

プログラム設計

<http://bit.ly/design4d>

MVCによるクラス設計 (2)

後期 第6週

2019/11/7

MVCとは

ソフトウェア（特にGUIやWebブラウザが伴うアプリケーション）を開発する際、クラスを3つのタイプに分けて設計します。

▶ Model

ソフトウェアで管理されるデータ持つクラス

▶ View

利用者のユーザインタフェースとなるクラス

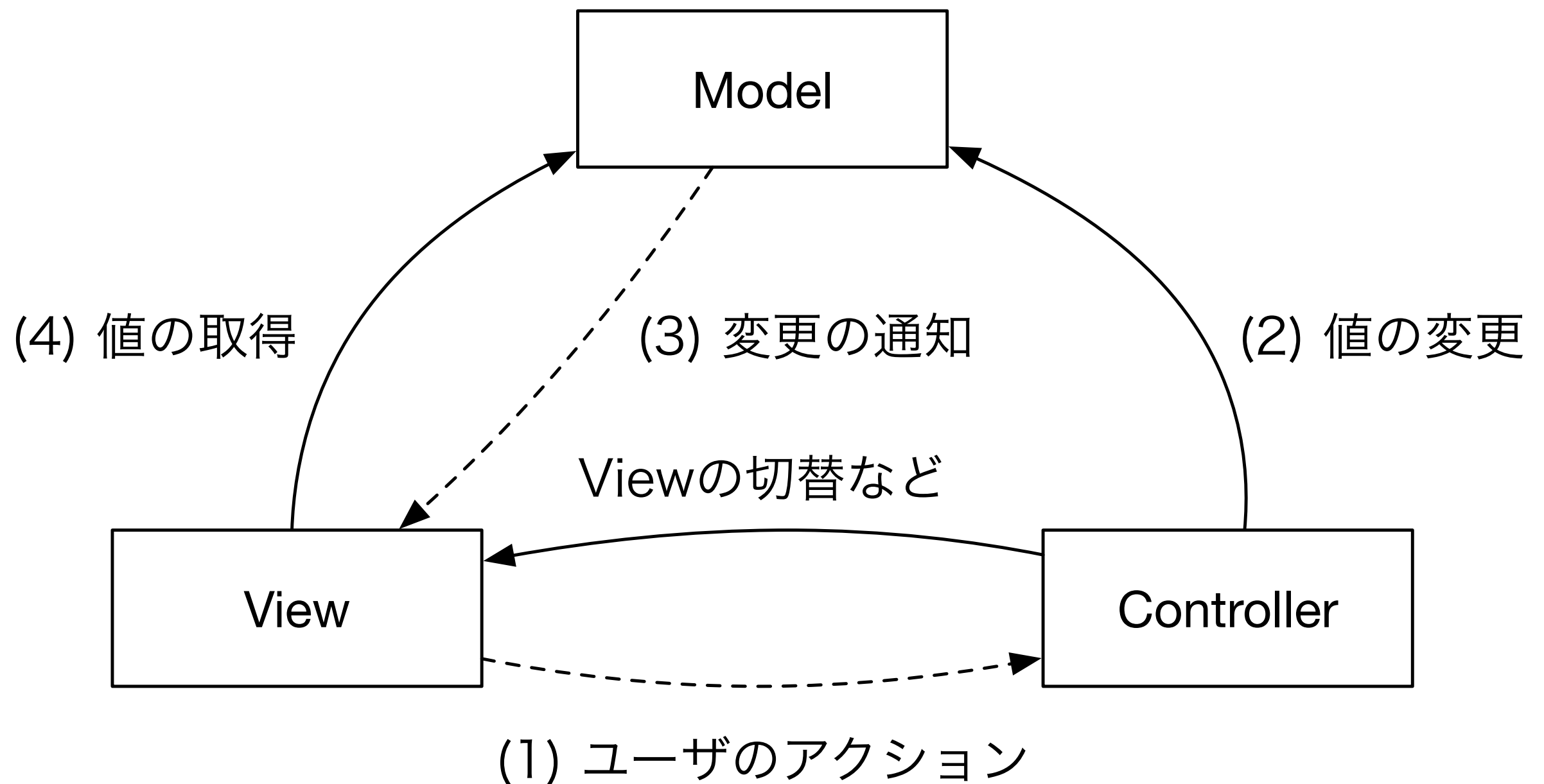
▶ Controller

利用者からの要求に応じてソフトウェアの処理を制御するクラス

前回より

MVCの関係

MVCの一般的な動きの流れは(1)~(4)のようになります。



今回は

前回までに作成したModelとViewの部分に、**Controllerを追加して**アプリケーションを完成させます。

前回未完成の人は、今回ダウンロードしたファイル2_05_todo_ans.astaを使えます。

【課題の準備】

演習室で作業する前に、以下のコマンドを
入れるだけで準備が完了する

```
$ mygitclone4d 「自分のGitHubユーザ名」  
$ cd prog4d-(ユーザ名)  
$ ./myconf
```

