JAVA 入門

Java は Oracle（オラクル）・Sun Microsystems（サンマイクロシステム）によって開発されたコンピュータ言語（プログラム言語）です。

コンピュータ言語（プログラム言語）

ソフトウエア（アプリ）作成に必要な構文と規則の総称です。主な言語は以下のものがあります。

C 言語、FORTRAN、BASIC、COBOL、Pascal、Python など

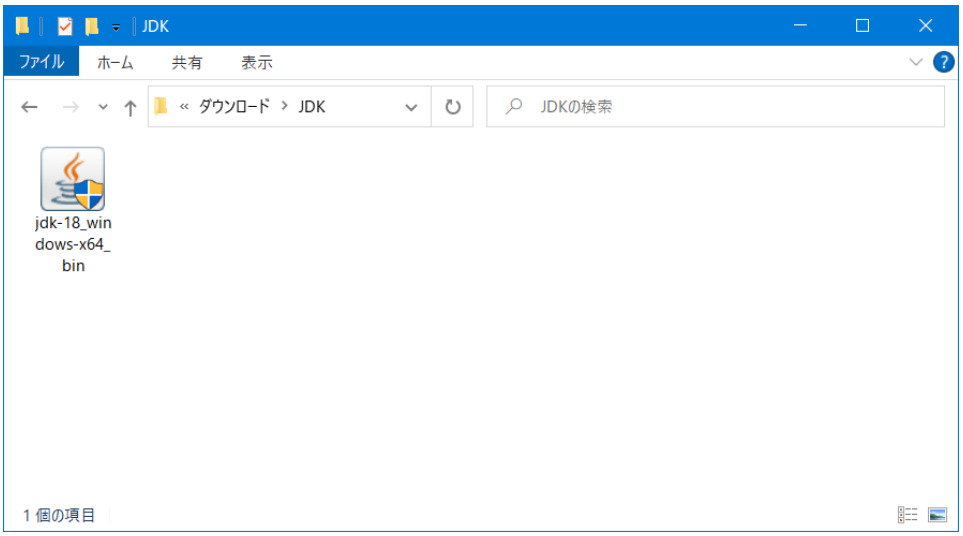
開発環境

ソフトウエアの作成を開発（Development）と呼びます。コンパイラ（変換器）を開発環境、デバッガ（入力ミスの発見と修正指示）、コード入力の補助などを加えたものを統合開発環境と呼びます。代表的な統合開発環境は Visual Studio(マイクロソフト）、Eclipse があります。今回は簡単にテキストエディタ（メモ帳）とコンパイラを使い、コマンド入力で開発をします。

1．開発環境の準備

Java の開発環境は Java SE Development Kit と呼ばれます。JDK と呼ばれることがあります。

開発環境は（https://www.oracle.com/java/technologies/downloads/）でダウンロードします。ダウンロードしたファイルをダブルクリックしてインストールを行います。



