

# Java



盛岡情報ビジネス&デザイン専門学校

# ★☆☆★本日の内容★☆☆★

1. クラスについて(復習)
2. APIとは
3. JavaMailライブラリ



# 型・戻り値とは (復習)



## ◆型・戻り値とは(復習)

クラスの基礎的な構成について復習します。

```
public class Human {
```

```
    String name;  
    public int age;
```

```
    public Human(String n , int a){  
        name = n;  
        age = a;  
    }
```

```
    public int getAge(){  
        return age;  
    }
```

```
}
```

### フィールド

(修飾子) 型 変数名

### コンストラクタ

(修飾子) クラス名

### メソッド

(修飾子) 戻り値の型 メソッド名



## ◆値の代入(復習)

値の代入は「型」を揃える必要があります。

int型へ代入する場合はint型の変数である必要があります。**※一部例外有**

```
String s = "abcd";
```

```
int i = 10;
```

```
int i2 = 20;
```

```
Human h = new Human("高橋");
```

```
Human h2 = new Human("山口");
```

```
s = i; //これはできません。
```

```
i = i2; //これはできます。代入先と代入元の型が一致しているため。
```

```
h = h2; //これもできます。代入先と代入元の型が一致しているため。
```



## ◆値の代入(復習)

値の代入は「型」を揃える必要があります。

メソッドの戻り値についても同様に型を合わせる必要があります。

```
Human = new Human("高橋", 27);
```

```
int a = getAge();
```

```
String s = getAge();
```

これはOK

これはNG

```
public class Human {  
    String name;  
    public int age;  
  
    public Human(String n, int a){  
        name = n;  
        age = a;  
    }  
  
    public int getAge(){  
        return age;  
    }  
}
```



# APIとは



## ◆APIとは

# Application Programming Interfaceの略。

Javaの場合はあらかじめ用意されたクラスがたくさん集まったクラス一覧のようなもの。





# ◆APIの読み方

Java(tm) Platform  
Standard Edition 8

概要 パッケージ クラス 使用 階層ツリー 非推奨 索引 ヘルプ

前のクラス 次のクラス フレーム フレームなし すべてのクラス

サマリー: ネストされたクラス | コンストラクタ | メソッド 詳細: フィールド | コンストラクタ | メソッド

compact1, compact2, compact3

java.util

クラスScanner

java.lang.Object  
java.util.Scanner

すべての実装されたインタフェース:  
Closeable, AutoCloseable, Iterator<String>

パッケージ名

クラス名

クラスの階層構造  
(この場合はObjectクラスを継承しているという意味)

