応用プログラミングII
第1回

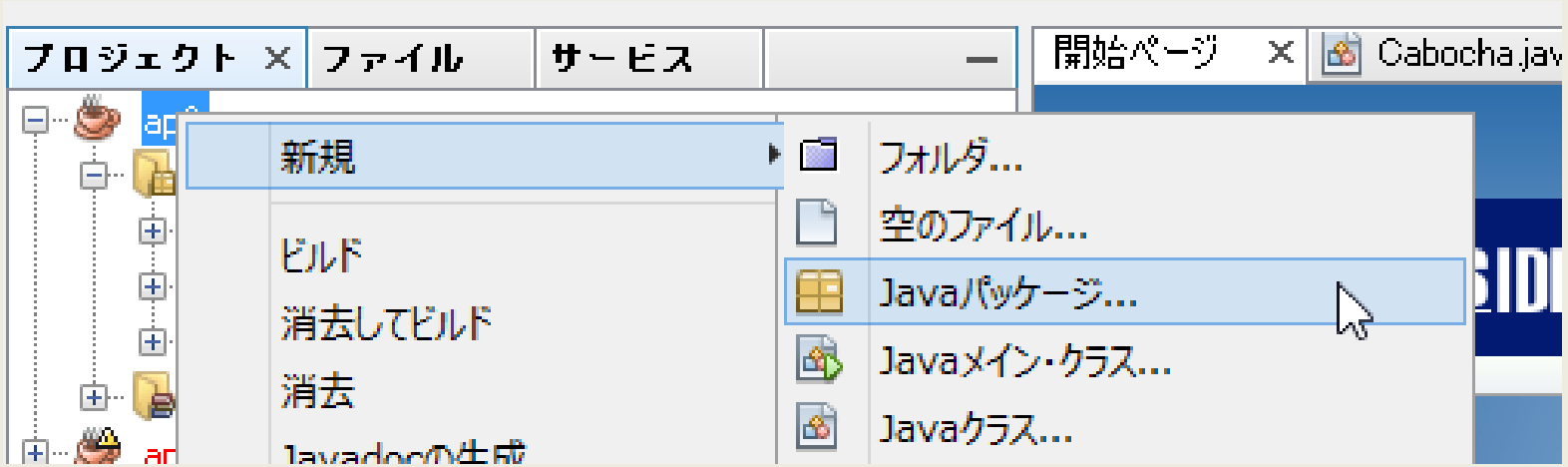
岩下，柴田

本日の内容

- 入出力（ファイル操作）
  1. ファイル入出力の基本
  2. ファイル情報の表示
  3. テキストファイルの読み書き
  4. GUIによるファイルの選択

