並列プログラミング
Parallel Programming

2018 2Q

演習　第4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**
  - Multi Thread Programming を体験
  - GUI のスレッド動作の理解
  - 並行サーバプログラム

- **題材**
  - 電卓プログラム
  - 映像送受信プログラム

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　para4.tar.gz を展開する

```
tar zxvf para4.tar.gz
```

```
tar xvf para4.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

Para4(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

```
javac -d bin -encoding UTF-8 -sourcepath src
  -cp bin src/para/calc/Calculator.java
```

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

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

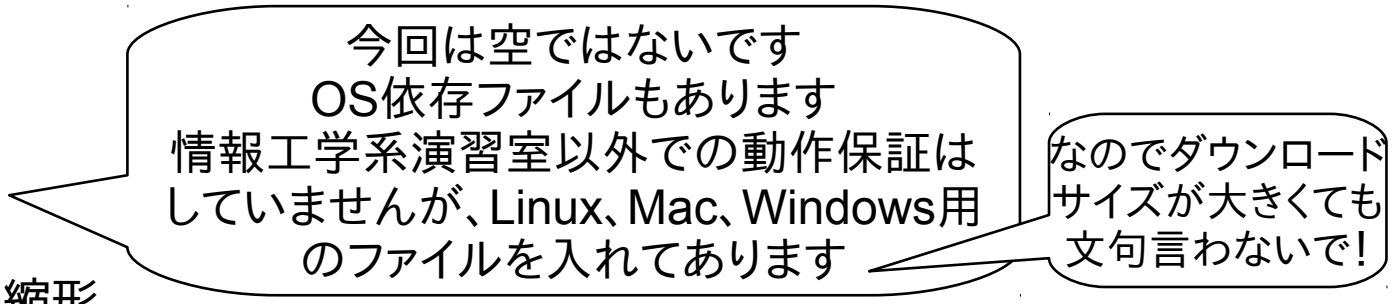
準備（3）

- 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.Main07
```

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.Main?? と para.calc.Calculator があります）



