

並列プログラミング Parallel Programming

2022 2Q

演習 第1回

情報理工学院 情報工学系

計算機室

大岡山西地区

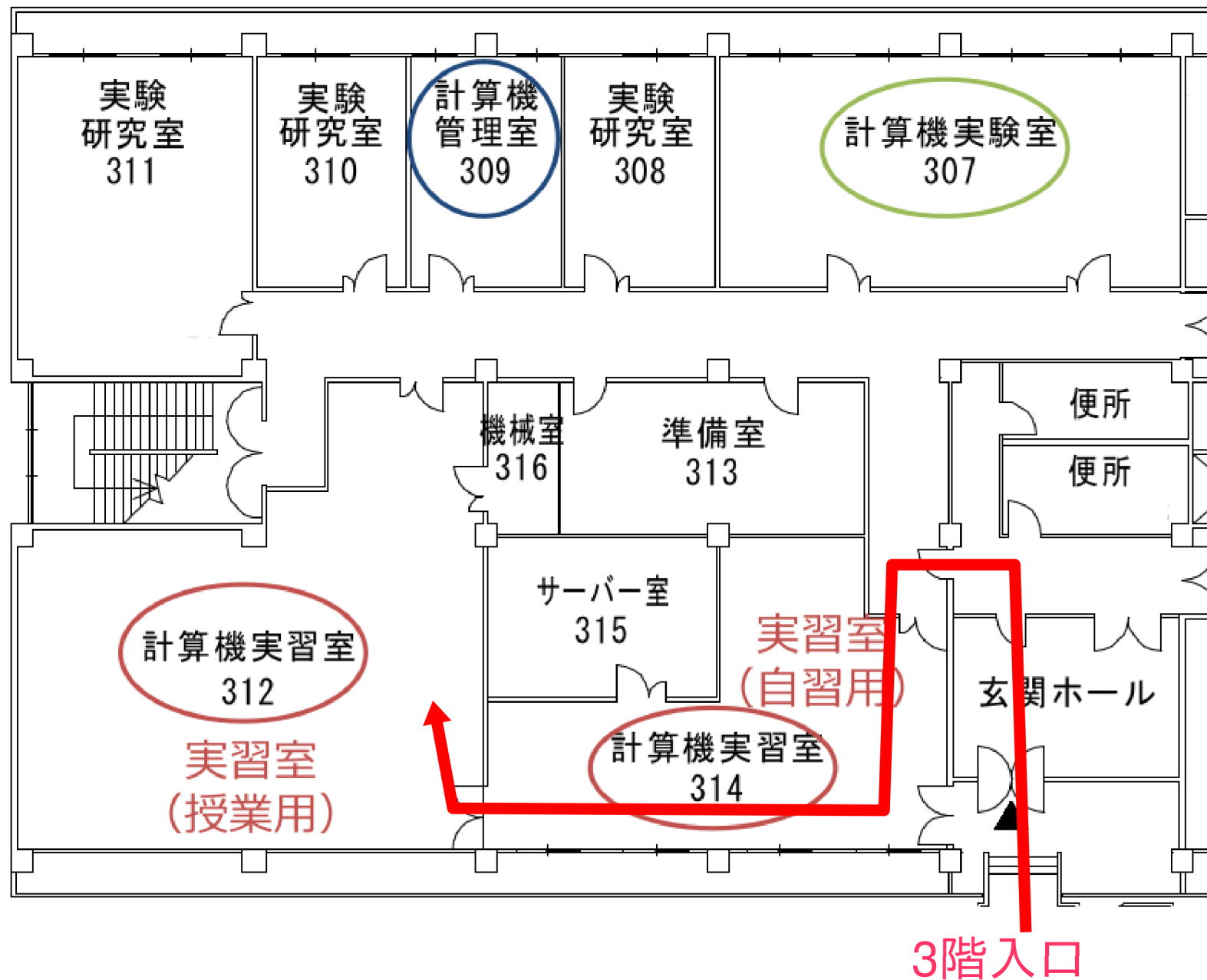
螺旋階段から
3階へ登る

大岡山東地区

正門



計算機室



本日の流れ

- 課題内容の説明
- 演習に取り組む

● 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

以下からダウンロードしてください
学内アクセス

● www.img.cs.titech.ac.jp/lecture/para/

The screenshot shows the Tokyo Tech Portal homepage. At the top, there is a header with the Tokyo Tech logo, the text "Tokyo Tech Portal", and the GSIC logo. Below the header, there is a "ログアウト" (Logout) button. On the left side, there is a "一般システム" (General System) section with three links: "Tokyo Tech Mail", "共通メール認証ID", and "学内ネットワークアクセス (SSL-VPN)". To the right of these links, there is a text box that says "Virtual Private Network 機能を使ってから、上のサイトにアクセスしてください" (Please use the Virtual Private Network function to access the site