プログラム設計

http://bit.ly/design4d

IDEによるJavaアプリケーション開発

後期 第15週

2020/1/20

本日は・・・

IDE（統合開発環境）の1つであるEclipseを使ってJavaアプリケーションを開発します。

IDEの特徴

- 編集・コンパイル・実行・デバックといった、
  実装の一連の作業を1つのツールで完結できる
- コードの色付けや自動字下げだけではない多くの機能
  - コード入力の補完機能
  - エラー解決方法の提案
  - 自動コンパイル
  - 視覚的なデバッグ
  - などなど
- 本格的なソフトウェア開発では必要不可欠
  - Java向け：Eclipse, NetBeansなど
  - iOS, macOSアプリケーション開発向け：Xcode
  - Androidアプリケーション開発向け：Android Studio

【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15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 week215
$ cd week215
```

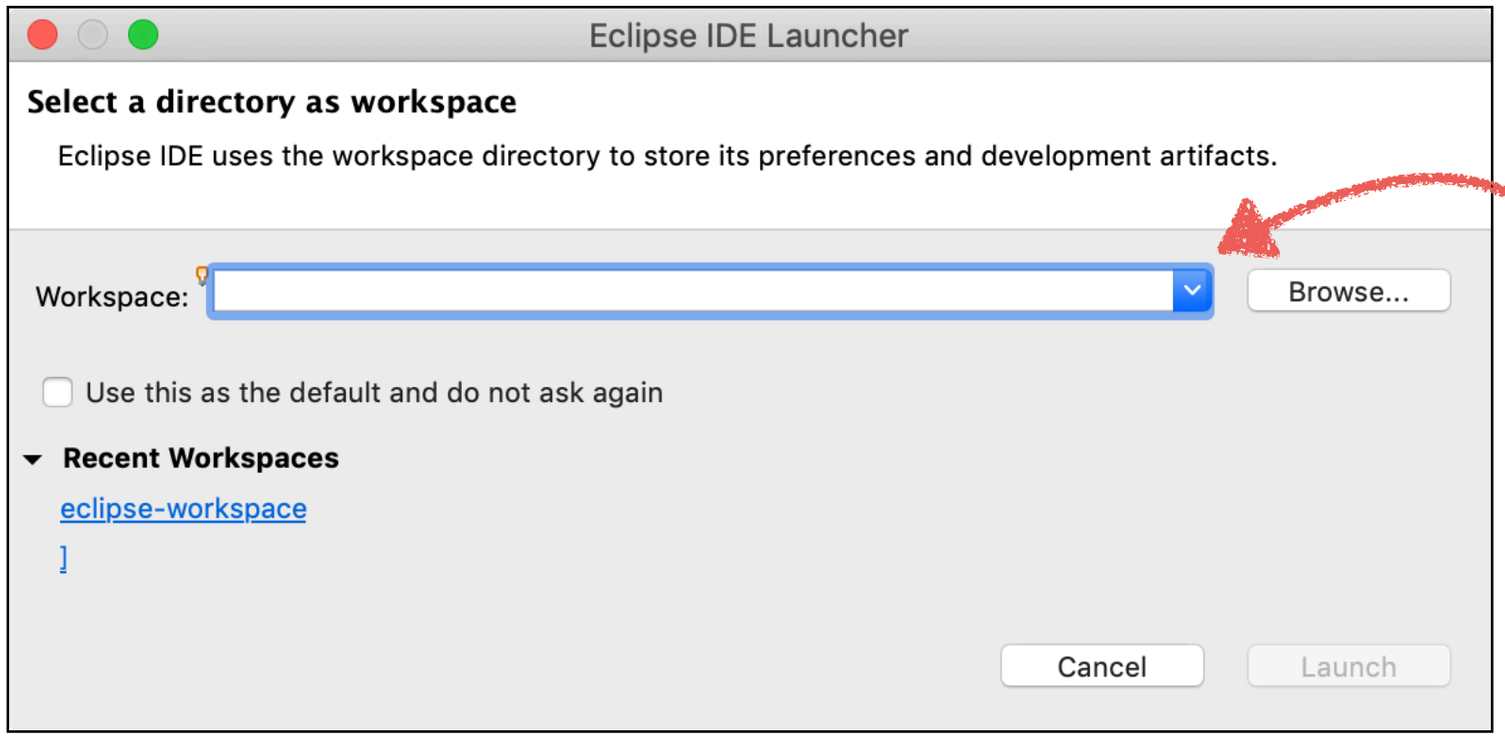
※課題で作るファイル名は各自で決めて構いません。

Eclipseの起動

演習室ではターミナルで以下のように実行する

