2022年度 プログラミング演習1
後半 (Java プログラミング)

情報理工学部 実世界情報コース

目 次

1　Java プログラミングをはじめるにあたって

1.1　おさらい：プログラミングとは

デジタル大辞泉（小学館）によると,「プログラム」という言葉は「コンピューターへ指示する，計算や仕事の手順を特定の言語や形式で書いたもの．また，それを作ること.」を意味する．

コンピュータはその構造上,「0」と「1」の 2 値で動作している．したがって，コンピュータは人間の言葉を直接は理解できず，コンピュータが分かる言葉である機械語で一連の指示を記述せねば，コンピュータへ仕事を命令することができない．しかしながら，我々のような普通の人間が機械語を理解・記述することは相当に困難である．

英語や日本語をはじめとする人間の言語をそのまま機械語に翻訳できれば良いが，残念ながら我々が日常で使っている言語には曖昧さ，行間といったものがあり，現代の技術では人間の言語を直接機械語に翻訳することも難しい．

そこで，人間が理解しやすく，かつ，機械語にも変換しやすい，人間と機械語の中間の言語である「プログラミング言語」を我々は学ぶのであった．これまでに提案されているプログラミング言語は 200 を超えるが，よく使われている言語は限られる．C/C++, Java, Python, Ruby, JavaScript, PHP, Swift などは良く使われる言語の一例である．中でも，この演習科目では Java, Python を学ぶのであった．他のコースでは C 言語を学んでいるところもある．

1.2　Java と Python の大きな違い

プログラミング言語間にはいくつかの違いがあり，向き不向きがある．利用するプログラミング言語は最終的に作りたい製品にあわせて選定することが，多くの場合で理想的であろう．

本講義ではメジャーな言語である Python, Java を取り上げている．皆さんが最初に直面する言語間の特徴は,「どのタイミングでプログラミング言語から機械語への翻訳が行われるか」ということである．

Python は言語仕様としてインタプリタ上で実行されることを前提としている．すなわち，プログラムを実行する際には，プログラミング言語を「逐次」解釈し，機械語に翻訳しながらプログラムは実行される．皆さんが Jupyter Notebook で記述したプログラムを「実行」ボタンで動作させていたが，その裏では 1 単位命令群ごとに逐次翻訳し，プログラムが実行されていたのである．

一方で，Java は言語仕様として,「事前に全て」プログラミング言語を機械語に翻訳してから，プログラムの実行を開始する特徴がある．この翻訳作業のことを「コンパイル」とよぶ．

「コンパイル」という操作を行わねばそもそものプログラムの実行すらできないため，みなさんによる Java プログラミング入門は，この「コンパイル」という操作を身に着けることから始まる．

1.3　web ブラウザ上で Java プログラムを実行できるサービス

Java は基本的にはコンパイル向けの言語だが，ブラウザ上で実行するサービスを提供している web ページがある．以下にそのいくつかを示す．

- Online GDB `https://www.onlinegdb.com/online_java_compiler` インタラクティブなキーボード入力，実行時引数指定可能
- paiza.io `https://paiza.io/ja/projects/new` 事前のキーボード入力可能，実行時引数指定不可
- www.compilejava.net `https://www.compilejava.net/` 実行時引数指定可能

1.4 Java 開発環境の構築法

macOS および Windows における Java 開発環境の構築方法を Panopto に動画としてまとめた。一部音声が切れているものがあるが、仕様である。なお、この動画では、JDK（Java Development Kit, Java 開発環境）として、オープンソースの Eclipse Temurin をインストールするが、そのほかの JDK を用いても構わない。

macOS への JDK のインストール　macOS 環境への JDK のインストール方法を動画で説明する。macOS を使っているものは参考にされたい。

Windows への JDK のインストール　Windows 環境への JDK のインストール方法を動画で説明する。Windows を使っているものは参考にされたい。

VS Code を用いた開発環境の構築　macOS にせよ Windows にせよ、コマンドラインを用いて Java プログラミングを行うのは非効率である。よく使われているエディタである Visual Studio Code[1]に Java 開発のためのプラグインを導入して、開発環境を構築する方法を動画で説明する。

1.5 RAINBOW 環境での Java プログラムの実行方法

プログラムの記述，コンパイル，実行の 3 ステップにて，Java で記述されたプログラムを実行しよう．

1.5.1 プログラムの記述

情報処理演習室のパソコンは，セキュリティ上の問題からどのフォルダに対しても自由にアクセスできるとは限らない．この実験では「D:\Temp」フォルダで作業することにしよう．図 1 で示す通り，デスクトップフォルダにある「一時保存用」ショートカットをダブルクリックする．

D:\Temp フォルダが開かれたことを確認したら，図 2 で示す通り，D:\Temp フォルダ内の適当な場所で，右クリックしてメニューを開き，**新規作成 (X)** →**テキストドキュメント**，の順番で操作を行い，空のテキストファイルを D:\Temp フォルダ内に作成する．

図 3 で示す通り，新しく作成したテキストファイルを選択し，右クリックして**名前の変更**と操作するか，「F2 キーを押す」操作を行うことで，ファイル名を Hello.java に変更する．

メモ帳や sakura エディタなどで Hello.java を開き，以下のプログラムをそのまま記述しよう．

```
class Hello {
    public static void main(String[] args) {
        System.out.println("Hello.");
    }
}
```

