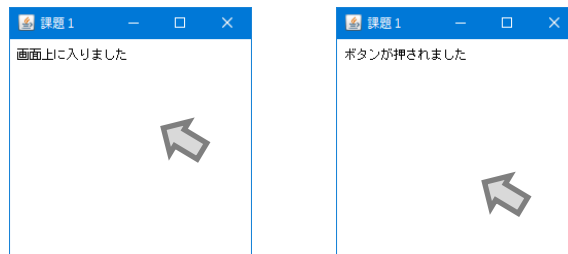


■難易度☆☆☆

課題 1 次のようにマウスのカーソルに同期しメッセージを表示するコードを作成しなさい。

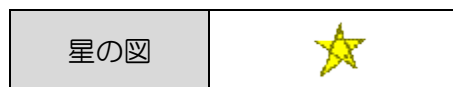
イベント	メッセージ
マウスをクリックした	クリックされました
マウスが画面に入った	画面上に入りました
マウスが画面から出た	画面から出ました
マウスのボタンを押した	ボタンが押されました
マウスのボタンをはなした	ボタンが離されました

〔実行例〕



■難易度☆☆☆

課題 2 星の画像がマウスカーソルを追従するコードを作成しなさい。次の星の図は HP よりダウンロードしてください。



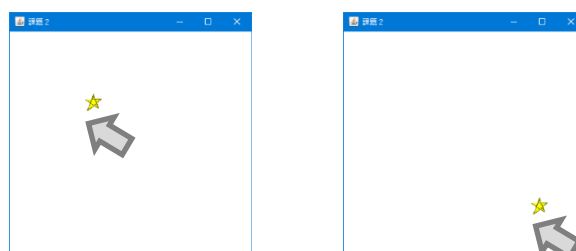
ヒント：MouseListener の代わりに MouseMotionListener を用いると、マウスカーソルが動いた時のイベントを受け取ることができます。以下のように宣言されています。

```
interface MouseMotionListener{
    void mouseDragged(MouseEvent e); // マウスボタンを押してドラッグしたとき
    void mouseMoved(MouseEvent e); // ボタンを押さずに移動したとき
}
```

このリスナーの登録には、Frame クラス（スーパークラスの Component クラスで宣言）から継承される次のメソッドを用います。

```
void addMouseMotionListener(MouseMotionListener l){…}
```

〔実行例〕



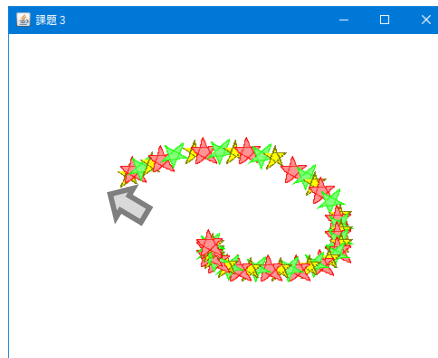
■難易度★★★

課題 3 色の異なる複数の星の画像がマウスカーソルを追従するコードを作成しなさい。星の図は HP から次に画像をダウンロードしてください。

星の図（黄）	星の図（赤）	星の図（緑）
		

ヒント 1：各星の座標を保持するために配列を用いなさい。マウスが動くたびに配列の値を 1 つ後方へシフトし、配列の先頭に現在のマウスカーソルの値をコピーすればよい。
ヒント 2：マウスカーソルが動いた時のイベントを受け取るには、課題 2 で説明しているインターフェース `MouseListener` を用いましょう。