```
$   eclipse   &
```

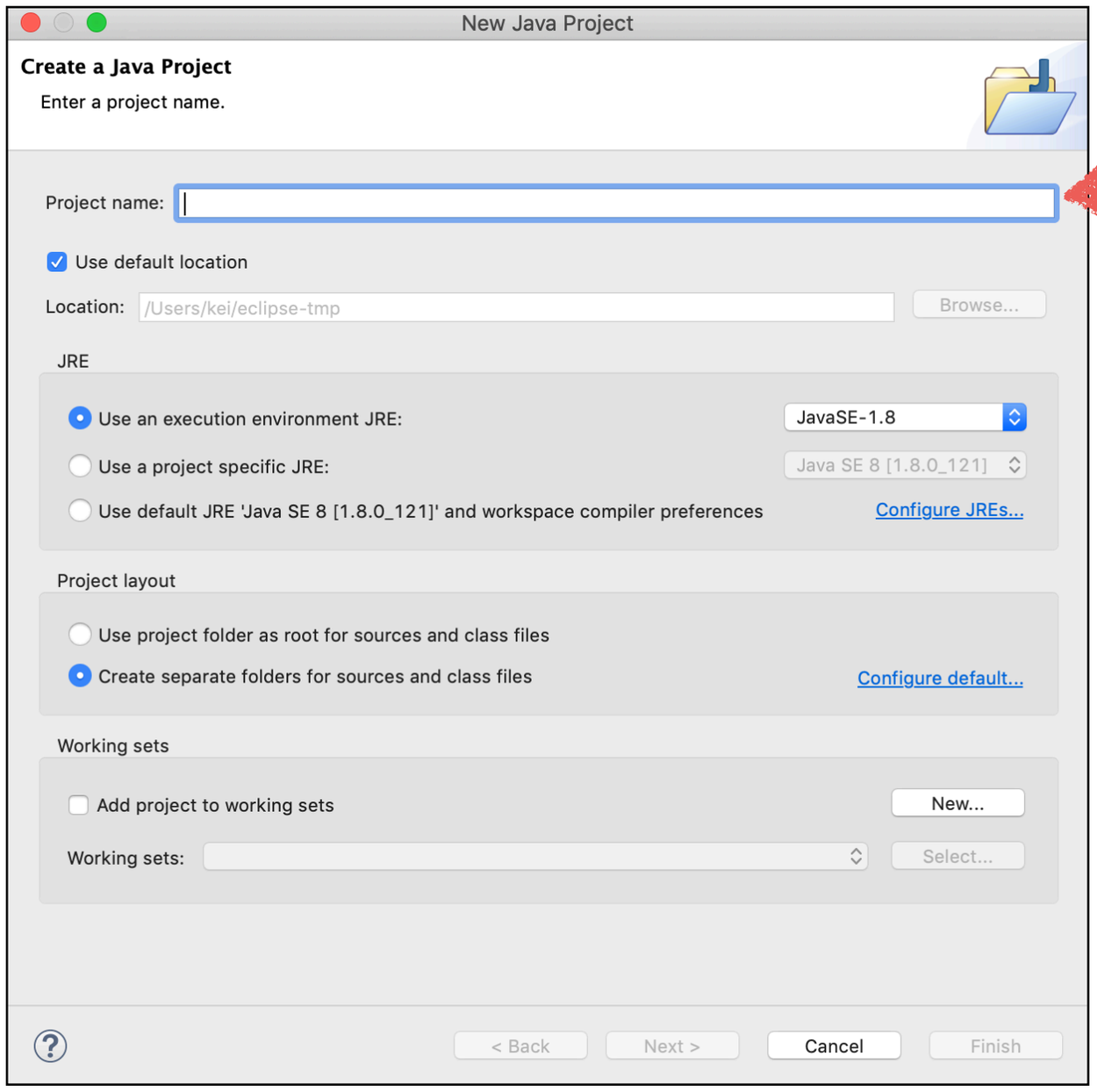
起動後、ワークスペース（プロジェクトを保存するフォルダ）の場所を選択する。



※演習室では、GitHubでcloneしたフォルダ以下を指定しないと消えてしまうことに注意

プロジェクトの作成

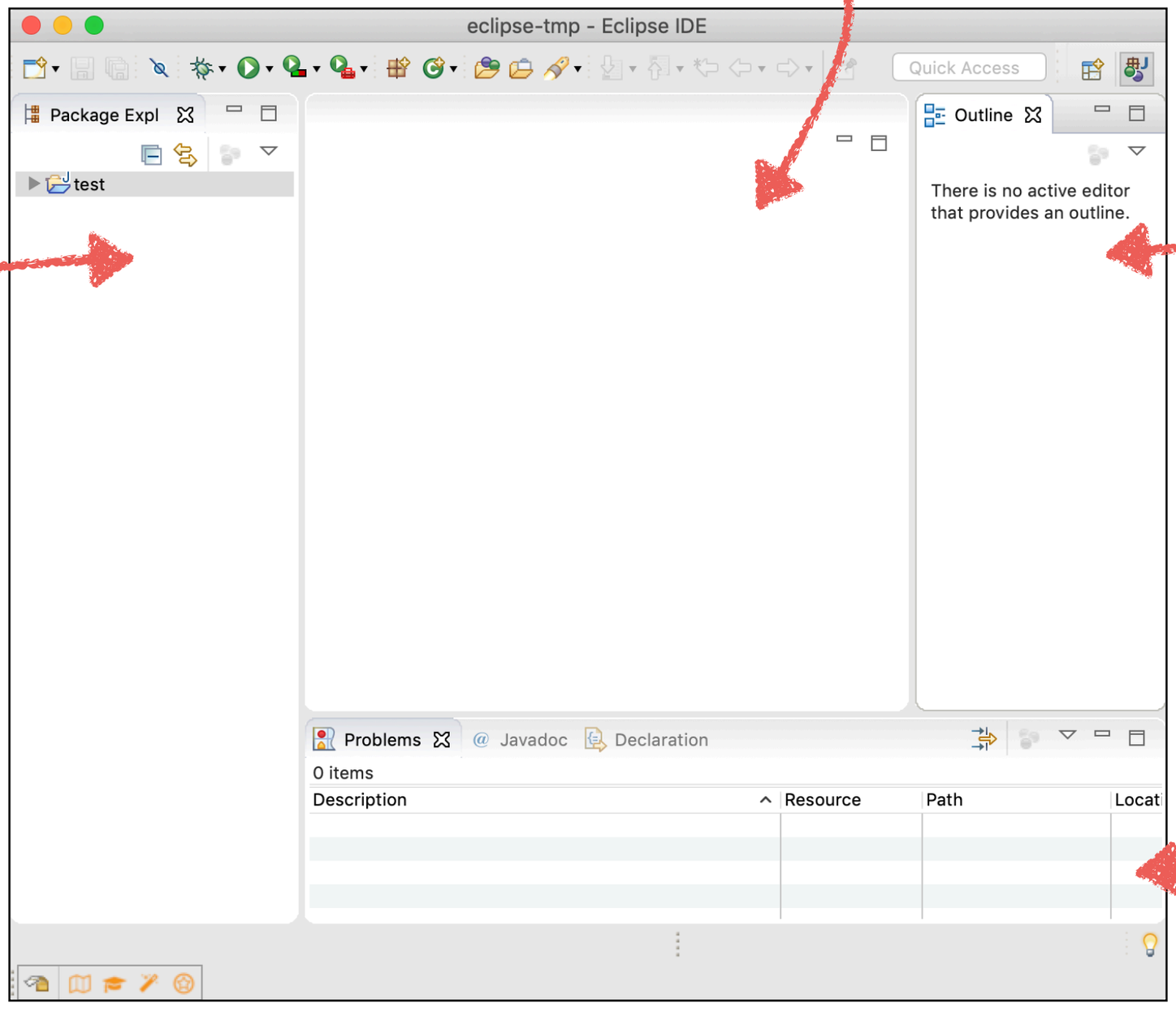
「New」ボタンメニュー→「Java Project」を選択する



プロジェクト名を
入力して
新規作成する

Eclipseの画面構成

プログラムコードを編集するエディタ



ワークスペース内にあるプロジェクトのリスト

編集しているクラスのアウトライン（属性・操作のリスト）

エラーメッセージや標準入出力用のコンソール

※配置や表示内容は自由にカスタマイズ可能

クラスの新規作成

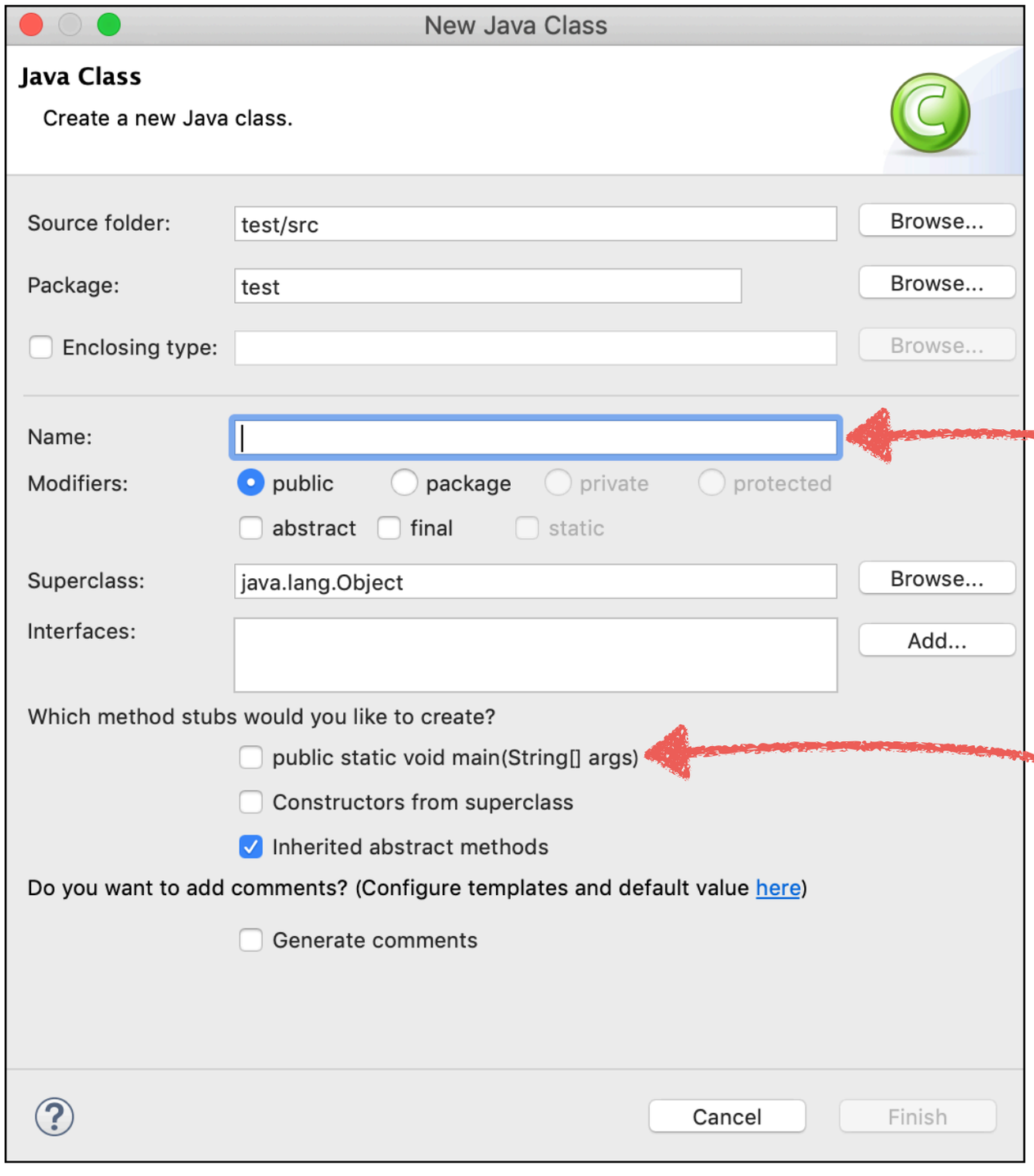
Eclipseでは、基本的に1つのファイルに1つのクラスを作成する（1つのファイルに複数クラスを作っても構わない）

① クラスを新規作成するプロジェクトを選択
②　「New」ボタンメニュー →「Class」を選択

または

① 対象プロジェクトで右クリック
②　「New」→「Class」を選択

クラスの新規作成



クラス名を入力する

mainメソッドを
作成時に付ける場合
はチェックを入れる

※ワークスペース外に保存されているクラスを後からプロジェクトに読み込むこともできる
（対象プロジェクトで右クリックして「Import」を選択）

【練習15-1】

次のようなクラスを 1 つ作ってみましょう。