above). Below this text box, there is a red starburst icon with the word "注意" (Attention) and a callout box that says "VPNの同時利用者数は限られています 利用後はすぐにVPNを解除して他の人に譲りましょう" (The number of simultaneous VPN users is limited. After use, please immediately cancel the VPN and let others use it).

Tokyo Tech Portal

English

ログアウト

一般システム

[Tokyo Tech Mail](#)

[共通メール認証ID](#)

[学内ネットワークアクセス \(SSL-VPN\)](#)

Virtual Private Network
機能を使ってから、
上のサイトにアクセスして
ください

注意

VPNの同時利用者数は
限られています
利用後はすぐにVPNを解除
して他の人に譲りましょう

準備 (0)

● 今後の演習回共通

- 演習用作業ディレクトリの作成
- java17 インストール

adoptiumopenjdk17.0.3 <https://adoptium.net/>

- JavaFX17 以降の sdk をインストール

JavaFX18.0.1 <https://gluonhq.com/products/javafx/>

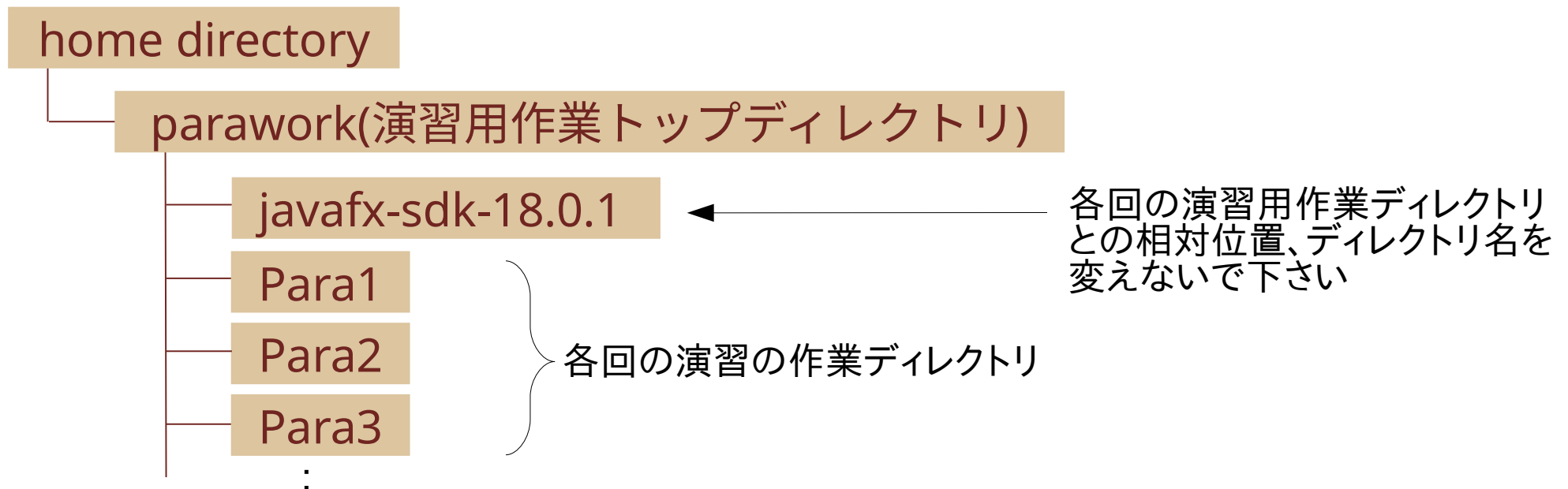
注意

展開場所は演習用作業ディレクトリの直下

注意

OS, CPUアーキテクチャを正しく選択

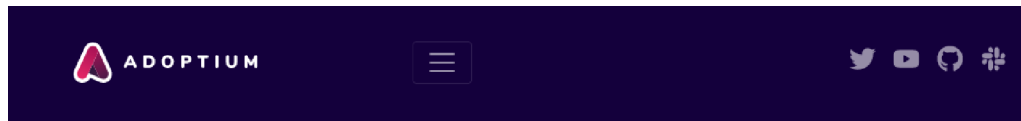
● 解凍後のディレクトリ



準備 (0.1)

