

## HCI プログラミング

### 9 回目 メニューとポップアップメニュー 課題

■課題 1 メディアプレイヤーのメニューを作成しましょう。実行例に示すように各メニュー項目にはアイコンを配置します。さらに機能ごとにセパレータを引きましょう。

メニュー構成は次の通りです。

[メニューバー] → [ファイル][再生][ヘルプ]

[ファイル] → [新規作成][開く][保存][設定][印刷][ログアウト]

[再生] → [再生][停止][一時停止]

[ヘルプ] → [ヘルプ][更新の確認][バージョン]

※ | はセパレータです

その他の設定は次のとおりです。参考にしましょう。

レイアウト → BorderLayout の Top にメニューバーを配置

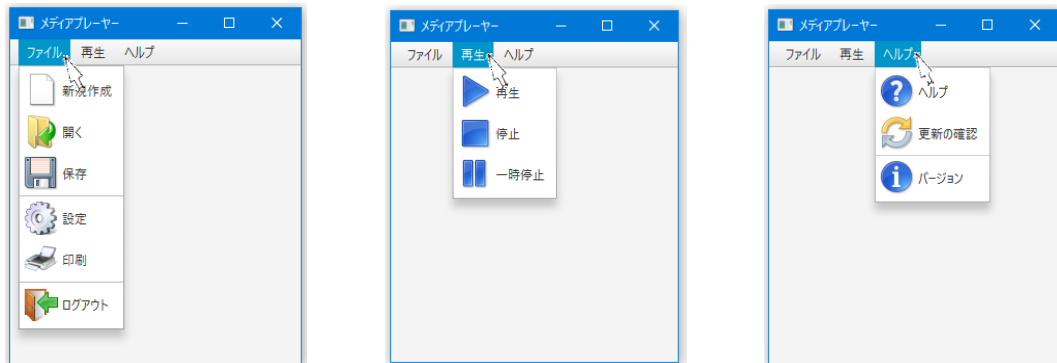
メニューバーの背景色 → 無し〔setBackground(...)は実行しません〕

ウィンドウのタイトル → メディアプレイヤー

※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〕



〔コマンドプロンプト〕

setting

open

pause

update

:

※各メニュー項目がクリックされたら識別子を表示します。識別子名はお任せです。

■課題2 画像レタッチアプリを作成しましょう。3枚の画像を準備します。メニューにより画像の選択とエフェクトの設定を行います。エフェクトはサブメニューで効果の強さを調整できます。

メニュー構成は次の通りです。

[メニューバー] → [画像の変更][エフェクトの設定]

[画像の変更] → [古い本棚][観葉植物][サンタさん]

[エフェクトの設定] → [セピア][ぼかし][輝き]

[セピア] → [強さ 0.2][強さ 0.5][強さ 0.8]

[ぼかし] → [ぼかし半径 2.0][ぼかし半径 8.0][ぼかし半径 32.0]

[輝き] → [強さ 0.2][強さ 0.5][強さ 0.8]

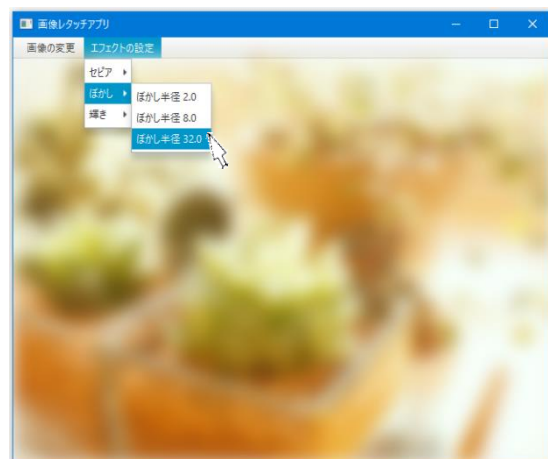
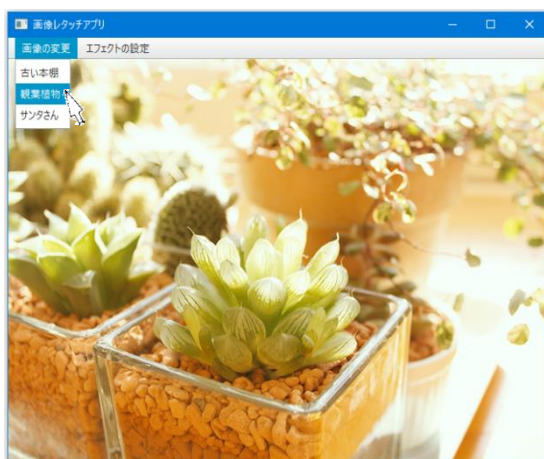
その他の設定は次のとおりです。参考にしましょう。

- レイアウト → BorderLayout の Top にメニューバーを配置  
BorderPane の Center に ImageView を配置
- メニューバーの背景色 → 無し〔setBackground(...)は実行しません〕
- ウィンドウのタイトル → 画像レタッチアプリ

ヒント：エフェクトを付けるときは、ImageView クラスの setEffect(...)メソッドを用いますが、エフェクトが重複しないようエフェクトをつける前に一度 setEffect(null)を実行して原画像に戻します。

※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 文字と背景の色合い調整アプリを作成しましょう。2つのウィンドウを表示します。1つは文字色と背景色の調整用で、もう1つは色見本の表示用です。

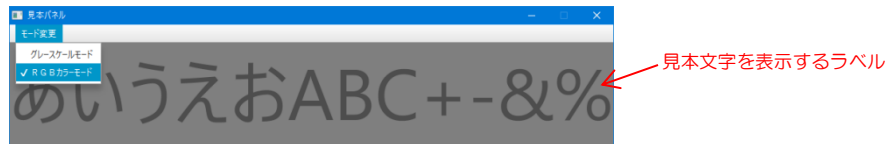
### 【色の調整用のウィンドウ】

赤、緑、青のRGBそれぞれに対応するスライダーがあり、これをドラッグすることにより各色要素の強度を調整できます。スライダーの右にあるラベルはスライダーの現在値を表示します。レイアウトは3行6列のGridPaneを用います。



### 【色見本用のウィンドウ】

スライダーによる色の調整がリアルタイムで反映されます。見本用のウィンドウは、モード変更のメニューをもちグレースケールとRGBカラーモードを変更することができます。



スライダーとは、つまみを左右に動かして数値を調整する GUI 部品です。



スライダーはクラス Slider により管理されており、次のように利用します。

```
Slider(){...} // 0 ~ 100 初期位置 0 のスライダーの生成
Slider(double a, double b, double v){...} // a ~ b 初期位置 v のスライダーの生成
void setMajorTickUnit(double v){...} // 主目盛間の距離の設定
void setMinorTickCount(double v){...} // 主目盛間の副目盛の数の設定
void setShowTickMarks(boolean b){...} // 主目盛と副目盛の描画
void setShowTickLabels(boolean b){...} // 目盛ラベルの描画
double getValue(){...} // スライダーの現在値の取得
void setValue(double v){...} // 値 v をスライダーの現在値に設定
```

また、クラス Slider はクラス Labeled をスーパークラスにもたない GUI 部品であり、文字列の指定や修飾を設定するメソッドは継承していません。識別子 ID の設定/取得は同様にできます。また Slider は MouseEvent を生成しますので、MouseEvent.MOUSE\_DRAGGED を受け取り、イベント処理を行います。

〔実行例〕