※本体をシャットダウンするまでは、
上記「mygitclone」と「myconf」の設定は有効です

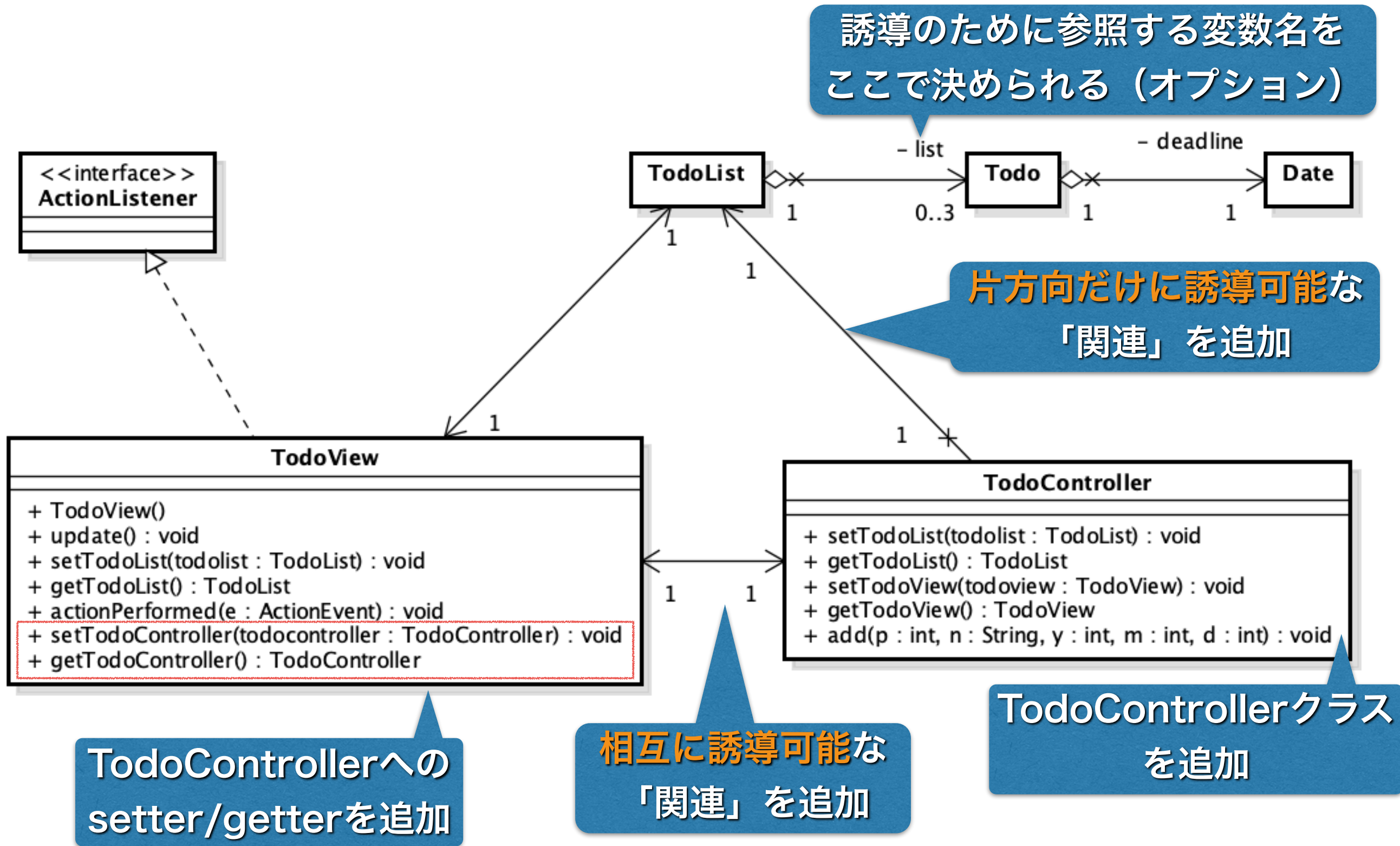
【課題の準備】

以下の流れで、課題のプログラムを作るためのフォルダを準備しましょう。

1. 端末を起動して、以下のコマンドを実行して後期第6週のフォルダを作る
\$ cd prog4d-(ユーザ名) (←既に移動しているなら不要)
\$ mkdir week206
\$ cd week206

※課題で作るファイル名は各自で決めて構いません。

① Controllerを作る



② スケルトンコードを生成する

スケルトンコードの作成で、前回作成したJavaコードを上書きしてしまわないように注意してください。（前回完成させている人は、TodoControllerのみのスケルトンコードを生成した方が良いです。）

③ クラス間の関係のsetter/getterを実装する
クラスTodoViewとTodoControllerが相互に参照できるように、また、TodoControllerからTodoListを参照できるようにするために、それぞれのクラスにsetter/getterを実装します。

▶ クラスTodoControllerに実装

setTodoView(), getTodoView()

