本現象を解決できるが、使用するためには描画処理は全て</p> <pre>public void paintComponent(Graphics g){}</pre> <p>内に記述されなければならない。課題 3-1 を修正し、ダブルバッファリングを機能させ、チラつきを改善せよ。</p>
3-3	問題設定	<p>（発展問題：採点対象外）</p> <p>課題 3-2 をベースに、表示させる図を動的に変更できるように拡張したい。まず、フレーム内上部に現在開発している JPanel を配置し、下部に 3 つのラジオボタンを配置する（ボタンのテキストはそれぞれ●, ☆, 画）。表示される図はラジオボタンでチェックされているものが表示される。</p> <p>●の場合は、単純な塗りつぶし円をマウスカーソル中心で描画する。</p> <p>☆の場合は、課題 1 で開発した☆をマウスカーソル中心で描画する。</p> <p>画の場合は、課題 3-2 同様に画像をマウスカーソル中心で描画する。</p> <p>なお、ダブルバッファリングを有効にすること。</p>
	動作画面例	<p>図 4. 動作画面例</p>