- クラス名は HelloWorld
- 「public static void main(String[] args)」にチェックを入れる

コードの入力

入力時にEclipseが支援してくれる主な内容

- { や (などの括弧は、自動的に閉じてくれる
- インデント（字下げ）は、改行時またはTabを押すと自動的に揃えてくれる
- コンパイルの必要なし（保存時に自動的にコンパイルしてくれる）
- エラーは入力時に教えてくれる（該当行の左にアイコン表示されるので、そのアイコンをクリックすると解決方法を教えてくれる）
- 変数名の後に「ピリオド」を入力すると、メソッド名を補完して候補リストを表示してくれる（絞り込み可能）
- 入力中に「alt+/」を入力すると補完リストを表示してくれる
- 「sysout」と入れて「alt+/」を入れると「System.out.println」を補完
- 変数のsetter, getterを自動生成してくれる
- 後からクラス, フィールド, メソッドなどをちゃんと変更できる
- 範囲選択してまとめてコメントにできる（コメントを外すこともできる）

【練習15-2】

先程のクラス HelloWorld のメソッド main に、
「Hello を出力する」処理を追加してみましょう。

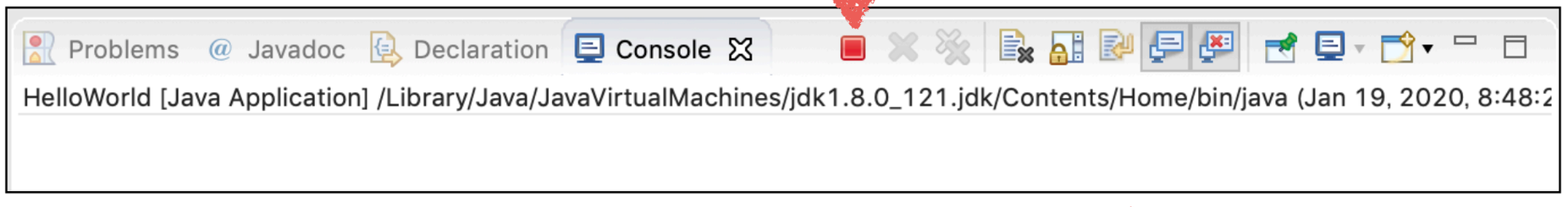
プログラムの実行

① 画面左側、mainを持つクラスを右クリックして
「Run As」→「Java Application」を選択
（以降、同じmainを実行する場合は、ツールバーの「Run As」ボタンを押すだけでよい）

② GUIアプリケーションの場合はフレームが表示される