-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、 Para4 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
para.graphic.camera
```

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 をコンパイルして、実行
`make Main07`	Main07 をコンパイルして、実行
`make Main08 PARAMETER=hostname`	Main08 をコンパイルして、実行
`make Main09`	Main09 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

ホスト名か192.168.0.1のようなIPアドレスの文字列で接続先を指定可能

省略すると、PARAMETER=localhost

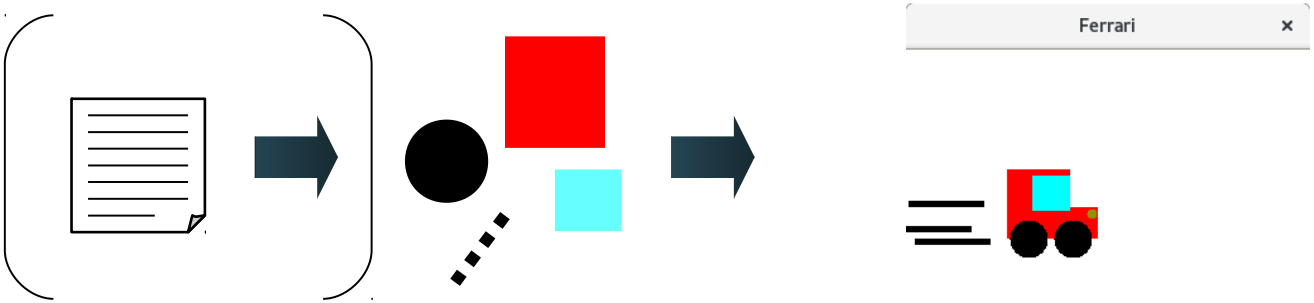
