JVMの内部構成

2002年5月13日

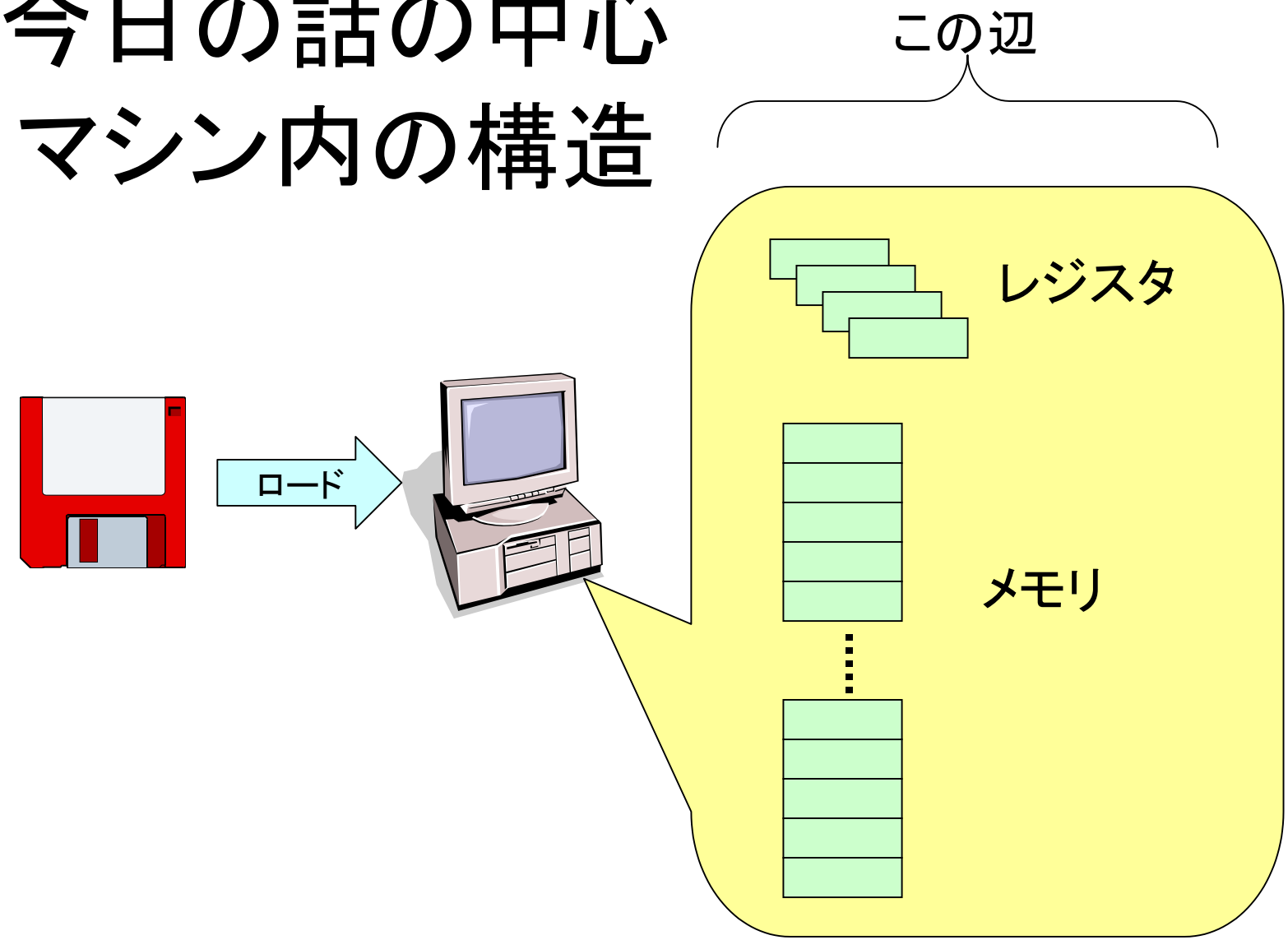
海谷 治彦

JVMという言葉の意味

- JavaVM仕様: JavaVMの動作等を規定した文書. リンク有
- JavaVM実装: 上記仕様に準拠して動くシステム(ソフト，ハード問わず).
- JavaVMランタイム: あるJVM実装の実行プロセス