実装しているインタフェースの一覧



## ◆APIの読み方

### フィールドのサマリー

フィールド

フィールドの一覧

修飾子と型

フィールドと説明

static int

BYTES

2の補数バイナリ形式でint値を表すために使用されるバイト数。

static int

MAX\_VALUE

変数名

intに設定可能な最大値 $2^{31}-1$ を保持する定数

変数の説明

「修飾子」と「型」

MIN\_VALUE

intに設定可能な最小値  $-2^{31}$ を保持する定数

static int

SIZE

2の補数バイナリ形式でint値を表すのに使用されるビット数で



# ◆APIの読み方

コンストラクタの一覧

## コンストラクタのサマリー

### コンストラクタ

#### コンストラクタと説明

**Scanner** (**File** source)

指定されたファイルからスキャンされた値を生成する新しいScannerを構築します。

**Scanner** (**File** source, **String** charsetName)

指定されたファイルからスキャンされた値を生成する新しいScannerを構築します。

コンストラクタ名と引数

コンストラクタの  
説明

**Scanner** (**InputStream** source)

指定された入力ストリームからスキャンされた値を生成する新しいScannerを構築します。

**Scanner** (**InputStream** source, **String** charsetName)

指定された入力ストリームからスキャンされた値を生成する新しいScannerを構築します。



# ◆APIの読み方

## メソッドのサマリー

メソッドの種類一覧

すべてのメソッド

インスタンス・メソッド

具象メソッド

修飾子と型

メソッドと説明

void

**close()**

現在のスキャナをクローズします。

Pattern

**delimiter()**

このScannerが区切り文字のマッチングに  
Patternを返し

戻り値の型

int

メソッド名と引数

**nextInt()**

メソッドの簡易説明  
※メソッド名クリックで  
詳細説明へジャンプ

入力の次のトークンをintとしてスキャンします。



# 外部ライブラリとは



## ◆外部ライブラリとは

標準ライブラリ …… 最初から用意されているクラスライブラリのこと  
ex) StringやScannerなど

外部ライブラリ …… インターネット上などで公開されているソースを  
個別にプロジェクトにインポートして使用するもの。  
ex) JavaFX、JavaMail、pdfbox、ZXing など

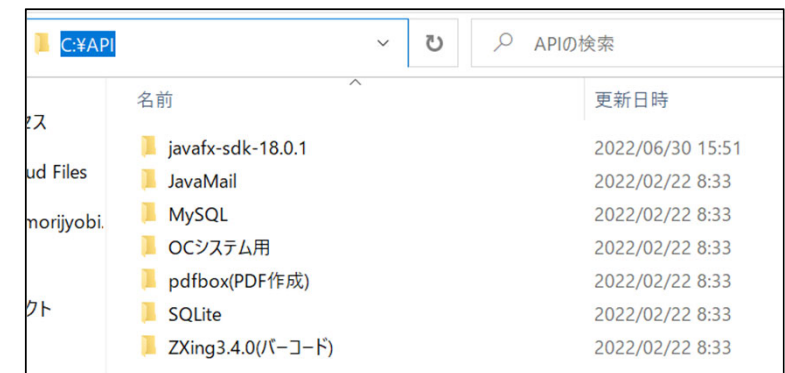


## ◆JavaMailとは

Javaプログラムからメールを送信するためのライブラリです。  
メール送信に必要なクラスがまとまっています。  
Slackで配布している「javax.mail.jar」というファイルです。

これから外部ライブラリを配布することが多数あります。  
受け取ったライブラリはダウンロードフォルダに入れっぱなしにするのではなく、整理しておくことをオススメします。

※次の手順でプロジェクトとライブラリの連携方法をレクチャします。  
連携はパスの参照になるため、元のフォルダから移動したり、  
削除したりすると参照できなくなります。  
そのため、パスが変わらないようにあらかじめ  
決まったフォルダに入れておくようにしましょう。  
参考までに、私はCドライブの直下にAPIフォルダを作って  
そこにサブフォルダを作成して入れています。



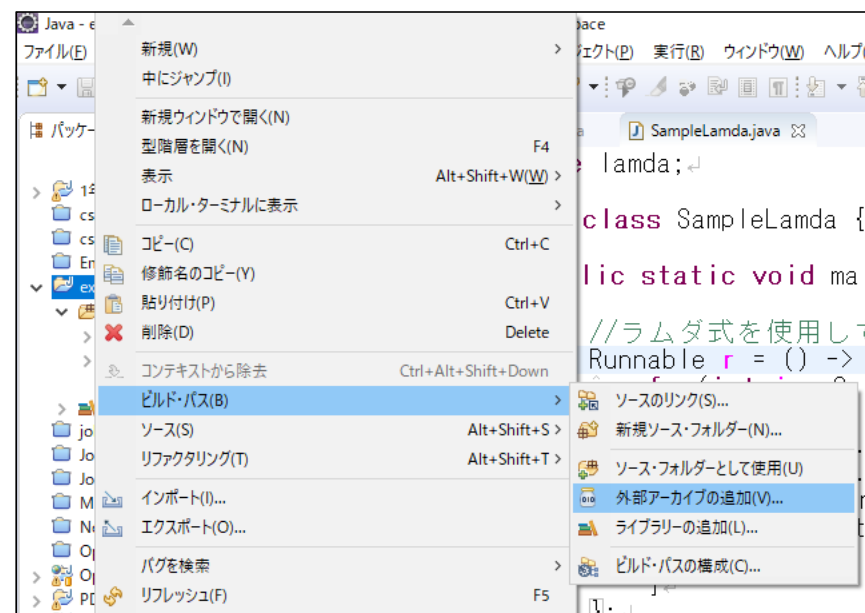
名前	更新日時
javafx-sdk-18.0.1	2022/06/30 15:51
JavaMail	2022/02/22 8:33
MySQL	2022/02/22 8:33
OCシステム用	2022/02/22 8:33
pdfbox(PDF作成)	2022/02/22 8:33
SQLite	2022/02/22 8:33
ZXing3.4.0(バーコード)	2022/02/22 8:33



