# 並列プログラミング
# Parallel Programming

2018 2Q

演習　第2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

# 演習課題概要

- **目的**
  - キーボードイベント駆動プログラムを作成する
  - Java の復習
- **題材**
  - 電卓プログラム
  - 図形表示プログラム

# 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　para2.tar.gz を展開する

```
tar zxvf para2.tar.gz
```

```
tar xvf para2.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

Para2(トップディレクトリ)

- Makefile （後述）
- README 説明文
- sweet.png
- 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

Calculator 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/calc/Calculator.java
```

実際は一行で書く

として下さい

| | |
|---|---|
| `-d bin` | コンパイル後のクラスファイルをディレクトリ bin に置く |
| `-encoding UTF-8` | ソースファイルの文字コードが UTF-8 であることを示す |
| `-sourcepath src` | ソースファイルがディレクトリ `src` 以下にあることをコンパイラに教える |
| `-classpath bin` | 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.calc.Calculator
```

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.calc.Calculator と para.Main0? 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.calc;

import javafx.application.Application;
        ⋮
/** JavaFXで作成する電卓プログラム. */
public class Calculator extends Application

{
    /** 入力文字列表示領域. */
    Label input;
```

外部の Javadoc 文書へのリンク

実際は一行で書く

今回の演習では、 Para2 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.calc para para.graphic.shape
para.graphic.target para.graphic.parser
```

HTML文書の出力先ディレクトリ

パッケージ名