開発環境は" C:¥Program Files¥Java¥jdk-18.0.1.1"にインストールされます。

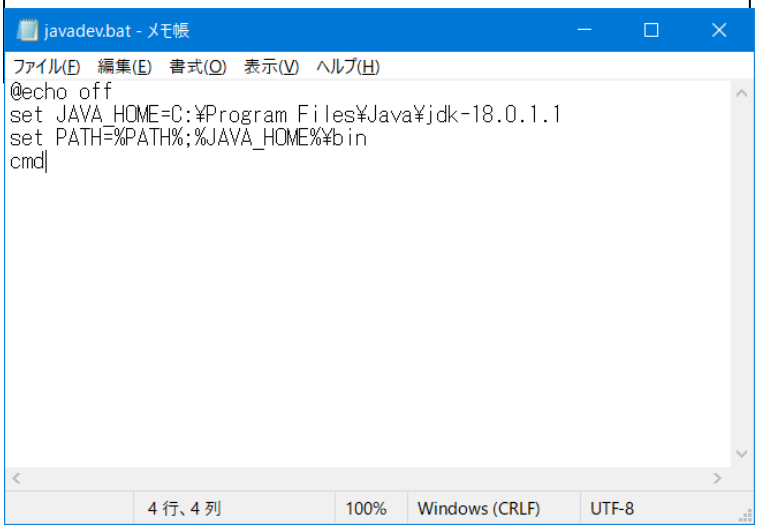
注意 バージョンによって数字が異なります。

（１）開発環境の実行

インストール後に開発環境を整えます。開発はコマンドプロンプトで行います。

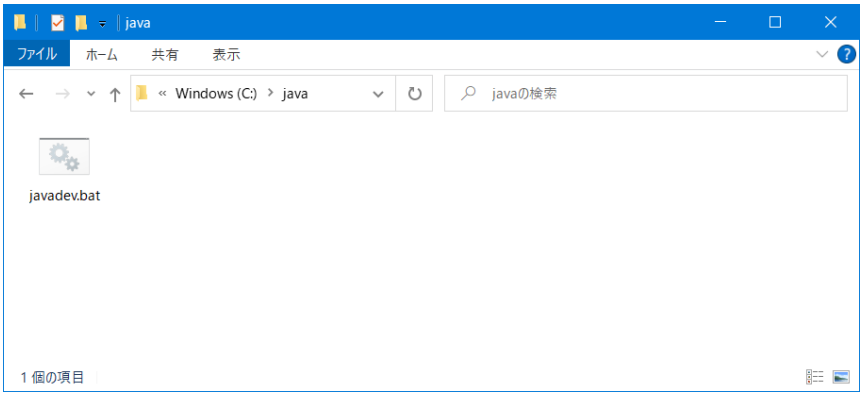
メモ帳を起動して以下の構文を記述します。バッチファイルと呼びます。

```
@echo off

set JAVA_HOME=C:¥Program Files¥Java¥jdk-18.0.1.1

set PATH=%PATH%;%JAVA_HOME%¥bin

cmd
```



