

並列プログラミング Parallel Programming

2018 2Q

演習 第1回

情報理工学院 情報工学系

本日の流れ

- 課題内容の説明
- 演習に取り組む
- 付録 統合開発環境について

演習課題概要

● 目的

- Java の復習
- イベント駆動プログラムを作成する
- Java の開発ツールに慣れる

● 題材

- お絵描きプログラム
- 電卓プログラム

課題のダウンロード

次のどちらかからダウンロードしてください

学内アクセス

- **OCW-i**

- **www.img.cs.titech.ac.jp/lecture/para/**

準備 (1)

- ダウンロードした `para1.tar.gz` を展開する

```
tar zxvf para1.tar.gz
```

```
tar xvf para1.tar
```

← ブラウザが勝手に一段階
解凍した場合

- 解凍後のディレクトリ

Para1(トップディレクトリ)

— Makefile (後述)

— README 説明文

bin

クラスファイル (*.class) が格納される (解凍直後は空)

javadoc

ドキュメントが格納される (解凍直後は空)

src

ソースファイル

lib

実行に必要なjavaのライブラリ集(jarファイル)

準備 (2)

- ソースファイルを `javac` コマンドでコンパイルしてクラスファイルを作る
 - 今回の演習では **トップディレクトリ**で

Paint が他のクラスに依存する場合、順次コンパイルしてくれる

```
javac -d bin -encoding UTF-8 -sourcepath src  
-cp bin src/para/paint/Paint.java
```

実際は一行で書く

として下さい

<code>-d bin</code>	コンパイル後のクラスファイルをディレクトリ <code>bin</code> に置く
<code>-encoding UTF-8</code>	ソースファイルの文字コードが <code>UTF-8</code> であることを示す
<code>-sourcepath src</code>	ソースファイルがディレクトリ <code>src</code> 以下にあることをコンパイラに教える
<code>-classpath bin</code>	依存するクラスやライブラリのありかをコンパイラに教える

※ `man javac` としてその他のオプションを確認すること

準備 (3)

- java コマンドでクラスファイルを実行する
 - 今回の演習ではトップディレクトリにて

```
java -cp bin:lib/* para.paint.Paint
```

パッケージ名

起点となるクラスの名前

として下さい（デモ用プログラムは para.paint.Paint と para.calc.Calculator があります）

-cp bin:lib/*

-cp は -classpathの短縮形

実行に必要なクラスファイルがディレクトリbin 以下に置かれていること、標準以外のjavaライブラリファイルがlib/に置かれているを java コマンドに教える

※ man java としてその他のオプションを確認すること

準備 (4)

- javadoc コマンドでソースファイルのコメント文から HTML のドキュメントファイルをつくる

```
package para.paint;
import javafx.application.Application;
...
/** JavaFXで作成するお絵描きプログラム. */
public class Paint extends Application
{
    /** 描画領域. */
    Canvas canvas;
```

外部の Javadoc
文書へのリンク

今回の演習では、Para1 直下で、

```
javadoc -sourcepath src -charset utf-8 -encoding UTF-8
        -link https://docs.oracle.com/javase/jp/10/docs/api
        -d javadoc -package para.paint para.calc
```

実際は一行で
書く

パッケージ名

としてください

HTML文書の出力先
ディレクトリ

※ man javadoc としてその他のオプションを確認すること

準備 (5)

- コマンドをいちいちタイプするのが面倒 ...
 - 今回は Makefile を用意したので make コマンドで `javac` , `javadoc` の実行が簡単に行える

<code>make Paint</code>	Paintについてコンパイルして、実行
<code>make Calculator</code>	Calculatorについてコンパイルして、実行
<code>make clean</code>	bin 以下のクラスファイルをすべて削除
<code>make javadoc</code>	javadoc コマンドを実行

上を実行すると実際に発行されたコマンドが表示される

Makefile を自分好みに変更してよいです

Makefileの記述ではタブは意味があります。スペースで置き換えると、makeが正しく解釈できません。
Makefileの書き方は各自調べて下さい

課題 1

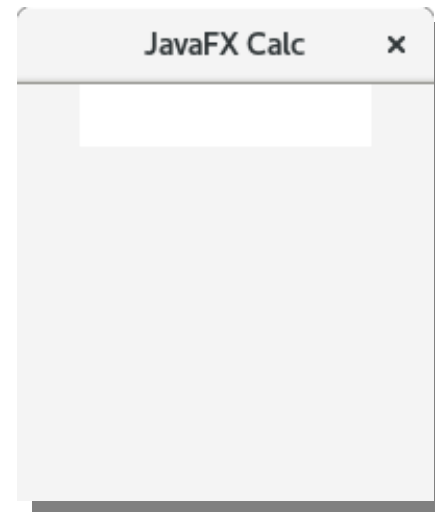
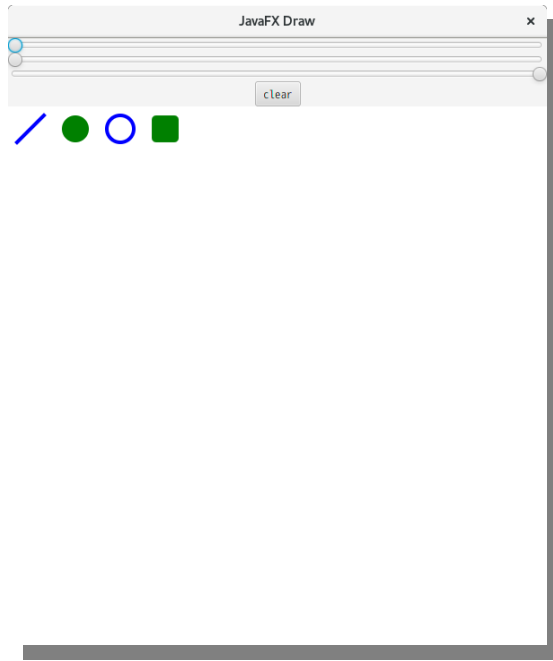
- Javadoc を読んでみる



- 1.1) <file:///usr/local/parallel-prog/javase10/docs/api/> をブックマークせよ
- 1.2) `Paint.java` の javadoc を開き、そこから `Button` クラスの継承関係を調べよ
- 1.3) `setOnAction` メソッドはどのクラスで定義されているか調べよ
- 1.4) `setOnAction` に与える引数からたどり、`handle` メソッドが呼ばれる時の引数のクラスにはどのような情報が含まれているか調べよ

課題 2

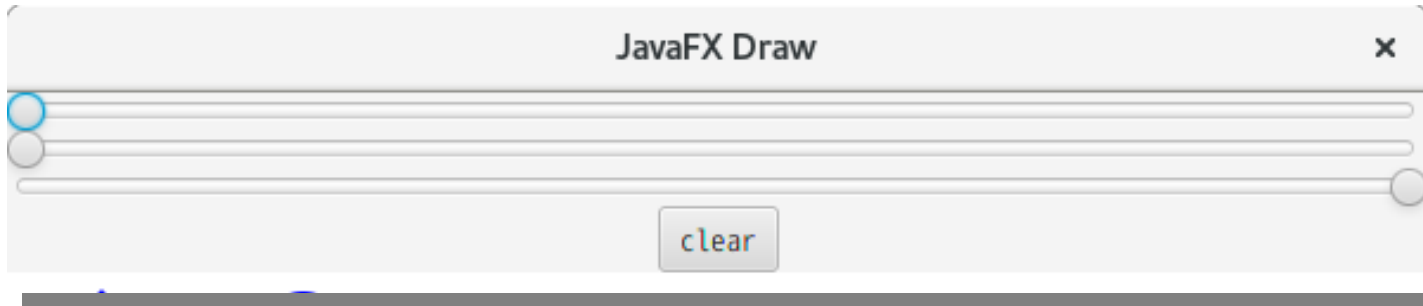
- 説明されたコンパイルの仕方に従って、`para.paint.Paint` と `para.calc.Calculator` をコンパイルして実行する



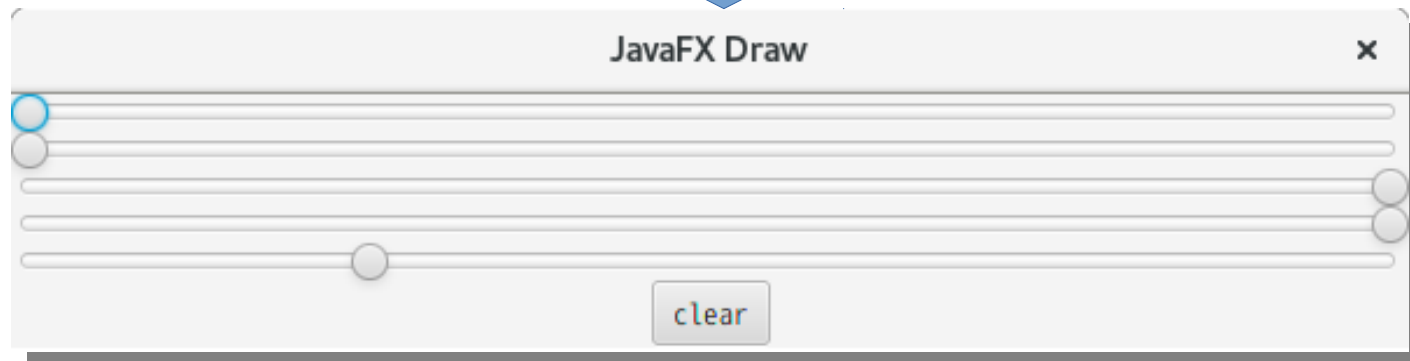
- 2.1) 作成されたclass ファイルが置かれたディレクトリ位置を、Para1からの相対パスで説明せよ。
- 2.2) ソースコード内のpackageで指定されたパッケージ名と、javac と java に与えたオプションと、2.1で答えたディレクトリ位置との関係を説明せよ。
- 2.3) ソースコード内のimport文で指定されたクラスが、正しくimportできるためには、そのクラスはどこにある必要があるかを説明せよ。

課題 3

- para.paint.Paint を改造せよ



3.1) 描画線の色に上から赤, 緑, 青の値を設定するスライダーとして機能するようにイベントハンドラを登録せよ

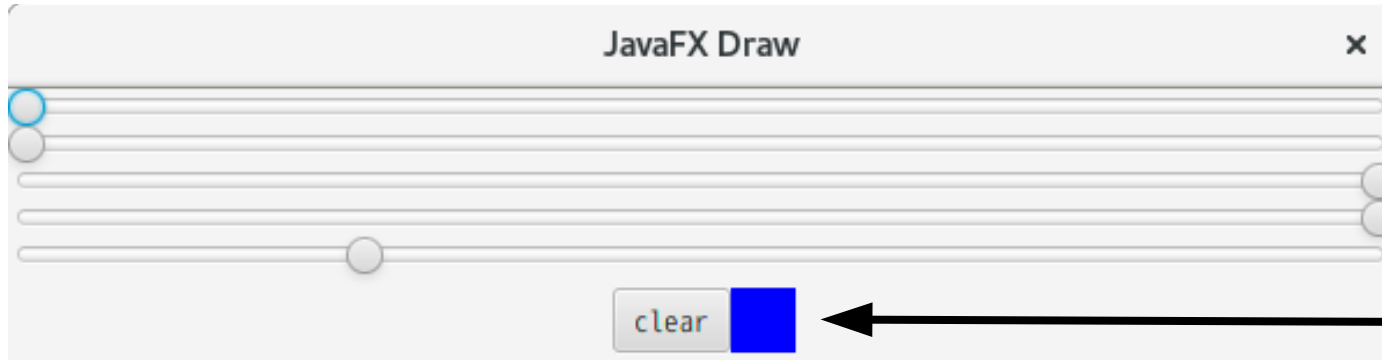


3.2) スライダーを2つ追加し、追加の1つ目は描画線の透明度を設定するスライダー、追加の2つ目は線幅を設定するスライダーとして機能するようにイベントハンドラを登録せよ

hint `javafx.scene.paint.Color`
hint `javafx.scene.canvas.GraphicsContext`

課題 3

- para.paint.Paint を改造せよ



3.3)現在設定されている描画色を表示する領域をclearボタンの右横に用意せよ



3.4) 機能拡張を”一つ以上”せよ

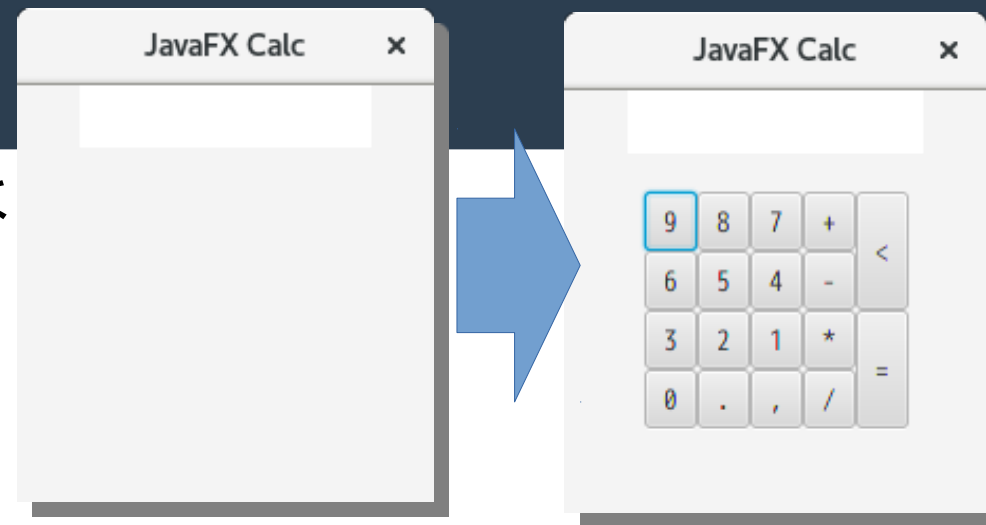
- 例えば、スタンプ機能の追加
- 例えば、画像の90度回転をさせるボタンの追加
- 例えば、一つ前の描画操作の取り消し機能の追加
- 例えば、画像ファイルの書き出し昨日の追加
- 例えば、画像ファイル読み込み機能の追加
- 例えば、レイヤー機能の追加