ソースコードを更新してもmakeが感知しない場合があるので、コードを書き換えても結果に変化がない場合は、一度 make clean して再コンパイルして下さい

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

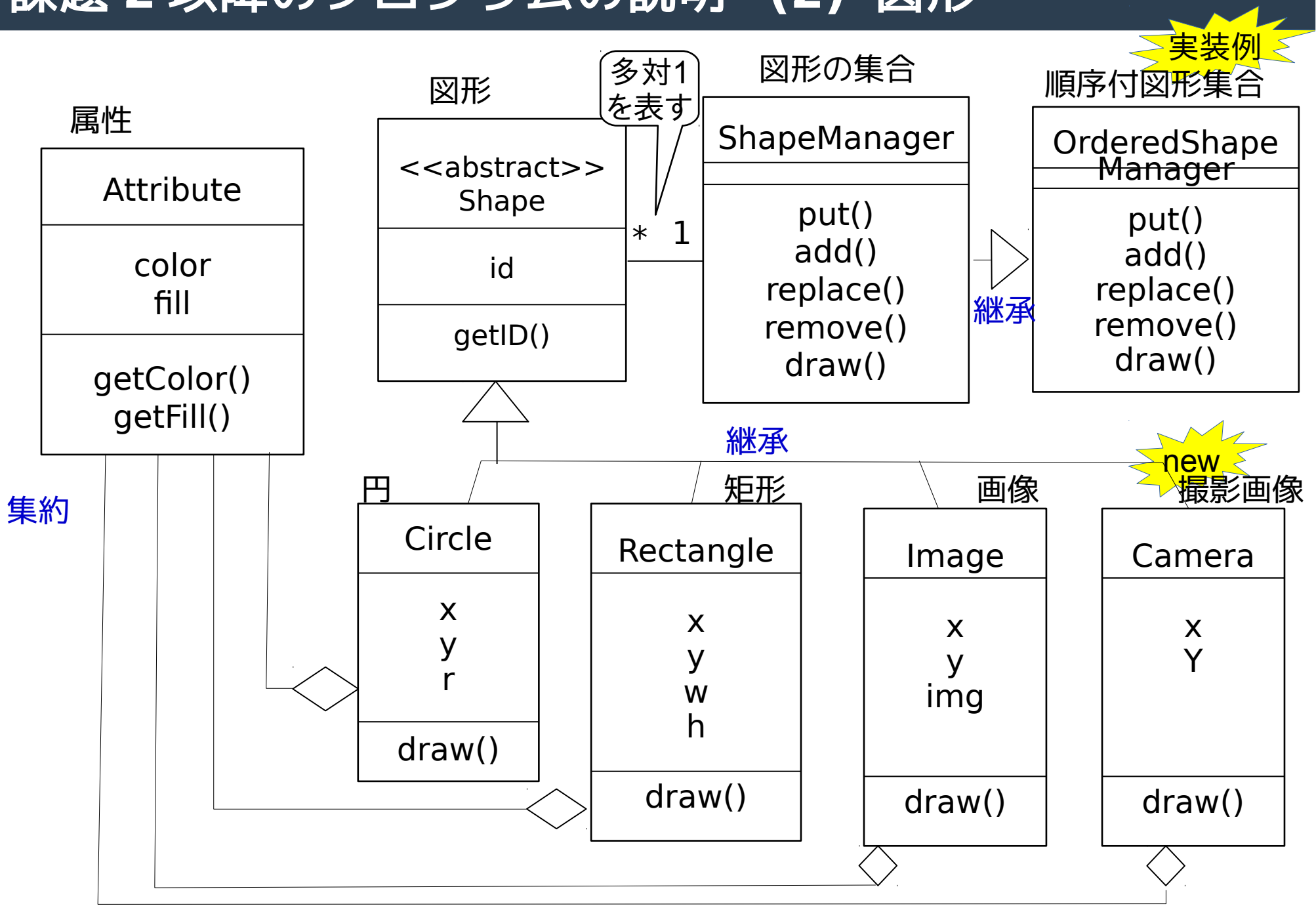
課題 2 以降のプログラムの説明 （1）概要

- プログラムは 5 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）
  - ウェブカメラ（ para.graphic.camera パッケージ）

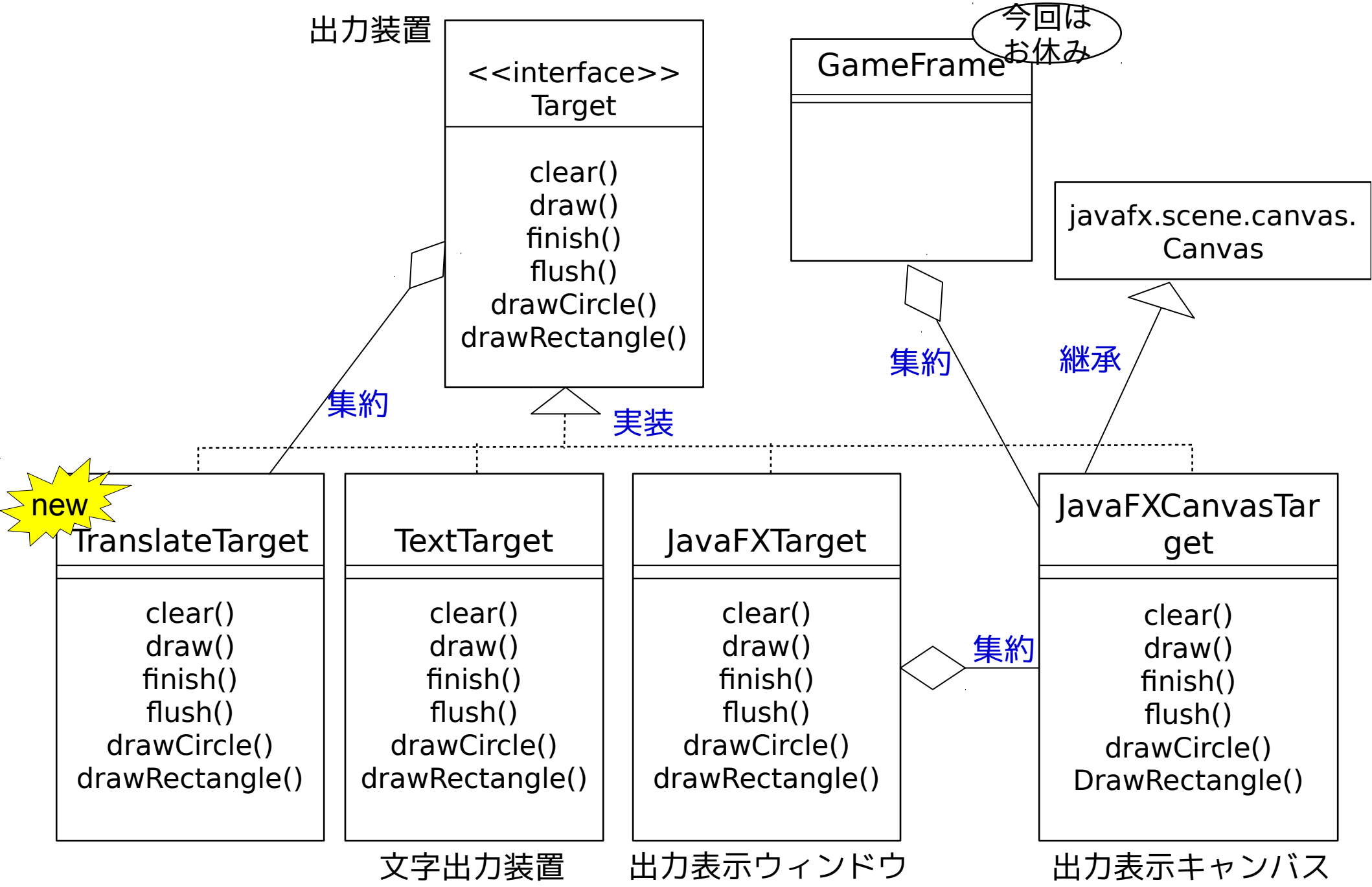


- 基本的流れ
  - メインプログラム内で図形オブジェクトを生成し，出力装置に対して出力する
  - 命令が書かれた文字列から構文解析器で図形オブジェクトを生成することもある

課題 2 以降のプログラムの説明 （2）図形

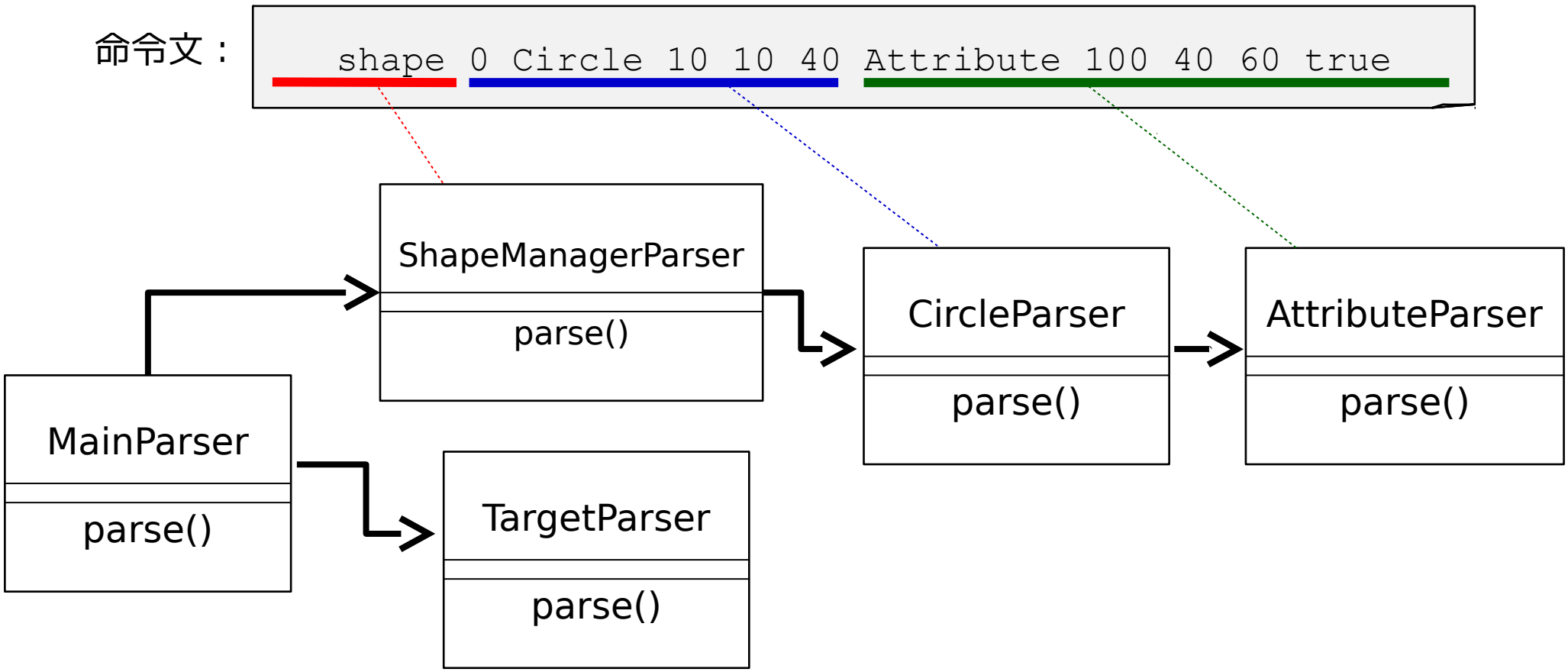


課題２以降のプログラムの説明 （3）出力装置