名前を付けてファイルを保存します。拡張子は.bat にします。（例 javadev.bat）

バッチファイルをダブルクリックします。コマンドプロンプトが表示されます。



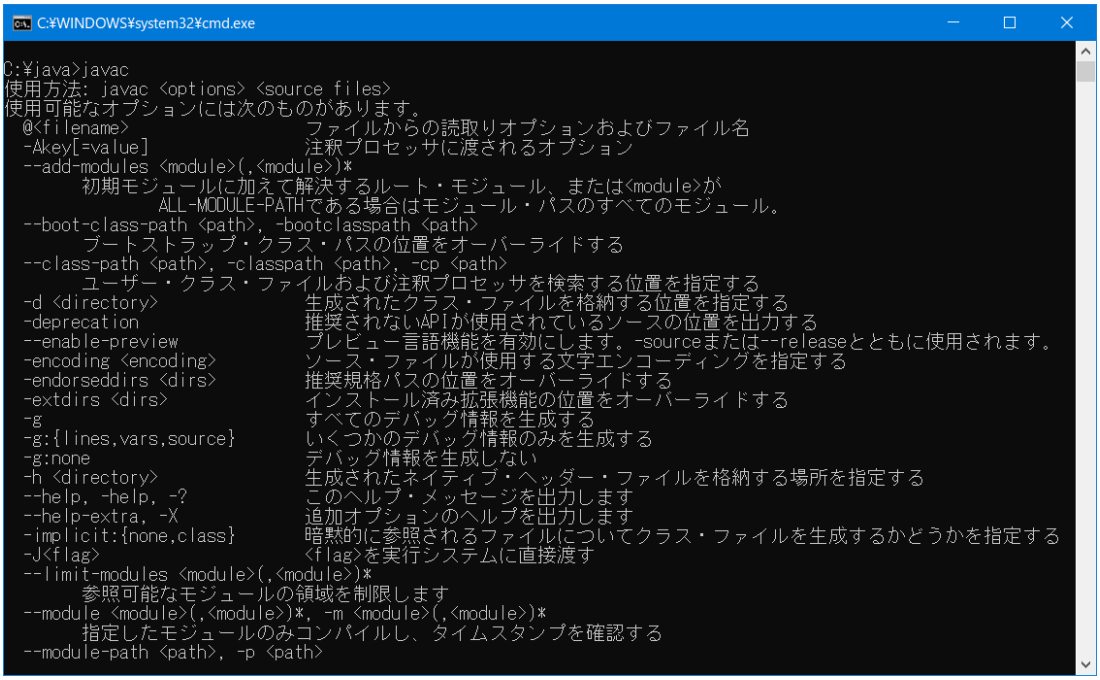
コマンドプロンプトに javac コマンドを入力します。

※「C:¥java>」の表記は各自のパソコンの状況により異なります。（カレントディレクトリと呼びます）

C:¥java>javac

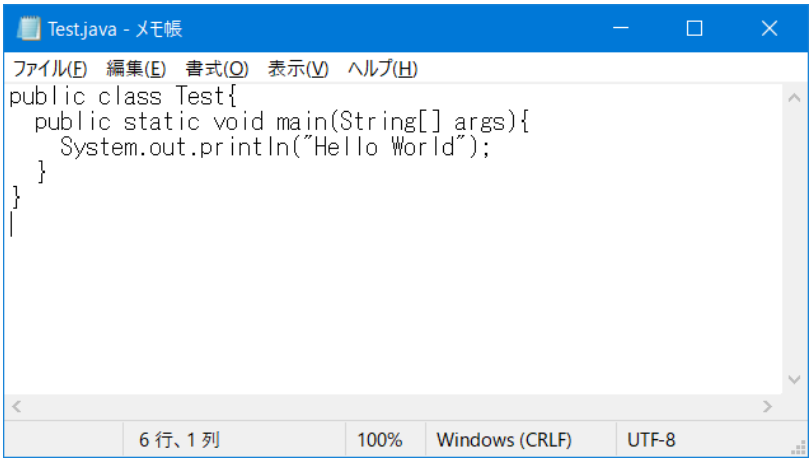
以下のメッセージが表示されます。

使用方法: javac <options> <source files>
使用可能なオプションには次のものがあります。



（２）プログラム（ソースコード）の記述

プログラムはソースコードと呼ばれます。メモ帳などのテキストエディタで記述します。



以下のソースコードを記述します。

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

名前を付けてソースファイルをバッチファイルと同じフォルダに保存します。

Java の場合はファイル名に規則があります。このソースファイルの場合は Test.java という名前で**保存してください。**学習すると分かりますが、Java ではファイル名=クラス名となっています。

（３）コンパイル

ソースファイルをコンピュータで実行可能な状態に変換することをコンパイルと呼びます。

Java では以下のコマンドでコンパイルを行います。

javac [ソースファイル]

例　javac Test.java

（４）実行

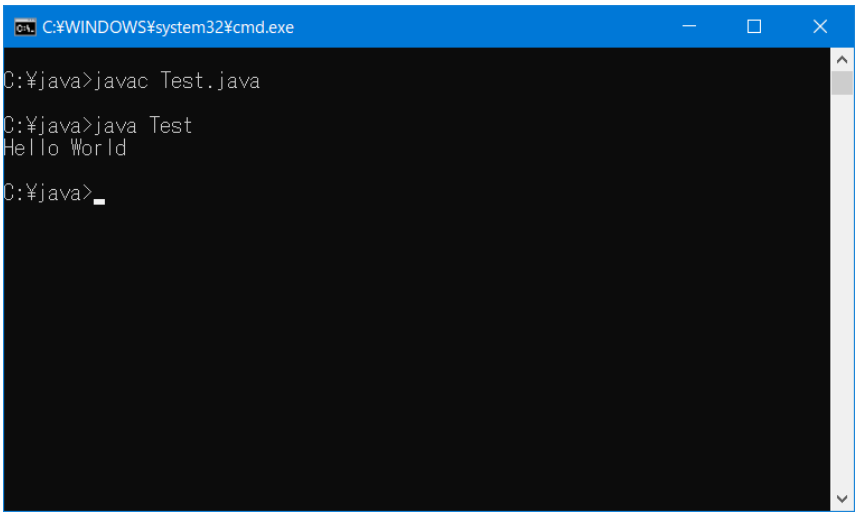
コンパイルによって実行ファイルが作成されます。Java の場合 class というファイルが作成されます。