標準入出力の情報はコンソール（画面下部）で入出力できる

③ ターミナルで「Control + C」で終了していたようなプログラムをEclipseで終了する場合は、コンソール画面右上にある「終了」ボタンを押す

プログラム実行中だけ
「終了」ボタンが現れる



画面下部の「Console」タブ

【練習15-3】

先程のクラスHelloWorldのメソッドmainを実行してみましょう。

【課題15-1】

以下の「時間」を表すクラスをEclipseで作ってください。

- クラス名：MyTime（作成と同時にmainメソッドも作っておく）
- フィールド：hour, minute（どちらもint型）
- メソッド：以下の3つ
  - public void setHour(int h)
    //フィールドhourのセッタ（フィールドhourにhを代入する）
  - public void setMinute(int m)
    //フィールドminuteのセッタ（フィールドminuteにmを代入する）
  - public void show()
    //hourとminuteの値を出力する

せっかくなので、できる人は以下の操作も利用してみて下さい。

- セッタの作成は、作成したいフィードで右クリックして、
  「Source」→「Generate Getters and Setters…」を選択する
- メソッドshowの出力処理は、「sysout」と入力して「alt+/」で補完する

【課題15-2】

mainメソッド内に以下のように入力して下さい。
（Eclipseでは、自分が作ったクラスでも、「t1.」
と入力すると、メソッドの候補を表示してくれる。）