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

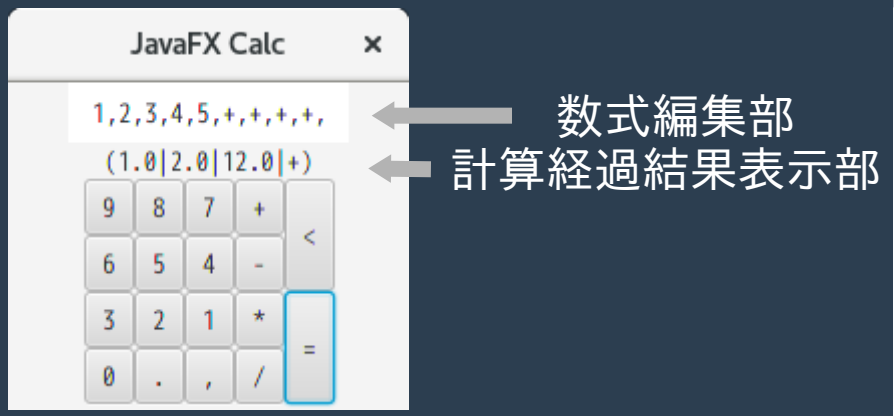


課題 0

- 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 1

第1回課題で登場した電卓プログラムを「式の評価途中の状態を結果表示部に逐次ゆっくり表示する。また計算途中であってもユーザが次の数式の入力を行える。」ように改造しよう。ただし、次の数式の処理は計算途中の数式処理が完全に終わって答えが出力された後に開始すること。



1.1)src/para/calc/Calculator.java 内の、
　　ex = new Executor1(); を ex = new Executor2(output);
に変更してみたところ、途中結果を表示することはできなかった。期待通りに動かない理由を説明せよ

hint Platform.runLater()は1.1の回答には関係ない

hint operation()を呼び出しているスレッド、writeState()を呼び出しているスレッドは何かを考える

1.2)”=”Buttonのクリックイベントを引き金としてスレッドを新たに起動するようにして期待通りに動くように改良せよ。また改良方針を文章で記述せよ

hint 計算の処理途中で次の計算処理が始まらないようにThread.join()を使う

1.3)改良後、javafx.application.Platform.runLater()を使わず、label.setText(state);を呼び出すとエラーが発生するはずである。そのエラー文を報告しなさい。また、runLater()メソッドの処理内容と、javafxにrunLater()が用意されている意味を考えて答えよ

課題 2

配布されたpara.Main07は反復サーバの実装であるが、これを改造して同時接続数を3まで許す並行サーバを作成せよ。

2.1)para.Main07はポート番号30000番へのTCP接続を受け付けるように待機している。次のようにtelnetコマンドで接続、送信、終了を行うことを一回と数え、それを4回行ってみよ。