コマンドを入力してプログラムを実行します。

java [実行ファイル名（拡張子なし）]

例　java Test

画面に Hello World と表示されます。



2．足し算

足し算を行います。メモ帳に以下のソースコードを記述します。

```
public class AdditionTest{
   public static void main(String[] args) {
      int a = 10;
      int b = 15;

      int c = a + b;

      System.out.println("c = " + c);
   }
}
```

名前を付けて保存します。名前は AdditionTest.java とします。

コンパイルします。以下のコマンドを入力します。

```
javac AdditionTest.java
```

エラーを確認します。エラーが出ない場合はコンパイル完了です。

実行します。以下のコマンドを入力します。

```
java AdditionTest
```

実行結果が画面に表示されます。

```
C = 25
```

コードの説明

```
public class AdditionTest{
   public static void main(String[] args) {
      int a = 10;
      int b = 15;

      int c = a + b;

      System.out.println("c = " + c);
   }
}
```

この main 文からプログラムが開始されます。

数値の入れ物を宣言します。"int"は整数を表現します。

数値を足します。

画面に c の値を表示します。

３．引数（ひきすう）の利用

前述のソースコードの場合は値が直接書いてあります。ハードコーディングと呼びます。実際に使うには不便ですので、ソースコードを修正して自由な値を入力できるようにします。

以下のソースコードを記述してコンパイルします。

```
public class ArgumentTest{
   public static void main(String[] args) {
      if (args.length > 1) {
         int a = 0;
         int b = 0;

         try{
            a = Integer.parseInt(args[0]);
         }catch(Exception ex){
         }

         try{
            b = Integer.parseInt(args[1]);
         }catch(Exception ex){
         }

         int c = a + b;
         System.out.println("c = " + c);
      }else{
         System.out.println("java ArgumentTest a b");
      }
   }
}
```

実行方法

以下のコマンドを入力します。

```
java ArgumentTest
```

計算は行われずに画面に

```
java ArgumentTest a b
```

と表示されます。次に以下のコマンドを入力します。

```
java ArgumentTest 10 15
```

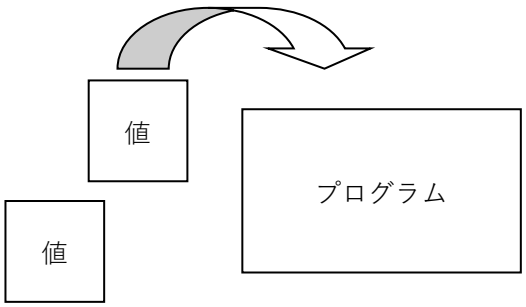
画面に

```
c = 25
```

と表示されます。

コードの説明

今回はコマンドから入力される値を使用しました。入力された値は引数と呼びます。




```
public class ArgumentTest{
   public static void main(String[] args) {
      if (args.length > 1){
         int a = 0;
         int b = 0;

         try{
            a = Integer.parseInt(args[0]);
         }catch(Exception ex){
         }

         try{
            b = Integer.parseInt(args[1]);
         }catch(Exception ex){
         }

         int c = a + b;
         System.out.println("c = " + c);
      }else{
         System.out.println("java ArgumentTest a b");
      }
   }
}
```