これで，パソコン上に皆さんが記述したプログラムを保存することができた．

コメント 1：上記プログラムは教科書に掲載されているリスト 2-01 のプログラムそのものである．

コメント 2：一般的なエディタは，エディタを開いた後，エディタの編集画面上に，編集したいファイルをドラッグ＆ドロップすることで，当該ファイルが編集できるようになることがほとんどである．

[1] https://code.visualstudio.com

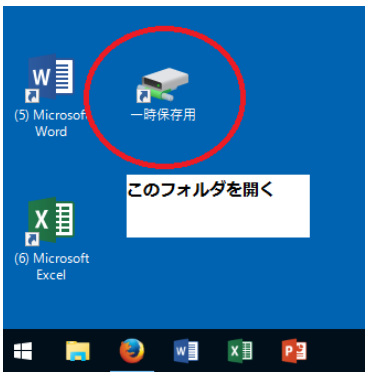

図 1: 一時保存用と書かれたショートカットを開く

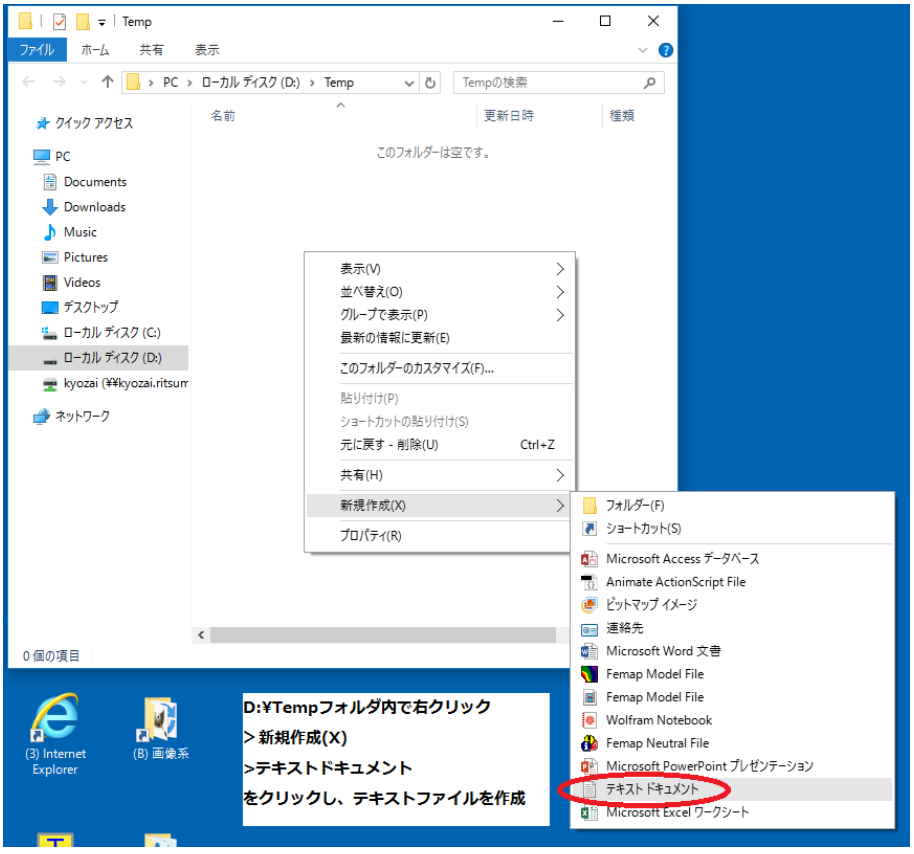

図 2: 空のテキストファイルを作成

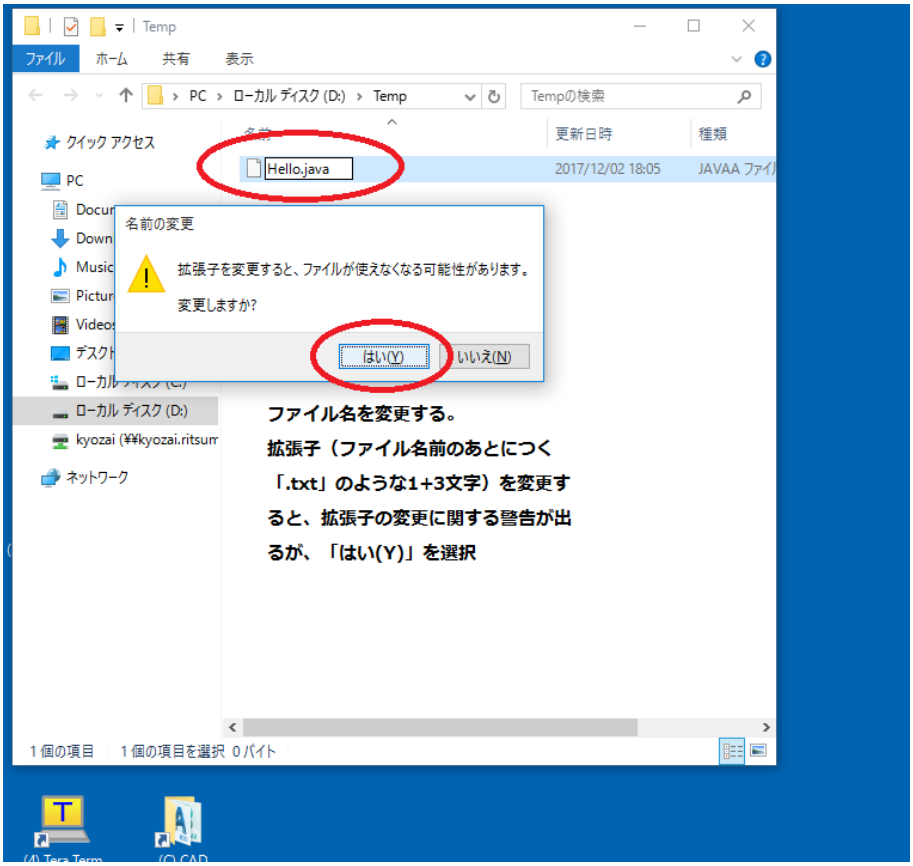
Temp
ファイル
ホーム
共有
表示
PC › ローカル ディスク (D:) › Temp
Tempの検索
クイック アクセス
PC
ローカル ディスク (D:)
kyozai (¥¥kyozai.ritsum
ネットワーク
名前
更新日時
種類
Hello.java
2017/12/02 18:05
JAVAA ファイ
名前の変更
拡張子を変更すると、ファイルが使えなくなる可能性があります。
変更しますか?
はい(Y)
いいえ(N)
ファイル名を変更する。
拡張子（ファイル名前のあとにつく
「.txt」のような1+3文字）を変更す
ると、拡張子の変更に関する警告が出
るが、「はい(Y)」を選択
1 個の項目
1 個の項目を選択 0 バイト

