

■課題 1 プロフィール登録の登録画面を作成しましょう。実行例に示すように 5 項目の入力フィールドを配置します。各フィールドへの入力例をプロンプトテキストとして表示します。文字列が入力されリターンキーが押されると、登録したとメッセージが表示されます。

項目名の一覧とプロンプトテキストは次の通りです。

項目名	プロンプトテキスト
苗字	福工大
名前	太郎
得意科目	プログラミング
特技・趣味	バドミントン
好きなスイーツ	エッグタルト

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 → GridPane を用いてラベルとテキストフィールドを配置
- レイアウト周りの空白エリア → 10ピクセル〔setPadding(new Insets(10));〕
- GUI 部品間の空白エリア (縦) → 10ピクセル〔setVgap(10);〕
- GUI 部品間の空白エリア (横) → 10ピクセル〔setHgap(10);〕
- ウィンドウのタイトル → プロフィール登録

ヒント：テキストフィールドに識別子を設定しましょう。アクションイベントを取得したとき、どのテキストフィールドかを識別子により判断し、そのフィールドに入力されている文字列を取得して表示しましょう。

※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〕

- 苗字を「情報」に設定しました ← 苗字項目に「情報」と入力しリターンを押す
- 名前を「科学」に設定しました ← 名前項目に「科学」と入力しリターンを押す
- 得意科目を「Java」に設定しました ← 得意科目項目に「Java」と入力しリターンを押す




:

■課題 2 BMI 計算アプリを作成しましょう。実行例のように GUI 部品を配置します。入力フィールドに身長と体重を入力し、判定ボタンを押すと結果が文字と画像で表示されます。

BMI の計算方法と判定基準、表示メッセージと画像です。

BMI 計算式 BMI = 体重 kg ÷ (身長 m × 身長 m) ※←単位に注意しましょう

BMI 判定

- BMI < 18.5 > やせていますね。 
- 18.5 <= BMI < 25 > 標準です。 
- 25 <= BMI > 太り気味です。 

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 VBox の設定

レイアウト周りの空白エリア → 10 ピクセル [setPadding(new Insets(10));]

GUI 部品間の空白エリア → 10 ピクセル [setSpacing(10);]

レイアウト GridPane の設定

GUI 部品間の空白エリア (縦) → 10 ピクセル [setHgap(10);]

GUI 部品間の空白エリア (横) → 10 ピクセル [setVgap(10);]

ボタンのサイズ → 横 200 ピクセル × 縦 75 ピクセル

シーンの色 → 黄緑 [Color.GREENYELLOW]
※レイアウト VBox の背景色を透明にします

レイアウト VBox の setBackground(null); を実行

ウィンドウのタイトル → BMI 計算

ヒント 1: イベントハンドラはボタンにのみ登録します。ボタンをクリックしたとき、入力フィールドの文字列を取得します。ラッパクラス Double の parseDouble() メソッドを用いて文字列を実数に変換し BMI を計算します。結果に応じてメッセージと画像を変更します。

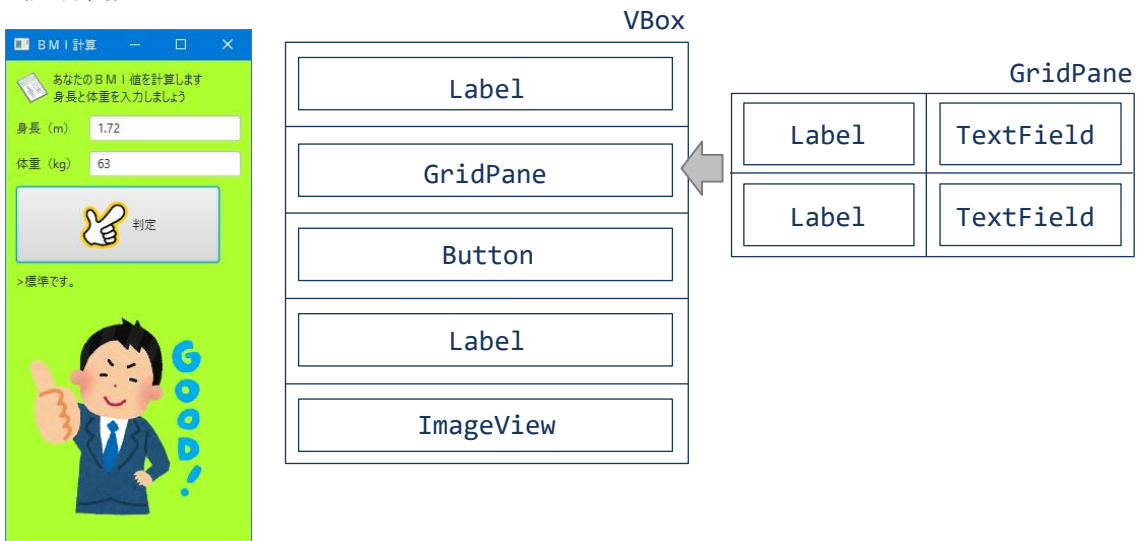
ヒント 2: Label クラスの文字列は setText(String str) メソッドで変更できます。

