

第7回 これまでの復習とJavaの開発環境

第7回は 対面授業 です。

1 第1回～第6回の復習

- **第1回から第6回までの復習をする**
 - これまでの内容を振り返る
- **第1回 プログラミングの基礎を復習**
 - Processingとは
 - Processingの基本
 - Processingで図形を描画する
 - 主な変数
 - setup() 関数
 - 単純な繰り返し
- **第2回 Javaの基礎と条件分岐**
 - JavaとProcessing
 - 数値、変数、演算子など (Lesson 2、3、4)
 - 条件分岐 (Lesson 5)
- **第3回 関数 (メソッド)**
 - Processingの ActiveMode と StaticMode
 - 関数の呼び出し
 - 変数の有効範囲
 - 関数の引数
 - 関数の戻り値
 - Processingの draw() 関数
 - Processingのマウス操作
- **第4回 繰り返し**
 - セミコロン、if文の「かつ」「または」、適切なインデント
 - 繰り返し
 - 乱数
- **第5回 配列 (その1)**
 - 配列の基礎
 - 文字列の変数 (配列変数)
- **第6回 配列 (その2)**
 - 配列変数
 - 配列の応用 `.length`、並び替え など
 - 多次元配列
 - 配列変数を引数にする

2 Javaの開発環境

・Javaの開発環境を整備する

- 次回（第8回）以降は、Javaの**JDK (Java Development Kit)** を使います
- **JDKを自分のPCにインストールする**必要があります
- Javaは コマンドライン でコンパイル・実行する
- コマンドラインの操作は「**コマンドプロンプト**」で行う（Macはターミナル）

・JDKを自分のPCにインストールする

- JDKはPCでJavaのプログラムを コンパイル・実行 する環境
- インストール手順の概要
 - (1) 「jdk.java.net」からzipファイルをDLする [JDKのリンク](#)
 - (2) 「C:\Program Files」の下に「\java」を作成する
 - (3) DLしたzipファイルの中身（\jdk-19???)を「C:\Program Files\java」にドラッグして展開する
???は適宜読み替えること
 - (4) システム環境変数の **path** に「C:\Program Files\java\jdk-19???\bin」を追加する
 - (5) コマンドプロンプトを開き、サンプルプログラムを実行してみる
- インストール手順の[説明動画を用意しました](#) [説明動画（音声なし）](#)
(説明動画ではバージョンが**18**ですが、現在の最新のバージョンは**19**です。適宜読み替えて作業してください。)

