

ネットワークアプリケーション

第2回 ネットワークアプリケーションの
プログラミングモデル
Javaによるウィンドウプログラミング

石井 健太郎

(423研究室・オフィスアワー火3限)

スケジュール

- 9月27日 第1回「TCP/IPプロトコルスイート」
 - 10月4日 第2回「Javaによるウィンドウプログラミング」
 - 10月11日 第3回「ネットワークアプリケーションのプログラミングモデル」
 - 10月18日 第4回「Javaによるネットワークプログラミング」
 - 10月25日 第5回「Javaによるネットワークプログラミング」
 - 11月8日 第6回「Javaによるネットワークプログラミング」
 - 11月15日 第7回「Javaによるネットワークプログラミング」
 - 11月17日** 第8回「ウェブプログラミングについて」
- 最終課題(1)**

スケジュール

- 11月22日 第9回「JavaScriptによるクライアントサイドウェブプログラミング」
- 11月29日 第10回「JavaScriptによるクライアントサイドウェブプログラミング」
- 12月6日 第11回「JavaScriptによるクライアントサイドウェブプログラミング」
- 12月13日 第12回「JavaScriptによるクライアントサイドウェブプログラミング」
- 12月20日 第13回「JavaScriptによるクライアントサイドウェブプログラミング」
- 1月10日 第14回「JavaScriptによるクライアントサイド...」 **最終課題(2)**
- 1月19日 第15回「まとめと演習」

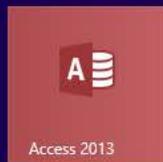
- **レポートを提出してください**

Eclipseで開発

- **本講義のJavaの演習にはEclipseを用いる**
 - EclipseはJava(以外の言語にも使用できるが)の統合開発環境 (IDE)
 - GUIの操作で様々な処理が可能, 補完・自動フォーマットなどの便利機能もある
 - コマンドを打たずとも自動コンパイル, ライブラリのリンクもプロジェクトで管理

