

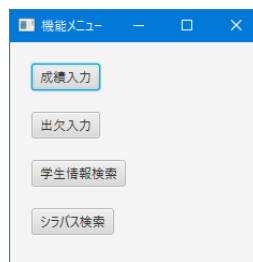
4 回目 レイアウト 課題

■課題 1 下のようにボタンを 4 つ配置し、機能メニューを表示するコードを書きなさい。レイアウト `VBox` を用いてボタンを配置します。各種設定は次のとおりです。

- レイアウト周りの空白エリア → 20 ピクセル [`setPadding(new Insets(20));`]
- ボタン間の空白エリア → 20 ピクセル [`setSpacing(20);`]
- ウィンドウのタイトル → 機能メニュー

※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 2 下のように「まちカフェ予約システム」のメニュー画面を表示するコードを書きなさい。ラベルとボタンはレイアウト `GridPane` を用いて配置します。ラベルは `Label` クラスを用います。使用方法は `Button` クラスと同様で以下のように生成し、レイアウトに配置することができます。

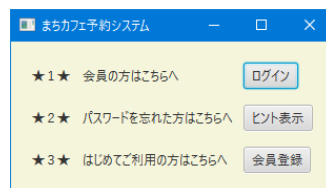
```
Label lb = new Label("表示する文字列");
```

各種設定は次のとおりです。

- レイアウト周りの空白エリア → 20 ピクセル [`setPadding(new Insets(20));`]
- ボタン横隣りの空白エリア → 10 ピクセル [`setHgap(10);`]
- ボタン縦隣りの空白エリア → 15 ピクセル [`setVgap(15);`]
- レイアウトの背景色 → 透明 [`setBackground(null);`] (※)
- シーンの色 → ベージュ色 (`Color.BEIGE`)
- ウィンドウのタイトル → まちカフェ予約システム

※おおよそ実行例のような画面になれば OK です

〔実行例〕

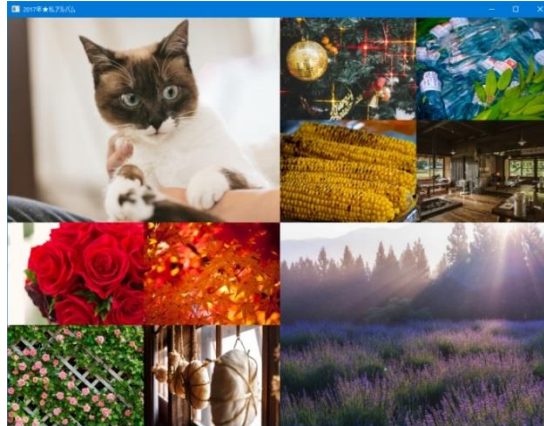


(※) レイアウトの背景色の初期値は不透明ですので、初期値のままではその下にあるシーンの色が見えません。レイアウトクラスのメソッドを用いて背景色を透明に設定します。

■課題 3 下のようなレイアウトで 10 枚の写真 (1024×768 px) を表示する「今年★私アルバム」を作成しなさい。2 行 2 列のレイアウト GridPane を重ね合わせます。画像は ImageView クラスを使い、使用方法は Button クラスと同様で以下のように生成してレイアウトに配置することができます。

```
(例) ImageView iv = new ImageView("img00.jpg"); // 画像を準備します
      iv.setFitHeight(); // 画像の縦幅を指定した高さにします
      iv.setFitWidth(); // 画像の横幅を指定した長さにします
```

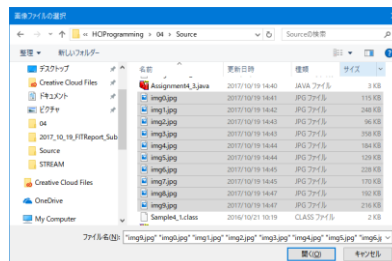
※画像は HP よりダウンロードしてソースファイルと同じフォルダに保存しておきましょう



さらに、10 枚の写真はファイル選択ダイアログクラス FileChooser を用いましょう。FileChooser クラスの使用方法は次のようにします。

```
(例) FileChooser fc = new FileChooser(); // オブジェクトの作成
      fc.setTitle("画像ファイルの選択"); // ダイアログタイトルの設定
      List<File> files = fc.showOpenMultipleDialog(st); // 自身のステージを指定して、
                                                    // ダイアログのオープンと選択
      System.out.println(files.size()); // 選択されたファイル数
      File f = files.get(0); // 引数 0 で最初のファイルの取得
      System.out.println(f.getName()); // そのファイル名の取得
```

※List<File>は File を要素にもつリストを管理するクラスです。List<String>とすれば String を要素にもつリストを管理するクラスとなります。このように<...>の中は引数のような働きをします。この引数を用いたクラスはジェネリクスと呼ばれます。



上記のクラスが使えるように import 文を追加しましょう。

```
import javafx.scene.image.*;
import java.io.*;
import java.util.*;
```