## ◆外部ライブラリの設定方法

外部ライブラリは基本的にプロジェクトごとに設定を行います。

- ① プロジェクト名を右クリック→「ビルドパス」→「外部アーカイブの追加」
- ② 追加したい外部ライブラリ(.jar)を選択
- ③ 参照ライブラリに追加され、  
そのプロジェクトで使用可能になる。



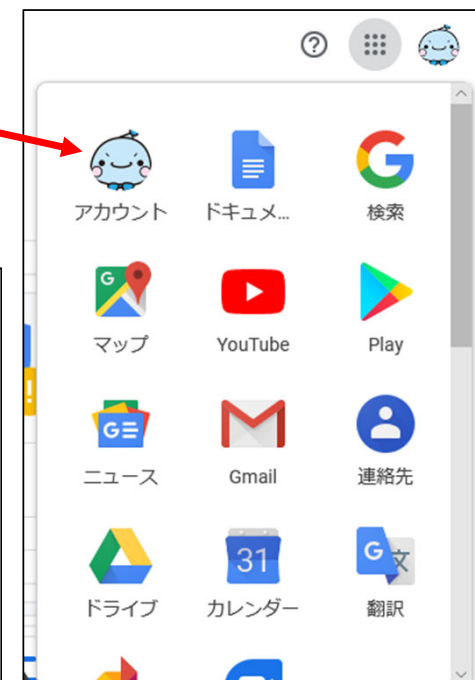


## ◆送信に使うGmailアカウント側での設定

今回は送信アカウントとしてGmailアカウントを使用します。

アプリからGmailアカウントを利用する際はアプリパスワードが必要です。

- ① Googleの右上のアプリ一覧から「アカウント」を選択
- ② 左のメニューから「セキュリティ」を選択し、  
その中の「アプリパスワード」を選択する。



## ◆送信に使うGmailアカウント側での設定

- ③ Googleアカウントへのログインを求められる。(ログイン済みでも求められます。)
- ④ 「アプリを選択」のプルダウンで「その他」を選びます。
- ⑤ 分かりやすい名前を付けます。 ex) JavaMail
- ⑥ アプリパスワードが発行されるので、これを利用します。

