

HCI プログラミング

8回目 ボタン、チェックボックス、ラジオボタン 課題

■課題1 将来のお仕事占いアプリを作成しましょう。4枚の写真を貼ったボタンを配置します。気になる写真のボタンを押すと、将来あなたが就くお仕事が表示されます。

お仕事の例：「建物」が好きなあなたは将来有名な建築家になるでしょう

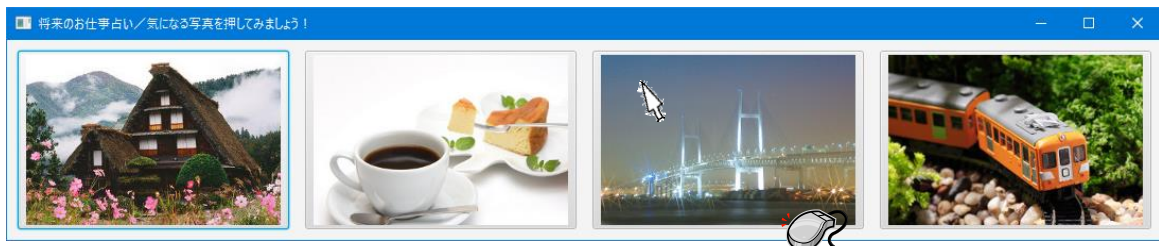
その他の設定は次のとおりです。参考にしましょう。

- レイアウト → 横配置 [HBox]
- レイアウト周りの空白エリア → 10ピクセル [setPadding(new Insets(10));]
- ボタン間の空白エリア → 15ピクセル [setSpacing(15);]
- ウィンドウのタイトル → 将来のお仕事占い／気になる写真を押してみましょう！

ヒント：画像のみのボタンは、画像のみのラベルと同時ようにして生成できます。

※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〕



〔コマンドプロンプト〕

「夜景」が好きなあなたは将来ツアーコンダクターになるでしょう

■課題2 パソコン Web ショップを作成しましょう。パソコンのパーツ毎にチェックボックスを配置します。購入したいパーツにチェックを入れて、購入確定ボタンを押すとお買い上げ合計金額が表示されます。

その他の設定は次のとおりです。参考にしましょう。

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

ボタンやチェックボックス間の空白エリア → 20ピクセル [setSpacing(20);]

文字のサイズ(すべて) → 24ポイント

文字の色(上のラベル) → アジュール [Color.AZURE]

文字の背景色(上のラベル) → 薄みどり [Color.LIGHTSEAGREEN]

文字の色(チェックボックス) → 薄みどり [Color.LIGHTSEAGREEN]

文字の色(下のボタン) → 黒 [Color.BLACK]

ボタンのサイズ → 横 300ピクセル×縦 100ピクセル

シーンの色 → アジュール [Color.AZURE]

※レイアウト VBox の背景色を透明にします

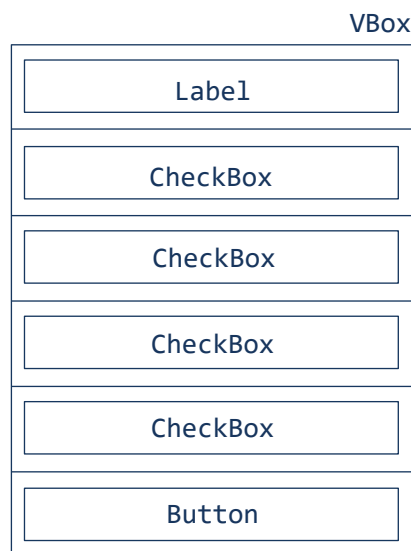
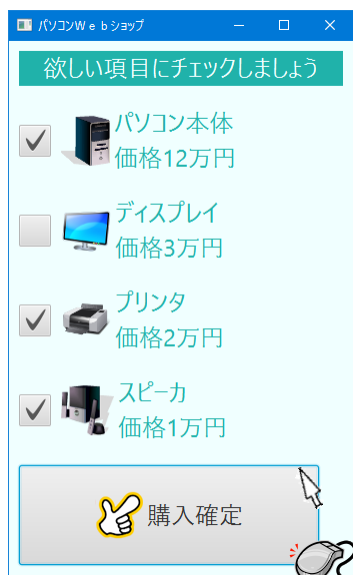
レイアウト VBox の setBackground(null); を実行

