```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.scene.canvas.*;
import javafx.scene.shape.*;
import javafx.stage.*;
import javafx.event.*;
import javafx.geometry.*;
import javafx.collections.*;

// 追加文のインポート（入出力ストリーム用）
import java.io.*;
// 追加文のインポート（HTMLEditor用）
import javafx.scene.web.*;
// 追加文のインポート（印刷用）
import javafx.print.*;

//////////////////////////////////////////////////////////
// 課題３　簡易HTMLエディタを作ってみよう
//////////////////////////////////////////////////////////

public class Assignment12_3 extends Application
{
  private HTMLEditor he;
  private Stage st;

  public void start(Stage stage)
  {
    st = stage;

    // HTMLエディタの生成
    he = new HTMLEditor();

    // メニューの生成
    MenuBar mb = new MenuBar();
    mb.setBackground(new Background(new BackgroundFill(Color.LIGHTGRAY,null,null)));
    Menu m = new Menu("ファイル _F");
    MenuItem[] mis = new MenuItem[6];
    mis[0] = new MenuItem("開く _O");
    mis[1] = new MenuItem("保存 _S");
    mis[2] = new SeparatorMenuItem();
    mis[3] = new MenuItem("印刷 _P");
    mis[4] = new SeparatorMenuItem();
    mis[5] = new MenuItem("終了 _Q");
    mis[0].setId("Open");
    mis[1].setId("Save");
    mis[3].setId("Print");
    mis[5].setId("Quit");

    // メニューにイベントハンドラを設定します
    MenuEventHandler mh = new MenuEventHandler();
    m.addEventHandler(ActionEvent.ANY, mh);

    // メニューの構成
    ObservableList<Menu> mlst = mb.getMenus();
    mlst.add(m);
    ObservableList<MenuItem> milst = m.getItems();
    milst.addAll(mis);

    // レイアウトの組み立て
    BorderPane bp = new BorderPane();
    bp.setTop(mb);
    bp.setCenter(he);

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

    // ステージを設定します
    stage.setScene(scene);
    stage.setTitle("簡易HTMLアプリ");

    // アプリアイコンの設定
    ObservableList<Image> ilst = stage.getIcons();
    ilst.add(new Image("htmleditor.png"));

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

  // イベントハンドラ（メニュー用）クラスの宣言
  private class MenuEventHandler implements EventHandler<ActionEvent>
  {
    public void handle(ActionEvent e)
    {
      MenuItem mi = (MenuItem)e.getTarget();
      String id = mi.getId();
```

```
94
95      if(id.equals("Open")){
96        // HTMLファイルの読み込み
97        FileChooser fc = new FileChooser();
98        fc.setTitle("HTMLファイルを開きます");
99        ObservableList<FileChooser.ExtensionFilter> flst = fc.getExtensionFilters();
100       flst.add(new FileChooser.ExtensionFilter("HTMLファイル", "*.html"));
101       File htmlfile = fc.showOpenDialog(st);
102       if(htmlfile != null){
103         BufferedReader br;
104         String str;
105         String htmlstring=null;
106         try{
107           br = new BufferedReader(new FileReader(htmlfile));
108           while((str = br.readLine()) != null){
109             if(htmlstring==null) htmlstring = str;
110             else htmlstring += str;
111           }
112           he.setHtmlText(htmlstring);
113           br.close();
114         }catch(IOException er){}
115       }
116     }else if(id.equals("Save")){
117       // HTMLファイルの保存
118       FileChooser fc = new FileChooser();
119       fc.setTitle("HTMLファイルを保存します");
120       ObservableList<FileChooser.ExtensionFilter> flst = fc.getExtensionFilters();
121       flst.add(new FileChooser.ExtensionFilter("HTMLファイル", "*.html"));
122       File htmlfile = fc.showSaveDialog(st);
123       if(htmlfile != null){
124         PrintWriter pw;
125         try{
126           pw = new PrintWriter(new BufferedWriter(new FileWriter(htmlfile, false)));
127           pw.println(he.getHtmlText());
128           pw.close();
129         }catch(IOException er){}
130       }
131     }else if(id.equals("Print")){
132       // HTMLファイルの印刷
133       PrinterJob pj = PrinterJob.createPrinterJob(Printer.getDefaultPrinter());
134       he.print(pj);
135       pj.endJob();
136     }else if(id.equals("Quit")){
137       // アプリを終了
138       Platform.exit();
139     }
140   }
141 }
142
143 public static void main(String[] args)
144 {
145   launch(args);
146 }
147 }
148
```