args が引数です。

args を判定します。
（引数の個数が 1 より多いので、2 個を対象とする）

引数を計算で使えるように整数に変換して a と b に入れます。

注意　JAVA で配列の数え方は 0 番目、1 番目、2 番目というように 0 から数えます。

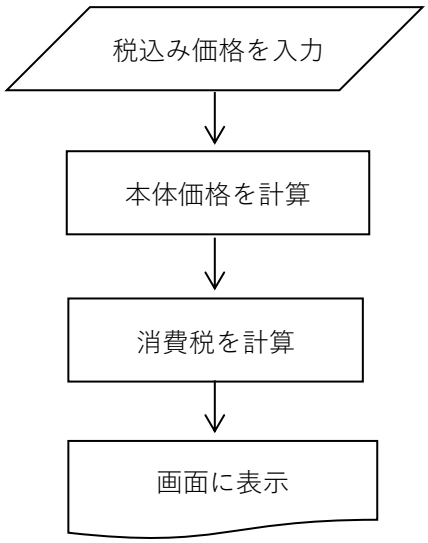
0 番目の引数　　args[0]

1 番目の引数　　args[1]

プログラム言語では 0 から数える方法が一般的です。

４．プログラムの設計

入力された金額を元に本体金額と消費税を計算するソフトウエアを作成します。
プログラムの流れ（フローチャート）は次のようになります。



ソースコードは以下のようになります。

```
public class TaxTest{
   public static void main(String[] args) {
      if (args.length > 0){
         float a = 0f;

         try {
            a = Float.parseFloat(args[0]);
         }catch(Exception ex) {
         }

         float b = a / 1.1f;
         float tax = a - b;
         System.out.println("本体 " + b + " 円");
         System.out.println("消費税 " + tax + " 円");
      }
   }
}
```

コードの説明

float b = a / 1.1f;　　割り算は / で表現します。

本体価格 = 税込み価格 ÷ 1.1　　※消費税が 10%の場合

消費税 = 税込み価格 - 本体価格

となります。

５．繰り返し処理（ループ処理）の実装

繰り返し処理を行うと効率のよいソフトウエアを作成することができます。

２の累乗を１０乗まで求める処理を実装します。

２の１乗

２の２乗

…

２の１０乗

考え方

計算の法則性を見つけます。この場合は前の計算結果にさらに×２を行っています。

２の１乗（２）　２の２乗（２×２）　２の３乗（４×２）　…

この計算を１０回繰り返します。この考え方から実装するとコードは以下となります。

```
public class ExpTest{
   public static void main(String[] args) {
      int b = 1;
      for (int i = 0; i < 10; i++) {
         b = b * 2;
         System.out.println("2 の" + (i + 1) + "乗  = " + b);
      }
   }
}
```

コードの説明

繰り返し処理は for という命令文を書きます。処理は{}でくくります。

```
for (int i = 0; i < 10; i++)
```

この場合は０から開始して１０より小さい整数（９）で終わります。変数 i は

0→1→2→3→4→5→6→7→8→9

と変化します。+1（インクリメント）と呼びます。インクリメントは i++と表記します。

代入演算子について（算数や数学とは異なる考え方なので、注意します。）

プログラム言語では=(イコール)は代入演算子を表現します。「変数 b に b * 2 の値が入る」という意味です。

１回目（i は 0）　b = 1 * 2　　1×2 の答え「2」が b に入ります。

２回目（i は 1）　b = 2 * 2　　2×2 の答え「4」が b に入ります。

３回目（i は 2）　b = 4 * 2　　4×2 の答え「8」が b に入ります。

6．クラスを使う　日時の表示

ソフトウエアを開発するときに日付や時刻を使うことがあります。Java は便利な機能をあらかじめ用意しています。この機能たちをクラス（Class）と呼びます。日付と時刻を扱うために日時クラス(LocalDateTime クラス)を使います。以下のコードを記述してコンパイルします。ファイル名は LocalDateTimeTest.java です。

```
import java.time.*;
public class LocalDateTimeTest{
   public static void main(String[] args){
      System.out.println(LocalDateTime.now(ZoneId.systemDefault()).toString());
   }
}
```

コンパイルして実行します。画面に

2022-05-28T19:07:03.112631

と表示されます。※日時は実行された時のものが表示されます。

コードの説明

コードの LocalDateTime.now(ZoneId.systemDefault())で日時クラスの現在日時を取得することができます。toString()という命令文により文字列で表示するように指示します。