としてください

※ 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

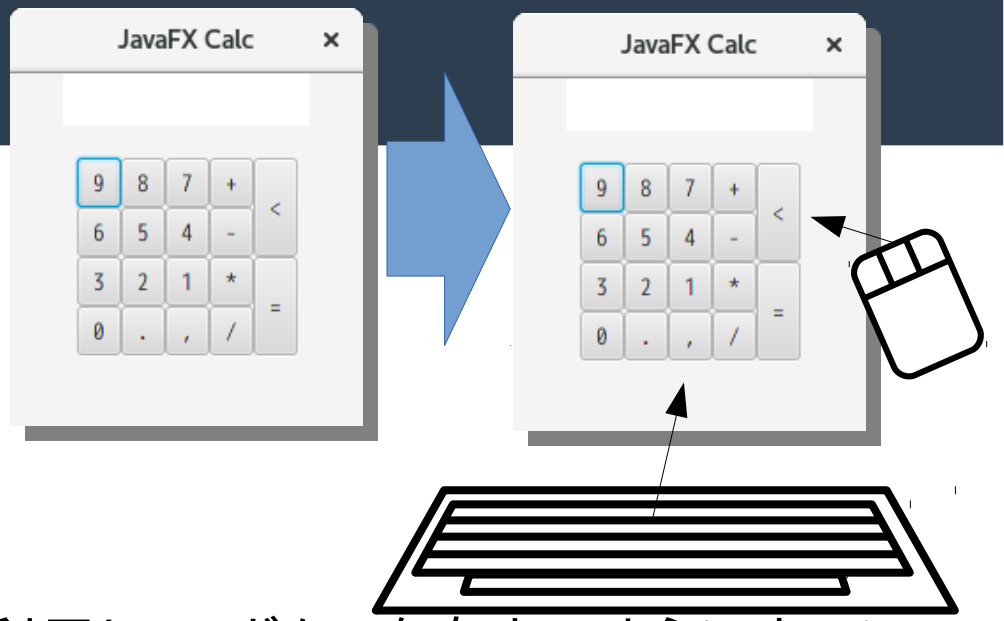
| | |
|---|---|
| `make Calculator` | Calculatorについてコンパイルして、実行 |
| `make Main00` | Main00 をコンパイルして、実行 |
| `make Main01` | Main01 をコンパイルして、実行 |
| `make Main02` | Main02 をコンパイルして、実行 |
| `make Main03` | Main03 をコンパイルして、実行 |
| `make clean` | bin 以下のクラスファイルをすべて削除 |
| `make javadoc` | javadoc コマンドを実行 |

ソースコードを更新してもmakeが感知しない場合があるので、コードを書き換えても結果に変化がない場合は、一度 make clean して再コンパイルして下さい

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\t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




- para.calc.Calculator にキーボード入力機能を追加せよ

小数も入力可能
“<” のキー入力は文字列の末尾の1文字を削除する機能とする

