

Javaプログラマ育成コース

第四回 JDKライブラリーの利用

ソフトシンク株式会社

代表取締役

周 順彩

zhousc@soft-think.com

目次

- JDKとは
- JDKの分類
- Java.lang
- Java.text
- Java.util
- 演習
- 練習課題
- 次回の予習タスク

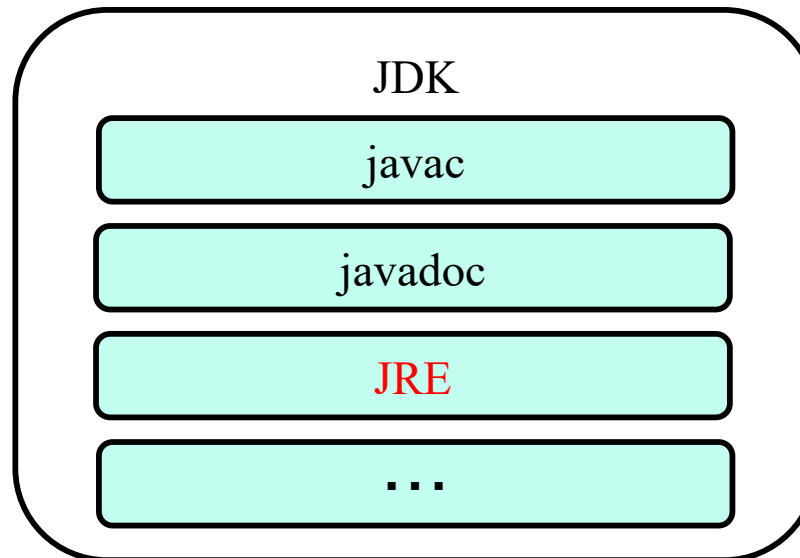
JDKとは？

◆ JDKとは？

Java Development Kit (JDK) は、Javaプログラムの開発、実行に必要なものの集合。
SUN(Oracleに買収されてからはOracle)より提供されるもの。
また、Oracleの方針変更より、JDK無償サポートは今後なくなり、フリを求めるなら、OpenJDKという選択肢しかない(半年ごとにバージョンが変わってしまうが)。

◆ JDKとJRE

JDK: Javaのプログラムを作るときに使うもの
JRE: Javaのプログラムを動かすときに使うもの



JDKのクラスライブラリ(Java9ベース)

◆ java.io

◆ java.lang

String、Math、スレッドやプロセスの基本的なランタイムサポートなど、Java プログラミング言語の設計の基礎となるクラスを提供。

◆ java.math

◆ java.net

◆ java.nio

◆ java.security

◆ java.text

テキスト、日付、数値、およびメッセージを自然言語に依存しない方法で処理するためのクラスとインタフェースを提供。

◆ java.time

◆ java.util

Collections Framework、書式付き印刷およびスキャン、配列操作ユーティリティ、イベントモデル、日付および時刻機能、国際化、その他のユーティリティークラスを提供。

java.langクラス: Object

◆ Object

クラス階層のルートです。すべてのクラスは、スーパークラスとして Object を持つ。つまりどのクラスも Object の子クラスとなる。

また、Object を継承する場合は、extends 句を書かなくても、継承が成立。明示的に書いても問題なし。

◆ よく利用するメソッド

- hashCode()

Object のハッシュコードを返す。

HashMap などハッシュ系のクラスのキーに利用される場合は、必ずオーバーライドする必要がある。また、本メソッドをオーバーライドした場合は、equals も同様にオーバーライドしなければならない。

※詳細原因は本メソッド一般契約事項に参照

- equals(Object)

二つのオブジェクトが "equals to" の関係が成立するかどうかを判定。

Object の子のメソッドは「return (this == obj)」でやっているだけなので、メモリ上の同一オブジェクトにしている場合のみ判定結果が true となる。

- toString()

オブジェクトの文字列の表現を返す。

ログ出力などで、情報を正しく得るには、このメソッドをオーバーライドする必要がある。

java.langクラス:String

◆ String

Stringは文字列を表すクラス。Stringで表す文字列が定数となる。一旦Stringのインスタンスを作成したら、内容を変更できない。Stringクラスはfinalクラスで継承できない。