準備

- プロジェクト「ap2」を使用
- ap2の下にパッケージ「kougi01」を作成

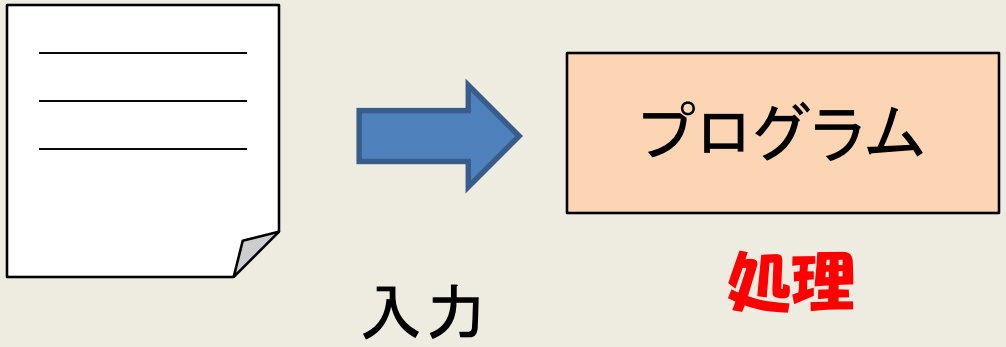


パッケージ「ap2」を右クリック ⇒ Javaパッケージ

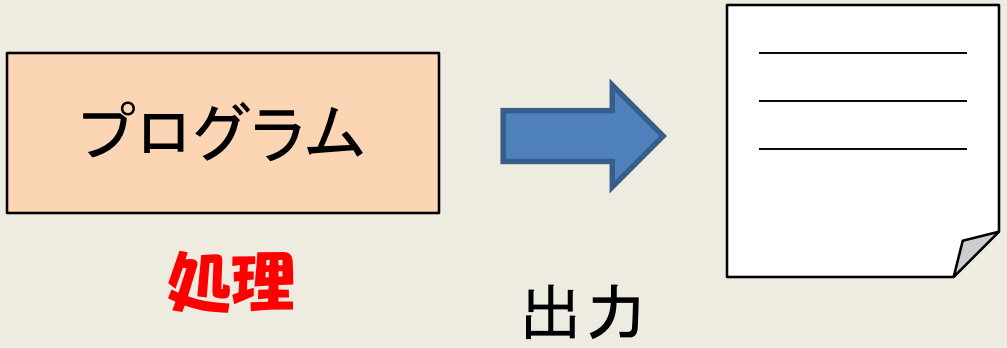
1. ファイル入出力

1. ファイル入出力

【ファイル入力】



【ファイル出力】



ファイルへのアクセス：Fileクラス

- 使用するクラス：java.io.File
- 主なコンストラクタ

```
File(String pathname)
File(File parent, String child)
File(String parent, String child)
```

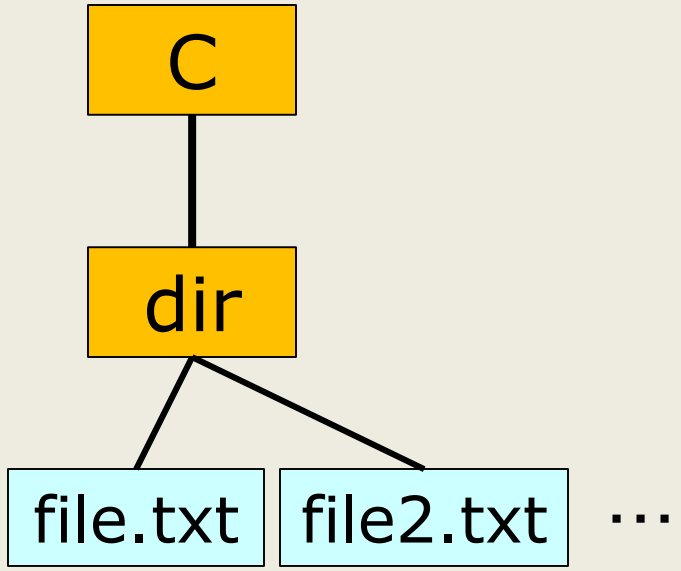
- 注意
  - ファイルもディレクトリもFileクラスで表現する

Fileクラスのコンストラクタ

- ファイルの絶対パスが下記の場合

C:¥dir¥file.txt

コンストラクタ File file1 = new ・・・	file1の絶対パス
File(“C:¥¥dir”)	C:¥dir
File(“C:¥¥dir¥¥file.txt”)	C:¥dir¥file.txt
File(“C:¥¥dir”, “file.txt”)	C:¥dir¥file.txt



ファイルもディレクトリもFileクラスで表現する

注意）絶対パスを指定する場合

- ファイルの絶対パスを指定して開く場合

```
File f2 = new File("C:¥¥tmp¥¥test.txt");
```

※パス内のスラッシュを2つ重ねる

2. ファイル情報の表示

2. ファイル情報の取得

- ファイル名 : String getName()
- 絶対パス : String getAbsolutePath()
- サイズ : long length()

