```java
import javafx.application.*;
import javafx.scene.*;
import javafx.scene.layout.*;
import javafx.scene.control.*;
import javafx.scene.paint.*;
import javafx.scene.image.*;
import javafx.scene.effect.*;
import javafx.scene.text.*;
import javafx.scene.input.*;
import javafx.stage.*;
import javafx.event.*;
import javafx.geometry.*;
import javafx.collections.*;

////////////////////////////////////////////////////////
// 問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…
////////////////////////////////////////////////////////

public class Assignment8_1 extends Application
{
  public void start(Stage stage) throws Exception
  {
    // ボタンを生成／設定します
    Button[] bts = new Button[4];
    for(int i=0;i<bts.length; i++){
      bts[i] = new Button();
      bts[i].setGraphic(new ImageView("photo"+i+".jpg"));
      bts[i].setId("p"+i);
    }

    // イベントハンドラを設定します
    MyEventHandler actionhandler = new MyEventHandler();
    for(int i=0;i<bts.length; i++)
      bts[i].addEventHandler(ActionEvent.ANY, actionhandler);

    // レイアウトHBoxを生成／設定します
    HBox hb = new HBox();
    ObservableList<Node> lst = hb.getChildren();
    lst.addAll(bts);
    hb.setPadding(new Insets(10));
    hb.setSpacing(15);

    // シーンを生成／設定します
    Scene scene = new Scene(hb);

    // ステージを設定します
    stage.setScene(scene);
    stage.setTitle("将来のお仕事占い／気になる写真を押してみましょう！");

    // ステージを表示します
    stage.show();
  }

  // イベントハンドラ（イベント処理）クラスの宣言
  private class MyEventHandler implements EventHandler<ActionEvent>
  {
    public void handle(ActionEvent e)
    {
      Button bt = (Button)e.getTarget();

      // 押されたボタンにより、対応するメッセージを表示します
      String id = bt.getId();
      if(id.equals("p0")){
        System.out.println("「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");
      }else if(id.equals("p1")){
        System.out.println("「カフェ」が好きなあなたは将来お店をもつことになるでしょう");
      }else if(id.equals("p2")){
        System.out.println("「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");
      }else if(id.equals("p3")){
        System.out.println("「玩具」が好きなあなたは将来夢を与えるお仕事に就くでしょう");
      }
    }
  }

  public static void main(String[] args)
  {
    launch(args);
  }
}

```

```java
import javafx.application.*;
import javafx.scene.*;
import javafx.scene.layout.*;
import javafx.scene.control.*;
import javafx.scene.paint.*;
import javafx.scene.image.*;
import javafx.scene.effect.*;
import javafx.scene.text.*;
import javafx.scene.input.*;
import javafx.stage.*;
import javafx.event.*;
import javafx.geometry.*;
import javafx.collections.*;

//////////////////////////////////////////////////////////
// 問２　パソコンWebショップを作成しましょう。パソコンのパ…
//////////////////////////////////////////////////////////

public class Assignment8_2 extends Application
{
  private CheckBox[] cbs;

  // 商品と価格の配列
  private String[] product = {"パソコン本体","ディスプレイ","プリンタ","スピーカ"};
  private int[] price = {12, 3, 2, 1};

  // 画像ファイルの配列
  private String[] files = {"cpu.png","display.png","printer.png","speaker.png","click.png"};

  public void start(Stage stage) throws Exception
  {
    // 画像を準備します
    ImageView[] ivs = new ImageView[files.length];
    for(int i=0; i<files.length; i++)
      ivs[i] = new ImageView(files[i]);

    // フォントを生成します
    Font ft = new Font(24);

    // ラベルを生成／設定します
    Label top = new Label("  欲しい項目にチェックしましょう  ");
    top.setFont(ft);
    top.setBackground(new Background(new BackgroundFill(Color.LIGHTSEAGREEN,null,null)));
    top.setTextFill(Color.AZURE);