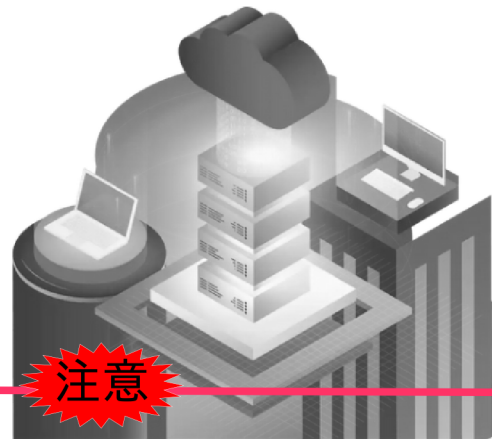
🍌 adoptopenjdk17.0.3 のインストール

<https://adoptium.net/>



Prebuilt
OpenJDK
Binaries for
Free!

Java™ is the world's leading programming language and platform. The Adoptium Working Group promotes and supports high-quality, TCK certified runtimes and associated technology for use across the Java™ ecosystem. Eclipse Temurin is the name of the OpenJDK distribution from Adoptium.



Download Temurin
for Linux x64



注意

OS CPUアーキテクチャの自動判定結果

正しい場合はここから
ダウンロード

正しくない場合はここから
自分の環境に合わせて
jdkを選択をする

準備 (0.2)

● adoptopenjdk17.0.3 のインストール先

- javac, java コマンドのインストールされるディレクトリは PATH に自動的に追加されるので、気にしなくても良いですが、複数の java 開発環境インストールしている場合は注意
 - Linux(debian,ubuntu) の場合 は PATH に自動登録されません
< 展開したディレクトリ >/jdk-17.0.3+7/bin
 - MacOS の場合
/Library/Java/JavaVirtualMachines/temurin-17.jdk/Contents/Home/bin
 - Windows の場合
C:\Program Files\Eclipse Adoptium\jdk-17.0.3.7-hotspot\bin

java のバージョンを表示させて AdoptiumOpenJDK が選択されていることを確認


```
%java -version
openjdk version "17.0.3" 2022-04-19
OpenJDK Runtime Environment Temurin-17.0.3+7 (build 17.0.3+7)
OpenJDK 64-Bit Server VM Temurin-17.0.3+7 (build 17.0.3+7, mixed mode, sharing)
```

準備 (0.3)

● javafx のインストール

<https://gluonhq.com/products/javafx>

- jmods と sdk があるが
必ず sdk を選択
- OS と CPU アーキテクチャを選んでダウンロード
 - Linux CPU アーキテクチャ別に
aarch64, arm32, x64
 - MacOS CPU アーキテクチャ別に
aarch64, x64
 - Windows CPU アーキテクチャ別に
x64, x86

Products ▾ Developers Pricing Services Insights ▾ Contact ▾

Home ▸ Products ▸ JavaFX

JavaFX

Roadmap

Release	GA Date	Latest version	Long Term Support	Extended or custom support
19	September 2022 (planned)	early access	no	
18	March 2022	18.0.1 (April 2022)	no	upon request
17	September 2021	17.0.3 (April 2022)	until September 2026	upon request
16	March 2021	16 (March 2021)	no	upon request
15	September 2020	15.0.1 (October 2020)	no	upon request
14	March 2020	14.0.2.1 (July 2020)	no	upon request
13	September 2019	13.0.2 (January 2020)	no	upon request
12	March 2019	12.0.2 (July 2019)	no	upon request
11	September 2018	11.0.15 (April 2022)	until September 2023	upon request

- All dates on this page are informative. [Contact us](#) for a personalised quote for our [JavaFX Long Term Support \(LTS\) service](#).
- Releases in grey don't receive updates anymore. We strongly encourage all our users to use either the latest version (currently 17) or the latest the LTS releases (currently 11 and 17).
- The JavaFX runtime is available as a platform-specific SDK, as a number of jmods, and as a set of artifacts in [maven central](#).
- The OpenJFX page at [openjfx.io](#) is a great starting place to learn more about JavaFX.
- This software is licensed under [GPL v2 + Classpath](#)