7．条件文とファイルの利用

特定の条件を判断して処理を変えたい場合があります。if 文を使用して条件判断を行います。

ファイルサイズを判断してメッセージを表示するソフトウエアを作成します。

ファイルサイズが 1MByte 以下のときは「メールに添付できます。」それ以外は「添付できません。」と表示します。※1MByte は参考例です。

ファイル(File)クラスを利用するとファイルの情報を取得することができます。

```
import java.io.*;
public class FileSizeTest{
   public static void main(String[] args) {
      if (args.length > 0) {
         File f = new File(args[0]);

         if (f.exists()) {
            long size = f.length();
            System.out.println("File Size = " + size + " byte");

            if (size <= 1000000L) {
               System.out.println("メールに添付できます。");
            }else{
               System.out.println("添付できません。");
            }
         }
      }
   }
}
```

コードの説明

if 文が 3 ヵ所あります。この条件を満たす場合{ }の中の処理が実行されます。

1 番目	if (args.length > 0)	引数の大きさが 0 より多いか？
2 番目	if (f.exists())	ファイルが存在するか？
3 番目	if (size <= 1000000L)	ファイルサイズが 1000000 以下か？

ファイルの有無を確認する場合はファイルクラスの exists()を使用します。

ファイルサイズの取得はファイルクラスの length()を使用します。

```
long size = f.length();
```

コンパイルして実行します。確認するファイルをフルパスで記述します。

例　C ドライブにある test.txt を確認したい場合。

```
java FileSizeTest c:¥test.txt
```

画面に結果が表示されます。

8．グラフィカル・ユーザー・インターフェイス（GUI）の利用

現在のパソコンは画面をマウスなどで操作します。この操作画面をグラフィカル・ユーザー・インターフェイス(GUI)と呼びます。Java には GUI を行うためのクラスがあります。