    // チェックボックスを生成／設定します
    cbs = new CheckBox[product.length];
    for(int i=0; i<cbs.length; i++){
      cbs[i] = new CheckBox(product[i]+"\n価格"+price[i]+"万円");
      cbs[i].setFont(ft);
      cbs[i].setGraphic(ivs[i]);
      cbs[i].setTextFill(Color.LIGHTSEAGREEN);
    }

    // ボタンを生成／設定します
    Button ok = new Button("購入確定");
    ok.setPrefSize(300,100);
    ok.setFont(ft);
    ok.setGraphic(ivs[4]);

    // ボタンにイベントハンドラを設定します
    MyEventHandler actionhandler = new MyEventHandler();
    ok.addEventHandler(ActionEvent.ANY, actionhandler);

    // レイアウトVBoxを生成／設定します
    VBox vb = new VBox();
    ObservableList<Node> lst = vb.getChildren();
    lst.add(top);
    lst.addAll(cbs);
    lst.add(ok);
    vb.setPadding(new Insets(10));
    vb.setSpacing(20);
    vb.setBackground(null);

    // シーンを生成／設定します
    Scene scene = new Scene(vb);
    scene.setFill(Color.AZURE);

    // ステージを設定します
    stage.setScene(scene);
    stage.setTitle("パソコンＷｅｂショップ");

    // ステージを表示します
    stage.show();
  }

  // イベントハンドラ（イベント処理）クラスの宣言
  private class MyEventHandler implements EventHandler<ActionEvent>
  {
    public void handle(ActionEvent e)
    {
      int sum = 0;
      for(int i=0; i<cbs.length; i++)
        if(cbs[i].isSelected()) sum += price[i];
```

```
95          System.out.println("合計"+sum+"万円のお買い上げ有難うございます");
96        }
97      }
98
99      public static void main(String[] args)
100     {
101       launch(args);
102     }
103 }
104
```

```java
import javafx.application.*;
import javafx.scene.*;
import javafx.scene.layout.*;
import javafx.scene.control.*;
import javafx.scene.paint.*;
import javafx.scene.image.*;
import javafx.scene.effect.*;
import javafx.scene.text.*;
import javafx.scene.input.*;
import javafx.stage.*;
import javafx.event.*;
import javafx.geometry.*;
import javafx.collections.*;

////////////////////////////////////////////////////////
// 問３　「宝探しゲーム」を作成しましょう。地図上の草を…
////////////////////////////////////////////////////////

public class Assignment8_3 extends Application
{
  private int counter;       // 掘った回数
  private int[] posTreasure; // お宝の位置
  private int[][] map;       // 地図の掘り起こし状態（0草,1穴,2宝）
  private Image[] imgs;      // 地図に貼るアイコン（草・穴・宝）
  private ImageView[][] ivs; // 地図に配置するアイコン用画像枠
  private Label msg;         // メッセージラベル

  public void start(Stage stage) throws Exception
  {
    // 掘った回数を初期化します
    counter = 0;

    // お宝の位置を乱数で決定します
    posTreasure = new int[2];
    posTreasure[0] = (int)(Math.random()*5);
    posTreasure[1] = (int)(Math.random()*5);

    // 掘り起こし状態をすべて草0にします
    // - デフォルト値は0
    map = new int[5][5];

    // 地図に貼るアイコンを準備します
    imgs = new Image[3];
    for(int i=0; i<imgs.length; i++)
      imgs[i] = new Image(i+".png");

    // アイコン用画像枠を準備します
    ivs = new ImageView[5][5];
    for(int j=0; j<ivs.length; j++)
      for(int i=0; i<ivs[j].length; i++){
        ivs[j][i] = new ImageView(imgs[map[j][i]]);
        ivs[j][i].setPreserveRatio(true);
        ivs[j][i].setFitWidth(50);
      }

    // フォントを準備します
    Font ft=new Font("HGPSoeiPresenceEB",20);

    // メッセージラベルを準備します
    Label top = new Label("宝があると思う場所を\nクリックしましょう");
    top.setTextFill(Color.BROWN);
    top.setFont(ft);
    ImageView sh = new ImageView("shovel.png");
    sh.setPreserveRatio(true);
    sh.setFitWidth(30);
    top.setGraphic(sh);
    msg = new Label("-");
    msg.setPrefHeight(40);
    msg.setTextFill(Color.BLUE);
    msg.setFont(ft);

    // ＵＩを作成します
    // - ボタンを生成し、アイコンを貼ります
    // - ボタンの位置が分かるように識別子を設定します