Downloads

JavaFX version: 18.0.1 ▾ Operating System: [any] ▾ Architecture: [any] ▾ Type: [any] ▾

☐ Include older versions

OS	Version	Architecture	Type	Download
Linux	18.0.1	aarch64	SDK	Download SHA256
Linux	18.0.1	aarch64	jmods	Download SHA256
Linux	18.0.1	arm32	SDK	Download SHA256
Linux	18.0.1	x64	SDK	Download SHA256
Linux	18.0.1	x64	jmods	Download SHA256
macOS	18.0.1	aarch64	SDK	Download SHA256
macOS	18.0.1	aarch64	jmods	Download SHA256
macOS	18.0.1	x64	SDK	Download SHA256
macOS	18.0.1	x64	jmods	Download SHA256
Windows	18.0.1	x64	SDK	Download SHA256
Windows	18.0.1	x64	jmods	Download SHA256
Windows	18.0.1	x86	SDK	Download SHA256
Windows	18.0.1	x86	jmods	Download SHA256
Javadoc	18.0.1		Javadoc	Download SHA256

準備 (0.4)

● make コマンドのインストール

- linux (debian, ubuntu)

`sudo apt-get update`

`sudo apt-get install make`

- mac

`xcode-select --install`

- windows

Complete package, except sources をダウンロードして実行

<http://gnuwin32.sourceforge.net/packages/make.htm>

コマンドを探すディレクトリを指定する環境変数に追記

PowerShellの起動毎に環境変数に追記するのが面倒なら、設定ファイルに記述
<https://stackoverflow.com/questions/12143245/powershell-configuration-with-an-rc-like-file>

準備 (1)

- ダウンロードした **para1.zip** を展開する

```
unzip para1.zip
```

- 解凍後のディレクトリ

home directory

parawork(演習用作業トップディレクトリ)

javafx-sdk-18.0.1

Para1

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) (今回は空)

準備 (2)

- **Para1 以下の ._* ファイルを再帰的に消去**

```
cd Para1  
make cleanall
```

- **MacOS では、 ._ で始まるファイルが作成されることがありますが、コンパイル時の障害になるので消去**

準備 (3)

● 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 Para1 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 --module-path ../javafx-sdk-18.0.1/lib --  
add-modules javafx.controls,javafx.swing -sourcepath src -classpath  
lib/*: src/para/paint/Paint.java
```

として下さい

Windowsでは:を;に置換

実際は一行で
書く

-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
--module-path	モジュールファイルがあるディレクトリを コンパイラに教える
--add-modules	依存するモジュールを列挙する
-classpath	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

※ <https://docs.oracle.com/javase/jp/17/docs/specs/man/javac.html>
でその他のオプションを確認すること

準備 (4)

● 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 Para1 ディレクトリにて

実際は一行で
書く

```
java  
--module-path ../javafx-sdk-18.0.1/lib --add-modules  
javafx.controls,javafx.swing -cp bin:lib/*: para.paint.Paint
```

Windowsでは:を;に置換

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）

--module-path javacのオプションと同じ役割

--add-modules javacのオプションと同じ役割

-cp bin:lib/*:

Windowsでは:を;に置換

-cp は -classpathの短縮形

実行に必要なコンパイル済み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(jarファイル)がlib/に置かれていることを java コマンドに教える

※<https://docs.oracle.com/javase/jp/17/docs/specs/man/java.html>
でその他のオプションを確認すること

準備 (5)

- 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;
```

HTML文書の出力先
ディレクトリ

実際は一行で
書く

今回の演習では、Para1 直下で、

```
javadoc -html5 -charset utf-8 -encoding UTF-8
-d javadoc -sourcepath src --module-path ../javafx-sdk-18.0.1/lib
--add-modules javafx.controls,javafx.swing
-link https://docs.oracle.com/javase/jp/17/docs/api
-link https://openjfx.io/javadoc/18
-package para.paint para.calc
```

パッケージ名
としてください

外部javadocのURI
リンクを作成する
には必要

※<https://docs.oracle.com/javase/jp/17/docs/specs/man/javadoc.html>
でその他のオプションを確認すること

準備 (6)