課題 4

- `para.calc.Calculator` を完成させよ

小数も入力可能

“<” ボタンは文字列の末尾の1文字を削除する機能とする



4.1) `javafx.scene.layout.GridPane` を利用して ボタンを右上のように配置せよ

4.2) ボタンを連続して押すことで、数値と四則演算記号を「,」区切りの**逆ポーランド記法**で表記する文字列データを作成する機能を付け加えよ、また一文字入力される毎に、更新された文字列が、Label inputに表示されるようにすること

hint 文字列データの保持には `java.lang.StringBuilder` を使うと便利

hint イベントハンドラが受け取るEventクラスには、イベント発生場所の情報が含まれている

4.3) “=” ボタンを押すことで、4.2)で作成した文字列をStringクラスのインスタンスにして、`para.calc.Executor`の継承クラスの`operation()`に渡し、演算結果をLabelクラスの `output`を使って表示する一連の処理が行われるようにせよ

提出方法 (1 of 3)

- para1.tar.gz を展開したディレクトリ構造を保ったまま, 課題 3 ~ 4 の変更作業を行う
- 課題 3 ~ 4 で変更した自分が変更したファイルの先頭には自分の名前と学籍番号を書いておく
 - プログラムの場合はコメント内に書く
- 課題 1 ~ 4 の回答文、工夫点および感想を書いた report1.txt を用意する (雛形は課題のウェブページ)

7桁の数字

学籍番号(トップディレクトリ)

— Makefile

— (もしあれば機能拡張で必要なファイル)

— **src**

ソースファイル

— report1.txt

回答プログラムと
レポートの作成終
了後、次のようにフ
ァイルを配置したディ
レクトリを作成

次ページに具体的な作業手順あり

提出方法 (2 of 3)

- 提出用ディレクトリを作成する 学籍番号から7桁の数字にすること

```
mkdir dir
```

- ソースファイルのディレクトリのコピーを作る

```
cp -R トップディレクトリ/src dir
```

- dir* に report1.txt もコピーする 今回はPara1

```
cp トップディレクトリ/report1.txt dir
```

- dir* に課題 2 で必要なその他のデータがあればコピーする

```
cp トップディレクトリ/otherfiles dir
```

- 次のコマンドを実行する

```
zip ex1-1612345.zip dir
```

学籍番号に対応する7桁の数字にすること

dir 以下の内容が圧縮され、ex1-1612345.zip が作られます

- 圧縮後に内容を “unzip ex1-1612345.zip” で確認すると提出ミスを防げて安全

提出方法 (3 of 3)

- 作成した zip ファイルを ocw にアップロードする
- 締め切り
 - 6月22日(金) 10:00am (JST)

質問したいときは ...

- メールアドレス

pro3report [AT] img.cs.titech.ac.jp

※ [AT] は @ に置き換え

– 佐藤、齋藤、担当 TA へメールが送られます

- あるいは西 8 号館 E 棟 401 号室の佐藤まで
直接質問に来て下さい

– 4F エレベータホールのインターホンにて
呼び出し

付録

統合開発環境 Eclipse

Emacsによる編集でも課題を解くことはできますが、クラス間の依存関係を把握するために統合開発環境を利用すると便利なこともあります。

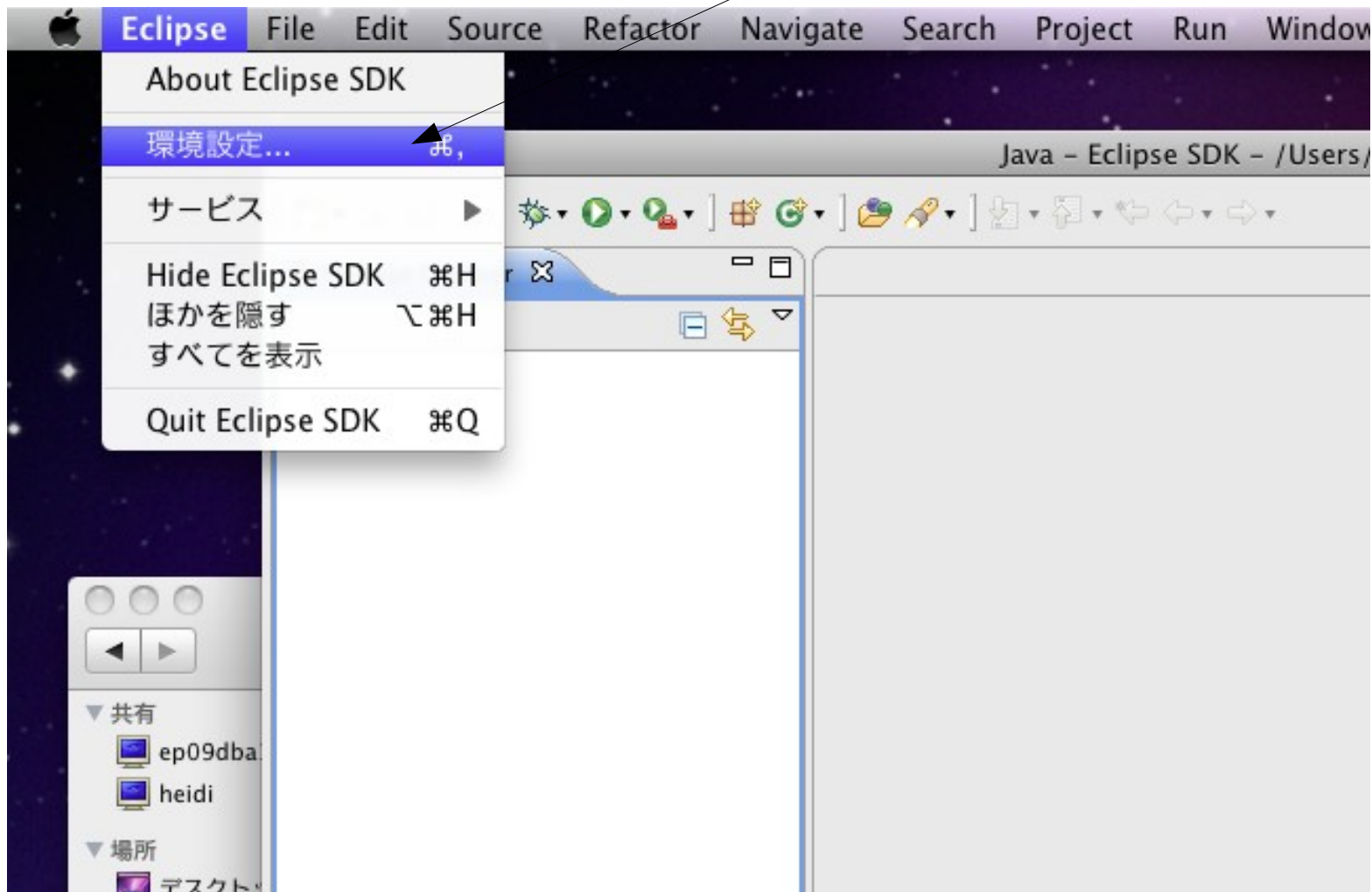
(Emacsでも機能拡張をすればとても強力に大規模なプロジェクトを把握出来ます)