```
MyTime t1;
t1 = new MyTime();
t1.setHour(15);
t1.setMinute(55);
t1.show();
```

【課題15-3】

課題15-2で作ったmainを実行して、コンソール画面で出力結果を確認してください。

【コンソールでの実行結果】

```
15:55
```

【課題15-4】

以下の手順でクラスTimeの名前を変更して下さい。

① プログラムのMyTime（どこの「MyTime」でもよい）をマウスで選択する
② マウスを右クリックして、
「Refactor」→「Rename…」を選択する
③ 別の名前を入力する（MyTime2など）
④ 関連している全てのMyTimeがMyTime2に置き換わる

※ 単なる文字列置換ではなく、プロジェクト内のプログラムの構造が損なわれないように全て置き換えてくれる

【課題15-5】

できる人は、課題15-4と同様に以下の名前変更をしてください。

- クラス型変数t1を「t2」に名前変更する
- フィールドhourを「hour2」に名前変更する
- メソッドshowを「show2」に変更する

名前変更後でもプログラムは正常に動作することを確認してください。

【課題の提出】

以下の流れで、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15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-(ユーザ名)

試験範囲

- 第8週～第14週
  - シーケンス図
  - ステートマシン図, アクティビティ図
  - ユースケース図
  - データフロー図
  - など

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