setTodoList(), getTodoList()

▶ クラスTodoViewに実装

setTodoController(), getTodoController()

(前回完成させている人は、スケルトンコードを生成せずに、手動で追加・実装した方が良いです。)

④ Controllerのメソッドを実装する

MVCモデルの「(2) 値の変更」に相当するメソッドを実装します。 (add.javaから入手可能)

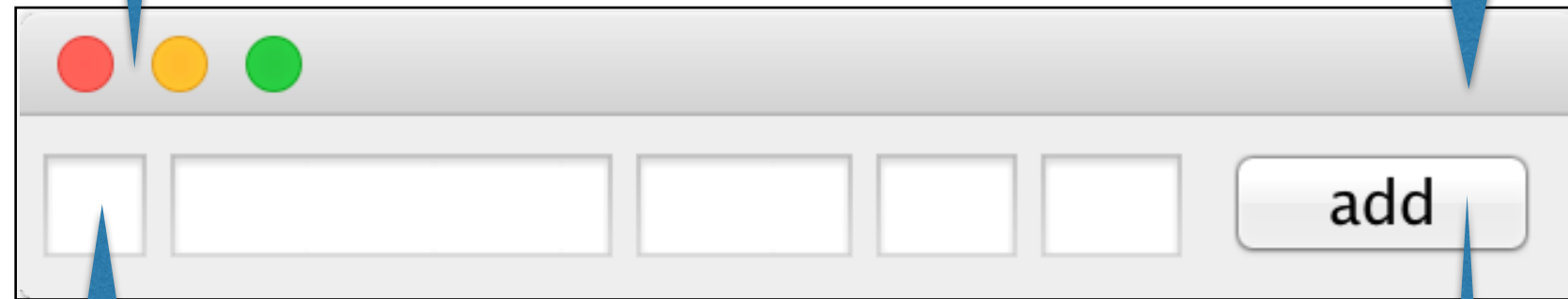
```
public void add(int p, String n, int y, int m, int d) {  
    //Dateインスタンスを作る  
  
    Date d1 = new Date();  
    d1.setYear(y);  
    d1.setMonth(m);  
    d1.setDay(d);  
    //Todoインスタンスを作る  
  
    Todo t1 = new Todo();  
    t1.setName(n);  
    t1.setPriority(p);  
    t1.setDeadline(d1);  
    //作ったTodoをリストに追加する  
  
    todolist.addTodo(t1);  
}
```