◆ よく利用するメソッド

- String(byte[]/char[]/String/StringBuffer/StringBuilder)
- length()
- charAt(int ch)
- indexOf/lastIndexOf(int ch)
- substring(int beginIndex[, int endIndex])
- replace(CharSequence target, CharSequence replacement)
- replaceFirst/replaceAll(String regex, String replacement)
- split(String regex)
- valueOf(Object/char[]/boolean/int/char/long/float/double)
- equals()
- isEmpty()
- trim()

java.langクラス: StringBuilder

◆ StringBuilder

文字列を効率よく変更できるクラス。本クラスはスレッドセーフではない。

◆ よく利用するメソッド

- `StringBuilder(int capacity/String)`
文字列の長さが分かるなら、容量を指定してインスタンスを作成した方がパフォーマンスが良い。
- `append(Object/String/char[]/boolean/int/char/long/float/double)`
- `indexOf/lastIndexOf(String)`
- `delete(int start, int len)`
- `insert(int offset, Object/String/char[]/boolean/int/char/long/float/double)`
- `replace(int start, int end, String str)`
- `reverse()`
- `toString()`

java.langクラス: primitive型のラッパークラス

◆ Byte/Char/Short/Integer/Long/Float/Double/Boolean

対象のprimitive型のデータを表すオブジェクト型のラッパークラス。

すべてのクラスはStringと同じように、値は定数で、finalクラスで継承不可。

Byte/ Integer/Long/Float/Doubleの共通の親クラスはNumber。

◆ よく利用するメソッド

- parseByte/Char... (String)
- valueOf (byte/char ...)
- valueOf(String)

オブジェクトの作成はコンストラクタではなく、上記のメソッドを利用すべき。

- compareTo(Byte)
- hashCode/equals()
- toString()

java.langクラス: Math

◆ Math

数値処理(指数関数、対数関数、平方根、および三角関数など)用メソッドを提供するユーティリティクラス。インスタンスの作成はできない。

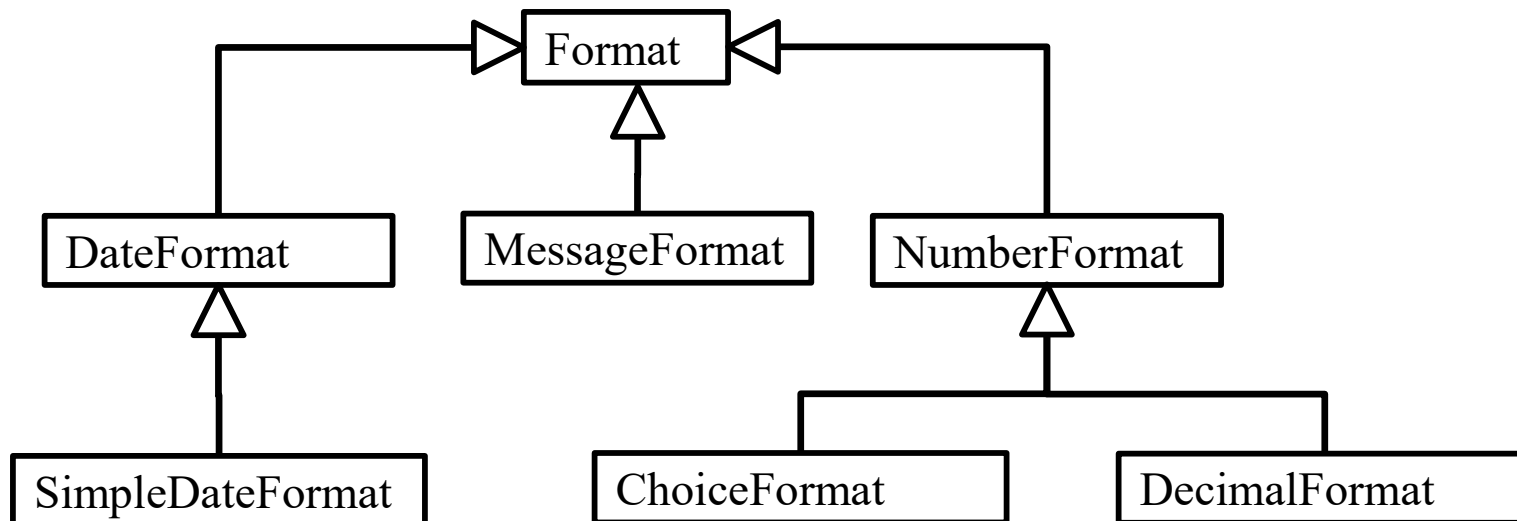
◆ よく利用するメソッド

- max/min(int/long/float/double a, int/long/float/double b)
- abs(int/long/float/double)
- floor/round()