- 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 Paint	Paintについてコンパイルして、実行
make Calculator	Calculatorについてコンパイルして、実行
make clean	bin 以下のクラスファイルをすべて削除
make doc	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\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

準備 (6.1)

● Makefile の編集

```
#####
## if you want to specify a directory java installed explicitly,
## set the java binary directory here
#####
JAVABIN=
# JAVABIN=$(HOME)/parawork/jdk-17.0.3/bin/
# JAVABIN=/usr/lib/jvm/default-java/bin

#####
## set javafx-sdk directory your installed
#####
JAVAFXMODULE=../javafx-sdk-18.0.1/lib

#####
## set proxy server address and port number, if your machine connects
## to the internet via proxy
JAVADOCPROXY=
# JAVADOCPROXY=-J-Dhttp.proxyHost=proxy.csc.titech.ac.jp -J-Dhttp.proxyPort=8080

#####
## csc room setting
#####
#JAVABIN=
#JAVAFXMODULE=/Library/Java/JavaFX/javafx-sdk-17.0.2/lib

# OS-dependent commands and separator
ifeq ($(OS),Windows_NT)
    SEP=;
    RM=rd /s /q
    FIND=echo
else
    SEP=:
    RM=rm -rf
```

5行目
定数 JAVABINにjavacコマンドがある
ディレクトリを絶対パスで設定する必要
がある場合もある

必要ならばインストールディレクトリを準備(0.2)を参考に
して記述

JAVABIN=/Library/Java/JavaVirtualMachines/
temurin-17.jdk/Contents/Home/bin

課題 1

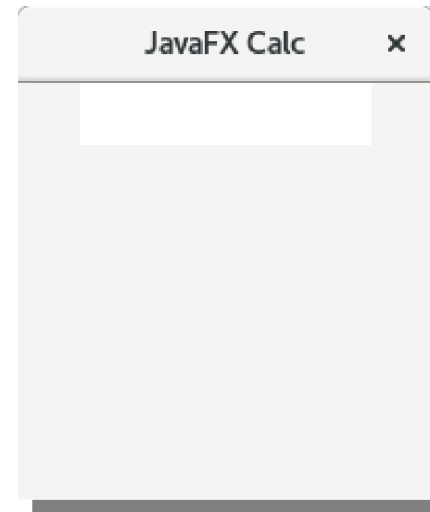
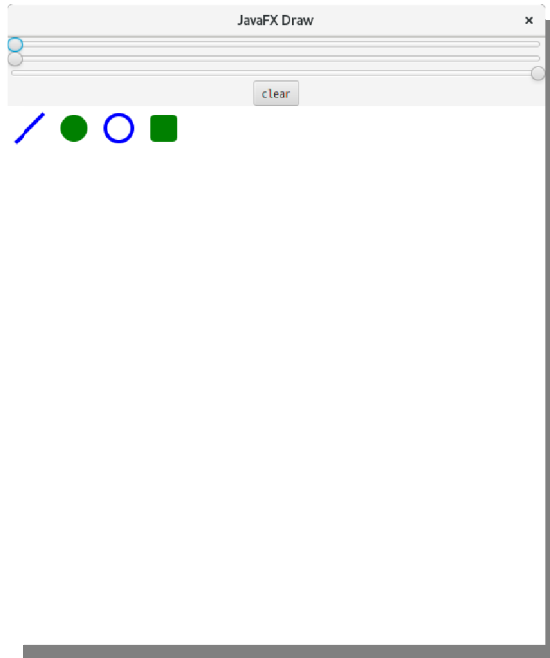
Javadoc を読んでみる



- 1.1) <https://openjfx.io/javadoc/18/> をブックマークせよ
- 1.2) `javafx.scene.control.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
- 1.5) 配付パッケージ `calc` に含まれるクラス、インタフェースの継承関係を調べよ