図 3: ファイル名を変更

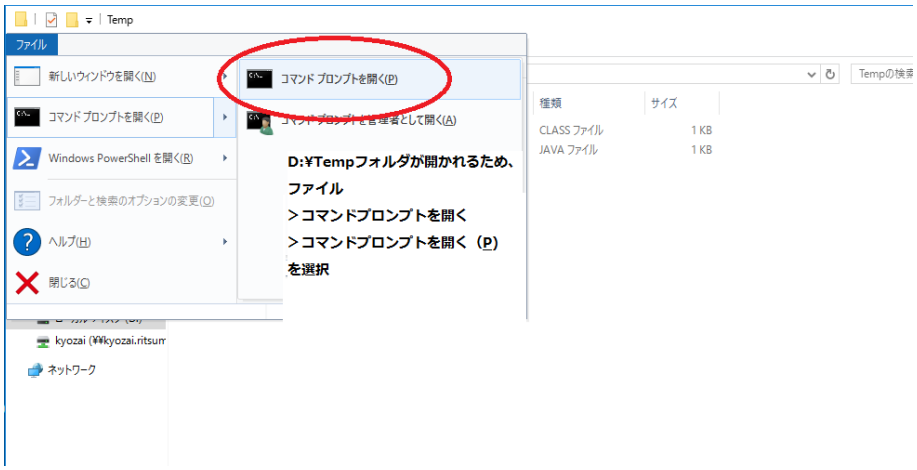

図 4: コマンドプロンプトの開き方

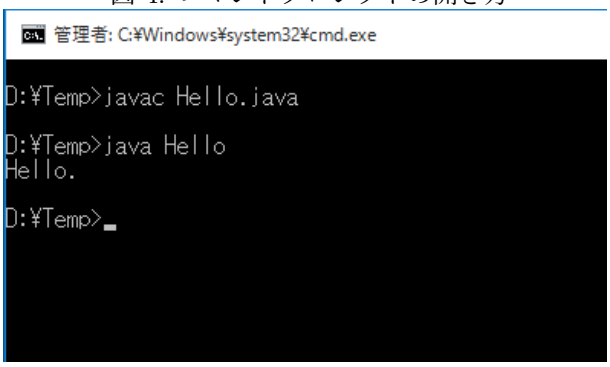

図 5: Hello.java のコンパイル・実行例

Sublime Text, Atom, 秀丸, Emacs, Vim, Notepad++など，世の中にはエディタがたくさんリリースされている．どのエディタを使うかは人により好みが分かれる．自分にあった便利な機能をたくさん組み込んだ，快適な開発環境を構築することも，プログラマーの実力のうちである．

1.5.2 コンパイル

D:\Temp フォルダの左上に「ファイル」というメニューが確認できるだろう．図 4 で示す通り,「**ファイル→コマンドプロンプトを開く（P）→コマンドプロンプトを開く（P）**」と操作して，コマンドプロンプトを開く．コマンドプロンプトでは，キーボードで命令を打ち込むことにより，パソコンに直接命令（コマンド）を与えることができる．

コマンドプロンプトが開けたら，以下のコマンドを実行しよう．

```
javac Hello.java
```

下記間違いがなければ正常にコンパイルが完了する．コンパイルが正常に完了した場合はメッセージは表示されない．

コメント：空ファイルをコンパイルしようとした場合も，メッセージが表示されないことに注意．

1.5.3 実行

コマンドプロンプトで以下のコマンドを実行しよう．

```
java Hello
```

コンパイルをするときは javac だったが実行するときには java となることに注意しよう．ファイル名の拡張子部分「.java」も不要である．図 5 で示す通り，Hello. と表示されたらあなたも java プログラマの仲間入りである．

2 よくハマる注意点

2.1 クラス名とファイル名

クラス名とファイル名は一致していなければ適切にコンパイルが実行されない．例えば，以下のコードを考える．

```
class hello {
    public static void main(String[] args) {
        System.out.println("Hello.");
    }
}
```