端数処理。ただ、端数処理なら、BigDecimalを利用する。

java.textクラス／インターフェース:Format

- ◆ 日付、メッセージ、数字などのロケールに依存する情報をフォーマットするクラス

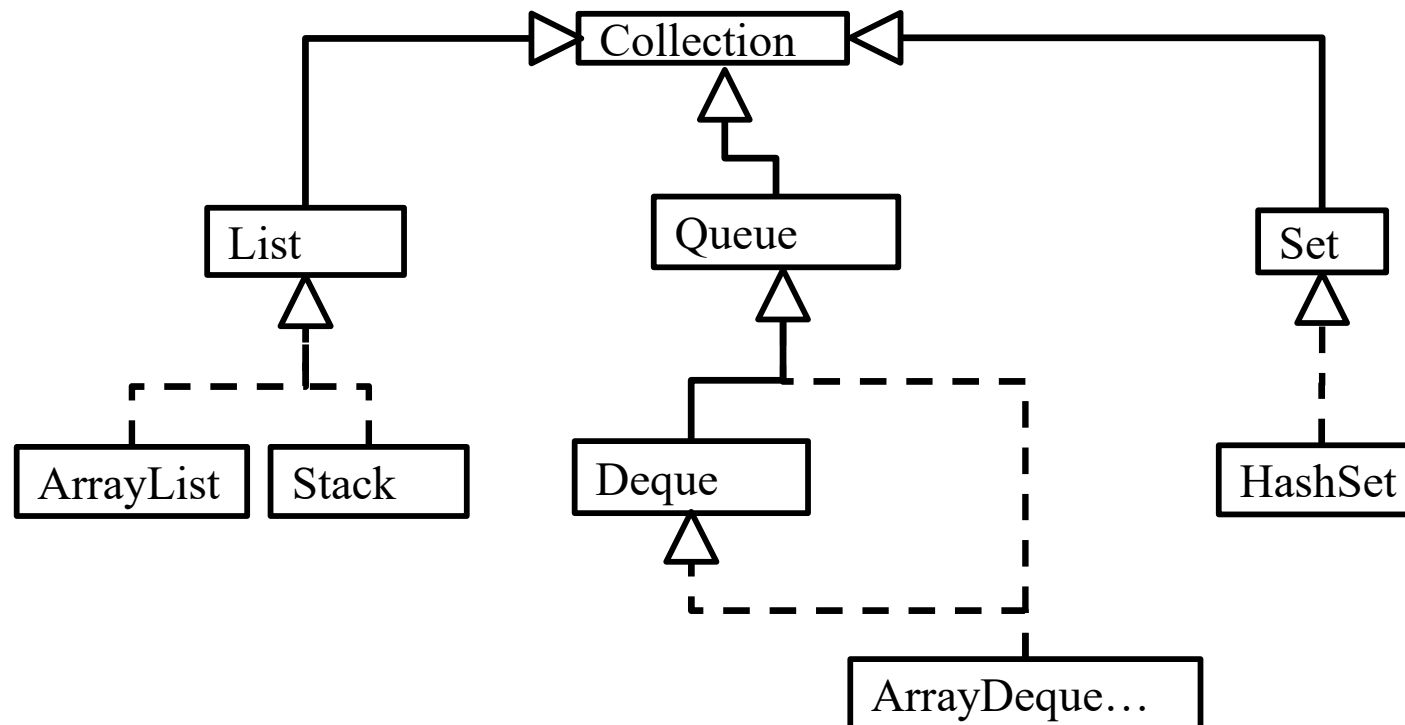


- ◆ 使用例

```
SimpleDateFormat sdf = new SimpleDateFormat("yyyy/MM/dd");
System.out.println("now:" + sdf.format(new Date()));
System.out.println("now:" + sdf.format(sdf.parse("2018/11/18")));
String messageTemplate = "{0}を入力してください。";
Object[] params = {"氏名"};
System.out.println("message:" + MessageFormat.format(messageTemplate, params));
```