教科書 p.4-5

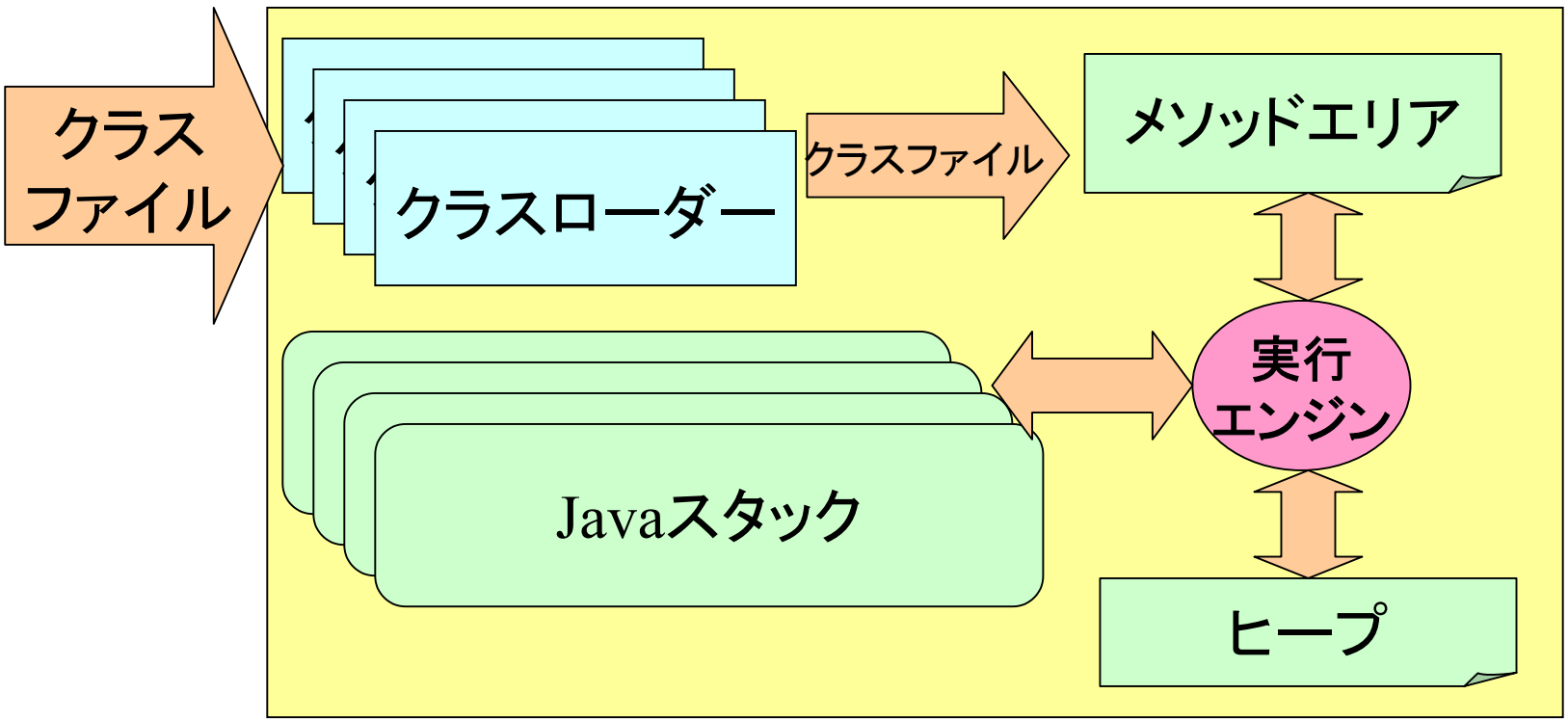
今日の話の中心
マシン内の構造



JVM内の基本構造(大雑把)

クラスファイルの
内容チェック

クラスデータを保存



各実行スレッドのローカルデータ
(実行経過)を保存

インスタンスデータ
を保存

教科書 p.15

* 原著および教科書p.15をベースに書いた.

リンク有

クラス・ローダー

- (クラスファイルに入っている)マシン語をマシン内(メモリ)に読み込むサブ・システム.
- (システムクラスローダーを除き)Javaプログラム自身で記述されているプログラムの一種.

- 詳細は本授業シリーズ後半にて.

教科書 p.14

メソッドエリア

- 読まれたクラスデータを保存しておく場所.
- 動的リンク，ゴミ集め(GC)，ネームスペース等の重要な技術が実装されている.
- 実体はクラスデータ保存用のメモリと思えばよい.

- 詳細は，「クラスファイルの構造」の回に.