このコードの 1 行目 class hello { 内にある hello がクラス名にあたる．クラス名が hello であるため，上記ソースのファイル名は hello.java でなければならない．

クラス名とファイル名が違う場合でもコンパイルは問題なく実行されるが，実行時にエラーが発生することになる．

2.2 大文字と小文字

ほとんどのプログラミング言語では大文字と小文字を区別する．Java も例外なく，ファイル名やクラス名において大文字と小文字は別の文字として処理される．

例えば，以下のプログラムを考える．

```
/* AbCdEf.java */
class AbCdEf {
    public static void main(String[] args) {
        System.out.println("Hello.");
    }
}
```

上記プログラムは javac abcdef.java というコマンドではコンパイルできない．javac AbCdEf.java というコマンドを使う必要がある．

3　小テスト範囲

小テストの出題項目と，それに相当する内容を記載している教科書**本格学習 Java 入門**のページは以下の通りである．

11 週目 (week11 小テスト)

出題項目 Java の特徴と利用方法, Java のプログラムの書き方, 型, リテラル, 変数, 変数のスコープ, 変数の初期化, 演算子 (四則演算や論理演算，ビット演算を含む)

教科書の範囲 [改訂新版] の p.1 - p.53, p.72 - p.102，[改訂 3 版] の p.11 - p.58, p.77 - p.108

12 週目 (week12 小テスト)

出題項目 配列の宣言と初期化，配列への要素の代入，条件分岐，繰り返し

教科書の範囲 [改訂新版] の p.54-p.70, p.103-p.136，[改訂 3 版] の p.59-p.73, p.109-144

13 週目 (week13 小テスト)

出題項目 メソッド，クラス，コンストラクタ，継承，関数の多重定義

教科書の範囲 [改訂新版] の p.139-p.186，[改訂 3 版] の p.145-p.202

14 週目 (week14 小テスト)

出題項目 例外，型変換，データの入出力

教科書の範囲 [改訂新版] の p.187-p.206 [改訂 3 版] の p.203-p.222

4　第 11 週課題

4.1　課題 11-1(教員と一緒にやろう)

学習教材動画「Java 入門」の**「3. 基本操作法について」**にある『一番簡単なプログラム「HelloWorld」』(https://jp.linkedin.com/learning/learning-java/528438) の通りに入力し，Hello World. が表示されることを確認せよ．作成した HelloWorld.java ファイルをそのまま manaba+R に提出せよ．

4.2　課題 11-2

課題 1-3 と同様のコードを Java で実装せよ．作成した.java ファイルを manaba+R に提出せよ．ただし，ファイル名は kadai11_2.java とする．

課題 1-3

変数 a,b,c,d の値がそれぞれ 2,3,4,5 の場合，計算結果が 10 になる計算式を考えよ．このとき、すべての文字 a,b,c,d を用いること.（四則演算（+,-,*,/)，小数点以下を切り捨てる割り算の演算子，累乗の演算子を使えばできるだろう.）

変数 a,b,c,d にそれぞれ 2,3,4,5 を代入した上で，自分で考えた計算式を実行するコードを記述・実行し，確かに計算結果が 10 になっていることを確認せよ.

4.3　課題 11-3

Python でも提示していた実行されない行を表現するコメントであるが，Java でも表現することができる．1 行だけのコメントを表記したい場合と複数コメントにしたい場合で書き方が異なる．

```
//1 行だけがコメント

/* 「/*」と「*/」に囲まれた範囲すべてがコメントとして処理される。
ここもコメント
ここもコメント
*/
```

さらに Java ではプログラムファイル内にプログラム本体以外のもの，例えばプログラムの仕様や著者名などの付加情報をコメントとして記述する Javadoc という仕組みがある．Javadoc の書式に従って付加情報を記述しておくとそのプログラムのドキュメント (HTML 形式) を自動生成することができ，個別に手動でドキュメントを作成する手間を省くことができる．Javadoc では情報の種類を表す「タグ」が定義されており，各タグについての情報を続ける形で付加情報を記述する．いくつかのタグの例とそれぞれの意味を下記に示す.

```
/** 「/**」と「*/」に囲まれた範囲すべてが Javadoc として処理される。
 * @author 著者名
 * @param パラメータに関する情報
 * @return メソッドの戻り値と範囲
 * @see プログラムに関連するキーワード
 * @version プログラムの現在のバージョン
 * @since 当該クラスまたはメソッドの導入されたバージョン
 */
```

課題 11-2 のコードに@author, @see, @since, @version タグを少なくとも含んだ Javadoc の主要タグをコメント文にて追加せよ．それぞれのタグには常識で考えた内容を記述すること（採点上はタグに何を記述しても正答とする）．

また可能であれば javadoc を実行し，作成したプログラムのマニュアルを作成せよ．その場合.java ファイルではなく，Javadoc で作成した html ファイルを提出せよ．例えば，java ファイルが kadai11_3.java ならば kadai11_3.html を提出せよ．間違ったファイルを提出した場合不正解として採点する．

コメント１ タグを指定する際には/**と*/でくくらなければならないことに注意せよ．/*と*/では javadoc コマンドから無視されてしまうのである．教科書記載の例とよく比較せよ（「技術評論社」のリンクの表示の有無がわかりやすい）．