スタート

Microsoft Office



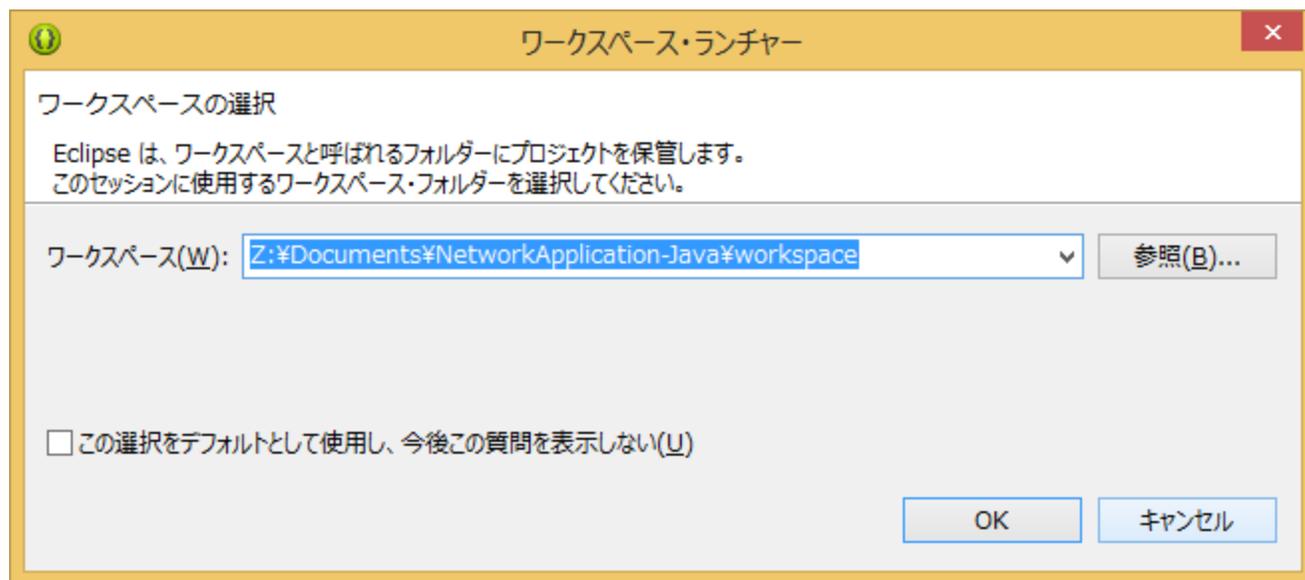
検索

すべての場所

eclipse



eclipse





パッケージエクスプローラ

- SimpleWindowApplication
 - src
 - JRE システム・ライブラリー [JavaSE-1.7.0_75]