※ポップアップを閉じるとPWが見れなくなりますので、気を付けてください。

アプリ パスワードを生成するアプリとデバイスを選択してください。

JavaMail用



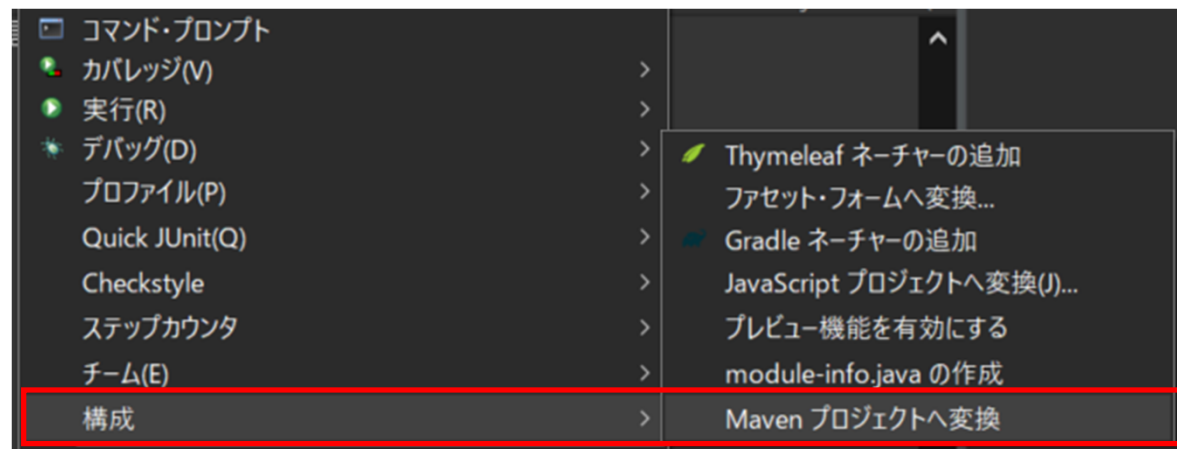
生成



## ◆Mavenプロジェクト化

Java 11 以上でJavaMailを使用するには下記の設定が必要です。

- ① プロジェクトを右クリック → 「構成」 → 「Mavenプロジェクトへ変換」
- ② そのまま完了を押す。
- ③ pom.xml が生成される。
- ④ pom.xml の<version>タグと<build>タグの間に下記の情報を追記します。



```
<dependencies>
  <dependency>
    <groupId>com. sun. mail</groupId>
    <artifactId>javax. mail</artifactId>
    <version>1. 6. 2</version>
  </dependency>
</dependencies>
```



## ◆練習問題 (必須課題)

問1: Stringクラスのconcat()を使って下記の2つの文字列を結合して、表示せよ。concat()の使い方はAPIで調べること。

```
String s1 = "abcd"
```

```
String s2 = "efgh"
```

```
abcdefgh
```



## ◆練習問題 (必須課題)

問2:「practice2.txt」に書かれている文字列から「q」が最初に出現する場所を探索して表示せよ。

※Stringクラスに用意されているメソッドを使用すること。

(APIを見て使えそうなメソッドを探すこと。)

※テキストファイルの中身をコピーしてString型の変数を作ってください。

26番目



## ◆練習問題 (必須課題)

問3: 標準入力から宛先、件名、本文を入力してメールを送信するアプリケーションを作成せよ。なお、送信元情報はプログラム中に直接記述して良いこととする。morijyobiアカウントを使用せず、このJava課題用にGmailアカウントを作成することをオススメします。

あくまで実行サンプルです。

動作確認は自分の持っているメールアドレスを使ってください。

※そうじゃないとうまく動いたか分からないよね？私にメールボムしないでね。

この注意書きを読まずに私にメールを送ってきた方は減点&晒します。

宛先: `y.takahashi2@morijyobi.ac.jp`

件名: `Javaの練習`

本文: `Java授業の課題でプログラムからメール送信をするロジックを作成！`

`メール送信完了`

※なお、Gmailからdocomo.ne.jpには送れない場合があります。



## ◆チャレンジ問題

問4: String型のArrayListを定義せよ。

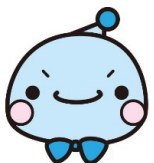
その後、下記の文字列を順番にaddせよ。

「first」「third」

最後に「second」という文字列を2つの「first」と「third」の間に挿入し、  
中身をすべて表示せよ。

末尾以外に追加する方法はArrayListのAPIから探すこと。

```
first  
second  
third
```



## ◆チャレンジ問題

問5: 標準入力から2つの値を入力し、1番目に入力された値の2番目に入力された値で累乗した値を表示せよ。

※Mathクラスのメソッドを使用すること。

1番目の入力: 2  
2番目の入力: 10  
結果: 1024





## ◆チャレンジ問題

問6: 下記の文字列の「a」すべて「e」に置換した文字列を表示せよ。

「I have a pen. I have a pineapple.」

※Stringクラスのメソッドを使用すること。

元の文字列: I have a pen. I have a pineapple.

置換後の文字列: I heve e pen. I heve e pineepple.



## ◆チャレンジ問題

問7:「practice7.txt」に書かれている文字列から「you」という単語の出現回数を検索するプログラムを作成せよ。

※テキストファイルの中身をコピーしてString型の変数を作ってください。

※文字列ではなく単語です。「yourself」等はカウント対象外です。

3回