コメント２ 文字化けしているようであれば，ブラウザのエンコードを変更すれば正しい表示に切り替わる．例えば，Firefox ブラウザであれば，メニューの**その他**を開き**テキストエンコーディング**から **Unicode** もしくは**日本語**を選択すれば多くのパソコンで解決するであろう．

4.4 課題 11-4

Java ではキーボードからの入力機能を実現することは少しだけ面倒である．以下のコードを実行してみよう．ただし，ファイル名は kadai11_4.java とする．

```
/* kadai11_4.java*/
import java.util.Scanner;

class kadai11_4 {
  public static void main(String[] args) {
    System.out.println("整数を入力してください");
    java.util.Scanner sin = new java.util.Scanner(System.in);
    int price = sin.nextInt(); /*nextInt() メソッドで int 型に */
    System.out.println("数字の" + price + "を入力しましたね");
  }
}
```

上記コードのポイントは，キーボードからの入力を sin 変数の nextInt() メソッドを使うことによって，int 型として読み込んでいることである．nextInt() メソッド部分を nextFloat() に改造すれば float 型を読み込めるし，next() に改造すれば String 型（文字列）を読み込むことができる．

そこで，課題 1-4 で出題した問題（の変形版）に取り組もう．名前をローマ字でユーザーから入力させ，文字列を格納するための String 型変数 name に保存，name の中身を表示した後，変数 name の長さを表示するプログラムを作成せよ．作成した .java ファイルを manaba+R に提出せよ．以下，どの課題も同じで .java ファイルを manaba+R に提出せよ．

コメント１ 文字列の長さを調べるメソッドを自分で調べてみよう.「java 文字列　長さ」で検索である．あるいは，次の週も含めてよく予習している人は既に知っていてもおかしくない．積極的に予習していこう．

コメント２ java.util.Scanner sin = new java.util.Scanner(System.in); の sin 部分は単なる変数名で，それ以外の部分は「おまじないである」と今のところは理解しておいて良い．最終的には厳格に全ての文法事項を押さえておいてほしいが, 最初の勉強（講義）では「おまじない」でひとまずスキップして文法事項の俯瞰を優先することにし，セメスター終了後，自習として行う学習にて詳細を学べばよいのである．

4.5 課題 11-5

課題 1-6 で出題した鶴亀算に関する問題（の変形版）に取り組もう．キーボードから鶴と亀の頭の数の和と足の数の和を入力させた上で，鶴の頭の数，亀の頭の数を表示するプログラムを Java で作成せよ．ただし，ファイル名は kadai11_5.java とする．

頭の数，足の数は正の整数で入力されることを仮定してよい．不正な入力に対する処理も記述しなくて良い．

4.6 課題 11-6

Python では字下げ（インデント）が文法事項として定めており，適切なインデントが行われたコードを書かねばエラーが出力される仕様であった．一方で Java では自由なインデントでプログラムを記述することが許されている違いがある．これを踏まえ，自由なインデントに関する課題に取り組んでみよう．

あなたは上回生の講義で Java によるプログラミングを使う実験科目を受講している．先生が実験科目で使うプログラムのサンプルコードを学生全員に提示しているが，そのプログラムの様子がどうもおかしい．明らかに読みにくい流儀でインデントが行われている．

そこで，以下の「先生のソースコード」を修正し,「先生！これならみんなで先生のコードを分かりやすく読めます．是非つかってください！」と提案してあげよう．

```
  /* kadai11_6.java*/
class kadai11_6 {
public static void main(String[] args){
for(int i=1;i<
10 ;i++){for(
int t=1;t<10 ;t++){
System.out.printf("%3d",i*t);/*printf 関数は教科書には記載がない．Google 検索をかけて，書
式指定した上で画面に文字を出力する関数であることを確認しよう */
System.out.
print(" ");}
System.out.println();
}}}
```

上記プログラムを打ち込み，まずは正しく動作が行われることを確認せよ．コメント文の部分は打ち込まなくて良い．その後,「K&R のスタイル」について Google 検索をかけ，ソースコードを K&R のスタイルで字下げを行い，字下げを行った.java ファイルを manaba+R に提出せよ．

注意）学生の皆さんに「K&R のスタイル」を推奨しているわけではなく，あくまで練習問題の例として挙げたにすぎない．これ以降の課題では，インターネットで調べることができるいくつかのスタイルの中から，好みのスタイルを選定・採用し，インデントを行ってほしい．ぐちゃぐちゃインデントは情報理工学部生として恥ずかしいからやめよう．

5　第 12 週課題

5.1　課題 12-1

以下のプログラムの空欄部分を埋めて変数 d に代入された値に対して下記の結果のように出力するプログラムを完成せよ.（同じ内容を羅列せず，必ず for 文を用いて作成すること）

```
class kadai12_1 {
  public static void main(String[] args) {
    int d = 3;

    ＜ここを埋める＞

  }
}
```

実行結果：

```
3 * 1 = 3
3 * 2 = 6
3 * 3 = 9
3 * 4 = 12
3 * 5 = 15
3 * 6 = 18
3 * 7 = 21
3 * 8 = 24
3 * 9 = 27
```

5.2　課題 12-2

以下のようなプログラムをコンパイルして実行する際に java test 123 abc と実行すると args[0] には"123"が，args[1] には"abc"が自動的に代入される．これを**コマンドライン引数**と呼び，プログラムの実行時にパラメータ指定などを行うためによく使われている．ちなみにコマンドライン引数の数は args.length で確認できる．なお，通常コマンドライン引数の変数名は args であるが任意の変数名に変えることもできる．

```
class test {
  public static void main(String[] args) {
  }
}
```