ウィンドウのタイトル → パソコンWebショップ

※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〕

合計 15 万円のお買い上げ有難うございます

■課題 3 「宝探しゲーム」を作成しましょう。地図上の草をクリックすると、穴を掘ります。もし宝があれば金塊が表示されます。穴を掘る回数はカウントされ、地図の下に表示されます。穴を掘った場所の上下左右斜め 8 近傍に金塊があれば探知機が反応します。

■作り方のしおり

変数設計

掘り起こし回数やお宝の位置、掘り起こし状態など地図の状態を表現する変数を宣言します。これらの変数は、イベントハンドラから参照したら値を変更したりしますのでメンバー変数として宣言しておきましょう。

- 掘り起こし回数 → int 型変数 counter として表現
- お宝の位置 → 1 行 2 列の int 型配列 posTreasure として表現
- 掘り起こし状態 → 5 行 5 列の int 型配列 map として表現



- アイコン → 1 行 3 列の Image 型配列 imgs として表現
- アイコン用画像枠 → 5 行 5 列の ImageView 型配列 ivs として表現
- 探知機メッセージ → Label 型変数 msg として表現

初期設定〔start()メソッド内にて〕

まず、上で宣言したメンバー変数に初期値を設定していきます。その後、このデータに沿ってレイアウトを構成していきます。

- 掘り起こし回数 counter → 0
- お宝の場所 posTreasure → 乱数
- 掘り起こし状態 map の全配列要素 → 0:草
- アイコンの準備 → 0.png 1.png 2.png
- アイコン用画像枠の全配列要素 → 草アイコン ※掘り起こし状態 map に応じて
- 探知機メッセージの初期文字列 → " - "など

- 宝探しの地図 → 5 行 5 列 Button 型配列
各配列要素ボタンに同行同列のアイコン用画像枠を設定
各配列要素ボタンに識別子を設定 (場所が特定できるように)
ボタン位置(x, y)〔各 int 型〕から id=""+(y*10 + x)〔String 型〕を識別子
ボタン位置は id を整数に変換した値 n を用いて y=n/10 と x=n-y*10 と計算 (※1)
各配列要素ボタンにイベントハンドラを登録

- レイアウト → 下図の例に
- フォント名 → HGPSoeiPresenceEB、20 ポイント
- フォント色 → 上ラベル：茶〔Color.BROWN〕
下レベル：青〔Color.BLUE〕
- シーンの色 → 土色〔Color.SANDYBROWN〕
- タイトル → 宝探しゲーム

※空白エリアの設定などはおおよそで結構です

イベントハンドラ処理

ボタンがクリックされたら実行されます。押されたボタンを特定して、掘り掘り起こし状態を掘った状態〔1:穴〕に変更します。もしお宝があったらその状態〔2:宝〕に変更します。

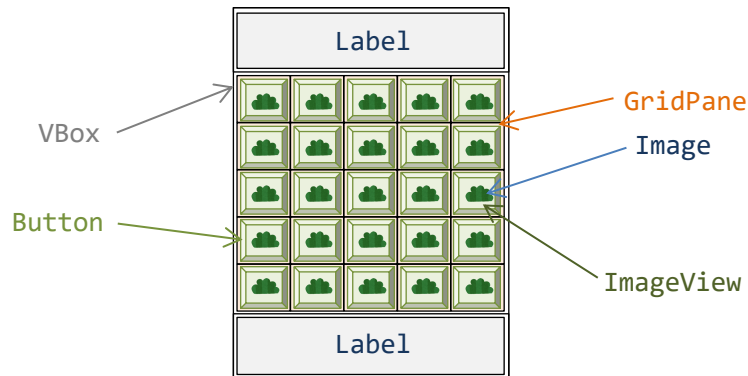
- 押されたボタンの位置を計算 (※1)
- 掘り起こし回数を 1 加算
- 押された位置にお宝があるかどうかで処理を分岐
 - * お宝がある場合 → 掘り起こし状態 map の該当位置を 2:宝に更新
該当ボタンのアイコンを変更
探知機メッセージを更新「〇掘り! ★見つけた!!★」
 - * お宝が無かった場合 → 掘り起こし状態 map の該当位置を 1:穴に更新
該当ボタンのアイコンを変更
探知機メッセージを更新
+8 近傍に宝がある場合「〇掘り! んっ、近いぞ!!」
+8 近傍に宝がない場合「〇掘り! 近くにはなさそうだ」

※その他メッセージは工夫してみましょう

※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

〔レイアウト例〕



〔実行例〕