```
% telnet localhost 30000
Trying 127.0.0.1...
Connected to localhost.
Escape character is '^]'.
reset
target clear
shape 10 Circle 160 120 100 Attribute Color 100 100 0 Fill false
shape   9 Circle 185 135 10 Attribute Color 255 100 160 Fill true
shape   8 Circle 120 100 30 Attribute Color 0 0 0  Fill true
shape   7 Circle 220 90 25 Attribute Color 0 0 0 Fill true
shape   6 Circle 128 100 21 Attribute Color 255 255 255 Fill true
shape   5 Circle 228 90 17 Attribute Color 255 255 255 Fill true
target draw
target flush
^]
telnet> close
Connection closed.
```

Ctrl キーと] キーの同時押し

課題 2

配布されたpara.Main07は反復サーバの実装であるが、これを改造して同時接続数を3まで許す並行サーバを作成せよ。途中で接続が切れた場合、新たな接続要求を処理できること。

2.2)改造前のプログラムで明示的に立ち上げられている2つのスレッドとその処理内容について説明せよ。

2.3)2.2で答えたスレッド間で共通して参照するデータに関して、どのような同期の配慮がされているかを答えよ。またその配慮は十分であるか、不十分であるかを検討し、十分であると考える場合はその理由を、不十分であると考える場合は問題となる具体的な状況を説明せよ。

2.4)同時に3接続を許す並行サーバをスレッドプールを使って実現せよ。
クライアントからの接続が切れて接続数が3未満になった場合には新たな接続要求を処理できること。
hint 接続が切れる順番は、接続開始の順番と同じとは限らないことに注意

課題 3

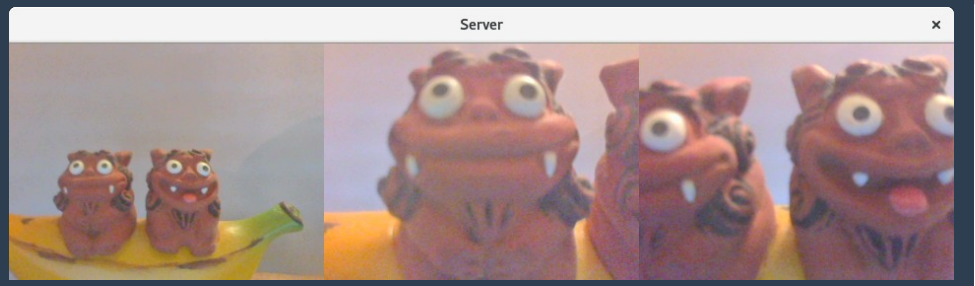
課題2で作成したプログラムを更に改造し、同時接続している全クライアントへサーバのウィンドウに表示される全データを送信するようにせよ。サーバのウィンドウには改造後も改造前と同様の表示がされること。
このプログラムはpara.Main09とすること。

3.1)改造を行い、その実装方針について説明せよ。特にスレッドをどのように割り当てたかとそのようにした理由を述べよ。

hint 開発中は、TextTargetのコンストラクタへソケットのストリームを入れるのではなく、System.outを使うと便利

課題 4

配布されたpara.Main08はCameraというShapeを継承した図形をサーバへ送る一方向通信をするプログラムである。これにサーバからの受信機能を追加しpara.Main09と送受信できるようにせよ。受信データの図形をウィンドウに表示するようにせよ。

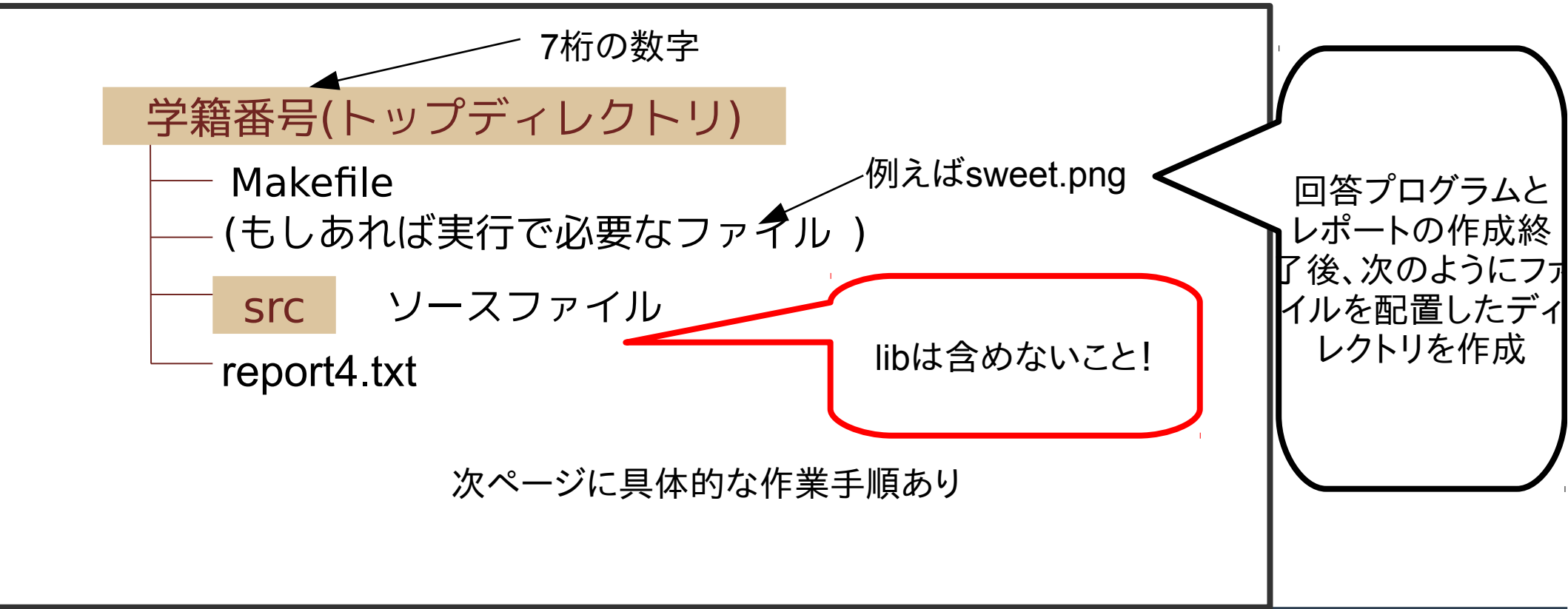


4.1)PrintStreamはOutputStreamなどの他のストリームと違い、IOExceptionを発生させない仕様である。そのため例外が発生したことの確認を逐次行う必要がある。どのように行うかを具体的に説明せよ。また、PrintStreamだけがこのように例外処理が異なる仕様である理由を調べて説明せよ。
hint コードを見よ

4.2)送信用と受信用を別々のスレッドとし、データを受信しそれを表示する機能を追加実装せよ。

提出方法 (1 of 3)

- para4.tar.gz を展開したディレクトリ構造を保ったまま，課題の変更作業を行う
- 各課題で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 から 4 の回答文、工夫点および感想を書いた report4.txt を用意する (雛形は課題のウェブページ)



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
```

今回はPara4

- 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 Makefile report4.txt もコピーする

```
cp トップディレクトリ/Makefile トップディレクトリ/report4.txt 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 ex4-1612345.zip -r dir
```

dir 以下の内容が圧縮され、 ex4-1612345.zip が作られます

- 圧縮後に内容を “ unzip ex4-1612345.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
  - 7 月 13 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