    Button[][] board = new Button[5][5];
    for(int j=0; j<board.length; j++)
      for(int i=0; i<board[j].length; i++){
        board[j][i] = new Button();
        board[j][i].setGraphic(ivs[j][i]);
        board[j][i].setId(""+(j*10+i));
      }

    // - ボタンにイベントハンドラを設定します
    MyEventHandler handler = new MyEventHandler();
    for(int j=0; j<board.length; j++)
      for(int i=0; i<board[j].length; i++)
        board[j][i].addEventHandler(ActionEvent.ANY, handler);

    // - レイアウトGridPaneを生成してボタンを配置します
    GridPane gp = new GridPane();
    for(int j=0; j<board.length; j++)
      for(int i=0; i<board[j].length; i++)
        gp.add(board[j][i], i, j);
```

```
94
95      gp.setBackground(null);
96
97      // - レイアウトVBoxにラベルやGridPaneを配置します
98      VBox vb = new VBox();
99      ObservableList<Node> lst = vb.getChildren();
100     lst.add(top);
101     lst.add(gp);
102     lst.add(msg);
103     vb.setBackground(null);
104     vb.setSpacing(5);
105     vb.setPadding(new Insets(5));
106
107     // シーンを生成／設定します
108     Scene scene = new Scene(vb);
109     scene.setFill(Color.SANDYBROWN);
110
111     // ステージを設定します
112     stage.setScene(scene);
113     stage.setTitle("宝探しゲーム");
114     stage.setResizable(false);
115     // ウィンドウサイズを固定する場合に
116     // 以下をしないと余白が勝手に出現します
117     // バグです！そのうち修正されると思います
118     stage.sizeToScene();
119
120     // ステージを表示します
121     stage.show();
122 }
123
124 // イベントハンドラ（イベント処理）クラスの宣言
125 private class MyEventHandler implements EventHandler<ActionEvent>
126 {
127   public void handle(ActionEvent e){
128     Button bt=(Button)e.getTarget();
129
130     // 押されたボタンの位置をIDより特定します
131     int id = Integer.parseInt(bt.getId());
132     int by = id/10;
133     int bx = id - by*10;
134
135     // 押されたボタン位置とお宝の位置の差を求めます
136     int tmpx = bx-posTreasure[0];
137     int tmpy = by-posTreasure[1];
138
139     if(tmpx==0 && tmpy==0){
140       // お宝がある場合
141       if(map[by][bx]==0){
142         // １度目
143         counter++;
144         map[by][bx] = 2;
145         ivs[by][bx].setImage(imgs[map[by][bx]]);
146         msg.setGraphic(null);
147         msg.setText(counter+"掘り！　★見つけた!!★");
148       }else{
149         // ２度押すと…
150         msg.setGraphic(null);
151         msg.setText("さっき見つけましたね");
152       }
153     }
154     else{
155       // お宝がない場合
156       if(map[by][bx]==0){
157         // １度目
158         counter++;
159         map[by][bx] = 1;
160         ivs[by][bx].setImage(imgs[map[by][bx]]);
161
162         if(map[posTreasure[1]][posTreasure[0]]!=2){
163           // まだお宝が見つかっていない場合
164           if( tmpx>=-1 && tmpx<=1 && tmpy>=-1 && tmpy<=1){
165             // 8近傍にお宝がある場合
166             ImageView det = new ImageView("detector.png");
167             det.setPreserveRatio(true);
168             det.setFitHeight(35);
169             msg.setGraphic(det);
170             msg.setContentDisplay(ContentDisplay.RIGHT);
171             msg.setText(counter+"掘り！　んっ、近いぞ!!");
172           }else{
173             // 8近傍にお宝がない場合
174             msg.setGraphic(null);
175             msg.setText(counter+"掘り！　近くにはなさそうだ");
176           }
177         }else{
178           // 既にお宝が見つかっている場合
179           msg.setGraphic(null);
180           msg.setText("お宝は１つなんです");
181         }
182
183       }else{
184         // ２度押すと…
185         msg.setGraphic(null);
```

```
          msg.setText("すでに掘ってあります");
        }
      }
    }
  }

  public static void main(String[] args)
  {
    launch(args);
  }
}
```