課題 2

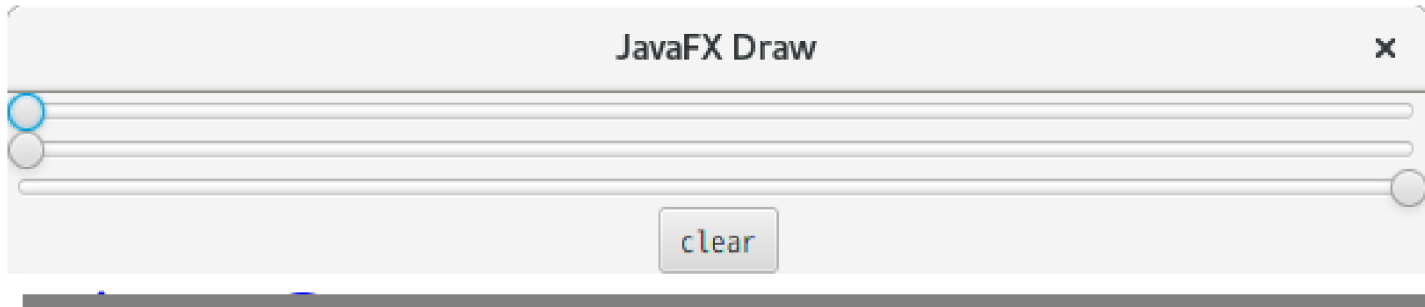
- 説明されたコンパイルの仕方に従って、`para.paint.Paint` と `para.calc.Calculator` をコンパイルして実行する



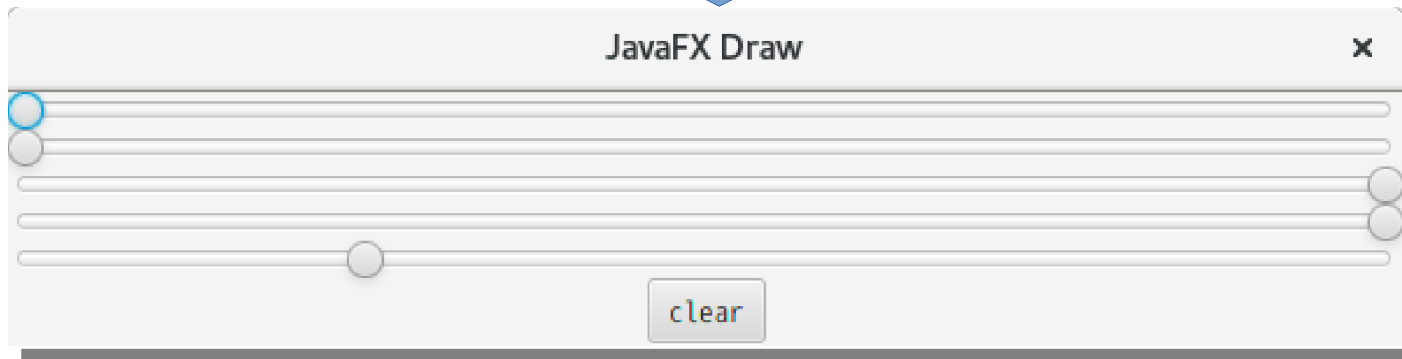
- 2.1) 作成されたclass ファイルが置かれたディレクトリ位置を、Para1からの相対パスで説明せよ。
- 2.2) ソースコード内のpackageで指定されたパッケージ名と、javac と java に与えたオプションと、2.1で答えたディレクトリ位置との関係を説明せよ。
- 2.3) ソースコード内のimport文で指定されたクラスが、正しくimport できるためには、そのクラスはどのディレクトリにある必要があるかを説明せよ。