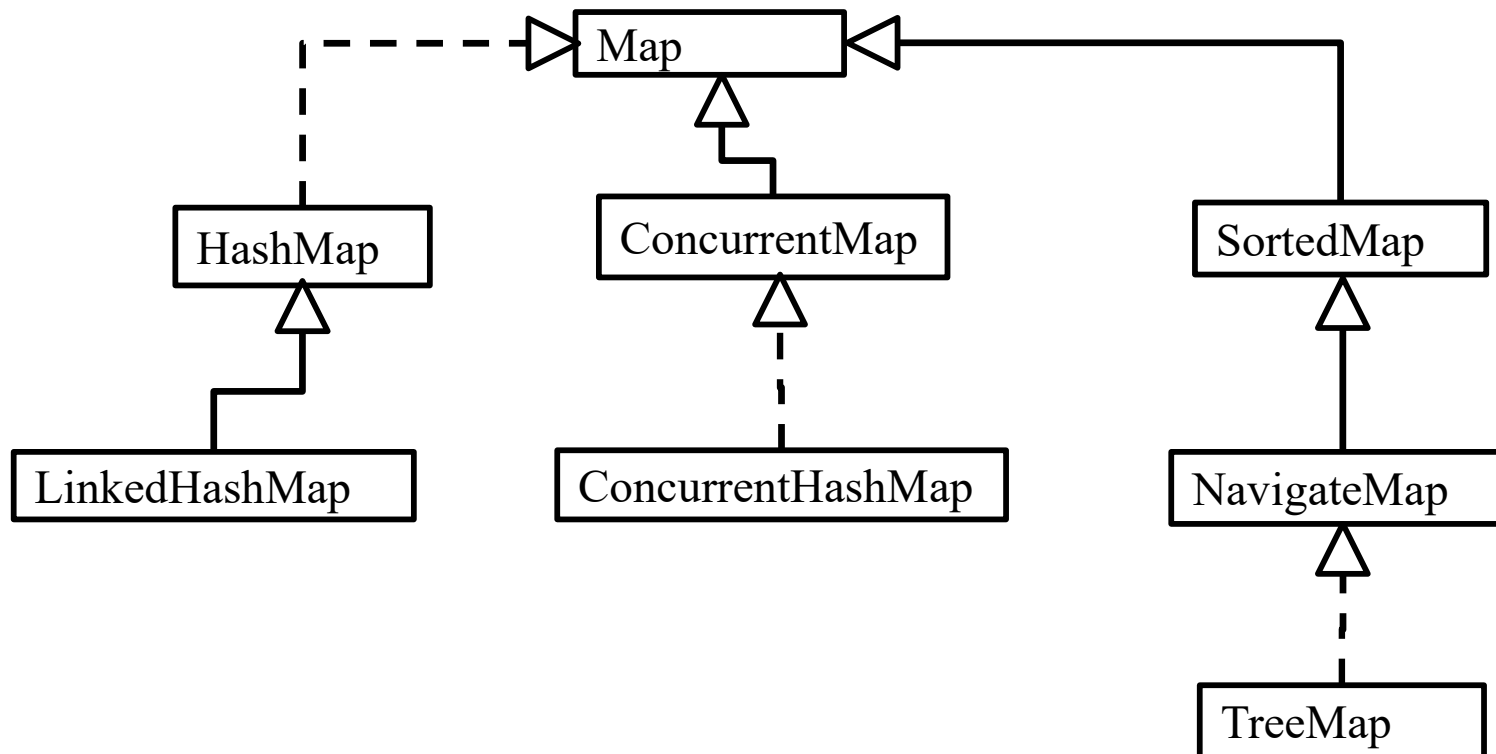
java.utilクラス／インターフェース: Collection

- ◆ コレクション階層のルート・インタフェース。その要素であるオブジェクトのグループを表す。



java.utilクラス／インターフェース：Map

- ◆ コレクション階層のルート・インタフェース。その要素であるオブジェクトのグループを表す。



HashMapのパフォーマンスの考え方はこちらにご参照：
<https://qiita.com/SoftThink/items/18cb839093a2600ea7cd>

演習

◆ FIFO仕組みの作成

- 対象オブジェクトが何番目なのか特定できる。

練習課題

◆ 四則演算器の拡張

- 括弧のサポート
- JDKライブラリの活用

次回の予習タスク

◆ 例外とJAVA I/O