コマンドライン引数を用いて課題 12-1 を拡張して，以下のような行動をするプログラムを二重 for ループを使用して完成せよ.（ただし，コマンドライン引数は String 型であるため Integer.parseInt(args[0]) のように Integer.parseInt() 関数を用いて整数に変換する必要がある）また，ファイル名は kadai12_2.java とする．

実行例：

```
>java kadai12_2 3 5
3 の段
3 * 1 = 3
3 * 2 = 6
3 * 3 = 9
3 * 4 = 12
3 * 5 = 15
3 * 6 = 18
3 * 7 = 21
3 * 8 = 24
3 * 9 = 27
4 の段
4 * 1 = 4
4 * 2 = 8
4 * 3 = 12
4 * 4 = 16
4 * 5 = 20
4 * 6 = 24
4 * 7 = 28
4 * 8 = 32
4 * 9 = 36
5 の段
5 * 1 = 5
5 * 2 = 10
5 * 3 = 15
5 * 4 = 20
5 * 5 = 25
5 * 6 = 30
5 * 7 = 35
5 * 8 = 40
5 * 9 = 45
```

5.3 課題 12-3

9.0 を何回 2 で割ったら 0.003 より小さくなるか，回数を出力するプログラムを do-while 文を用いて作成せよ．ただし，ファイル名は kadai12_3.java とする．

5.4 課題 12-4

既存のメソッドを使わずに equals() メソッドと同じ機能をするクラスを作成せよ．なお，文字列 2 つはコマンドライン引数で入力された文字列 2 つとする．結果は一致すれば true，一致しなければ false を出

力する．文字列の長さは `args[0].length()` のようにメソッドを使用して確認できる．ただし，ファイル名は `kadai12_4.java` とする．

ヒント 「Java toCharArray()」で検索をかけてみよ

5.5 課題 12-5

指定された数に等しい段数を持つピラミッドを出力するプログラムを作成せよ (必ず繰り返し文を使用すること)．以下の実行例のようにコマンドライン引数を通じて段数を指定できるようにすること．また，ファイル名は `kadai12_5.java` とする．

実行例：

```
>java kadai12_5 10
          *
         ***
        *****
       *******
      *********
     ***********
    *************
   ***************
  *****************
 *******************
```

5.6 課題 12-6

二つの 2×2 行列 A, B の掛け算 $(A \times B)$ をするプログラムを作成せよ．行列は 2 次元配列で表現し，プログラム内で初期化する．掛け算の結果は別の 2 次元配列に代入し，画面に行列に見えるように出力せよ．ただし，ファイル名は `kadai12_6.java` とする．(なるべく `for` 文を用いて作成する．3 重ループが必要であることに注意せよ.)

```
A = [ 1  2 ]    B = [ 5  6 ]
    [ 3  4 ]        [ 7  8 ]
```

6　第 13 週課題

6.1　課題 13-1

double 型の 2 つの変数 x,y を受取り，受け取った変数の和を double 型として返す関数 add を実装せよ．同様に，差 (sub)，積 (mul)，商 (div) を返す関数を適切な関数名のもとに実装せよ．ただし，ファイル名は kadai13_1.java とする．

これらの関数を使い，キーボードから入力された 2 つの実数に対する和，差，積，商を出力するプログラムを作成せよ．不正な入力への対応は考えなくてよい．

コメント Python の関数定義では引数・戻り値の型を明示する必要はなかった．一方で，Java では引数・戻り値の型を明示する必要があることに注意しよう．

ヒント static double add(double x, double y) が関数宣言部の一例である．

6.2　課題 13-2

0 個の引数を取り，キーボードからの数字を入力させ，帰り値として入力された数字を double 型でかえす関数 input_double を実装せよ．

実装した input_double 関数と for 文の機能を使い， 3 個の実数を入力させ，それらの平均を表示するプログラムを作成せよ．

```
class kadai13_2 {
  static double input_double(){

      ＜ここを埋める＞

  }
  public static void main(String[] args) {

    ＜ここを埋める＞

  }
}
```

6.3　課題 13-3

学習教材動画「Java 入門」の**「8. メソッドについて」**にある「オーバーロードについて」(https://jp.linkedin.com/learning/learning-java/528474) を参考にして，ひとつの変数を受け取り値を返さない関数で，受け取った引数の型を画面に表示する関数を作成せよ．ただし，ファイル名は kadai13_3.java とする．作成した関数に対し，int, double, int 型の配列, double 型の配列, String 型を引数に与えた場合の動作をするように main 関数を記述せよ．

例えば，int 型変数を受取った場合は,「int 型が引数として指定されました」などと表示するようにせよ．

型は int, double, int 型の配列， double 型の配列，String 型の 5 種類に対応していればよい．

コメント 本課題はオーバーロード機能の練習を意図している．理解がよく進んでいる学生は,「オーバーロード機能なんかでいちいち実装せずとも，instanceof 演算子あたりを使えばいいのでは？」など,もっと良い方法を多数思い浮かぶことであろう.