前に作成したファイル判定ソフトを GUI 対応ソフトに修正します。

改良点１　フルパスの入力が困難なので画面でファイルを選択するようにします。

改良点２　画面上に判定メッセージを表示するようにします。

※制限するファイルサイズを引数で指定します。

GUI を利用する場合、ボタンやファイル選択画面など多くのコードを記述する必要があります。

あらかじめ用意された画面の部品（ボタンなど）を利用して作成すると効率的です。

例では Java で用意されたファイル選択画面を利用しています。

ファイル選択画面(JFILECHOOSER)クラス

ファイル選択画面を表示します。選択されたファイルはファイルクラスとして取得できます。

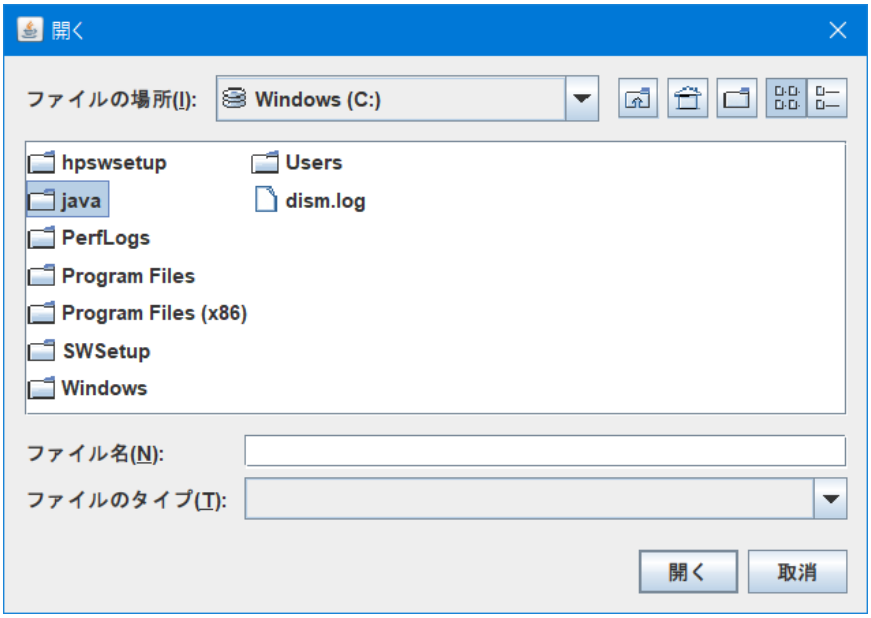


図３．ファイル選択画面

図４．判定画面（JFrame クラス）

ファイル選択画面（図３）が表示されます。ファイルを選択すると判定画面（図４）が表示されます。

ソースコードは以下のようになります。少し長いですが、これが最後となります。

```
import java.io.*;
import javax.swing.*;
public class GUIFileSizeTest{
   public static void main(String[] args){
      if (args.length > 0){
         long limit = 0L;
         try{
           limit = Long.parseLong(args[0]);
         }catch(Exception ex){}
         JLabel label = new JLabel();
         JFrame dlg = new JFrame();
         dlg.setTitle("ファイルサイズの判定");
         dlg.setDefaultCloseOperation( JFrame.EXIT_ON_CLOSE );
         dlg.add(label);
         dlg.setBounds(0, 0, 320, 120);
         dlg.setVisible(true);
         JFileChooser chooser = new JFileChooser();
         chooser.setFileSelectionMode( JFileChooser.FILES_ONLY );
         chooser.setDialogTitle( "開く" );
         chooser.setAcceptAllFileFilterUsed(false);
         int ret = chooser.showOpenDialog( dlg );
         if ( ret == JFileChooser.APPROVE_OPTION ){
            File f = chooser.getSelectedFile();
            if (f.exists()) {
               long size = f.length();
               if (size <= limit) {
                  label.setText("メールに添付できます。" + size + " byte");
               }else{
                  label.setText("添付できません。" + size + " byte");
               }
            }
         }
      }
   }
}
```

コンパイルして実行します。

```
java GUIFileSizeTest 1000000
```

バッチファイルの利用

判定ソフトを使いやすいように起動用バッチファイルを作成します。以下の構文を記述して GUIFileSizeTest.bat と名前を付け、GUIFileSize.class があるフォルダに保存します。

```
@echo off
java GUIFileSizeTest 1000000
```

このバッチファイルをダブルクリックするとファイル判定ソフトが起動します。

実行環境の利用

自分で作成したソフトを他のユーザの PC で使用したい場合は Java の実行環境をインストールします。
実行環境は Java を実行するためのもので Javac などの開発環境はインストールされません。
Java Runtime Environment(JRE)と呼ばれます。
JRE をインストールした後に class ファイルとバッチファイルをコピーします。
バッチファイルをダブルクリックするとソフトが起動します。

終

あとがき

おつかれさまでした。簡単なプログラムから GUI プログラムまでを早足で説明しました。入門とありますが、初めての方だけでなく、プログラミングに一度挑戦したが身につかなかったかたにも取り組んでほしい内容としました。
少ないページ（14 ページ）で覚えていただきたいことを書いてしまいましたので、ところどころにわからない単語がでてきます。これらを細かく解説すると市販されている入門書と同じようなページ数になってしまうため、説明を省略しました。詳しく知りたいかたはインターネット検索などで新たな知識を身に着けてください。
多くの入門書がプログラム言語仕様の解説書となっています。入門書は単なる How-to 本で良いと思います。自分でプログラムをしていくなかで言語仕様を理解し、必要な構文を使えるようになります。
プログラミングの目的はソフトウエアを開発・保守することです。アルゴリズムやオブジェクト指向、論理的な考え方などはあとから身についていくと感じます。まずは自分の手でコードを記述して、コンパイルを行うこと。エラーが出たらコードを直すこと。実行して思うように動かない場合はどうしてなのか考えること。それらの行動がプログラミングの上達だと考えます。
あなたが簡単なメモを取るようにコーディングをして、小説を読むかのようにソースコードを読めるようになると幸いです。

2022 年 6 月 1 日