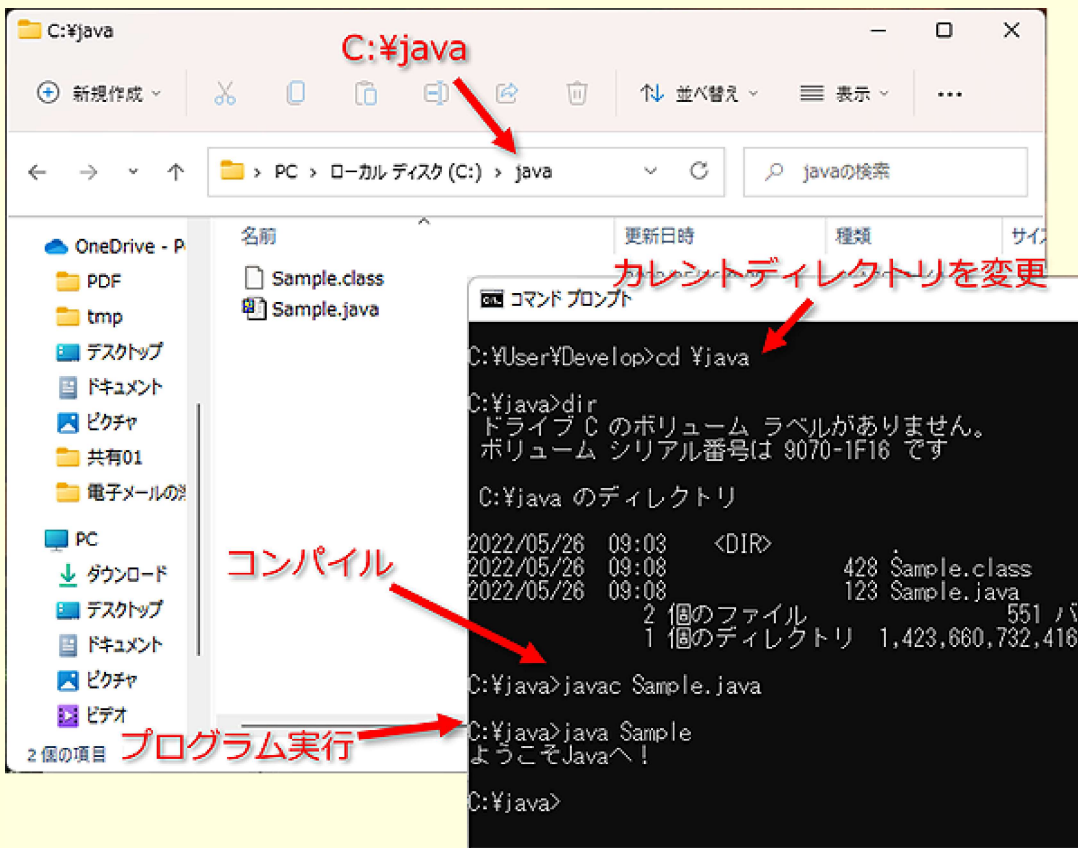
・サンプルプログラムを実行してみる

- フォルダのコピー、システム環境変数の設定が完了したら、コンパイル・実行して確認する
- 簡単なサンプルプログラム「**Sample.java**」を作成して、コンパイル・実行する

<Sample.java>

```
class Sample
{
    public static void main(String[] args)
    {
        System.out.println("ようこそJavaへ!");
    }
}
```

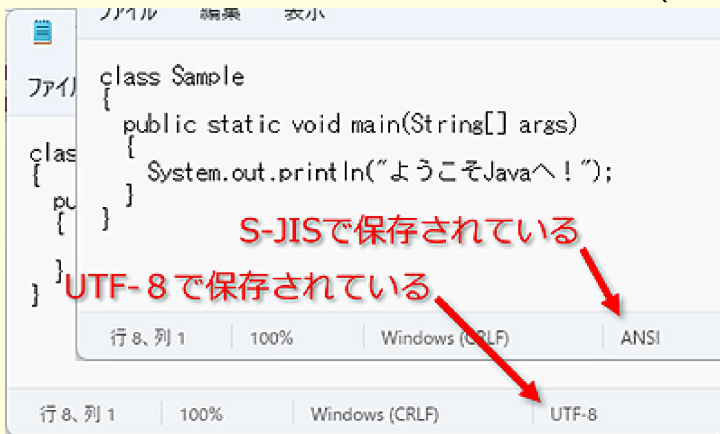
- Javaのソースファイルを保存するフォルダ「**C:\java**」を新規作成して開く
- 「**C:\java**」にソースファイル「**Sample.java**」を新規作成して、上記のように入力・保存する
- コマンドプロンプトを開き、カレントディレクトリを「**C:\java**」に変更する
- 「**javac Sample.java**」でコンパイルする
- 「**java Sample**」で実行する
- 環境設定からサンプルプログラム実行までの [説明動画（音声なし）](#)



・ソースファイルのエンコードに注意

- ソースファイルのエンコードによって、コンパイル時にエラーが出ることがあります
- デフォルトのエンコードは **UTF-8**
- エディタによって、**S-JIS (Shift-JIS)** になっている場合があるので注意