⑤ Viewのユーザインターフェースを作る

クラスTodoViewの**コンストラクタ**に、次のような「Todoを追加する」用のGUIを作る処理を追加します。（前回⑤の続きに追加する）

前回⑤のフレームとは別のフレームをもう一つ作る

レイアウトはFlowLayoutにして部品を追加する



5つのTextFieldを作る（左から順に、「優先順位」「名前」「年」「月」「日」を入力するための部品）

このボタンを押すと、TextFieldに入力したTodoが追加される（addActionListenerで自分自身をリスナに登録しておく）

⑥ Viewのイベント処理を実装する

MVCモデルの「(1) ユーザのアクション」に相当するメソッドを実装します。(action.javaから入手可能)

```
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  
  
        //name, priority, year, month, dayは、TextFieldの変数であり、  
        //それぞれ「名前」「優先順位」「年」「月」「日」が入力されているとする  
  
        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「Todoを追加する」メソッドを呼び出す  
        todocontroller.add(p, n, y, m, d);  
    }  
}
```

⑦ プログラムを実行する

mainを持ったファイル2_06_TodoMain3.javaも含め、これまで作った全てのファイルをコンパイルして、**完成したアプリケーションの動作**を確認してみます。（mainでは、MVCのクラス間の関係を作るだけの処理をしています。）

【課題の提出】

以下の流れで、作ったCプログラムをGitHubにプッシュして、Webサイトで確認してみましょう。

1. 端末内で、以下のコマンドで課題を提出

```
$ git add -A
```

```
$ git commit -m “課題6-1提出”
```

```
$ git push origin master
```

2. 自分のリポジトリを開いて、提出したファイルがプッシュされているか確認する

[https://github.com/nit-ibaraki-prog4d-2019/prog4d-\(ユーザー名\)](https://github.com/nit-ibaraki-prog4d-2019/prog4d-(ユーザー名))

今回の小テスト

第2週までの内容で小テストを実施します。

試験範囲

▶ 第1週～第6週

▶ クラス図

▶ astahによるJavaスケルトンコード生成

▶ MVCに基づいたクラス設計

▶ GUIアプリケーションの開発

個人PCでのastahの利用について

- ▶ 電子情報工学科の学生なら、個人のPCにastahを自由にインストールして使える
- ▶ 以下の「無料トライアル」からダウンロードする
<http://astah.change-vision.com/ja/product/astah-professional.html>
- ▶ ライセンスを登録する
(ライセンスはUSBメモリまたはPCを持参してくれば、直接コピーして渡します。)

定期試験の実施について

試験中に使用できるもの

- 筆記用具

(メモ用紙が必要な人には試験中に配布する)

- 演習室のコンピューター台

(一つの机に一人の配置で、座る場所はどこでもよい)

定期試験の実施について

試験中に参照できるもの

- 自分のホームディレクトリ（ホームフォルダ）以下に保存されているファイル
（定期試験では紙媒体のものは参照不可）
- 授業の資料や自分のGitHubリポジトリなどは事前にダウンロードまたはコピーしておく
- 上記以外の情報を参照することは不正行為とする
例：USBで接続された機器に保存されているファイルの参照
Webブラウザ、ネットワークを介した情報の参照
自分のPCを使用する、など

ネットワークの遮断について

- 試験開始5分後に演習室外へのネットワーク接続を切断する
- 試験開始60分後にネットワーク接続を戻す
- それ以降は、GitHubへの提出のためのコマンドに限ってネットワーク利用が可能（それ以外は不正行為とする）

講義資料のダウンロードについて

演習室で作業する前に、以下のコマンドを入
れると講義資料のリポジトリがダウンロードされる

```
$ mygitclone-pd
```

ダウンロードが完了すると、
ホーム以下に作られた「lecture」フォルダの中に
資料などが保存されています

※本体をシャットダウンするまではPCに残ります