6.4 課題 13-4

複素数とは，２つの実数のペア $z = (x, y)$ からなるもので，以下の加法，スカラー倍，乗法と呼ばれる二項演算を導入したものであった (式中の := は左辺の記法を右辺の内容で定義する，という意味である).

$$(x_1, y_1) + (x_2, y_2) := (x_1 + x_2, y_1 + y_2) \tag{1}$$

$$k(x, y) := (kx, ky) \tag{2}$$

$$(x_1, y_1) * (x_2, y_2) := (x_1x_2 - y_1y_2, y_1x_2 + x_1y_2) \tag{3}$$

複素数であることを明示するために $z = (x, y)$ は $z = x + iy$ と表記されることも多い.

ここで,「有理数」を表す Java のクラスとして以下のようなものが考えられる (「本格学習 Java 入門 改訂 3 版 p.175」より).

```
class Fraction {
  int numerator;
  int denominator;

  Fraction() {
      numerator = 0;
      denominator = 1;
  }

  Fraction(int numerator, int denominator) {
      this.numerator = numerator;
      this.denominator = denominator;
  }

  void add(Fraction f) {
    numerator = numerator * f.denominator + denominator * f.numerator;
    denominator = denominator * f.denominator;
  }

  void add(int n) {
    numerator = numerator  + denominator * n;
  }

  void add(int numerator, int denominator) {
    this.numerator = this.numerator * denominator + this.denominator * numerator;
    this.denominator = this.denominator * denominator;
  }

  public String toString() {
      return  numerator + "/" + denominator;
  }
}
```

このクラス Fraction を参考にしつつ，複素数を扱うクラス Complex を実装せよ．Complex クラスには加法を意味する add メソッド，スカラー倍を意味する scalar_mul メソッド，乗法を意味する mul メソッドを実装せよ．これらメソッドの機能を活用して $5*(3+4i)*(2+3i)+(3+4i)$ の計算を行い，計算結果を表示するプログラムを作成せよ．計算結果は toString() 機能を実装・利用することで表示すること．例えば,「System.out.println("計算結果は"+z+"です");」というコードで,「計算結果は 3+4i です」のように表示されるようにせよ．

ただし，Complex クラスを用いるファイル名は kadai13_4.java とする．

コメント 1 余裕のある者はコンストラクタで複素数型変数 Complex z の初期値を指定できるようにせよ．省略された場合は $0+0i$ が代入されるようにせよ．

コメント 2 複素数を定義するにはいくつかの流儀がある．このテキストで紹介しているものはその一例で

ある．代数学でいうところのイデアルを利用した定義もあるようである．興味があるものは Google 検索をかけてみよ．

コメント 3 Complex 型の変数 z1,z2,z3 に対し，z3=z1+z2 と記述すればよいように + 演算子をオーバーロードできないの？と思う学生もいるかもしれないが，残念ながら Java は演算子のオーバーロードをサポートしていないため，不可能なようである．C++言語なら z3=z1+z2 とかけば自作の複素数クラス Complex 同士の加法を行えるようにする実装が可能である．

ヒント

それぞれのメソッドの関数宣言部は `void add(Complex z)`, `void scalar_mul(double k)`, `void mul(Complex z)` のように書くことができる．

6.5 課題 13-5

複素数を扱う場面では，絶対値計算や割り算はよく行われる．絶対値，割り算は以下で定義される．

$$|(x,y)| := \sqrt{x^2+y^2} \tag{4}$$

$$(x_1,y_1)/(x_2,y_2) := \left(\frac{x_1x_2+y_1y_2}{x_2^2+y_2^2}, \frac{y_1x_2-x_1y_2}{x_2^2+y_2^2}\right) \tag{5}$$

ただし，割り算の定義を行う際には，以下の高校生の時に習った計算を参考にした：

$$\frac{x_1+iy_1}{x_2+iy_2} = \frac{(x_1+iy_1)(x_2-iy_2)}{(x_2+iy_2)(x_2-iy_2)} \tag{6}$$

$$= \frac{x_1x_2+y_1y_2+i(y_1x_2-x_1y_2)}{x_2^2+y_2^2} \tag{7}$$

$$= \frac{x_1x_2+y_1y_2}{x_2^2+y_2^2} + i\frac{y_1x_2-x_1y_2}{x_2^2+y_2^2} \tag{8}$$

そこで，絶対値演算と割り算演算に対応した複素数のクラス Complex2 を作成せよ．ただし，Complex2 クラスは継承機能を使って実装すること．スーパークラスは課題 13-4 で作成した Complex クラスとすること．

このとき Complex2 クラスを用いるファイル名は kadai13_5.java とする．

コメント 平方根の実装法は各自で調べること．文法事項に限らず，これからの（上回生の）講義を受ける中で，多少先生の説明にわからなない点があった場合は自分で調べる癖をつけることが大切である．

6.6 課題 13-6

課題 6-6 では,「関数 f」を引数に取り,「微分 f' の近似値をかえす関数 g」を返す作用素 d を作成することに取り組んだ．

$f(x) := \sin(x)$, $f'(x) = \cos(x)$, $(df)(x)$ の $x = 3$ における値を出力するプログラムを Java で作成すると，以下の通りとなる．

```
class kadai13_6 {
  @FunctionalInterface
  interface my_function{
    public double value(double x);
  }
  static my_function d(my_function f){
    return (double x) -> {
      return (f.value(x+0.00001)-f.value(x))/0.00001;
    };
  }
  public static void main(String[] args) {
    my_function f =  (double x) -> {
      return Math.sin(x);
    };
    my_function df = d(f);

    System.out.println(f.value(3));
    System.out.println(Math.cos(3));
    System.out.println(df.value(3));
  }
}
```