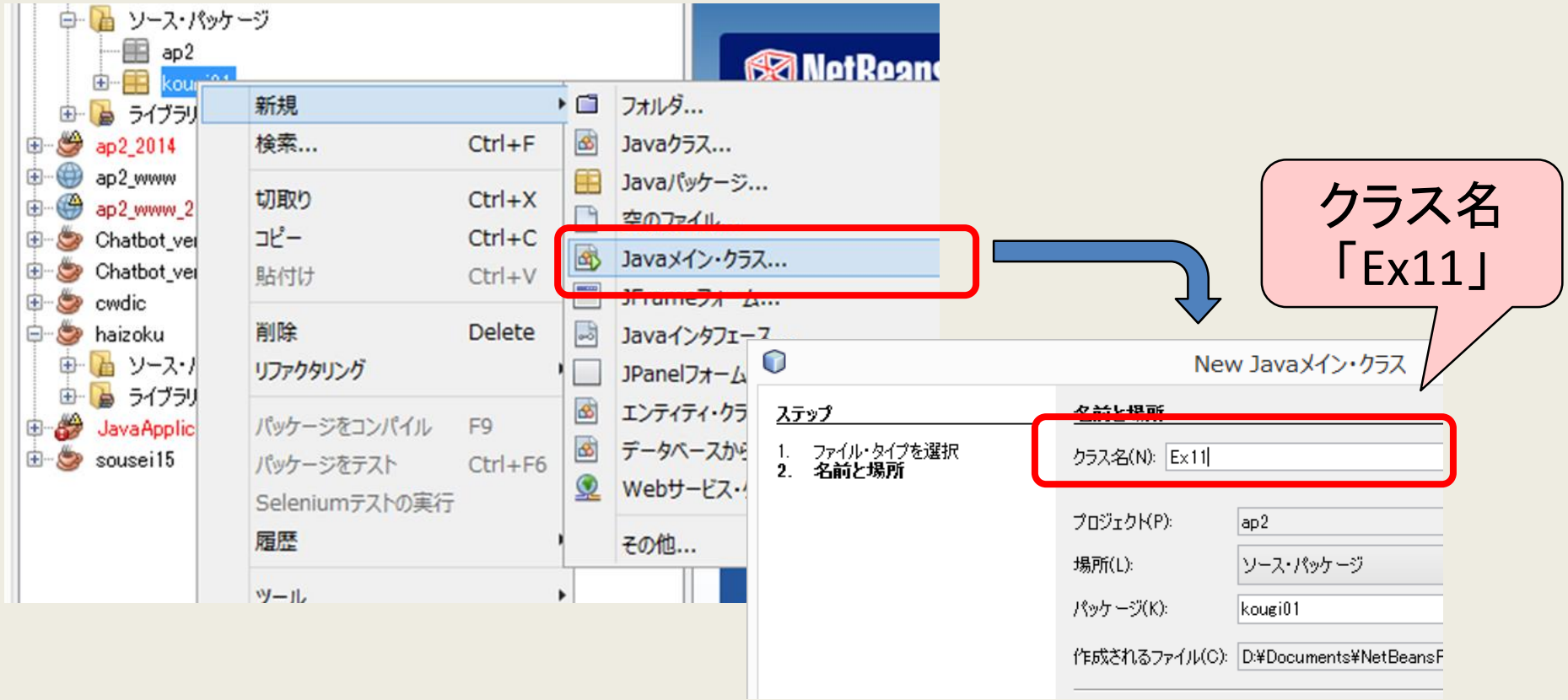
```
File f1 = new File("test.txt");
System.out.println("ファイル名：" + f1.getName());
System.out.println("絶対パス：" + f1.getAbsolutePath());
System.out.println("サイズ：" + f1.length() + "バイト");
```

練習問題【Ex11.java】

- テキストファイル（”test.txt”）とディレクトリ（”C:¥Program Files”）をFileクラスのインスタンスとして指定し，その情報を出力するプログラムを作成してみよう
- FileクラスのJavadocを参照し，他のメソッドも試してみよう