今回はjavafx.scene.layout.GridPane を利用して ボタンを右上のようにすでに配置済みです

1.1)ボタンのクリックを繰り返すことで、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前回付け加えましたが、同様にキーボード入力で文字列データを作成する機能**も追加せよ**

今回も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

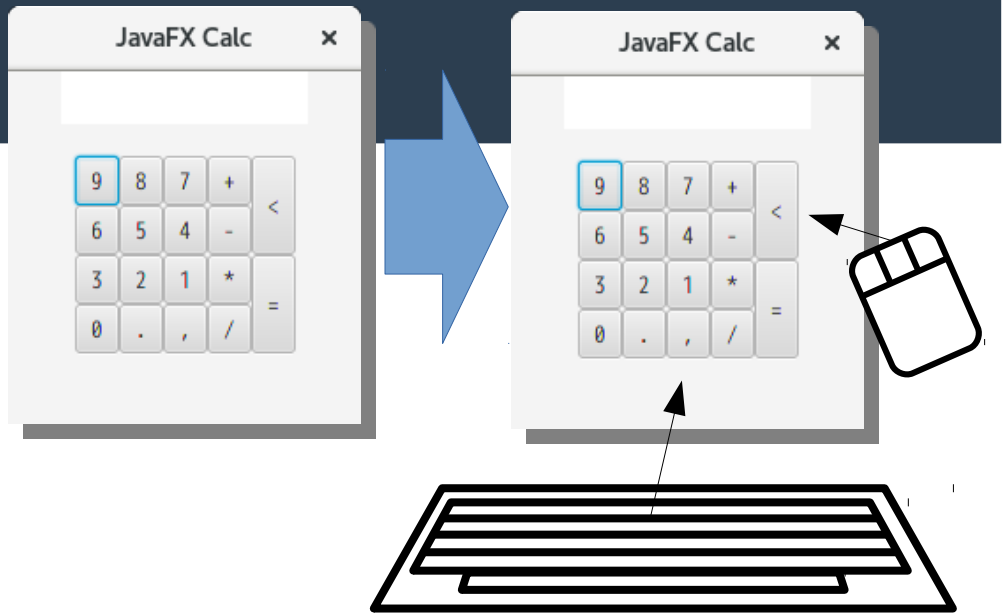
hint シーングラフのどのオブジェクトにイベントハンドラを登録すればユーザにとって便利かを考える

hint Eventクラスが持っているキー入力情報を使う

またキーボードの”=”を打つことで、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

# 課題 1

- para.calc.Calculator にキーボード入力機能を追加せよ



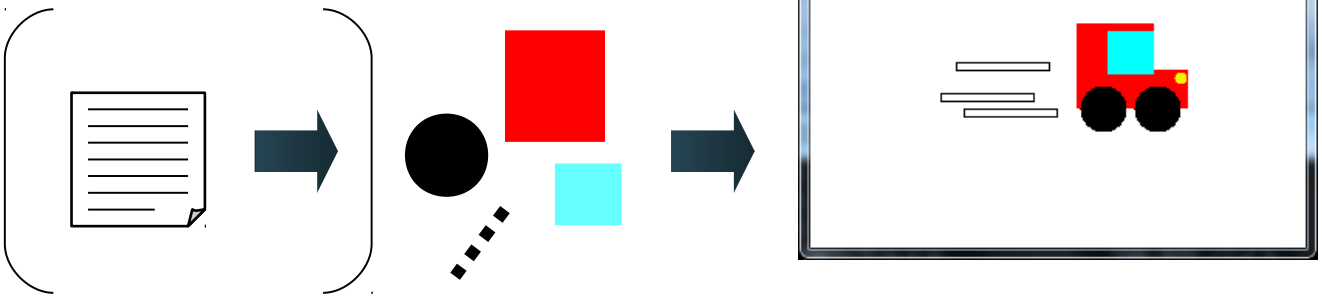

1.2) シーングラフのどのオブジェクトにイベントハンドラを登録したかを述べて、その実装法がユーザの様々な操作において便利であることの理由を説明せよ

# 課題 2 以降のプログラムについて

- 今後の演習で利用するプログラムに慣れるため、オブジェクト指向のプログラミングの復習と、作られたプログラムの部品を使った短いプログラムの改造をします