上記コードの部分の中でも，以下のコードが，double 型を受け取って double 型を受け取る関数を保存するための，新しい型の宣言とも解釈できることに注意しよう．

```
@FunctionalInterface
interface my_function{
  public double value(double x);
}
```

以下のコードは，関数を保存する新しい型に,「変数 x を受取り $\sin(x)$ を返す関数」を代入している操作にあたる．

```
my_function f =  (double x) -> {
  return Math.sin(x);
};
```

関数を保存する新しい型 my_function に保存された関数の値を計算する場合は，f.value(3); というコードを記述すればよい．f に sin(x) 関数が代入されていた場合は sin(3) の計算結果の値を得ることができる．

上記文法を踏まえ，関数 f，実数 a, 実数 b を受取り，f の区間 $[a, b]$ における定積分の近似値を返す関数 static double integral(my_function f, double a, double b) を実装せよ．入力は $a < b$ が常に満たされると仮定してよい．また，ファイル名は kadai13_6.java とする．

実装した関数を用いて，$\exp(x/2)$ の区間 $[2, 4]$ における定積分の近似値の計算結果を表示せよ．

ただし，関数 f の定積分の近似値は以下で求めることができることを利用せよ．

$$\int_a^b f(t)dt = \lim_{\delta \to 0} \sum_{t=a,a+\delta,\ldots,b-\delta} f(t)\delta \tag{9}$$

$$\simeq \sum_{t=a,a+h,\ldots,b-h} f(t)h \tag{10}$$

$$h := 0.00001 \tag{11}$$

コメント 1 手計算による計算結果とプログラムの出力を比較し,「異なる 2 手法で同一の結果だからそこそこの信頼度がある結果が得られた」という実感をしてみよ．

コメント 2 余裕のある者は $a < b$ を仮定せずに定積分を求める関数を実装せよ．$a > b$ の場合の定積分の定義を数学の講義を復習することで確認せよ．

コメント 3 この課題を Python を用いて実装してみよ．

7　第 14 週課題

7.1　課題 14-1

コマンド引数を 3 個受けて演算を行う簡単な計算機プログラムを作成せよ．コマンド引数 3 個は順に，数値，演算子，数値とし，加算，減算，乗算，除算の演算子はそれぞれ+, -, x, /とする．(注意．*は正規表現でワイルドカードとして使われるものであるため代わりに x を使用する)

実行例:

```
>java kadai14_1 4.5 + 3.5
= 8.0

>java kadai14_1 4.5 - 3.5
= 1.0

>java kadai14_1 4.5 x 3.5
= 15.75

>java kadai14_1 4.5 / 3.5
= 1.2857143
```

7.2　課題 14-2

文字列"ABCDEFG"に対して，ユーザが入力したコマンドライン引数の番号に該当する文字を出力する．下記の空欄を埋めて完成せよ.(ただし，'A' を 0 番目とする)

```
class kadai14_2 {
    public static void main(String[] args) {
        char[] str = "ABCDEFG".toCharArray();

        ＜ここを埋める＞

    }
}
```

7.3　課題 14-3

課題 14-2 はコマンドライン引数として 7 など存在しない文字番号を入力するとエラーが出てプログラムが終わってしまう．このようなエラーが起こらないように try, catch を用いて 14-2 を修正し，存在しない配列を指した場合は Number is out of the range と出力するようにせよ．ただし，ファイル名は kadai14_3.java とする．

7.4 課題 14-4

1. 以下のソースコードを打ち込み，ソースコードと同じ場所に input.txt という日本語文字を含むテキストファイルを設置した状態でソースコードのプログラムを実行して結果を確認せよ．
2. また，InputStreamReader(in, "UTF-8") の部分を InputStreamReader(in, "Shift_JIS") に変更して結果を比較し,「どのような違いがあるか」及び「なぜそのような違いが生じるか」について説明したテキスト文章ファイルを課題として提出せよ（テキストファイルは Microsoft Word を使わずに秀丸などテキストエディタを用いて作成せよ）．提出するファイル名は kadai14_4.txt とする．

```
import java.io.FileInputStream;
import java.io.InputStreamReader;
import java.io.BufferedReader;
import java.io.IOException;

class FileTest {
    public static void main(String[] args) {
        try {
            FileInputStream in = new FileInputStream("input.txt");
            InputStreamReader is = new InputStreamReader(in, "UTF-8");
            BufferedReader br = new BufferedReader(is);
            String s;

            while ((s = br.readLine()) != null) {
                System.out.println(s);
            }
        } catch (IOException e) {
        }
    }
}
```

7.5 課題 14-5

任意の数のコマンドライン引数として入力した単語群（文章）をファイルとして保存するプログラムを作成せよ．単語と単語の間には空白文字を入れて区切るものとする．なお，出力されるファイル名は test.txt とし，この出力結果とプログラムファイルの両方を提出せよ．プログラムファイル名は kadai14_5.java とする．

7.6 課題 14-6

テキストファイル 14-6.txt を読み込んで全ての行を逆順にして保存するプログラムを作成せよ．なお，保存先のファイル名を reverse.txt とし，この保存結果とプログラムファイルの両方を提出せよ．プログラムファイルは kadai14_6.java とする．