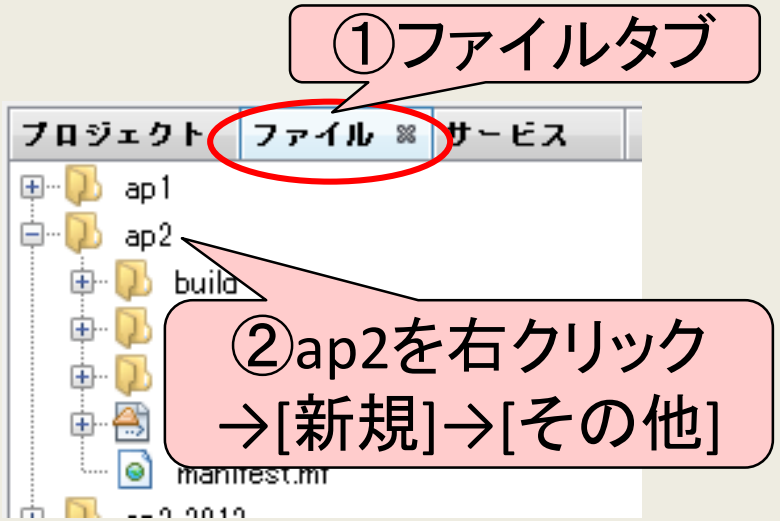
【Ex11】Javaメインクラスを作成

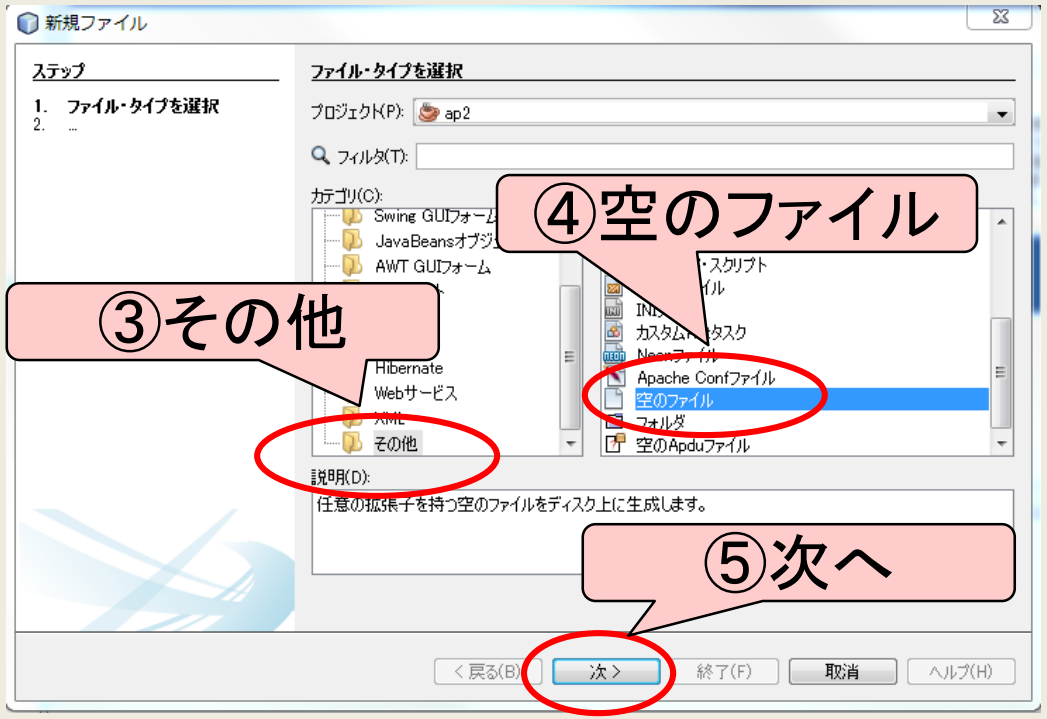
- 「kougi01」パッケージを右クリック⇒「新規」⇒「Javaメイン・クラス」（無ければ「その他」⇒カテゴリ「Java」内にある）



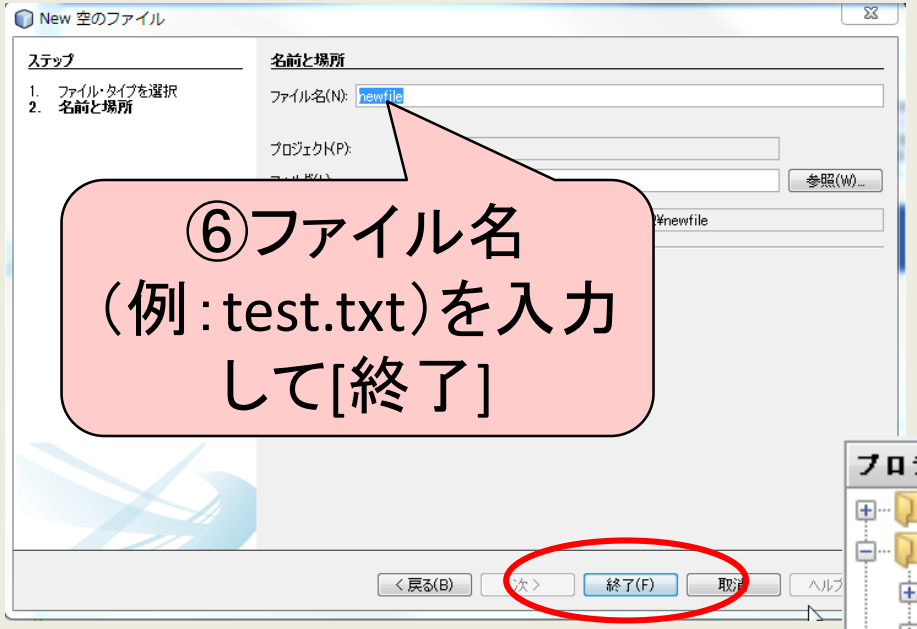
【test.txtの作成】NetBeansでのテキストファイルの作成(1)

- ファイルを相対パスで指定した場合，NetBeansProjects¥ap2 の下にあるファイルを参照する
- NetBeans上でテキストファイルを作成できる