利用する場合には、javacやjavaのオプションとEclipseのproject設定の関係を把握しておきましょう。

注意 Eclipse version Oxygen.3A は
初回起動時、設定ファイルがコピーされるので、
多人数で同時に作業するととても遅くなります

統合開発環境 Eclipse

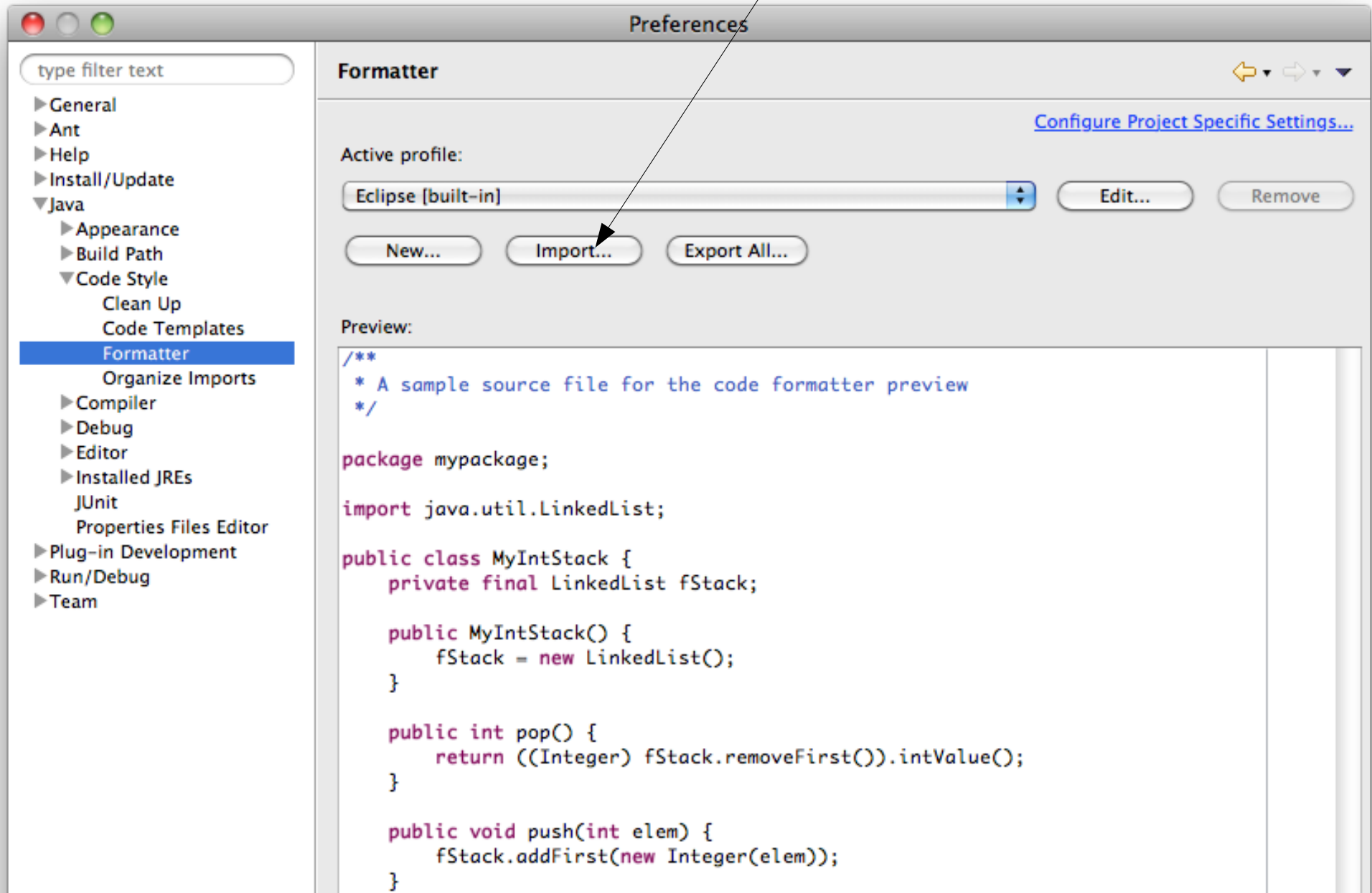
準備0 Eclipseの環境設定



統合開発環境 Eclipse

準備0 Eclipseの環境設定
コードスタイルの設定

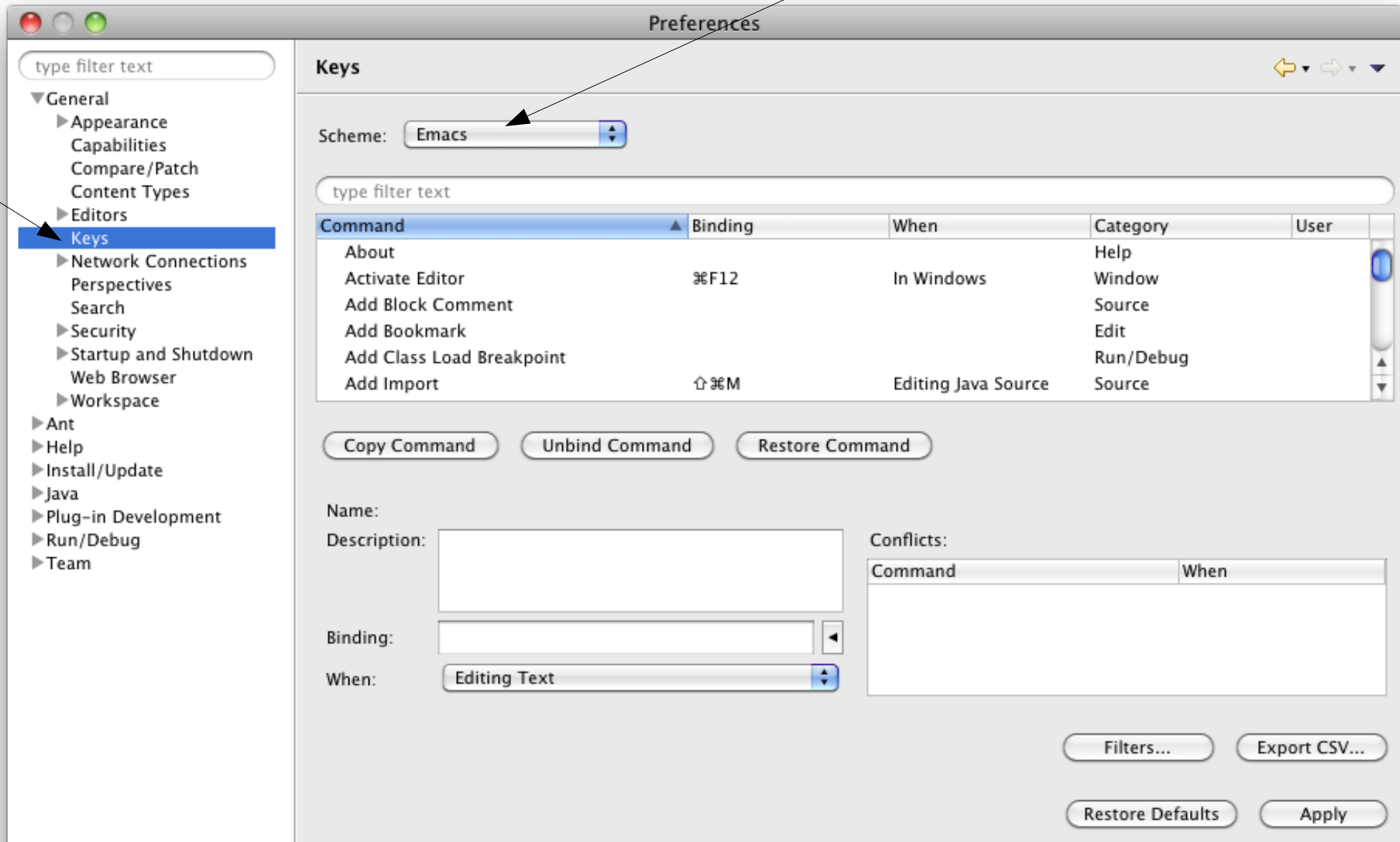
課題のWebpageからpro3styleEclipse.xmlを
ダウンロードして読み込む