ヒント 3: ImageView クラスの画像は setImage(Image img) メソッドで変更できます。

※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 Google 検索を用いたストラックアウトアプリを作成しましょう。1 番から 9 番までのパネルが配置されています。検索キーワードを入力してボタンを押すと Google 検索が実行され、そのヒット件数をもとに次のような対応でパネルを打ち抜くことができます。ここで検索できる回数の上限は 15 回とします。現在の残り回数はボタンのラベルに表示します。

〔検索ヒット件数とパネル番号の対応〕

1 番パネル → 10^0 以上 ~ 10^1 未満

2 番パネル → 10^1 以上 ~ 10^2 未満

3 番パネル → 10^2 以上 ~ 10^3 未満

⋮

i 番パネル → $10^{(i-1)}$ 以上 ~ 10^i 未満

※検索ヒット件数からパネル番号への対応は `Math.log10()` を用いると便利です。

〔パネルの種類〕

数値パネル	最新ヒットパネル	ヒットパネル
最初に配置する 9 枚のパネル (100x100 ピクセル画像、右の各パネルも同サイズ) です。3 行 3 列の格子状に配置します。	一番最近に打ち抜いた位置を表すパネルです。	これまでに打ち抜いた位置を表すパネルです。

〔Google 検索支援クラス `GoogleSearch` / インタフェース `GoogleSearchRespond`〕

【`GoogleSearch` クラスの利用】

Google 検索は `GoogleSearch` クラスを用いて以下の 4 ステップで処理します。

- | | |
|-----------------------|--|
| 1. インスタンスの生成 | <code>GoogleSearch gs=new GoogleSearch();</code>
※変数 <code>gs</code> はクラスのメンバーとして宣言 |
| 2. 検索完了の通知を受けるハンドラの登録 | <code>gs.setHandler(new MyHandler());</code> |
| 3. 検索キーワードの設定 | <code>gs.setKeywords("福岡太郎");</code> |
| 4. 検索の実行 | <code>gs.run();</code> |

【ハンドラの宣言】

ハンドラは `GoogleSearchRespond` インタフェースを実装して、継承される 3 つの抽象メソッドをオーバーライドします。各抽象メソッドは以下の役割をもちます。

```
class MyHandler implements GoogleSearchRespond{
    public void keywords(String k){...} // 検索キーワードの受け取り
    public void hits(long c){...} // 検索ヒット件数の受け取り
    public void docs(Document d){...} // 検索結果ページ DOM の受け取り
}
```

※次の 3 つのクラスファイルを同じフォルダに置きます。

`GoogleSearch.class`

`GoogleSearch$TempListener.class`

`GoogleSearchRespond.class`

※インポート文「`import org.w3c.dom.Document;`」を追加します。

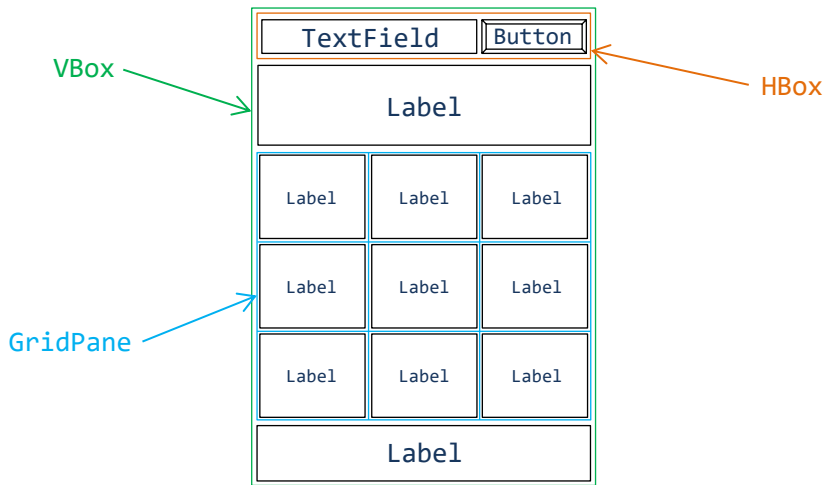
※使用例として `HowToUse.java` を準備しておりますのでご覧ください。

〔処理の組み立て方〕

このアプリはボタンを押したときに処理を開始するようにします。ボタンに表示される検索残回数を 1 減らし、テキストフィールドの内容を読み取って Google 検索を実行します。もし、検索残回数が 0 になったらボタンを無効にします。

実行結果がハンドラを通して戻されたら、検索ヒット件数を取得し、下のメッセージラベルにその件数を表示します。さらに、打ち抜いたパネルには最新ヒットパネルを、打ち抜いていないパネルには対応する数字パネルを、すでに打ち抜いたパネルにはヒットパネルを表示します。もし、すべてのパネルが打ち抜かれた場合はボタンを無効にします。

〔レイアウト例〕



※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〕