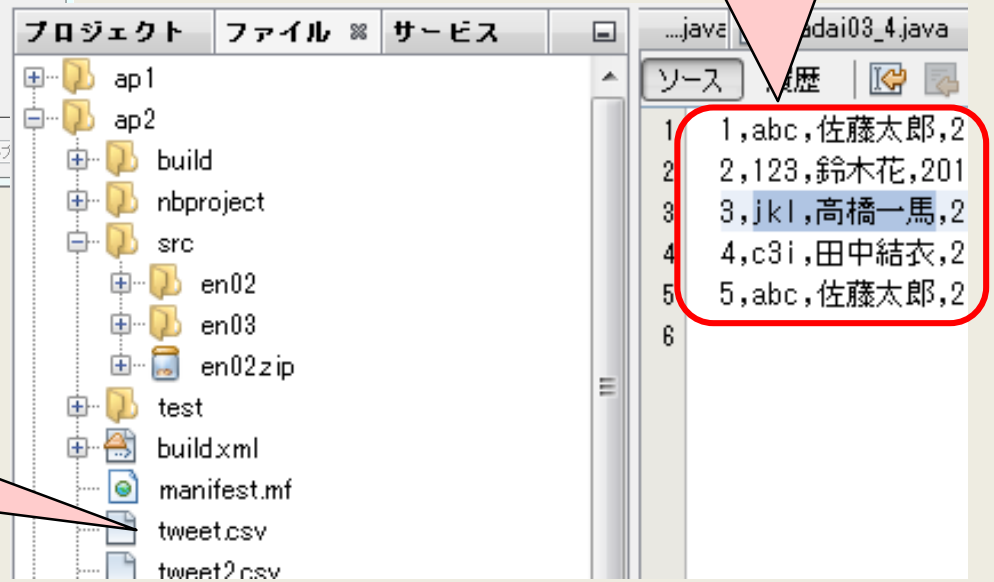
NetBeansでの
テキストファイルの作成(2)



⑥ファイル名（例：test.txt）を入力して[終了]

※作成したファイルの文字コードはUTF-8になる

⑧編集して保存



⑦作成されたファイルをダブルクリック

3. テキストファイルの読み書き

3. テキストファイルの読み書き

- 文字ストリームとバイトストリーム

	扱うデータ	入力	出力
文字ストリーム	文字	Reader	Writer
バイトストリーム	一般データ	InputStream	OutputStream

ファイルからの文字データ入力

- FileReaderとBufferedReaderによる方法

読み込みファイルの設定

```
File inFile = new File("test.txt”);
FileReader fr = new FileReader(inFile);
BufferedReader br = new BufferedReader(fr);
```

読み込みとBufferedReaderのクローズ

```
String line;
while ((line = br.readLine()) != null) {
    System.out.println(line);
}
br.close();
```

1行ずつ読み込み
lineに代入

1行で記述する方法（読み込み）

```
File inFile = new File("test.txt");
FileReader fr = new FileReader(inFile);
BufferedReader br = new BufferedReader(fr);
```

```
BufferedReader br
    = new BufferedReader(new FileReader(new File("test.txt"));
...

br.close();
```

※引数にしか使用しないインスタンスinFileやfrを省略できる

ファイルへの文字データ出力

- FileWriterとBufferedWriter, PrintWriterによる方法

書き込みファイルの設定

```
File outFile = new File("test.txt");
FileWriter fw = new FileWriter(outFile);
BufferedWriter bw = new BufferedWriter(fw);
PrintWriter pw = new PrintWriter(bw);
```

書き込みとPrintWriterのクローズ

```
pw.println("test");
pw.close();
```

1行で記述する方法（書き込み）

```
File outFile = new File("test.txt");
FileWriter fw = new FileWriter(outFile);
BufferedWriter bw = new BufferedWriter(fw);
PrintWriter pw = new PrintWriter(bw);
```

```
PrintWriter pw
    = new PrintWriter (new BufferedWriter (new FileWriter
      (new File("test.txt"))));

...
pw.close();
```

※引数にしか使用しないインスタンスoutFileやfw, bwを省略できる

ファイルのクローズ

- 読み込み，書き込みが終わったら，BufferedReaderやPrintWriterをクローズしなければならない
- BufferedReaderやPrintWriterをクローズすると関連するFileReaderやFileWriterなどのリソースも解放される

例外処理(1)

- オブジェクトの生成，write, read, closeなどのメソッドは全て例外処理が必要

```
public void write(String str)
            throws IOException

    文字列を書き込みます。

    パラメータ:
        str - 書き込まれる文字列
    例外:
        IOException - 入出力エラーが発生した場合
```