統合開発環境 Eclipse

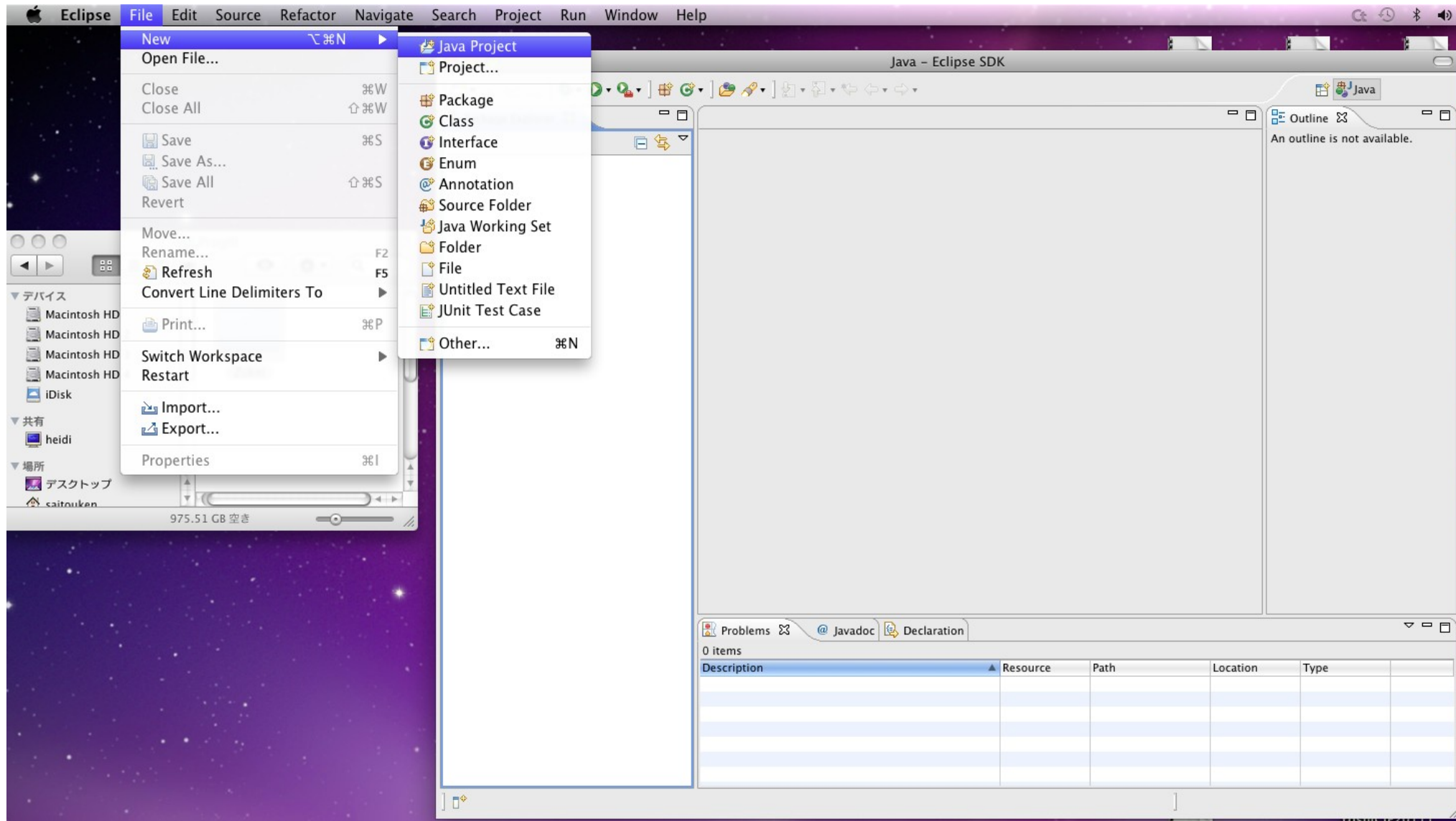
準備0 Eclipseの環境設定
キーバインドの設定

Emacsのスタイルを選択



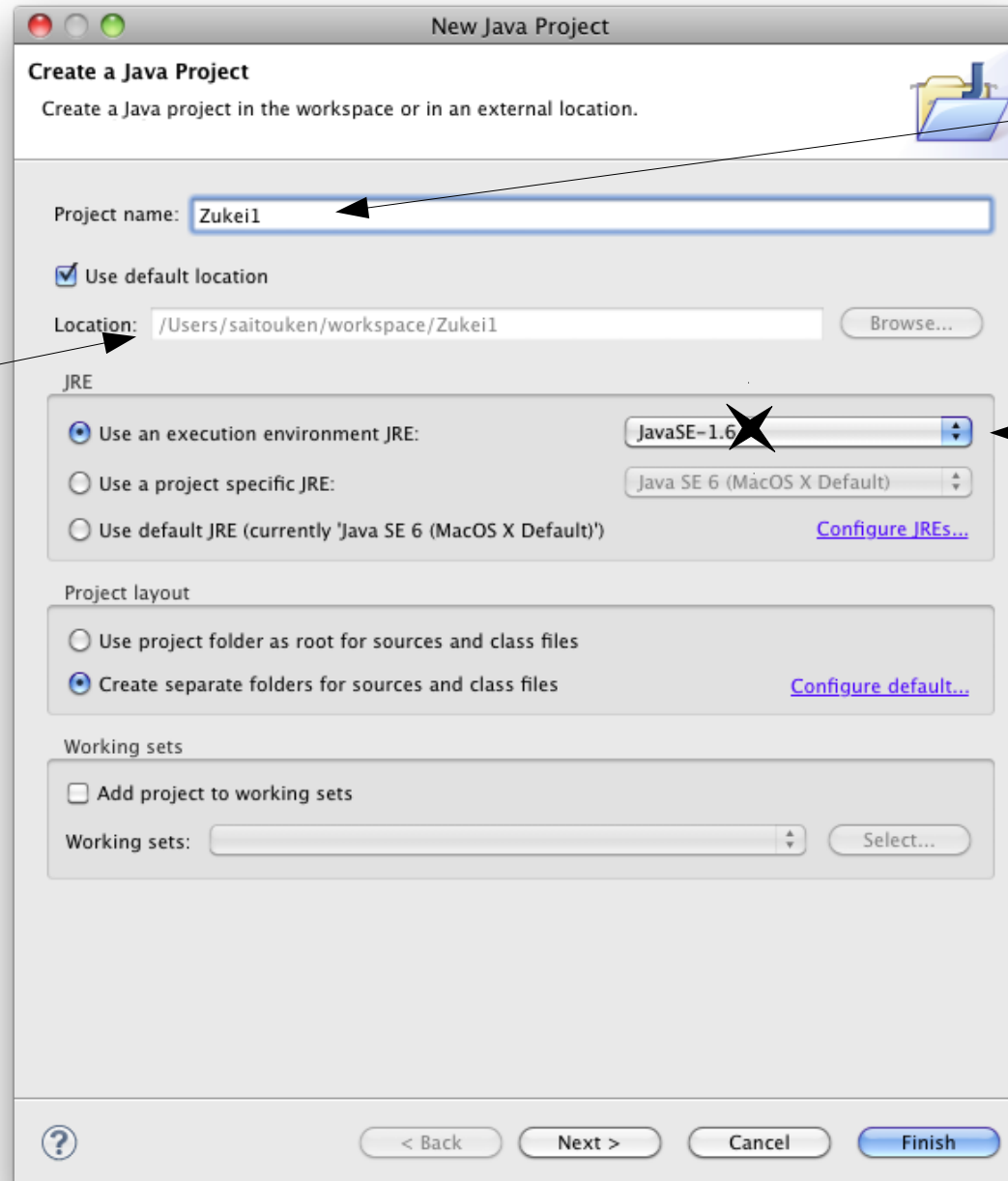
統合開発環境 Eclipse

準備その1 Java projectを作る



統合開発環境 Eclipse

準備その1 Java projectを作る



例えばZukei1 という名にする

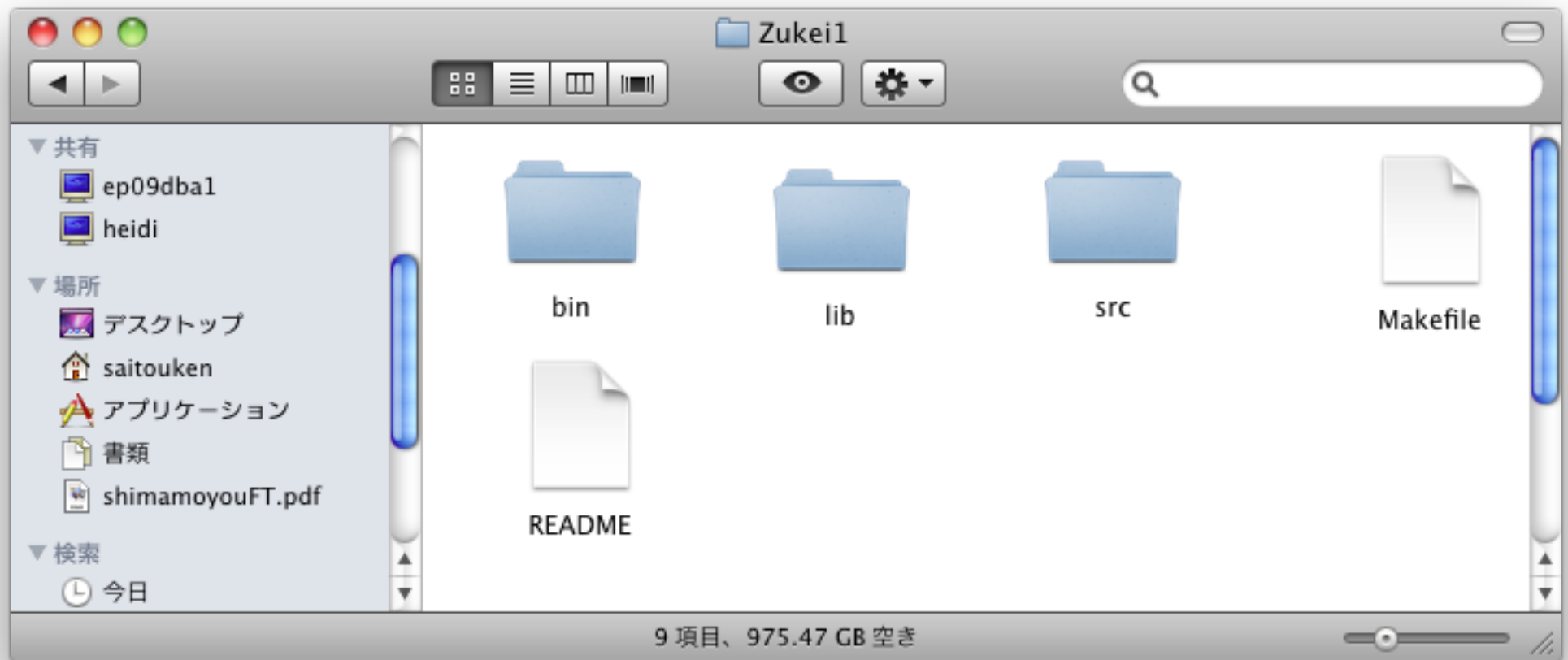
Javaのバージョンは10にする

workspace
ディレクトリの下に
プロジェクト
ディレクトリが
作られるので認識
しておく

統合開発環境 Eclipse

準備その2 Eclipseの作業領域にファイルをコピー

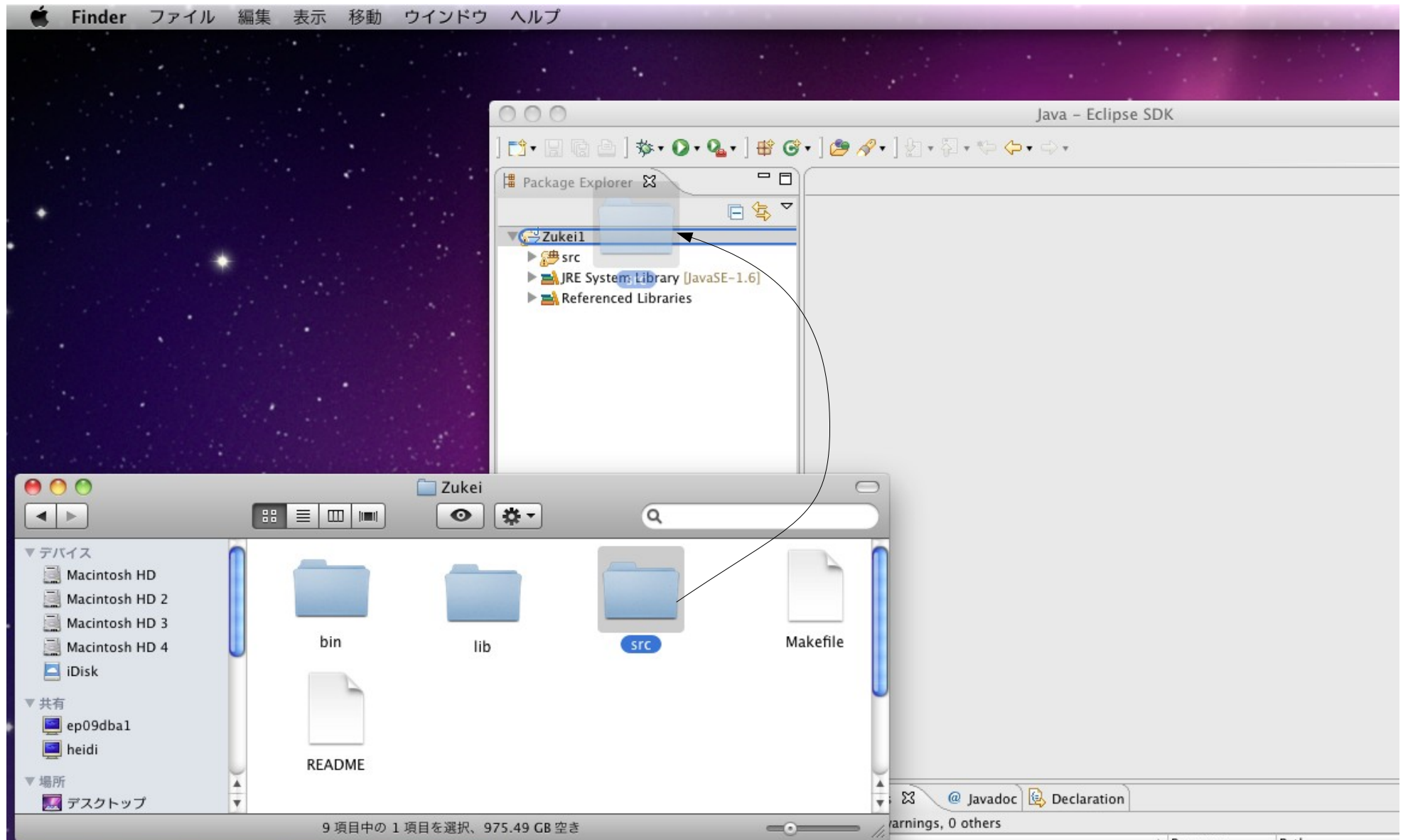
ダウンロードして展開したディレクトリZukei1をFinderで開く



統合開発環境 Eclipse

準備その2 Eclipseの作業領域にファイルをコピー

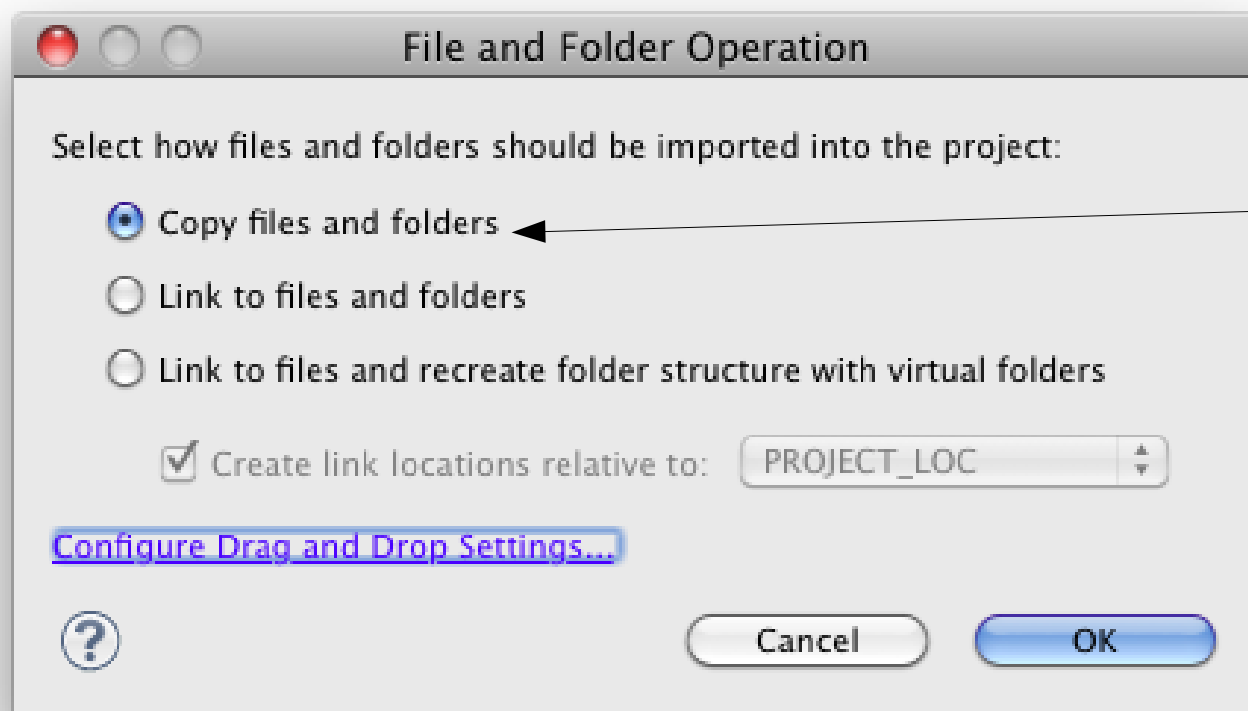
Finderの中のsrcアイコンをEclipseのプロジェクトZukei1のところに
ドラッグアンドドロップ



