

問題 1

```
import java.awt.*;
import java.awt.event.*;

class MenuFrame2 extends Frame
implements ActionListener, ItemListener {
    TextArea ta;

    public static void main(String args[]) {
        MenuFrame2 mf2 = new MenuFrame2("MyFrame");
        mf2.setSize(200, 200);
        mf2.setVisible(true);
    }

    MenuFrame2(String title) {
        // スーパークラスのコンストラクタを呼び出す
        super(title);

        // ウィンドウイベントを受け取るために登録する
        addWindowListener(new WindowAdapter() {
            public void windowClosing(WindowEvent we) {
                System.exit(0);
            }
        });

        // メニューバーを作成して設定する
        MenuBar mb = new MenuBar();
        setMenuBar(mb);

        // メニューAを作成する
        Menu a = new Menu("A");
        mb.add(a);
        MenuItem a1 = new MenuItem("A1");
        a1.addActionListener(this);
        a.add(a1);
        MenuItem a2 = new MenuItem("A2");
```

```
a2.addActionListener(this);
a.add(a2);
MenuItem a3 = new MenuItem("A3");
a3.addActionListener(this);
a.add(a3);

// メニューBを作成する
Menu b = new Menu("B");
mb.add(b);
MenuItem b1 = new MenuItem("B1");
b1.addActionListener(this);
b.add(b1);
MenuItem b2 = new MenuItem("B2");
b2.addActionListener(this);
b.add(b2);

// B3のサブメニューを作成する
Menu b3 = new Menu("B3");
b.add(b3);
MenuItem b31 = new MenuItem("B31");
b31.addActionListener(this);
b3.add(b31);
MenuItem b32 = new MenuItem("B32");
b32.addActionListener(this);
b3.add(b32);
MenuItem b33 = new MenuItem("B33");
b33.addActionListener(this);
b3.add(b33);

// メニューCを作成する
Menu c = new Menu("C");
mb.add(c);
MenuItem c1 = new MenuItem("C1");
c1.addActionListener(this);
c.add(c1);
MenuItem c2 = new MenuItem("C2");
```

```

c2.addActionListener(this);
c.add(c2);

// メニューDを作成する
Menu d = new Menu("D");
mb.add(d);
CheckboxMenuItem d1 = new CheckboxMenuItem("D1");
d1.addItemListener(this);
d.add(d1);
CheckboxMenuItem d2 = new CheckboxMenuItem("D2");
d2.addItemListener(this);
d.add(d2);

// テキストエリアを作成し、フレームに追加する
ta = new TextArea(10, 20);
add(ta, "Center");
}

public void actionPerformed(ActionEvent ae) {
    ta.append("ActionEvent: " +
        ae.getActionCommand() + "¥n");
}

public void itemStateChanged(ItemEvent ie) {
    CheckboxMenuItem cbmi = (CheckboxMenuItem)ie.getSource();
    ta.append("ItemEvent: " +
        cbmi.getLabel() + "¥n");
}
}

```

問題 2

```
import java.awt.*;
import java.awt.event.*;
import java.util.*;

public class DateDialogDemo extends Frame
implements ActionListener {
    public static void main(String args[]) {
        DateDialogDemo ddd = new DateDialogDemo();
        ddd.setSize(200, 100);
        ddd.setVisible(true);
    }

    DateDialogDemo() {
        super("Date Dialog Demo");

        // レイアウトマネージャを設定する
        setLayout(new FlowLayout());

        // ボタンを作成する
        Button b = new Button("Date Dialog");
        b.addActionListener(this);
        add(b);

        // 無名インナクラスがウィンドウイベントを処理する
        addWindowListener(new WindowAdapter() {
            public void windowClosing(WindowEvent we) {
                System.exit(0);
            }
        });
    }

    public void actionPerformed(ActionEvent ae) {
        DateDialog dd =
            new DateDialog(this, "Date Dialog", true);
        dd.setVisible(true);
    }
}
```

```
}  
}
```

```
class DateDialog extends Dialog  
implements ActionListener {  
    static String monthNames[] = {"January",  
        "February", "March", "April", "May",  
        "June", "July", "August", "September",  
        "October", "November", "December" };  
  
    DateDialog(Frame parent, String title, boolean mode) {  
        super(parent, title, mode);  
  
        // 日付情報を用いて文字列を作成する  
        Calendar calendar = Calendar.getInstance();  
        int month = calendar.get(Calendar.MONTH);  
        int date = calendar.get(Calendar.DAY_OF_MONTH);  
        int year = calendar.get(Calendar.YEAR);  
        String str = monthNames[month] + " ";  
        str += date + " ";  
        str += year;  
  
        // Center パネルを作成して追加する  
        Panel pc = new Panel();  
        Label label = new Label(str, Label.CENTER);  
        pc.add(label);  
        add(pc, BorderLayout.CENTER);  
  
        // South パネルを作成して追加する  
        Panel ps = new Panel();  
        Button ok = new Button("OK");  
        ok.addActionListener(this);  
        ps.add(ok);  
        add(ps, BorderLayout.SOUTH);  
  
        // コンポーネントをレイアウトし、このダイアログボックスの
```

```
// 初期サイズを設定する
pack();

// 無名インナクラスがウィンドウイベントを処理する
addWindowListener(new WindowAdapter() {
    public void windowClosing(WindowEvent we) {
        dispose();
    }
});
}

public Insets getInsets() {
    return new Insets(40, 20, 20, 20);
}

public void actionPerformed(ActionEvent ae) {
    dispose();
}
}
```