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

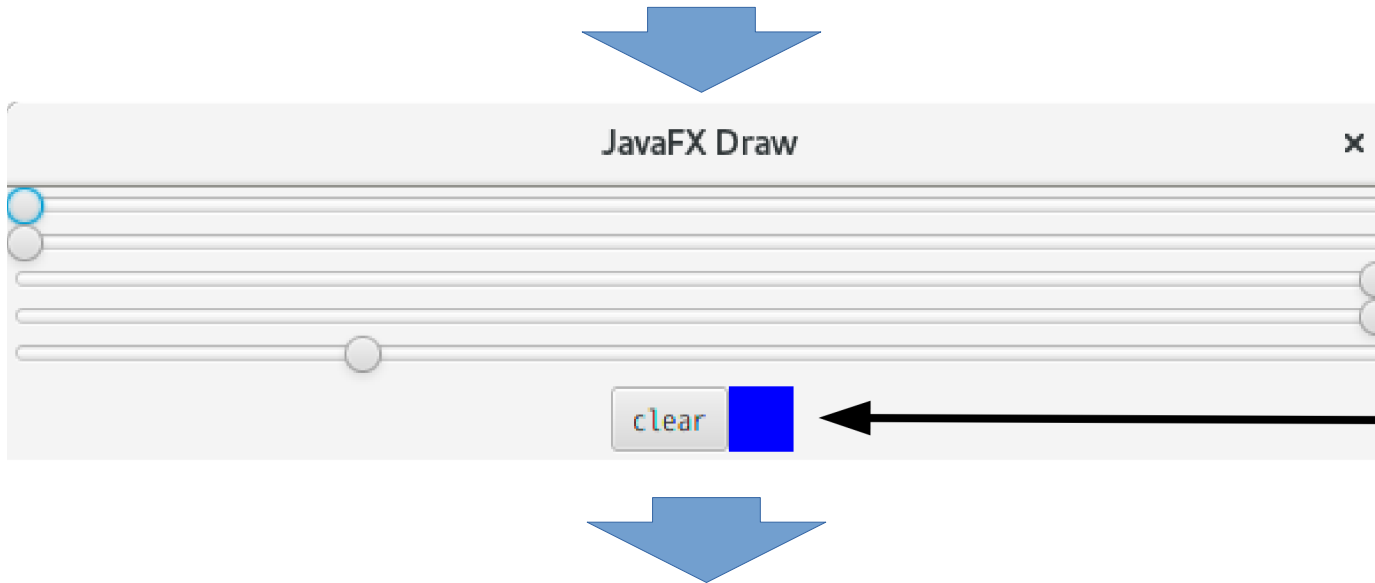


3.2) スライダーを2つ追加し、追加の1つ目は描画線の透明度を設定するスライダー、追加の2つ目は線幅を設定するスライダーとして機能するようにイベントハンドラを登録せよ

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

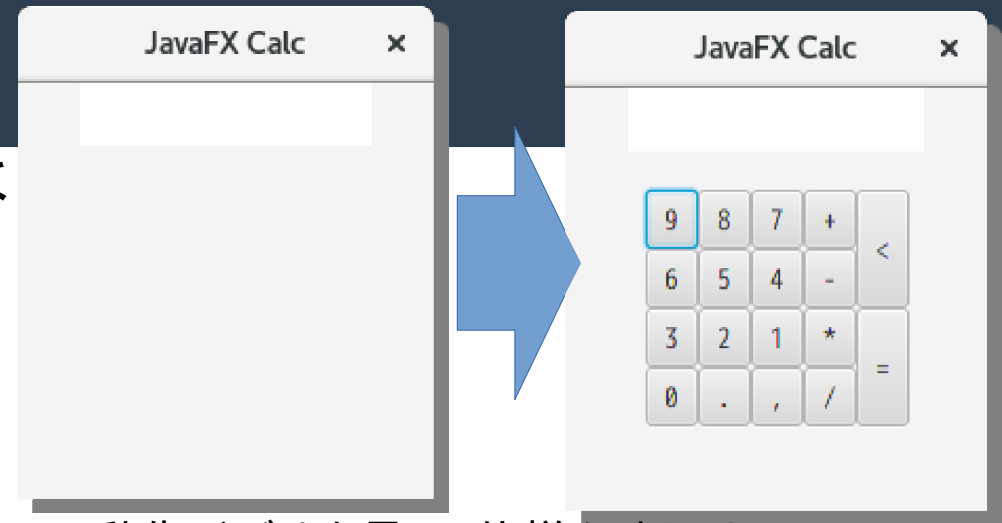
- 例えば、スタンプ機能の追加
- 例えば、画像の90度回転をさせるボタンの追加
- 例えば、一つ前の描画操作の取り消し機能の追加
- 例えば、画像ファイルの書き出し昨日の追加
- 例えば、画像ファイル読み込み機能の追加
- 例えば、レイヤー機能の追加

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



動作ビデオを見て、仕様を確認すること

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、イベント発生源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.zip 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
```

- ソースディレクトリの階層を再帰的にコピーする

```
cp -R Para1/src dir
```

- *dir* に report1.txt もコピーする

```
cp 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 Para1/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 ex1-2012345.zip -r dir
```

- *dir* 以下の内容が圧縮され、**ex1-2012345.zip** が作られます

- 圧縮後に内容を“unzip ex1-2012345.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 T2SCHOLA にアップロードする
- 締め切り
 - 6 月 23 日 (木) 23:59 (JST)