統合開発環境 Eclipse

準備その2 Eclipseの作業領域にファイルをコピー

Finderの中のsrcアイコンをEclipseのプロジェクトZukei1のところにドラッグアンドドロップ



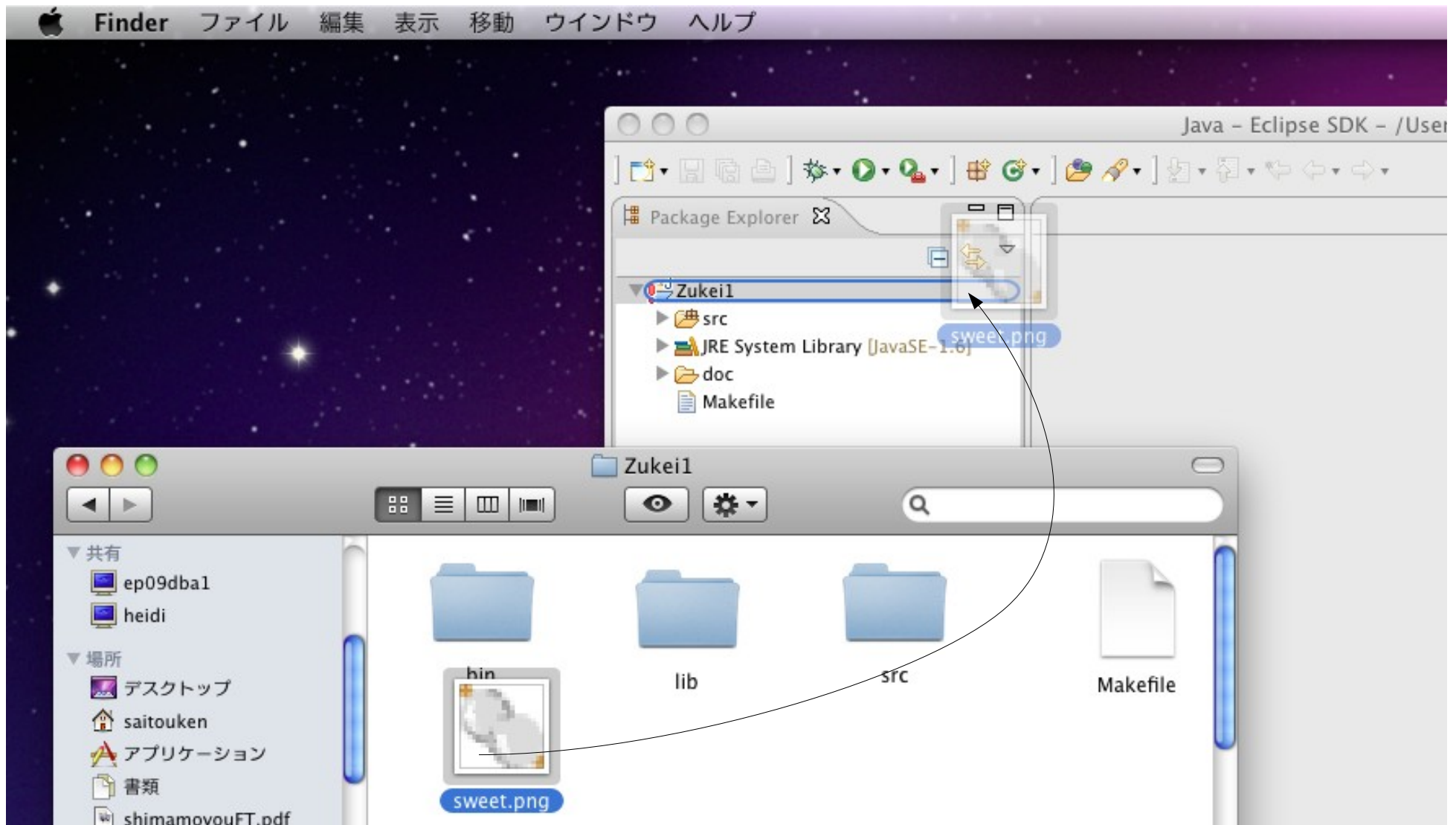
コピーするを選択

統合開発環境 Eclipse

準備その2 Eclipseの作業領域に関連するファイルをコピー

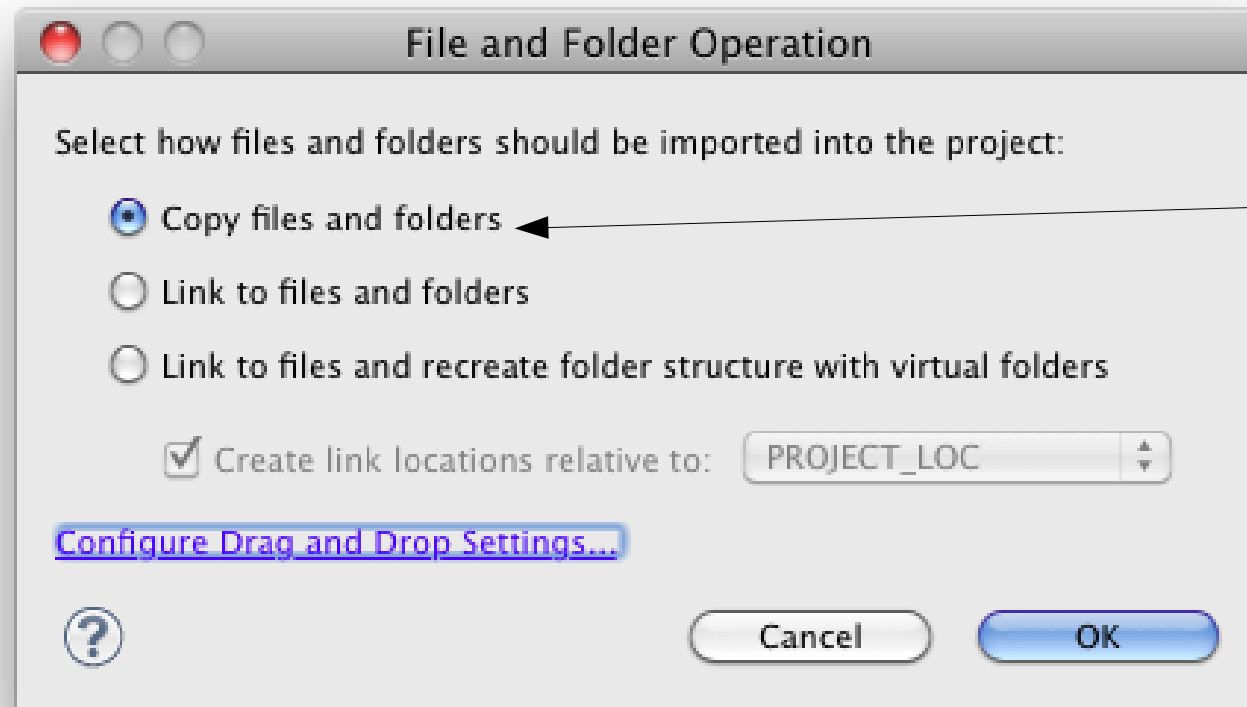
例えばFinderの中のsweet.pngアイコンをEclipseのプロジェクトZukei1のところにドラッグアンドドロップ

(sweet.pngが実行するときに読み込んで使う画像という想定)



統合開発環境 Eclipse

準備その2 Eclipseの作業領域にファイルをコピー



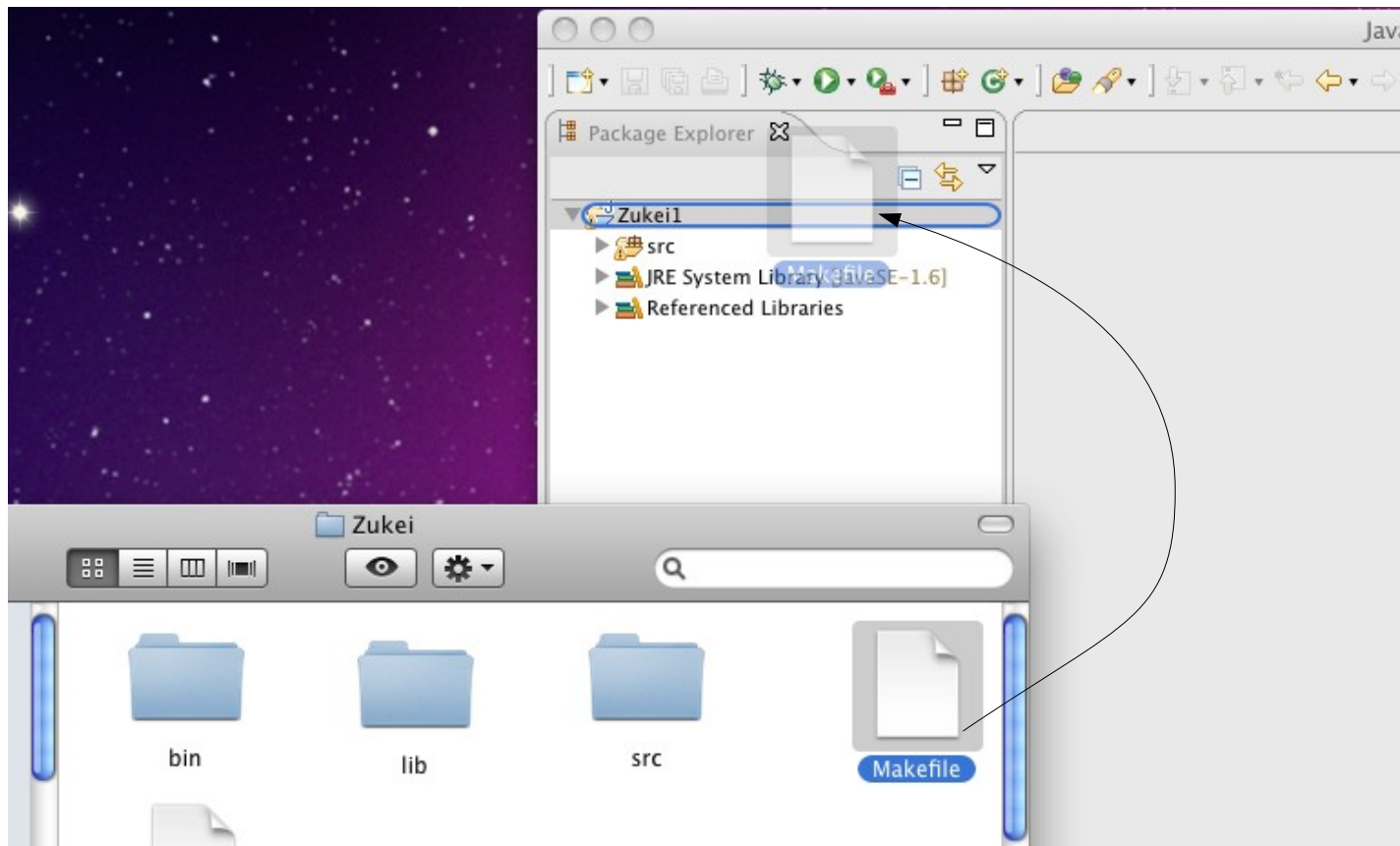
コピーするを選択

統合開発環境 Eclipse

準備その2 Eclipseの作業領域にファイルをコピー

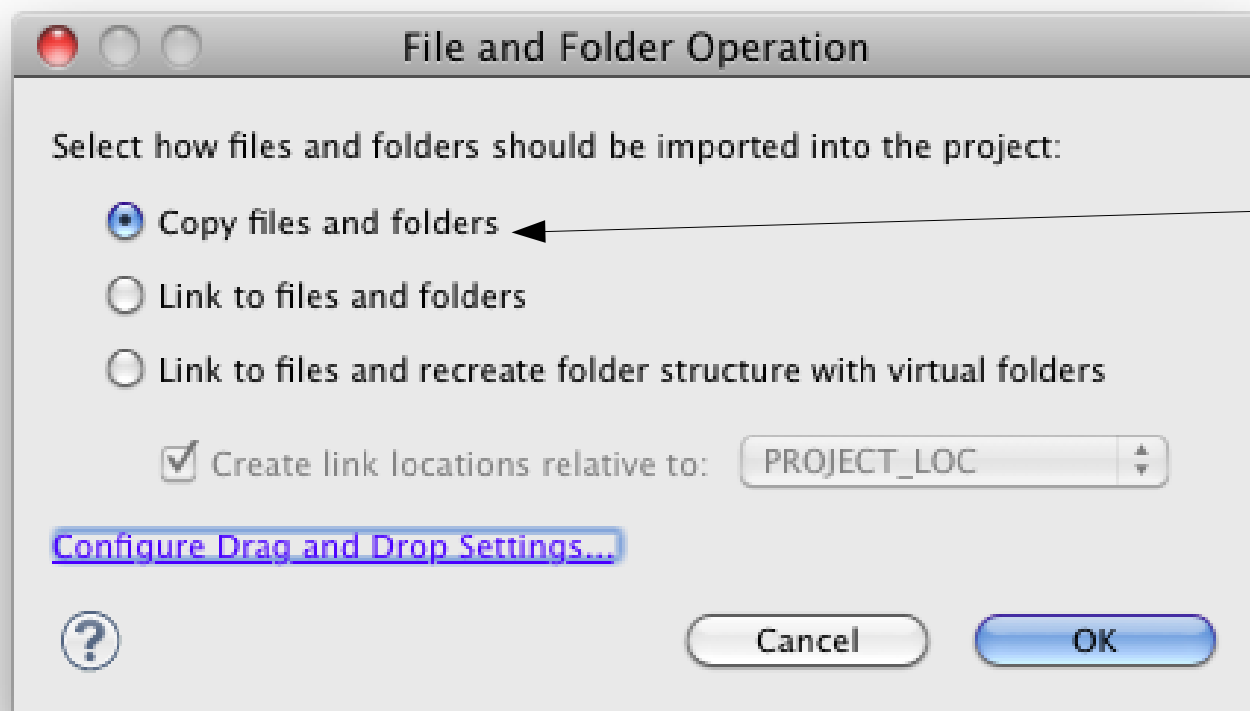
Finderの中のMakefileアイコンをEclipseのプロジェクトZukei1のところに
ドラッグアンドドロップ

