

Java プログラムの例（ファイルに毎にファイルを分けて保存する）

```
//
//クラス Date, Todo, TodoList は、前回と同じ
//

//Controller クラス
public class TodoController {
    private TodoView todoview;
    private TodoList todolist;

    public void setTodoView(TodoView tdv) {
        this.todoview = tdv;
    }
    public TodoView getTodoView() {
        return todoview;
    }
    public void setTodoList(TodoList tdl) {
        todolist = tdl;
    }
    public TodoList getTodoList() {
        return todolist;
    }
    public void add(int p, String n,
                    int y, int m, int d) {
        //Todo を作る
        // (TodoMain1 の main での追加処理を、
        // 仮引数を使うように変更した)
        //Date インスタンスを作る
        Date d1 = new Date();
        d1.setYear(y);
        d1.setMonth(m);
        d1.setDay(d);
        //Todo インスタンスを作る
        Todo t1 = new Todo();
        t1.setName(n);
        t1.setPriority(p);
        t1.setDeadline(d1);
        //作った Todo をリストに追加する
        todolist.addTodo(t1);
    }
}

//View クラス
import java.awt.*;
import java.awt.event.*;

public class TodoView implements ActionListener {

    private TodoController todocontroller;

    private TodoList todolist;
    //リストを表示するための部品
    private Frame f1;
    private List l1;
    //Todo を追加するための部品
    private Frame f2;
    private Button b1;
    private TextField name, priority, year, month, day;
    TodoView() {
        //リストを表示するための画面を作る
        f1 = new Frame();
        l1 = new List(10);
        f1.add(l1, BorderLayout.CENTER);
        f1.setSize(500, 300);
        f1.setVisible(true);
        //Todo を追加するための画面を作る
        f2 = new Frame();
        b1 = new Button("add");
        priority = new TextField(1);
        name = new TextField(10);
        year = new TextField(4);
        month = new TextField(2);
        day = new TextField(2);
        f2.setLayout(new FlowLayout());
        f2.add(priority);
        f2.add(name);
        f2.add(year);
        f2.add(month);
        f2.add(day);
        f2.add(b1);
        f2.pack();
        f2.setVisible(true);
        //add ボタンにリスナを登録する
        b1.addActionListener(this);
    }
    public void setTodoList(TodoList tdl) {
        todolist = tdl;
    }
    public TodoList getTodoList() {
        return todolist;
    }
    public void setTodoController(TodoController tdc) {
        todocontroller = tdc;
    }
    public TodoController getTodoController() {
        return todocontroller;
    }
    public void actionPerformed(ActionEvent e) {
        int p, y, m, d;

```

```

String n;
//押されたボタンがどれなのかを調べる
if(e.getActionCommand().equals("add")) {
    //テキストフィールドから情報を取得する
    //メソッド parseInt は取得した文字列を
    //int 型に変換する
    p = Integer.parseInt(
        priority.getText());
    y = Integer.parseInt(
        year.getText());
    m = Integer.parseInt(
        month.getText());
    d = Integer.parseInt(
        day.getText());
    n = name.getText();
    //Controller のメソッドを呼び出す
    todocontroller.add(p, n, y, m, d);
}

public void update() {
    //Todo をリストに出力する
    System.out.println("update");
    //リストの内容を全て消す
    l1.removeAll();
    Todo[] l = todolist.getList();
    //TodoList の Todo を繰り返し出力する
    for(int i=0; i<l.length; i++) {
        //リストに追加する Todo の情報の
        //文字列を作る
        String str = "";
        str += "[" + String.valueOf(i) + " ] ";
        if(l[i]!=null) {
            //Todo を str に追加する
            str += "(" + l[i].getPriority() + " ) ";
            str += l[i].getName() + " ";
            //期限日を str に追加する
            Date d = l[i].getDeadline();
            str += d.getYear()
                + "/" + d.getMonth()
                + "/" + d.getDay();
        }
        l1.add(str);
    }
}

//Main クラス
public class TodoMain3 {
    public static void main(String[] args) {
        //View を作成する
        TodoView todoview = new TodoView();
        //Model を作成する
        TodoList todolist = new TodoList();
        //Controller を作成する
        TodoController todocontroller
            = new TodoController();

        //View と Model を相互に関係付ける
        todolist.setTodoView(todoview);
        todoview.setTodoList(todolist);

        //View と Controller を相互に参照する
        todoview.setTodoController(todocontroller);
        todocontroller.setTodoView(todoview);

        //Controller から Model を参照する
        todocontroller.setTodoList(todolist);

        //最初に View を更新しておく
        todoview.update();
    }
}

```