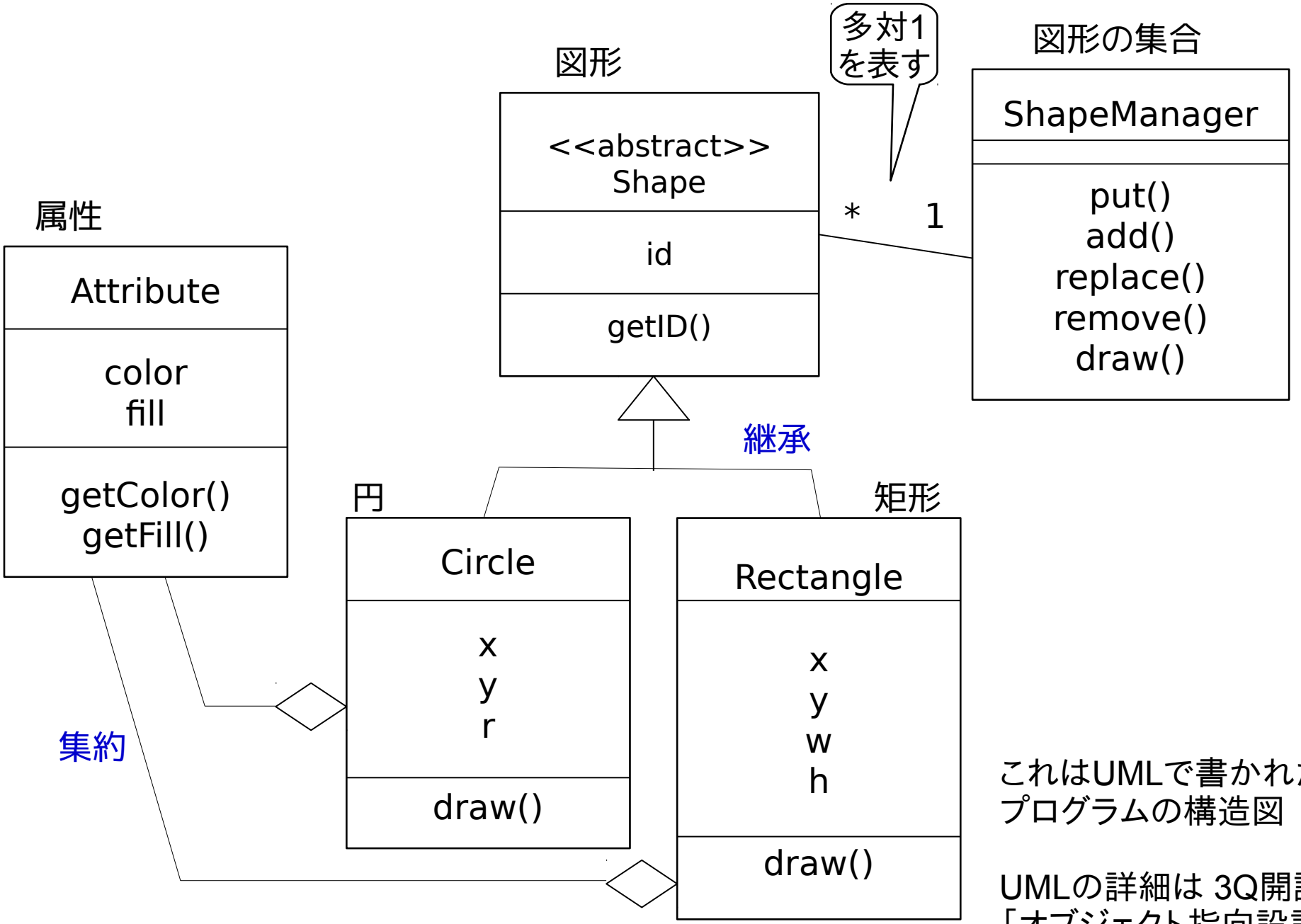
# 課題２以降のプログラムの説明 （1）概要

- プログラムは４つのパッケージから成る
  - メインプログラム（ para パッケージ）
  - 図形（ para.graphic.shape パッケージ）
  - 出力装置（ para.graphic.target パッケージ）
  - 構文解析器（ para.graphic.parse パッケージ




- 基本的流れ
  - メインプログラム内で図形オブジェクトを生成し，出力装置に対して出力する
  - 命令が書かれた文字列から構文解析器で図形オブジェクトを生成することもある

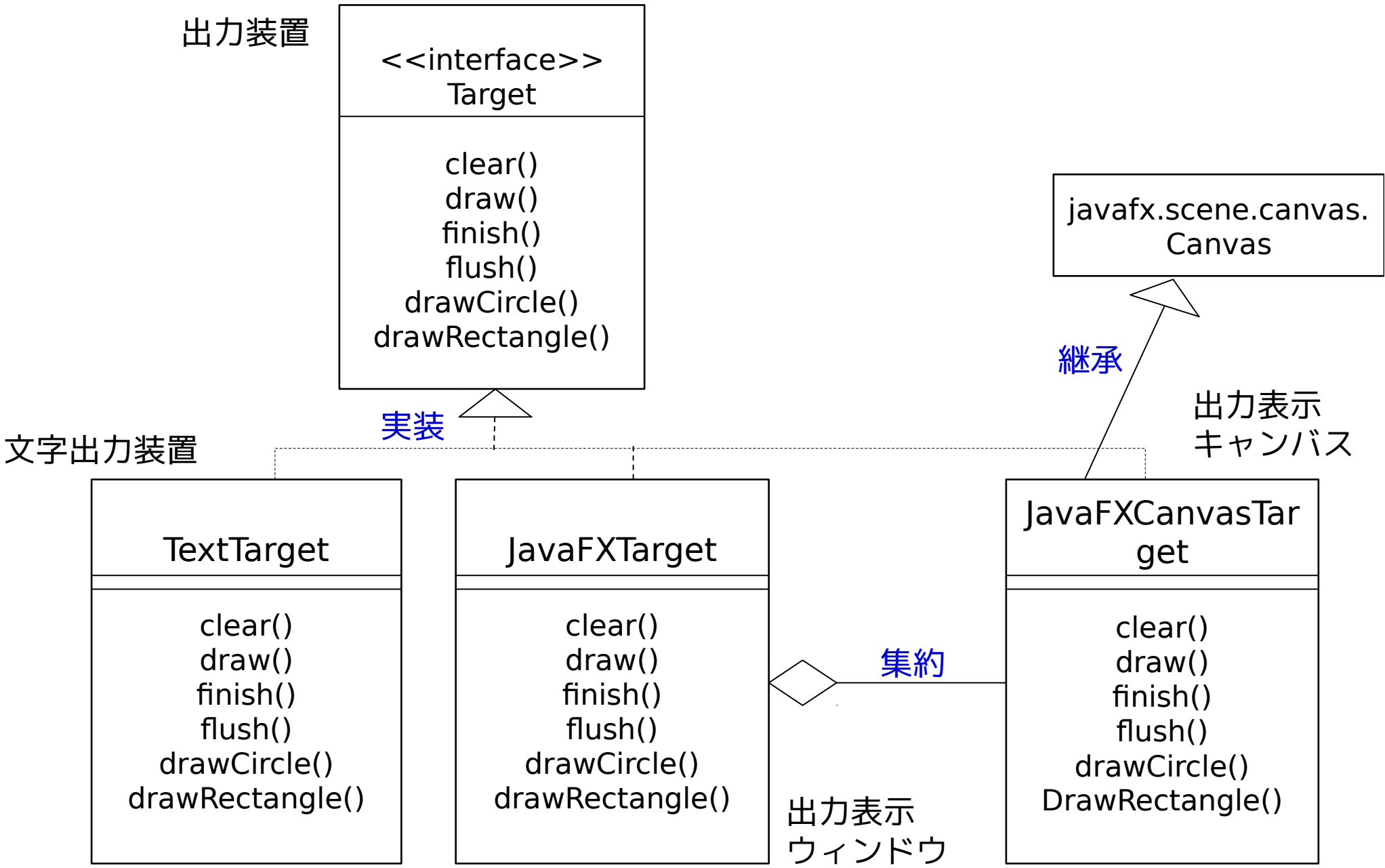
# 課題 2 以降のプログラムの説明 (2) 図形




これはUMLで書かれた
プログラムの構造図

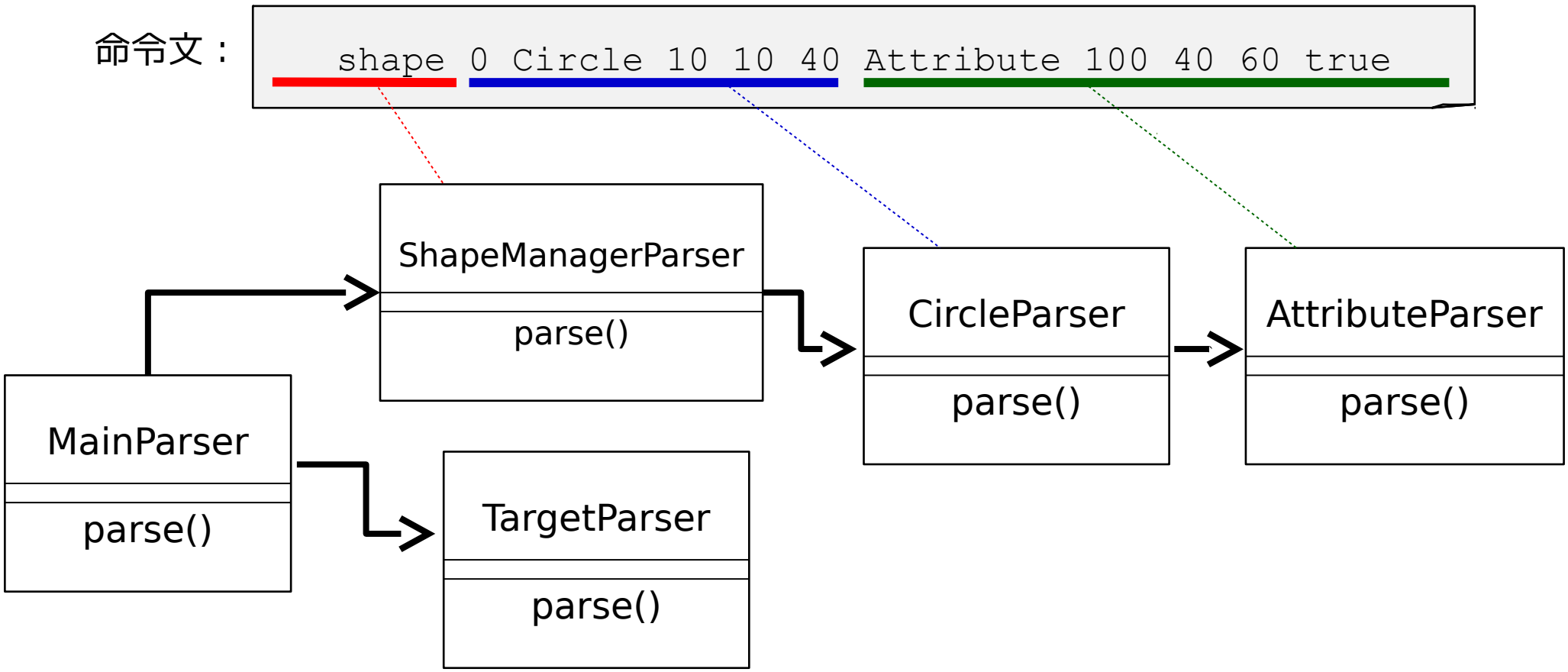
UMLの詳細は 3Q開講の
「オブジェクト指向設計」で

# 課題 2 以降のプログラムの説明 (3) 出力装置



# 課題 2 以降のプログラムの説明 (4) 構文解析器

- 命令文を解析しながら対応するオブジェクトを生成



# 課題 2

- Main00, Main01, Main02, Main03 というデモプログラムをコンパイル・実行する
  - Main0?.java の中で使われている Target 型変数の指すインスタンスを TextTarget 型と JavaFXTarget 型それぞれにして実行してみる
- javadoc コマンドを実行して HTML 文書を生成し，ブラウザで閲覧する
  - Mac OS X では `open` HTML ファイル名 とすればブラウザが起動する
  - 各クラスのパッケージ名などを確認する
  - ブラウザのエンコーディングの設定は UTF-8 にする
  - コンパイルエラーが起こる場合は展開直後にトップディレクトリで一度 make clean とタイプする

# 課題 3

- 新たな図形として三角形を追加しなさい
- 具体的には：
  - Triangle クラスを追加する
  - Triangle クラスの中には javadoc 用のコメントも記入する
  - 三角形を指定するための命令文を導入し，その命令文を解析できるよう構文解析プログラムを適宜変更する
  - Main02.java で円の代わりに三角形を用いた表示が行われるように変更する

※ Java Platform Standard Edition の標準 API を適宜利用すること

https://docs.oracle.com/javase/jp/10/docs/api/overview-summary.html

# 課題 4

- **package parser** の動作を次の文字列 **data** を解析する場合を例にとり説明しなさい
- 特にこの実装方法が **if-else** 文を並べた実装方法と比較して実装する上で優れていると気づいた点について述べなさい。ただし、オブジェクト指向言語が持つ大事な特徴に関係付けて回答すること