MakefileはEclipse環境では不要ですが、workspace内でmakeしたい場合のために
コピー



統合開発環境 Eclipse

準備その2 Eclipseの作業領域にファイルをコピー



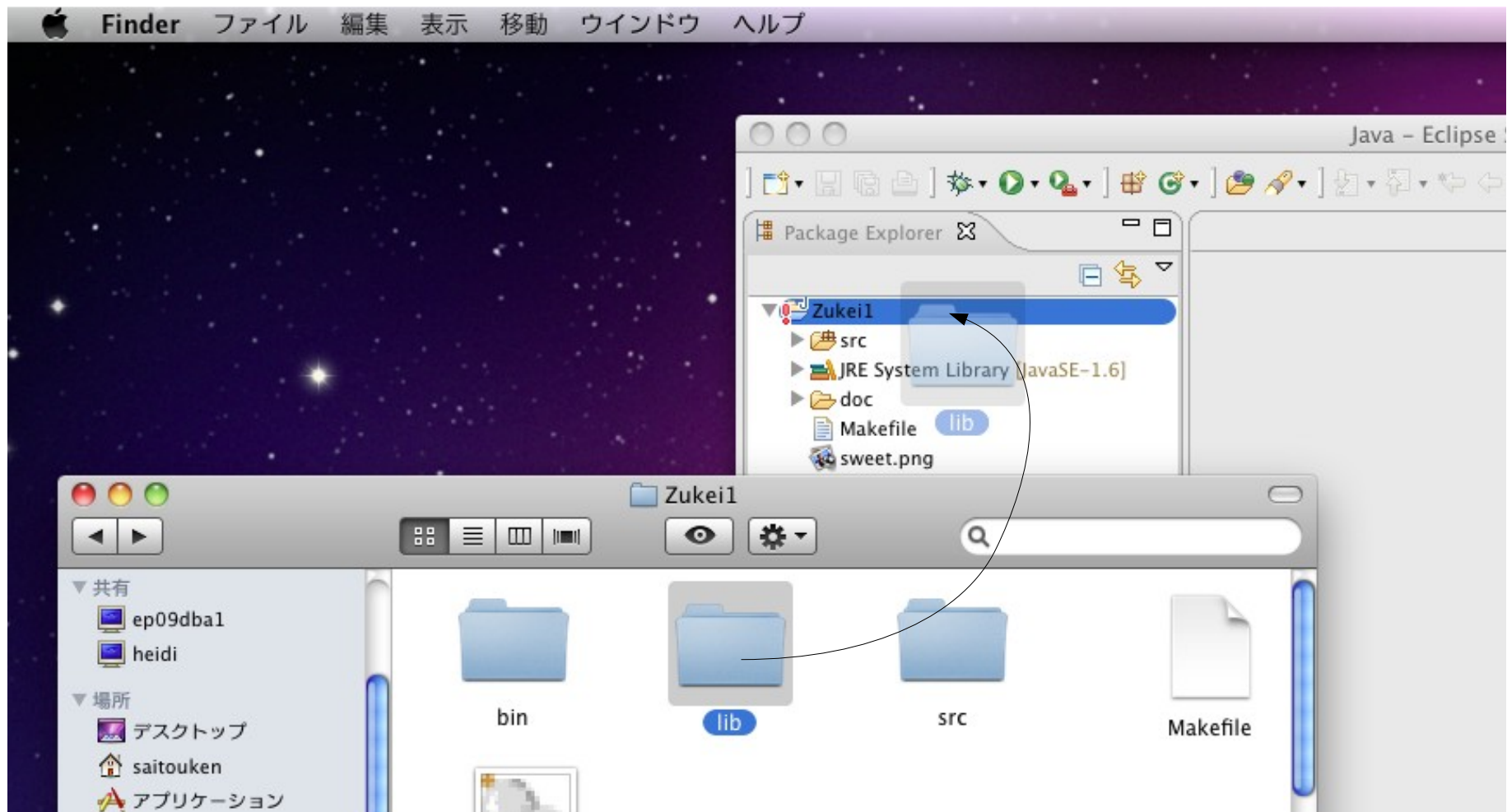
コピーするを選択

統合開発環境 Eclipse

準備その2 Eclipseの作業領域にファイルをコピー

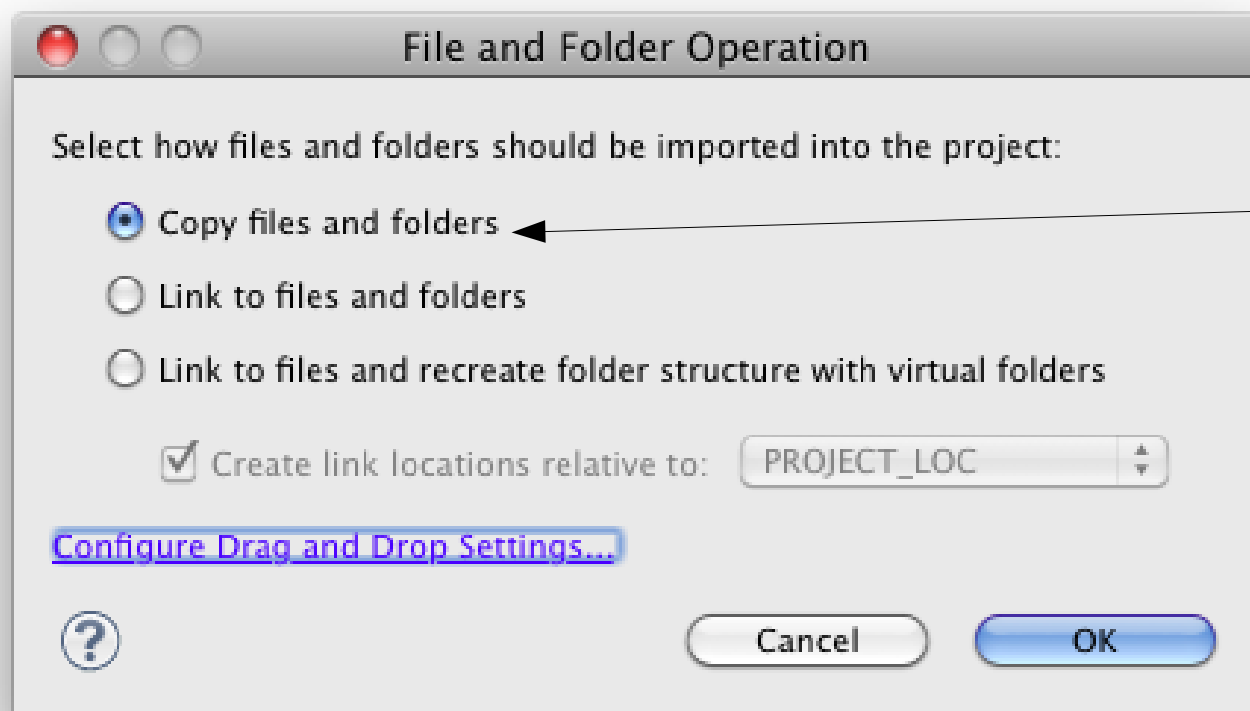
Finderの中のlibアイコンをEclipseのプロジェクトZukei1のところに
ドラッグアンドドロップ

libの中にはjavaのライブラリ jarファイルが入っています (今回の課題では空ですが)