〔実行例〕



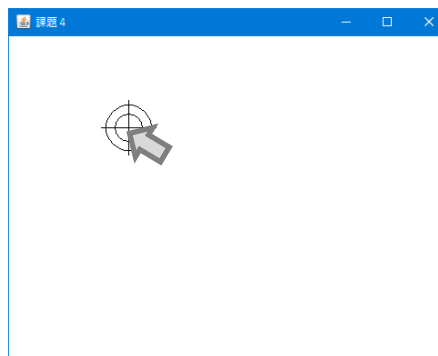
【注 意】デモでは星のちらつきが気になりますが、これは画面の描画にシングルバッファを用いているためです。シングルバッファは画面の表示に 1 枚の画用紙を使う方式です。画用紙のクリアと描画を交互に行うため、繰返し高速に描画が行われる場合にちらつきが発生します。画用紙を 2 枚用いる方式をダブルバッファといいます。1 枚は表示用でもう 1 枚は描画用と使い分けます。描画用の画用紙にクリアと描画を行います。描画が終わると 2 枚を入れ替えて描画の終了した画用紙を表示します。これを繰り返します。ダブルバッファ方式ではスムーズな描画効果を得ることができます。興味ある方は是非調べてみましょう！

■難易度★★☆

課題 4 マウスをクリックした場所にクロスヘア（照準）を表示するコードを作成しなさい。



〔実行例〕



■難易度★★☆

課題 5 2つのボタンを配置し、ボタンが押されたらそのボタンのラベルを●に、他方のラベルを○にするコードを作成しなさい。

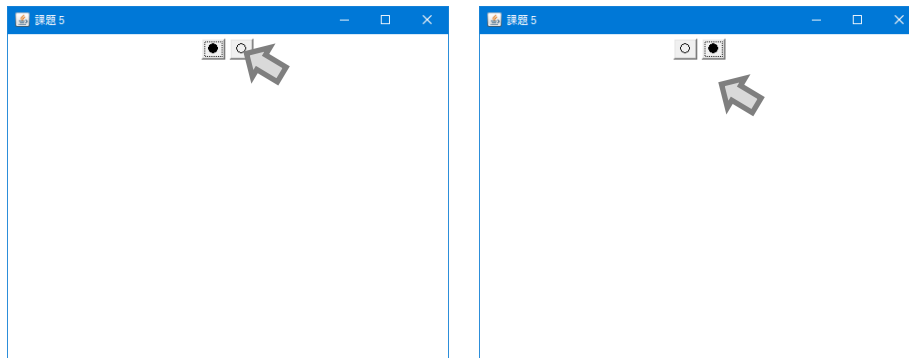
ヒント 1: Frame クラスから継承される `add(Component c){…}` はその Frame クラスが管理するウィンドウに 1 つの GUI 部品しか配置できません。2 つのボタンのように複数の GUI 部品を配置するには、Panel クラスを 사용합니다。Panel クラスは複数の GUI 部品を配置できるパネルです。このパネルに 2 つのボタンを配置し、このパネルをウィンドウに配置します。

```
Panel p = new Panel();
p.add(new Button("ボタン 1")); // パネルに 1 つ目のボタンを配置
p.add(new Button("ボタン 2")); // パネルに 2 つ目のボタンを配置
add(p); // Frame にパネルを配置
```

その他、レイアウトクラスを用いるとより柔軟な配置の指定が可能になります。

ヒント 2: 複数のボタンを 1 つのイベントリスナーで判別するには Button クラスの `void setActionCommand(){…}` メソッドで識別子 (String 型文字列) を設定し、リスナー側で受け取る `ActionEvent` クラスのオブジェクトの `String getActionCommand(){…}` メソッドで識別子を受け取り判断します。

〔実行例〕



■難易度★★☆

課題 6 血液型が選択できるチョイス (プルダウンメニュー) を配置し、選択されたらその血液型を表示するコードを作成しなさい。

ヒント 1: Choice クラスとその選択結果を表示する Label クラスは前課題で説明している Panel クラスを用いて配置しましょう。Choice クラスは次のように使用します。

```
Choice c = new Choice();
c.add("選択肢 1");
c.add("選択肢 2");
c.add("選択肢 3");
```

ヒント 2: ItemListener を用いると項目が選択された時にイベントを受け取ることができます。次のように宣言されています。

```
interface ItemListener{
    void itemStateChanged(ItemEvent e); // 項目が選択 or 解除されたとき
}
```

このリスナーの登録には Choice クラスの次のメソッドを用います。

```
void addItemListener(ItemListener l){...}
```

ヒント 3 : Choice クラスの String `getSelectedItem()`{...}を実行すると選んだラベル文字列を取得します。

〔実行例〕