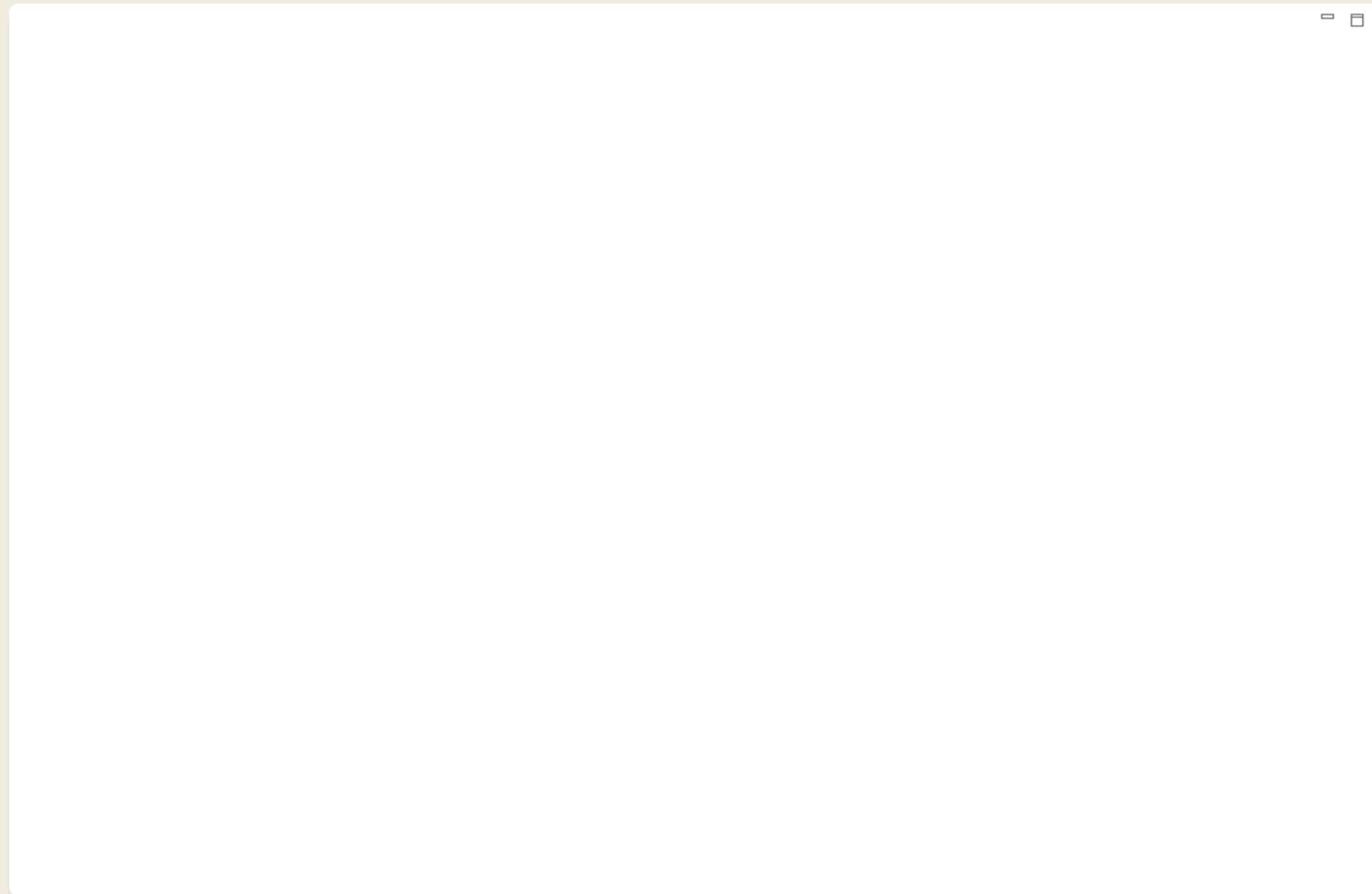
プロジェクト

DB ツリービュー

Filter:

Column:

- DBViewerPlugin
 - お気に入り



タスクリスト

検索

すべて アクティブにする...