統合開発環境 Eclipse

準備その2 Eclipseの作業領域にファイルをコピー

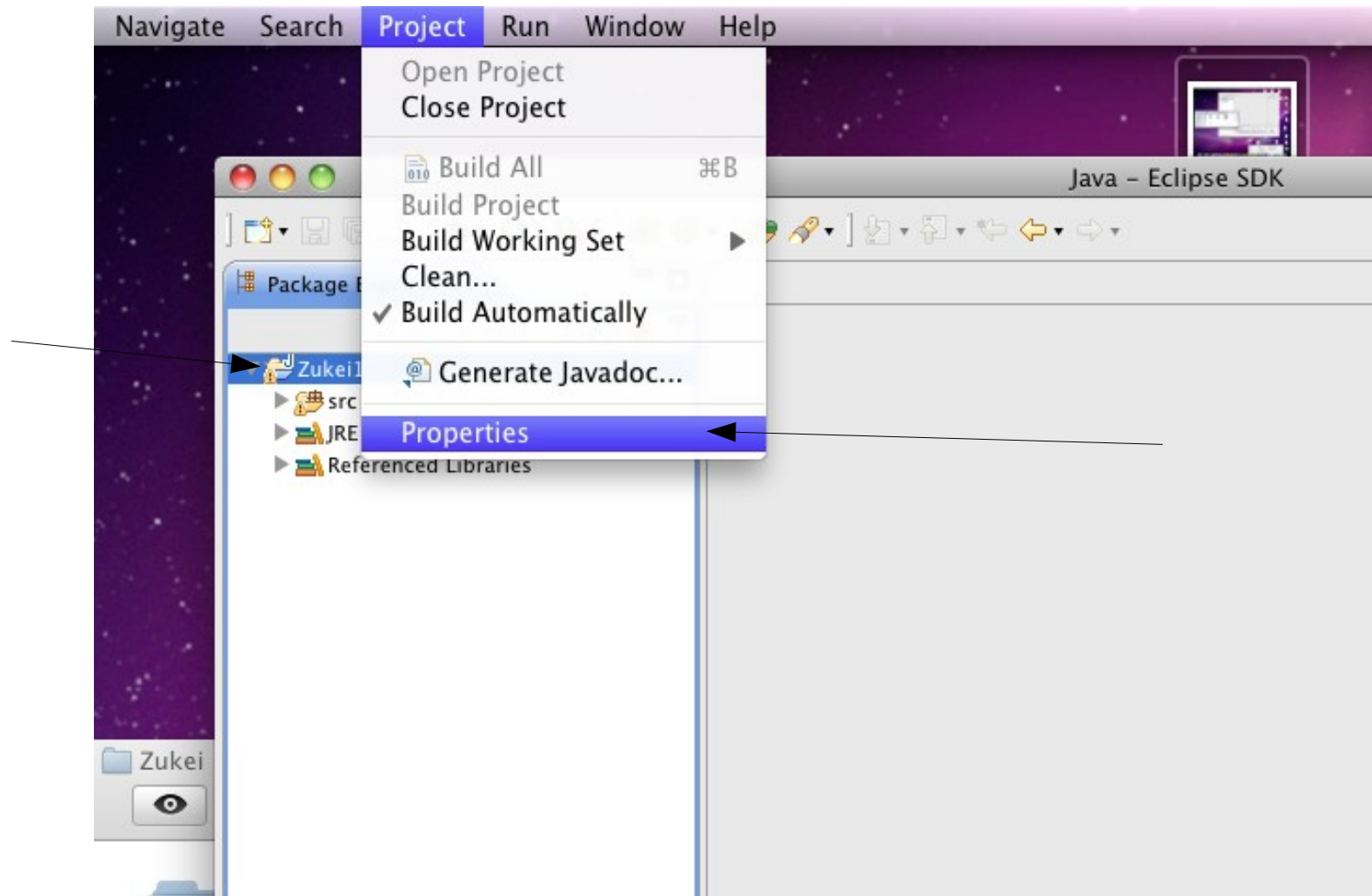


コピーするを選択

統合開発環境 Eclipse

準備その3 プロジェクトの属性を設定

このプロジェクトで使うjavacやjavaコマンドのオプションを設定

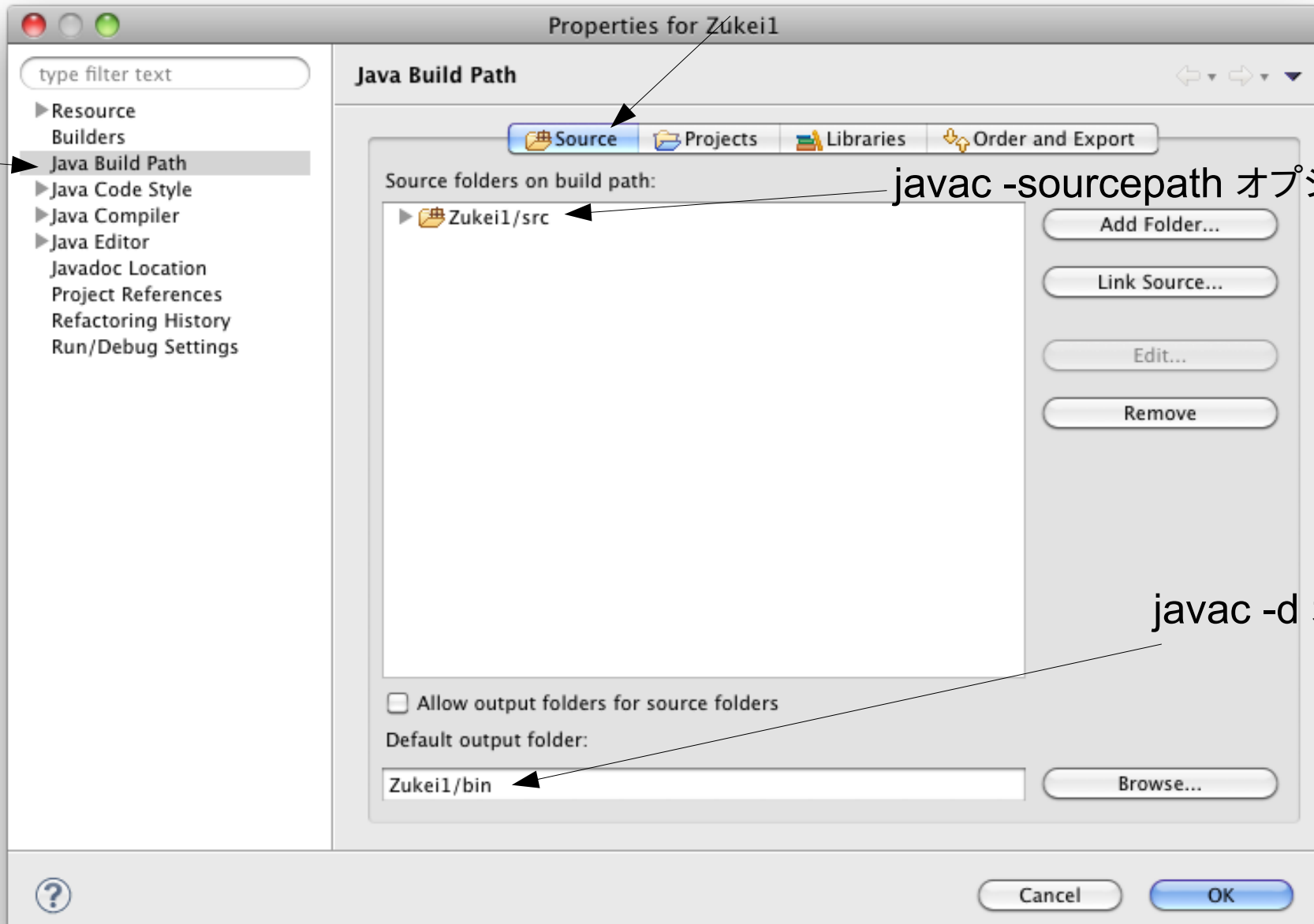


統合開発環境 Eclipse

準備その3 プロジェクトの属性を設定

このプロジェクトで使うjavacやjavaコマンドのオプションを設定

自動で設定されるはずなので確認するだけ



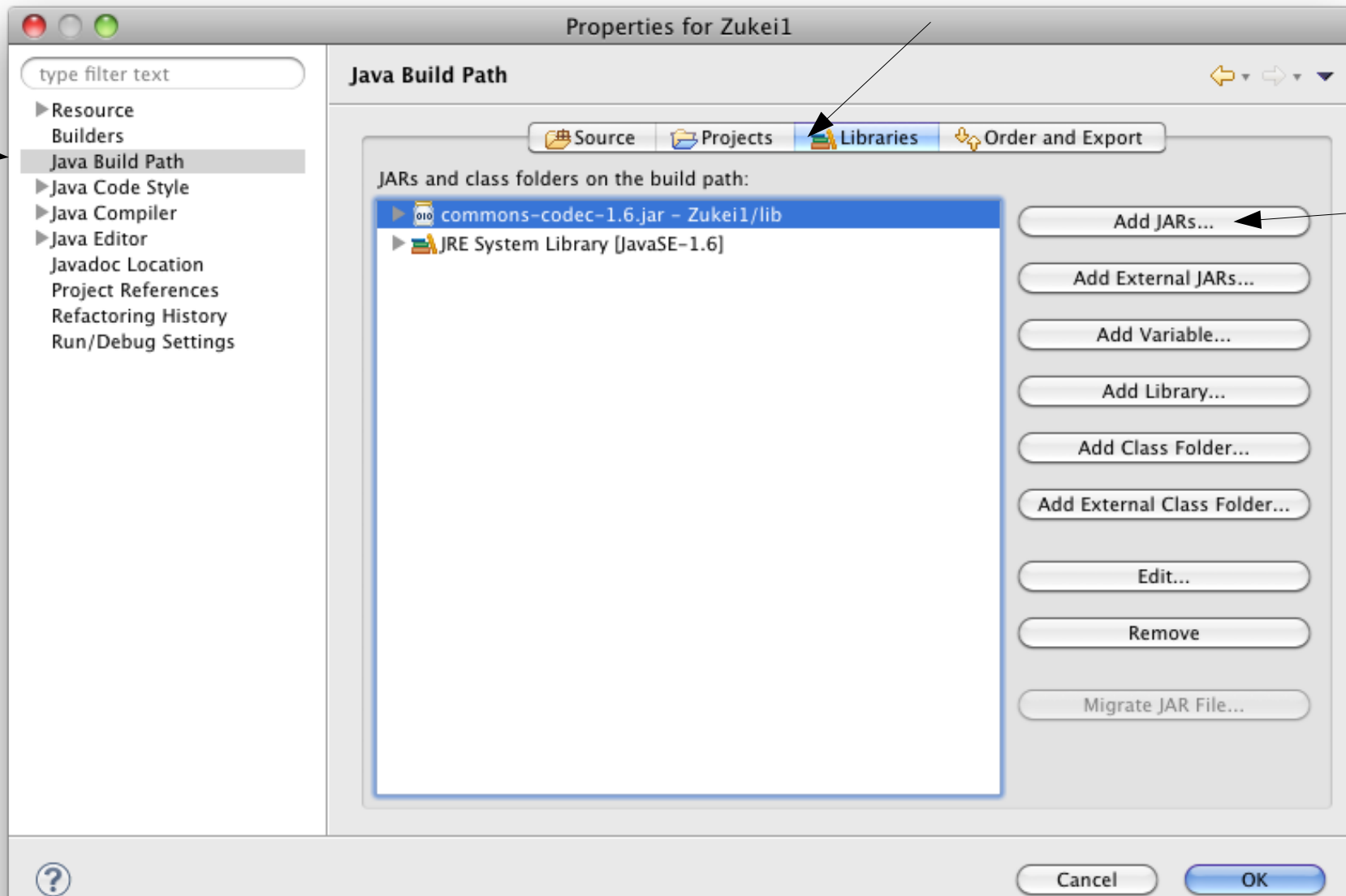
統合開発環境 Eclipse

準備その3 プロジェクトの属性を設定

このプロジェクトで使うjavacやjavaコマンドのオプションを設定

-classpath オプション用

java javacにjarファイルとクラスのあるディレクトリ位置を伝えるため



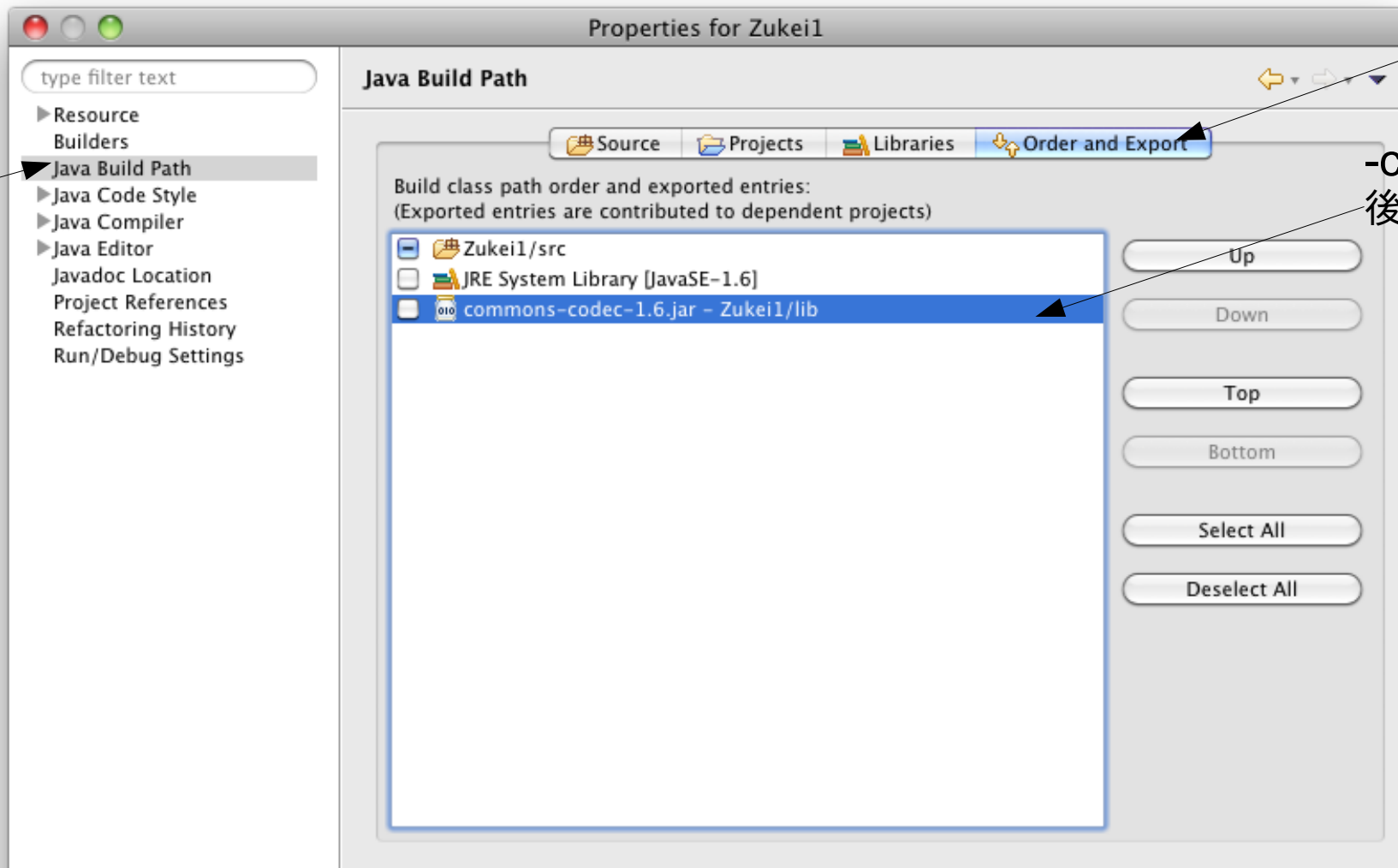
統合開発環境 Eclipse

準備その3 プロジェクトの属性を設定

このプロジェクトで使うjavacやjavaコマンドのオプションを設定

-classpathオプションの後ろに与える情報の順序を指定する

同一パッケージ名同一クラスが複数ある場合、最初に見つかったものを使うため



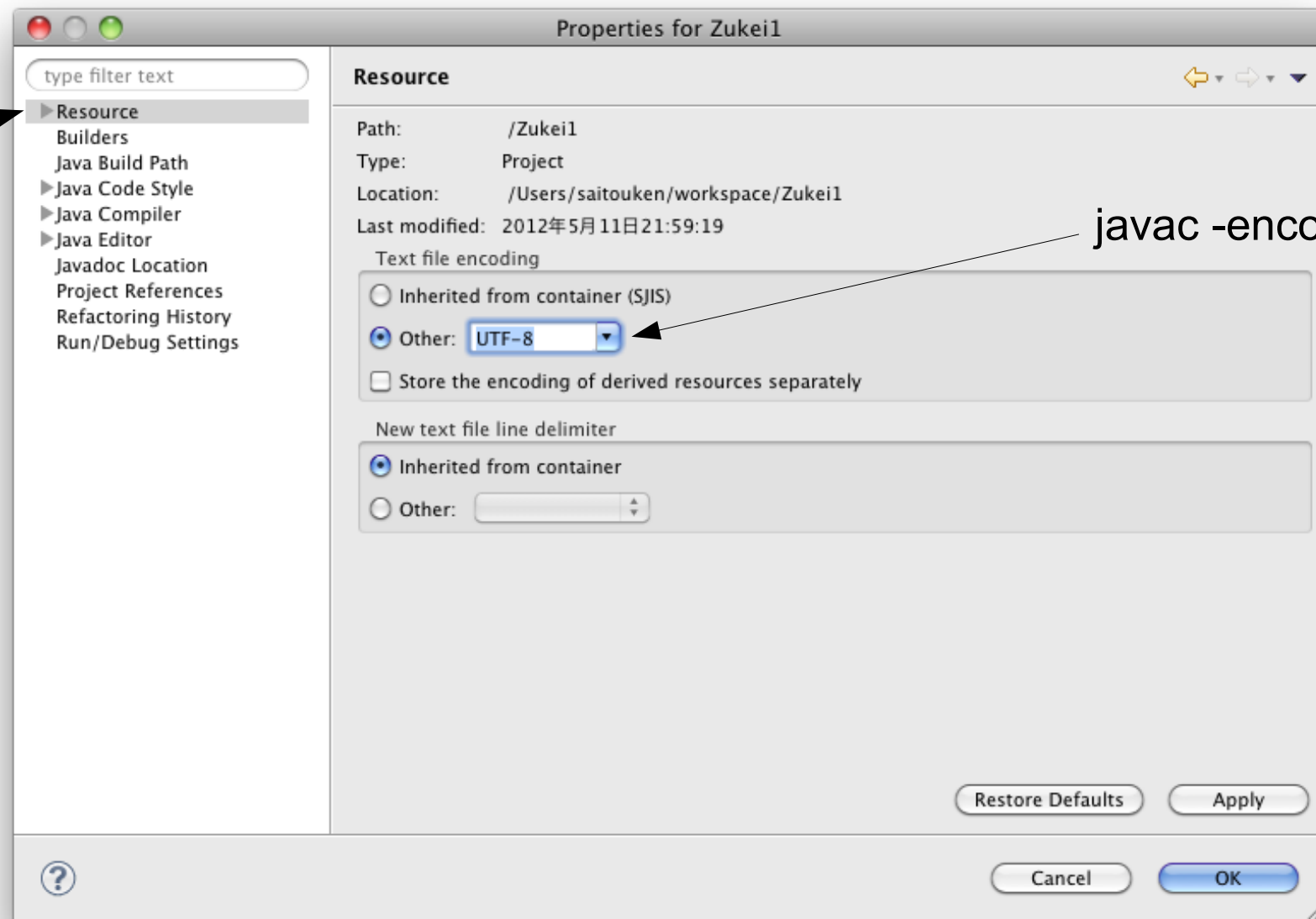