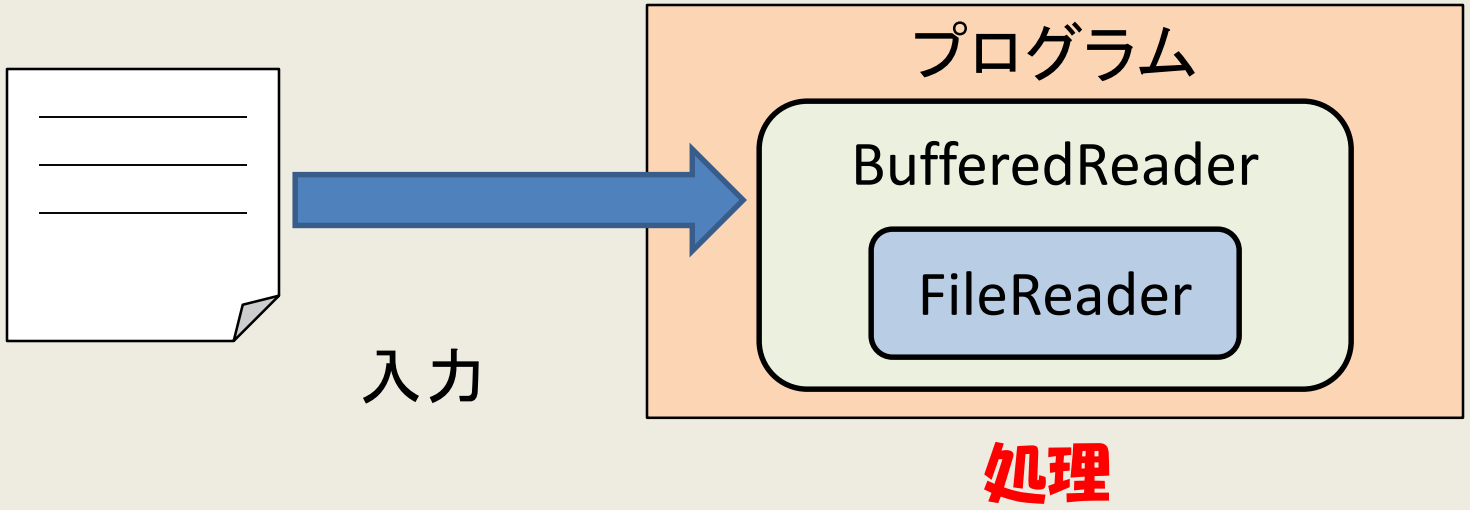
例外処理(2)

- try～catchで例外処理

```
try {
    BufferedReader br = new BufferedReader(
            new FileReader(new File("test.txt")));
    String line;
    while ((line = br.readLine()) != null) {
        System.out.println(line);
    }
    br.close();
} catch (IOException e) {
    System.out.println("読み込みに失敗");
}
```

FileReader と BufferedReader
FileWriter と BufferedWriter

- JavadocでBufferedReader, BufferedWriterについて調べてみる
- 何故，FileReader, FileWriterだけでなくBufferedReader, BufferedWriterと組み合わせて使うのか？
  ※プロ基礎2の内容をもう一度確認しよう