Mylyn 接続

タスクおよび ALM ツールへ [接続](#)、またはローカル・タスクを [作成](#) します。

アウトライン

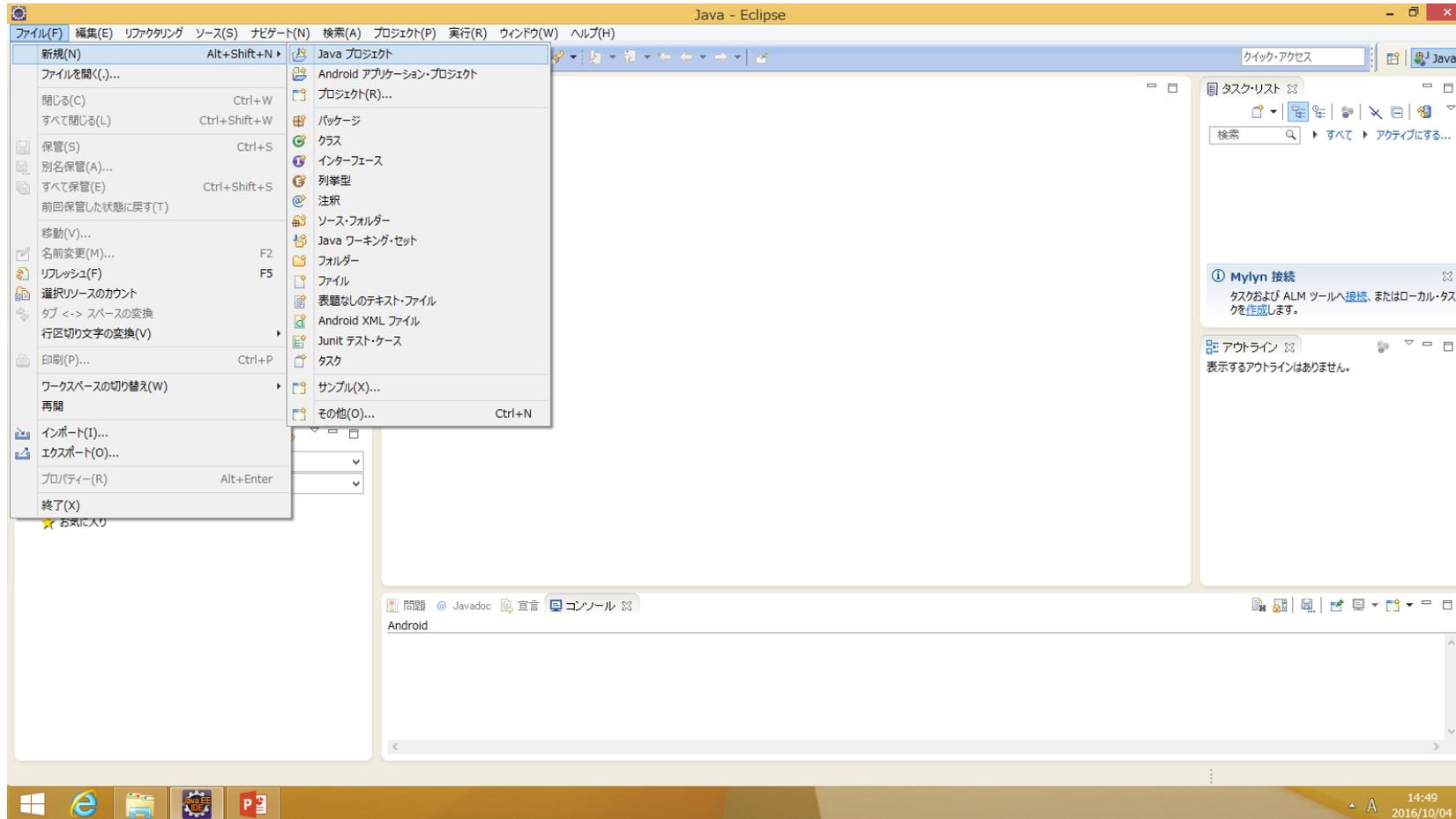
表示するアウトラインはありません。

問題 @ Javadoc 宣言 コンソール

Android

|

プロジェクトの作成



プロジェクトの作成



Graphicsクラス

- **画面の描画を行うクラス**

- 文字列の描画 `drawString ()`
- 線の描画 `drawLine ()`
- 色の指定 `setLine ()`
- 楕円(円)の描画 `drawOval ()`
- など
- <https://docs.oracle.com/javase/jp/8/docs/api/java/awt/Graphics.html>

- **(発展)**

GraphicsクラスをGraphics2Dクラスにキャストするとより強力な描画の指定ができる

- Javaでは(ほとんどの言語も社会に出ると),
APIドキュメントを読んで必要な機能を探ることができなければいけない
- <https://docs.oracle.com/javase/jp/8/docs/api/>

マウスイベント

```
MouseAdapter mouseAdapter = new MouseAdapter() {  
    @Override  
    public void mousePressed(MouseEvent me) {  
        int x = me.getX();  
        int y = me.getY();  
  
        repaint();  
    }  
  
    @Override  
    public void mouseDragged(MouseEvent me) {  
    }  
};  
this.panel.addMouseListener(mouseAdapter);  
this.panel.addMouseMotionListener(mouseAdapter);
```

- <https://docs.oracle.com/javase/jp/8/docs/api/java/awt/event/MouseListener.html>
- <https://docs.oracle.com/javase/jp/8/docs/api/java/awt/event/MouseMotionListener.html>
- <https://docs.oracle.com/javase/jp/8/docs/api/java/awt/event/MouseEvent.html>

(発展)キーイベント

```
KeyAdapter keyAdapter = new KeyAdapter() {
    @Override
    public void keyPressed(KeyEvent ke) {
        switch (ke.getKeyCode()) {
            case KeyEvent.VK_ESCAPE:
            case KeyEvent.VK_Q:
                quit();
                break;

            default:
                break;
        }
    }
};
addKeyListener(keyAdapter);
```

- <https://docs.oracle.com/javase/jp/8/docs/api/java/awt/event/KeyListener.html>
- <https://docs.oracle.com/javase/jp/8/docs/api/java/awt/event/KeyEvent.html>

演習

- ウィンドウアプリケーションを起動できたら、Graphicsクラスのメソッドを調査して使って、様々な文字列・図形を描画する
 - 基本課題
 - 最低限、学生番号と名前を描画すること
 - 文字列の描画以外のメソッドを使って、見た目におもしろいウィンドウの描画を行おう
 - 発展課題
 - 今回、アニメーションしている必要はないが、わかる人は挑戦しよう
 - マウスイベントを利用して、マウスに反応する描画を行おう
- プログラム(SimpleWindowApplication.java)の内容を、**テキストファイルにコピーして**、**次回の講義開始時まで**に、提出フォルダ(Xドライブ)にファイルで提出
 - X:¥IN科専門¥石井講師¥ネットワークアプリケーション¥第2回
 - ファイル名は「<学生番号>.txt」とする(ハイフンなし) 例: N14999.txt