```
String data =
"shape 0 Circle 10 10 40 Attribute Color 100 40 60 Fill
true\n"+
"target draw\n"+
"target flush\n";

MainParser mp = new MainParser(...);
mp.parse(data);
```

# 課題 5

- ダウンロード直後の状態では，para.graphic.parser パッケージにある構文解析用のクラスは全て public 修飾子がついている
すなわち、 para.graphic.parser の外にあるクラス（以下，外部クラス）から全て参照可能である

- パッケージの外から見える必要のないクラスと見える必要のあるクラスを分け、 それぞれに適したアクセス修飾を付けなさい

- なぜそのようにアクセス修飾を付けのか、その理由を説明しなさい

javaのアクセス修飾は次の4種類

public、protected、private、無修飾

hint　スコープルールの質問です

# 課題 6

- 余力のある人は，para.graphic.parser.ImageParser の仕組みを説明しなさい

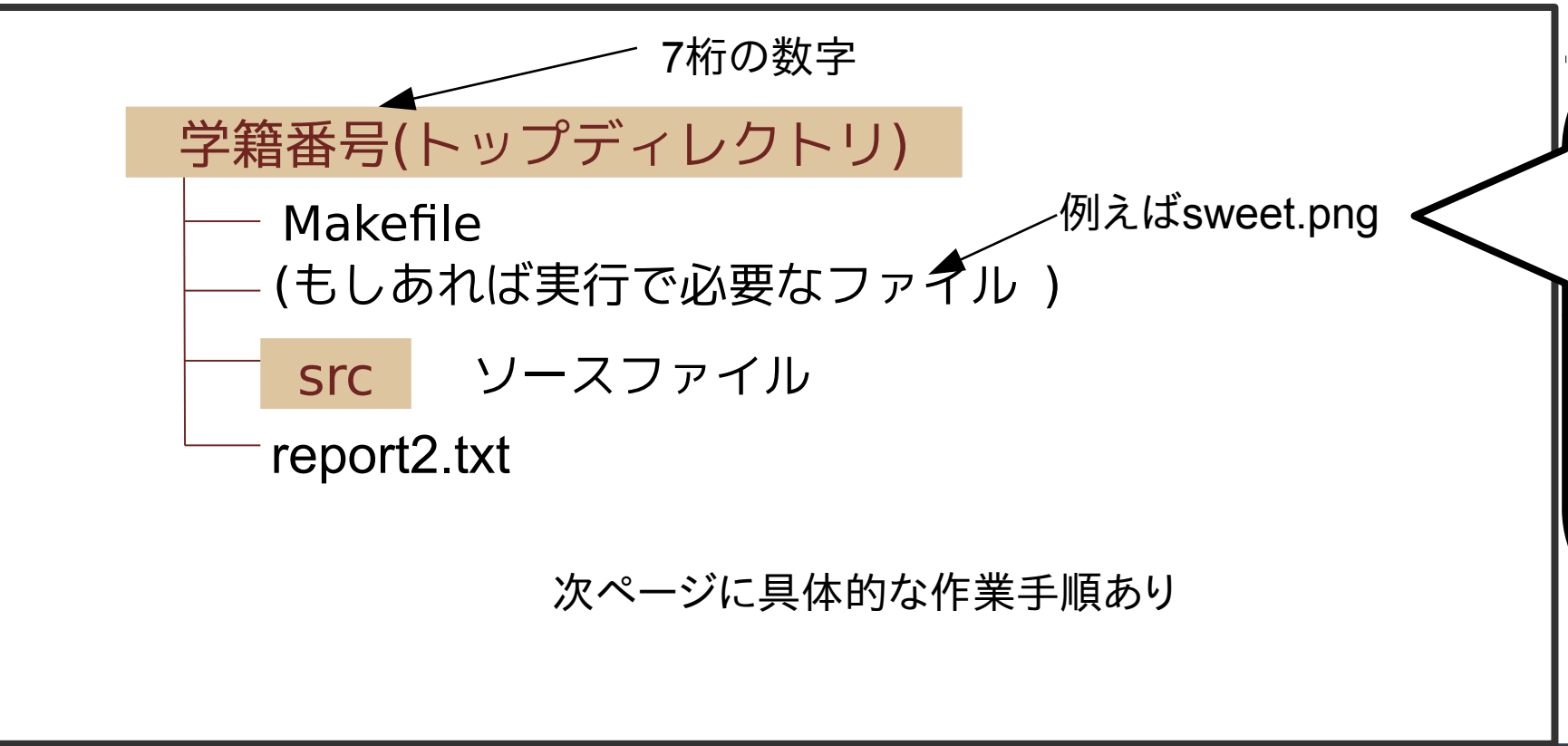
hint: javadoc で使用されているクラスのメソッドが行う処理を調べましょう

追加点が手に入ります

回答しましょう!!

# 提出方法 (1 of 3)

- para2.tar.gz を展開したディレクトリ構造を保ったまま，課題 1,3,5 の変更作業を行う
- 課題 1,3,5 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 1,4,5,6 の回答文、工夫点および感想を書いた report2.txt を用意する ( 雛形は課題のウェブページ )



# 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 report2.txt もコピーする

今回はPara2

```
cp トップディレクトリ/report2.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

例えばsweet.png

```
cp トップディレクトリ/otherfiles dir
```

- 次のコマンドを実行する

学籍番号に対応する7桁の数字にすること

```
zip ex2-1612345.zip -r dir
```

*dir* 以下の内容が圧縮され、 ex2-1612345.zip が作られます

- 圧縮後に内容を “ unzip ex2-1612345.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 29 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