練習問題【Ex12.java】

- 先ほど作成したテキストファイル(“test.txt”)を読み込み，
  その内容を別のファイル(“out.txt”)に書き込むプログラムを
  作成しよう

4. GUIによるファイル選択

4. GUIによるファイル選択

ボタンを押すとファイル選択画面が開く

練習問題【Ex13.java】

- ファイルチューザーで選択したファイルの情報を表示するプログラムを作成しよう
- ファイルの拡張子を出力しよう

filename.txt

拡張子

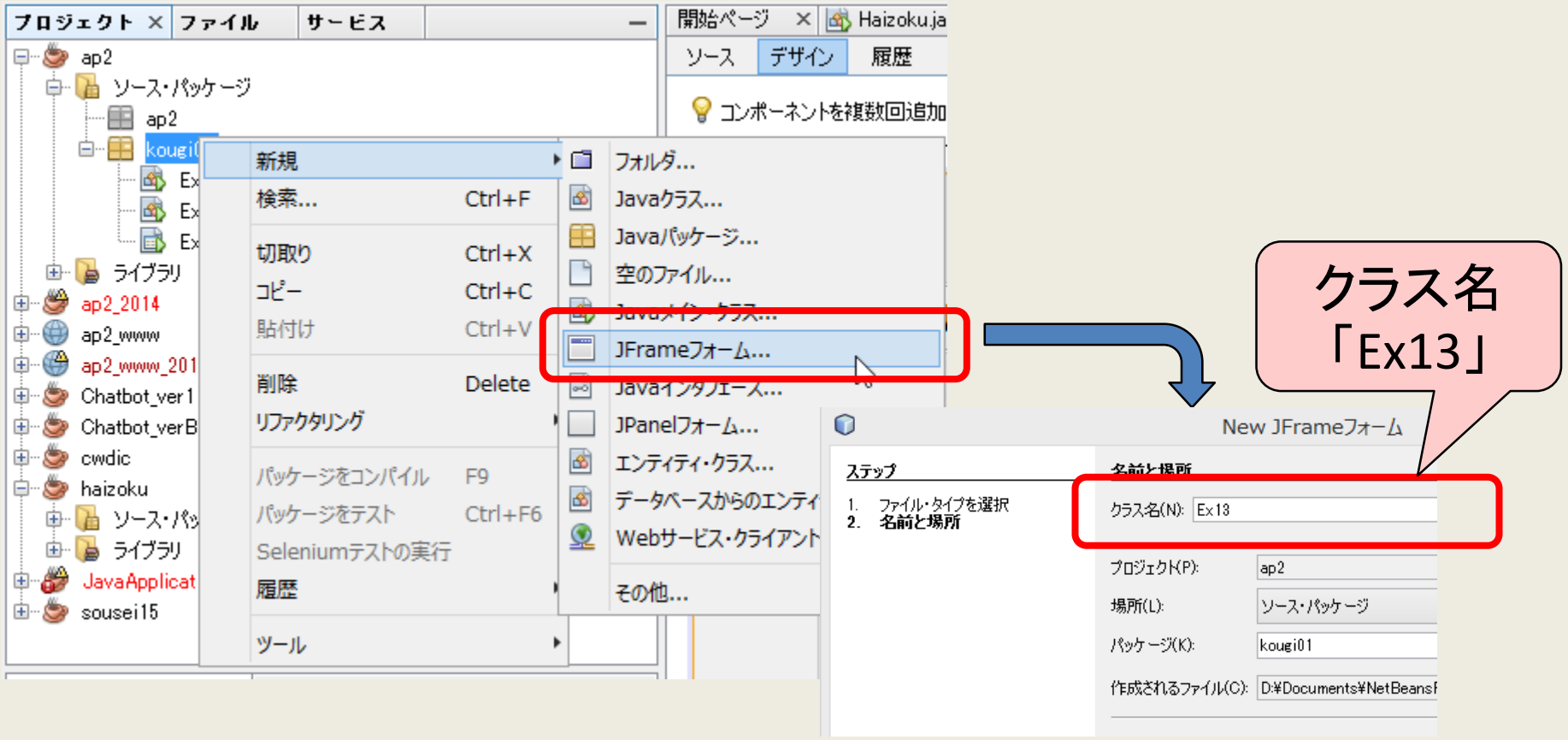
※ ファイル名の文字列に対してStringクラスのメソッドを適用

解答例）

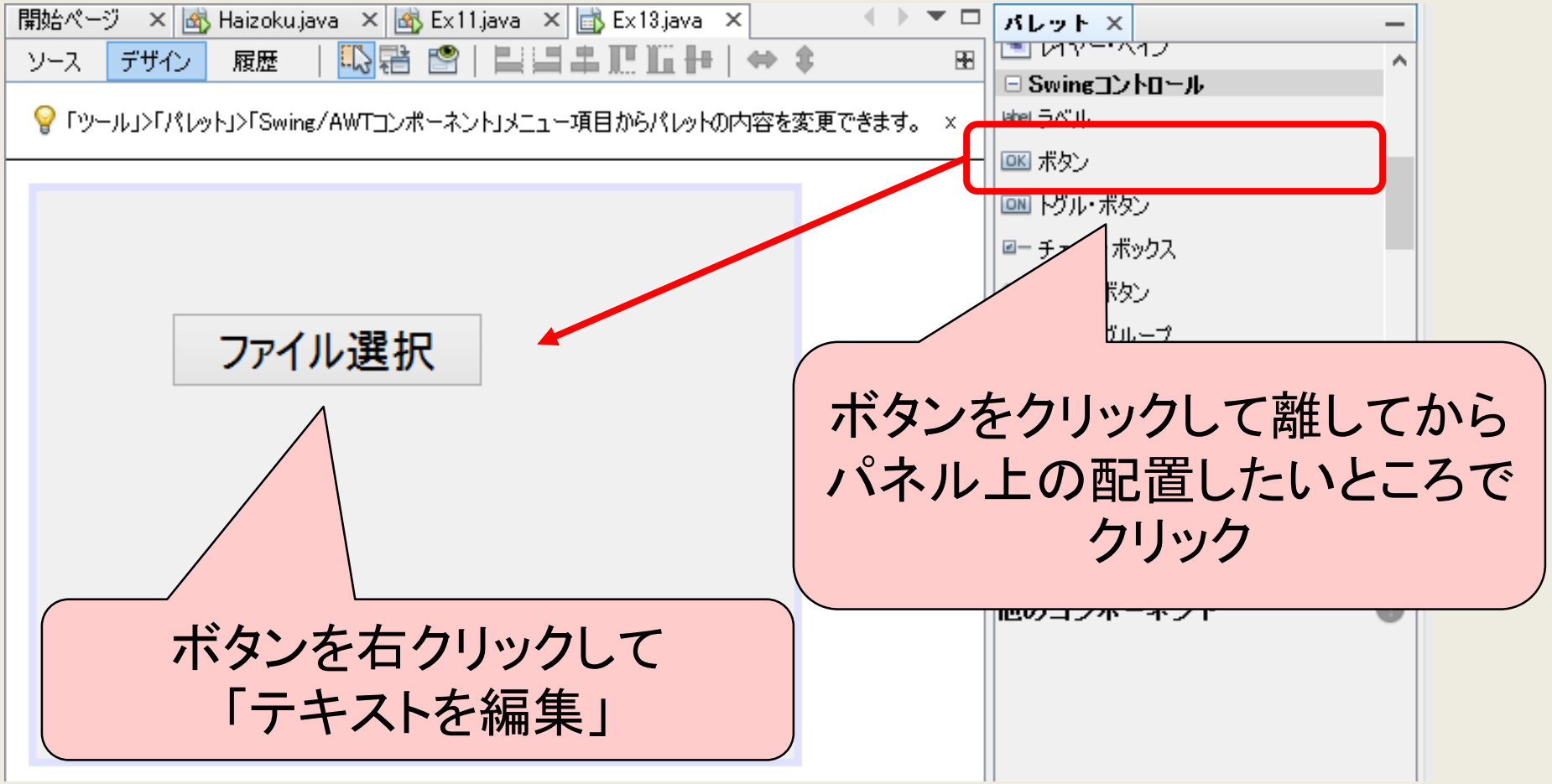
```
String fileName = f1.getName();
String ext = fileName.substring(fileName.lastIndexOf(".")+1);
```