<メモ帳で、UTF-8で保存されているものと、S-JIS(ANSI)で保存されているもの>



- エンコードによるエラーが出る場合、**コンパイルオプション**をつければコンパイルできる

```
コマンドプロンプト
C:\>java>javac Sample.java
Sample.java:5: エラー: この文字(0x82)は、エンコーディングUTF-8にマップできません
    System.out.println("?????Java??I");
Sample.java:5: エラー: この文字(0x82)は、エンコーディングUTF-8にマップできません
    System.out.println("?????Java??I");
Sample.java:5: エラー: この文字(0xB1)は、エンコーディングUTF-8にマップできません
    System.out.println("?????Java??I");
Sample.java:5: エラー: この文字(0x82)は、エンコーディングUTF-8にマップできません
    System.out.println("?????Java??I");
Sample.java:5: エラー: この文字(0x82)は、エンコーディングUTF-8にマップできません
    System.out.println("?????Java??I");
エラー6個
C:\>java>javac -encoding SJIS Sample.java
C:\>java>
```

ソースファイルのエンコード
によって、オプションを付ける

3 Javaの基本 (教科書)

- ・ 教科書の内容を確認する
 - 教科書 Lesson1、Lesson2

1.1 Javaのプログラム (教科書 p.2)

- ・ プログラムのしくみ
- ・ プログラム言語Java
 - コンパイラ (javacコマンド)
 - インタープリタ(javaコマンド)

1.2 コードの入力

- ・ JDKをインストールする (完了)
- ・ コードのしくみを知る
 - プログラムはテキストエディタ (メモ帳、秀丸エディタなど)で入力する
- ・ テキストエディタにコードを入力する
 - プログラムをテキストエディタで入力する
 - Sample1.java** で保存する

```
class Sample1
{
    public static void main(String[] args)
    {
```

```
System.out.println("ようこそJavaへ!");
}
}
```

- Javaのソースファイルの拡張子は、必ず **.java** にする
- ソースファイル名は、class名の「**Sample1**」に拡張子「**.java**」を追加したもの「**Sample1.java**」にしなければいけないので注意！
- Javaでは、行末(ステートメント終端)に必ず**セミコロン「;」**が必要です (※Pythonでは基本つけない)
- Windowsでファイルの**拡張子が表示されていない場合は、コントロールパネル-->エクスプローラーのオプション-->表示タブ** で設定を変更すること

1.3 プログラムの作成

・コンパイルのしくみを知る

ソースファイル --[コンパイラ]--> classファイル --[実行コマンド]-->プログラム実行

・コンパイラを実行する

- 下記のようにコンパイラのコマンドを実行する、カレントディレクトリに注意

```
C:\java>javac Sample1.java <--- コンパイル
C:\java> <--- エラーがなければ、コマンド入力待ちになる
```

- コンパイルが無事完了すると、中間ファイル **Sample1.class** が作成される

1.4 プログラムの実行

・プログラムを実行する

中間ファイル「*.class」があれば、実行できる

- 下記のようにインタプリタのコマンドを実行する

```
C:\java>java Sample1 <--- クラス名を指定して実行する
ようこそJavaへ！
C:\java>
```

2.1 画面への出力

・新しいコードを入力する

- 下記のように **Sample1.java**に青文字の1行を追加する

```
class Sample1
{
```

```
public static void main(String[] args)
{
    System.out.println("ようこそJavaへ!");
    System.out.println("Javaをはじめましょう!");
}
}
```

- ・ 画面に出力する

- System.out.println("ようこそJavaへ!"); // 改行あり
- System.out.print("ようこそJavaへ!"); // 改行なし

- ・ いろいろな出力方法を知る

2.2 コードの内容

- ・ コードの流れを追ってみる

- ・ main()メソッド

- プログラムが実行されると、一番はじめに処理されるメソッド

```
public static void main(String[] args)
```

- ・ 1文ずつ処理する
- ・ コードを読みやすくする
- ・ コメントを記述する

```
class Sample1
{
    public static void main(String[] args)
    {
        System.out.println("ようこそJavaへ!"); // スラッシュ2個でコメント
        System.out.println("Javaをはじめましょう!"); /* これもコメント */
    }
}
```

- ・ クラスをみわたす

演習

<演習 1>

JDKのインストールを完了して、下記 **Ensyu0101.java** のコンパイル、実行ができるようにしなさい。

<Ensyu0101.java のソース>

```
class Ensyu0101
```