

■課題 1 マウスイベントを可視化してみましょう。イベントが発生したら以下のように種類に応じて文字を出力するようにします。

- MouseEvent.MOUSE\_MOVED → + 改行なし
- MouseEvent.MOUSE\_PRESSED ● + 改行なし
- MouseEvent.MOUSE\_RELEASED ○ + 改行なし
- MouseEvent.MOUSE\_CLICKED ! + 改行あり
- MouseEvent.MOUSE\_DRAGGED △ + 改行なし
- MouseEvent.DRAG\_DETECTED ▲ + 改行なし

その他の設定は次のとおりです。参考にしましょう。

ウィンドウのタイトル → マウスイベントの可視化

〔実行例〕

→→→→→●△△△▲△△○!  
●○!  
→→→→→→→→→→→●△△△▲△△△△△△△△△△△△△△△△△△△○!  
→→→→→→●○!

■課題 2 写真閲覧アプリを作成しましょう。5枚の写真を読み込み、写真の上でマウスポタンが押されたら (MOUSE\_PRESSED) 次の写真へ進みます。最後の写真の上でマウスポタンが押されたら、最初の写真に戻ります。

その他の設定は次のとおりです。参考にしましょう。

ウィンドウのタイトル → 画像閲覧アプリ

ヒント 1 : 5枚の写真は Image クラスの配列で準備します。各 Image クラスのオブジェクトは画像ファイルをコンストラクタに渡して次のように生成します。

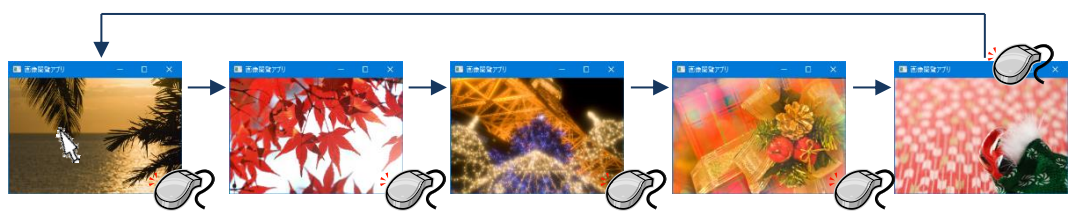
```
new Image("○○○.jpg");
```

ヒント 2 : 写真の表示は ImageView クラスを扱います。ImageView クラスに写真を設定したり、それを変更するには次のメソッドを利用します。

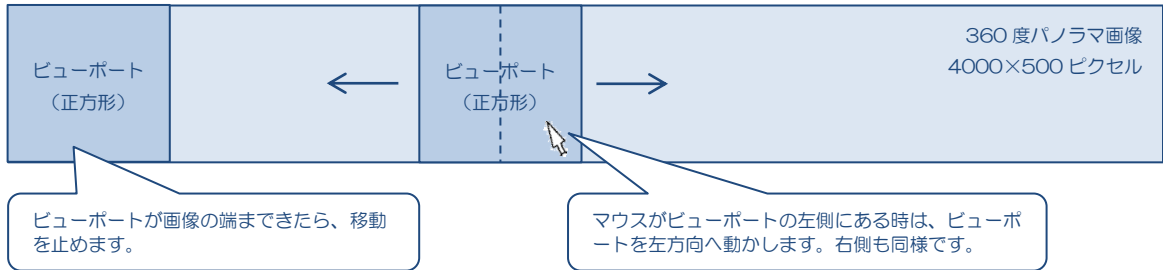
```
void setImage(Image img){...} // 引数の画像 img を現在の ImageView に設定します。
```

※画像は HP よりダウンロードしてソースファイルと同じフォルダに保存しておきましょう  
※おおよそ実行例のような画面になれば OK です

〔実行例〕

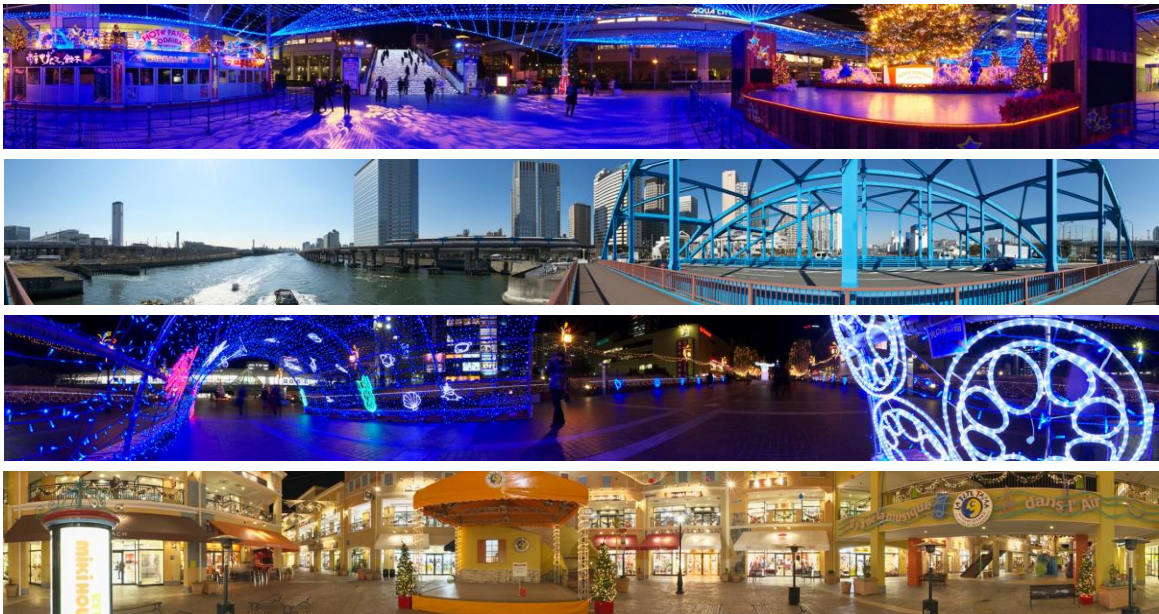


■課題3 「パノラマ画像閲覧アプリ」を作成しましょう。横 4000×縦 500 の 360 度パノラマ画像 1 枚を読み込み、ビューポートで画像の一部を切り出します。表示された画像上でマウスを左半分に着くと画像は左から右へ流れ、右半分に着くと画像は右から左へ流れます。



ヒント：マウスが動いたら (MOUSE\_MOVED) その位置が左側か右側かをメンバー変数に保存します。例えば、メンバー変数として `boolean slideLeft;` を宣言し、左側にあるときには `true` を代入するようにします。次に、ビューポートを移動する処理をスレッドで実現します。スレッドは `start()` メソッドが終わる前に起動します。内容は無限ループであり、毎回変数 `slideLeft` を参照して適切な方向にビューポートを少し移動します。これを繰り返します。

〔360度パノラマ画像例 4000×500 ピクセル〕



※画像は HP よりダウンロードしてソースファイルと同じフォルダに保存しておきましょう  
 ※おおよそ実行例のような画面になれば OK です

〔実行例〕