【Ex13】JFrame上にボタンを作成(1)

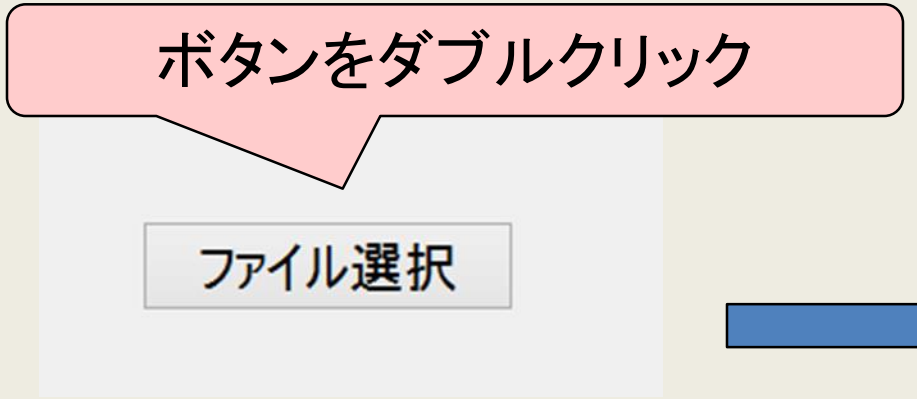
- 「kougi01」パッケージを右クリック⇒「新規」⇒「JFrameフォーム」
  （無ければ「その他」⇒カテゴリ「Swing GUIフォーム」内にある）



JFrame上にボタンを作成(2)



ボタンをクリックしたときのアクションを追加




```
65      private void jButton1ActionPerformed(java.awt.event.ActionEvent evt) {
66          // TODO add your handling code here:
67
68      }
69
```

ボタンをクリックしたときの処理をここに記述

ファイルチューザー(1)

- ダイアログを表示してファイルを選択
- 使用するクラス：javax.swing.JFileChooser

コンストラクタ

```
JFileChooser()

//開くディレクトリを指定
JFileChooser(File directory)
JFileChooser(String directoryPath)
```

ファイルチューザー(2)

```
private void jButton1ActionPerformed(java.awt.event.ActionEvent evt) {
    //ファイルチューザー作成
    JFileChooser fc = new JFileChooser();

    //ファイルチューザーの表示
    int selected = fc.showOpenDialog(this);

    //ファイルが選ばれたら，そのファイルをFileとして設定
    if (selected == JFileChooser.APPROVE_OPTION) {
        File f1 = fc.getSelectedFile();

        String fileName = f1.getName();
        System.out.println("ファイル名：" + fileName);
        System.out.println("絶対パス：" + f1.getAbsolutePath());
        System.out.println("サイズ：" + f1.length() + "バイト");
    }
}
```

ファイルチューザー(3)

- showOpenDialogメソッドの戻り値
  → ダイアログで押されたボタンの種類
  - JFileChooser.APPROVE_OPTION：0
    「開く」が押された時
  - JFileChooser.CANCEL_OPTION：1
    「取り消し」が押された時
    ウィンドウの×が押された時
  - JFileChooser.ERROR_OPTION：-1
    エラーが起きた時

その他

バイナリファイル

- コンピュータ内部で扱われる形式のままにデータを扱う
- 出力には

  FileOutputStream と BufferedOutputStream

  入力には

  FileInputStream と BufferedInputStream

  を用いる

※今回は説明しない