-classpath の
後の順番設定

統合開発環境 Eclipse

準備その3 プロジェクトの属性を設定

文字コードの確認

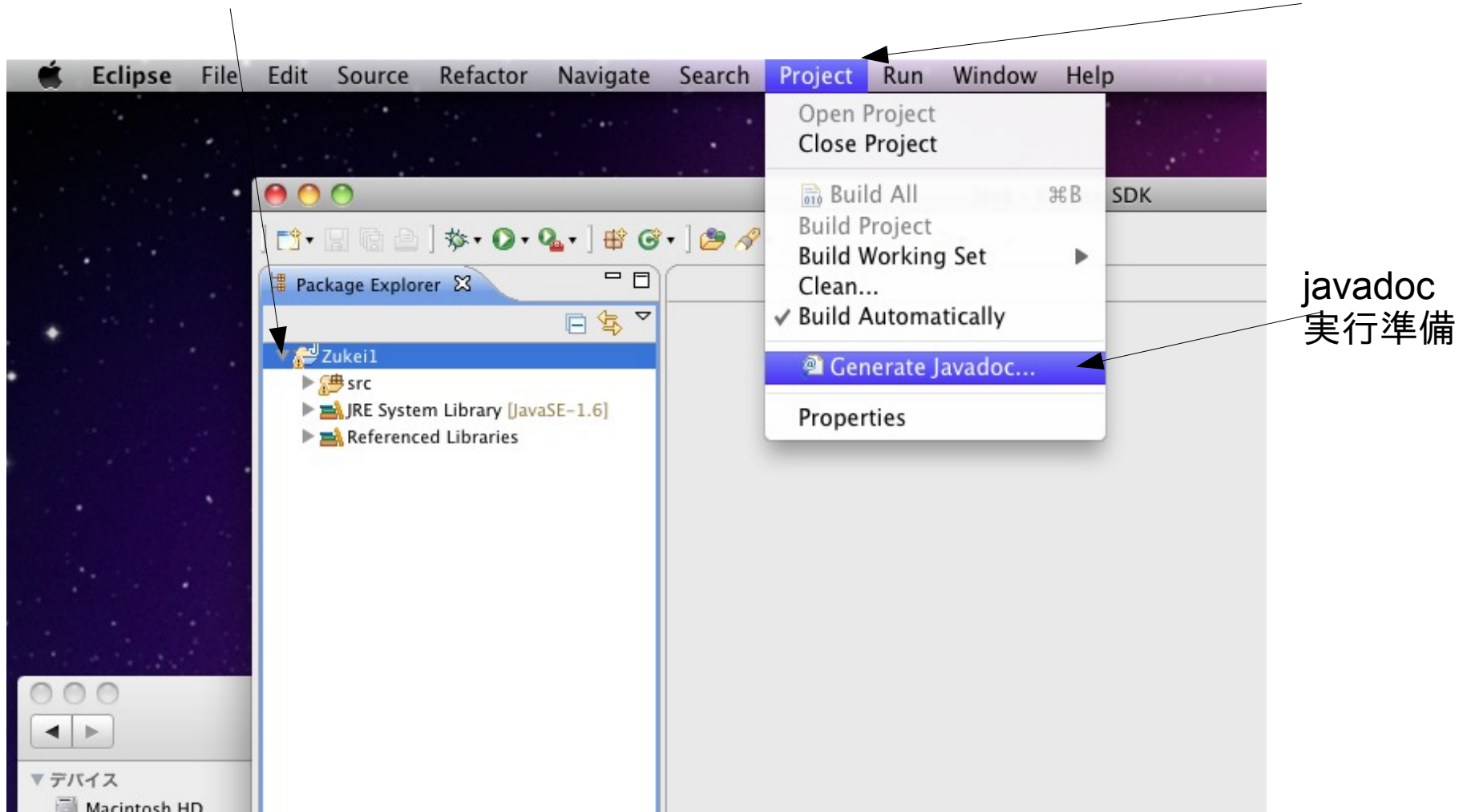
日本語のコメントなどがある場合に文字コードが異なるとコンパイルできません
文字コードは必ずUTF-8としてください。



統合開発環境 Eclipse

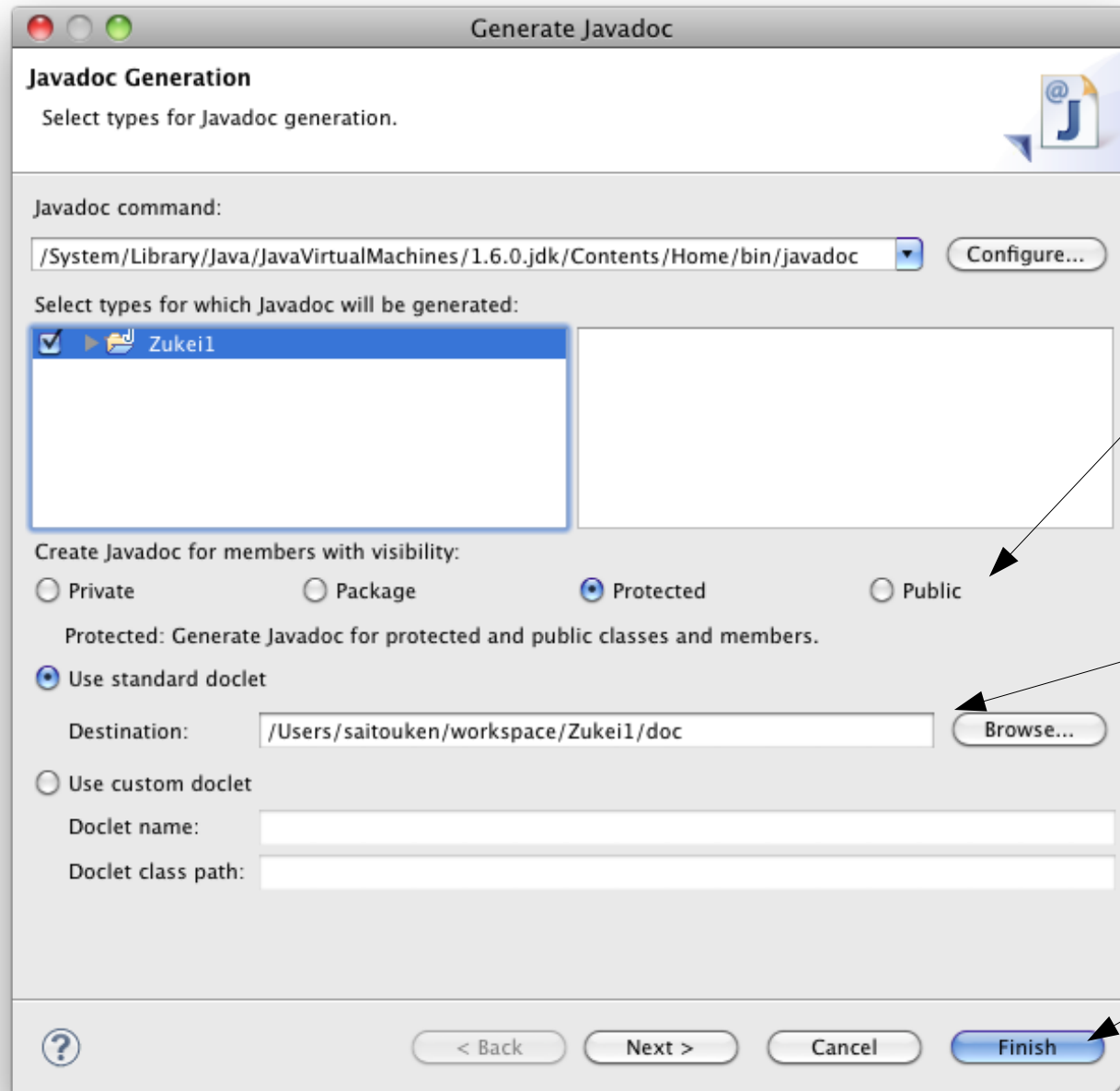
javadoc を作る方法

プロジェクトのトップを選択



統合開発環境 Eclipse

javadoc を作る方法



-private
-package
-protected
-public
のどれかを選択

-d オプション用

javadoc実行

統合開発環境 Eclipse

javaプログラムの実行

1) mainメソッドのあるクラスを開く

2) javaコマンドの通常実行

実行時の引数等を設定するときはこの▼をクリック

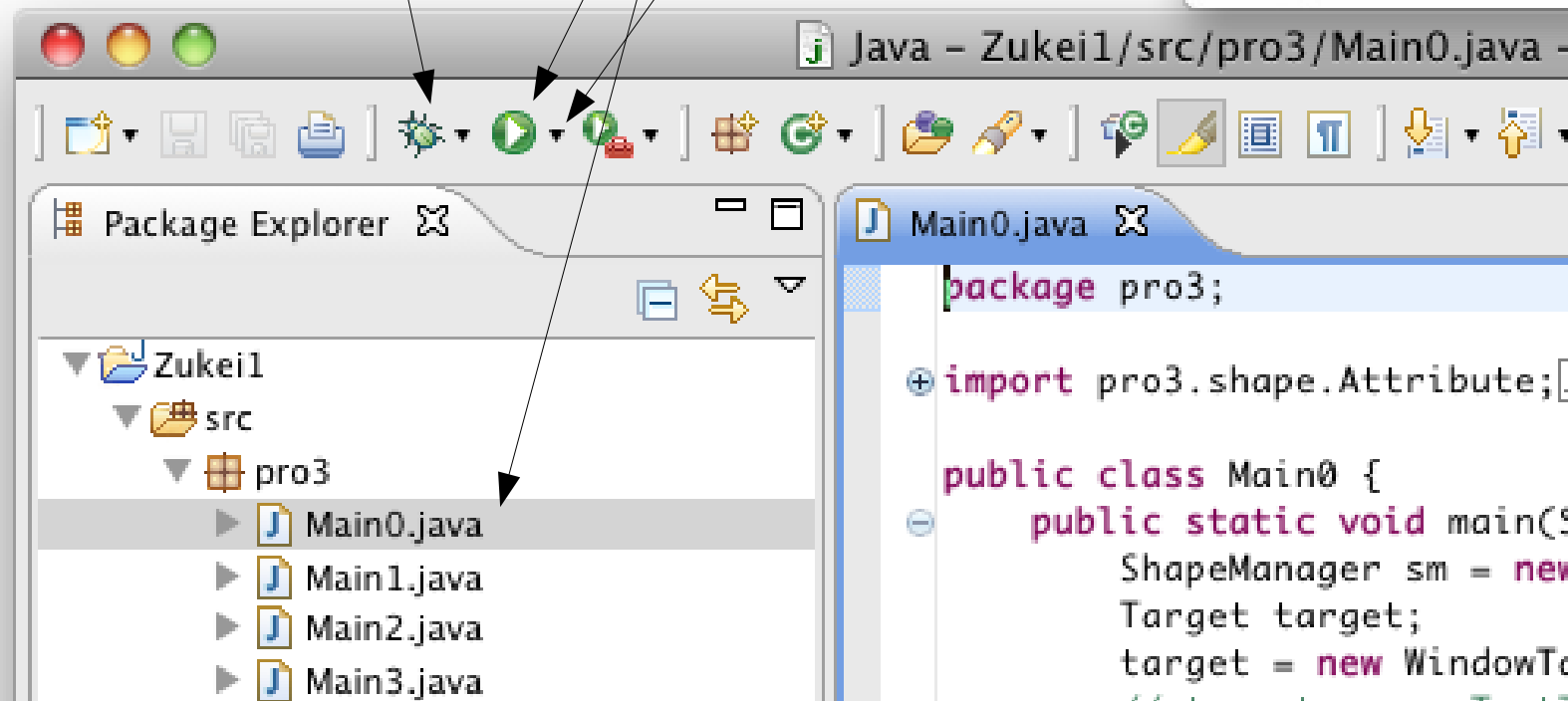
デバッグモードで実行するとき

(no launch history)

Run As ▶

Run Configurations...

Organize Favorites...



統合開発環境 Eclipse

Javaプログラムの実行

プログラムを実行するための設定

プロジェクトの設定を引き継いでいるので、常に設定する必要はないです

どのクラスのmainメソッドを使うか

引数を与えるとき

環境変数を設定するとき