教科書 p.16

ヒープ

- インスタンスデータの集積場所.
- インスタンス属性を保存する領域と思えばよい.
- よって，保存されるデータには型がある.
- 各メソッド内のローカル変数はここには置かれない. (Javaスタック内に置かれる)

教科書 p.16

例

```
public class GasStation implements Refuelable, Tradable{
    private int unitPrice;
    private int cbalance;
    private int remain;

    GasStation(){ remain = 0; }

    GasStation(int u, int b, int r){
        this(); initTrade(u, b);  inc(r);
    }

    public int dec(int s){
    int r=0;
        if(remain >= s){ remain -= s; r=s;}
        return r;
    }
```

青字の変数はヒープに入る.

赤字の変数はJavaスタックに入る.

Javaスタック

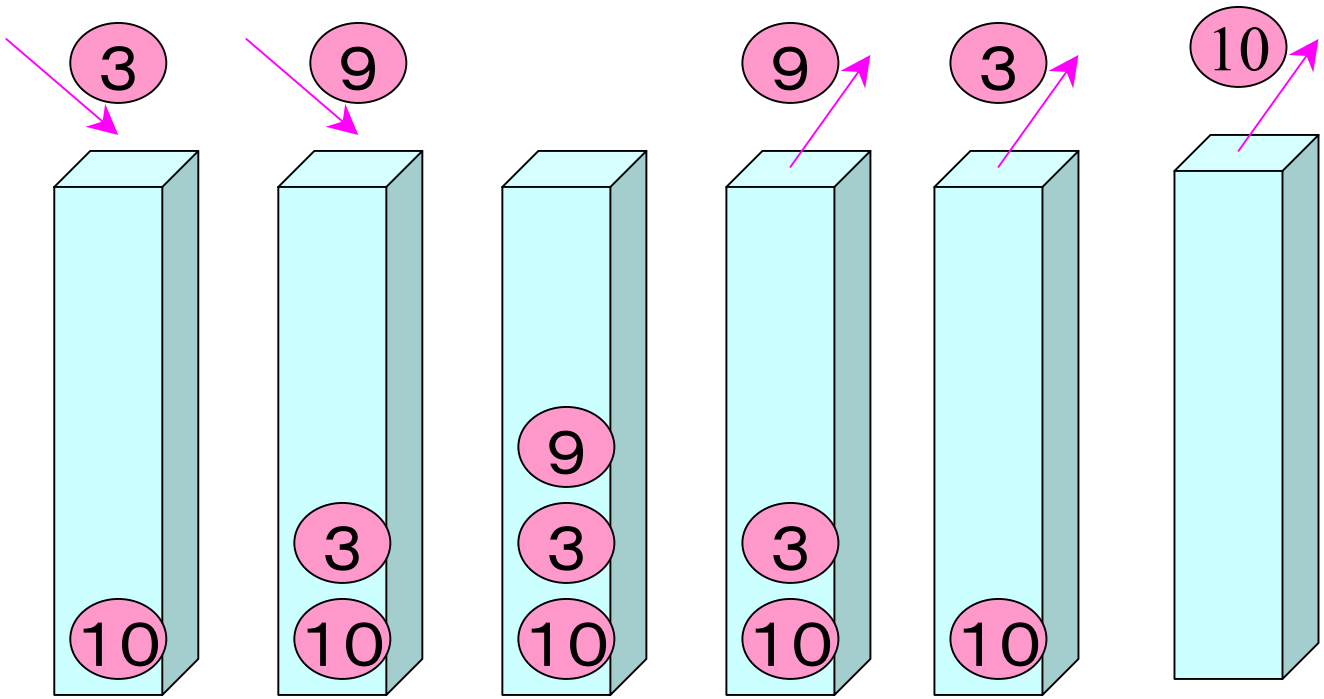
- 各スレッドが計算を行うためのメモリ領域.
- スレッドに1対1対応しているので動作中にその数は不定.
- スタックの名から分かるように，スタックに基づく計算を行う. (多分システムプログラムでやったでしょう.)
- 計算(関数呼び出しを含む)は逆ポーランド記法を利用している.

教科書 p.16-

スタックとは (多分，復習)

- 「積み重ねる」の意味.
- 計算機科学的には，「後から格納したデータが先に出てくる」集合型のデータ構造を指す.
- スタックの用語(というか基本操作)
  - push スタックに値を格納する
  - pop スタックから値を出す

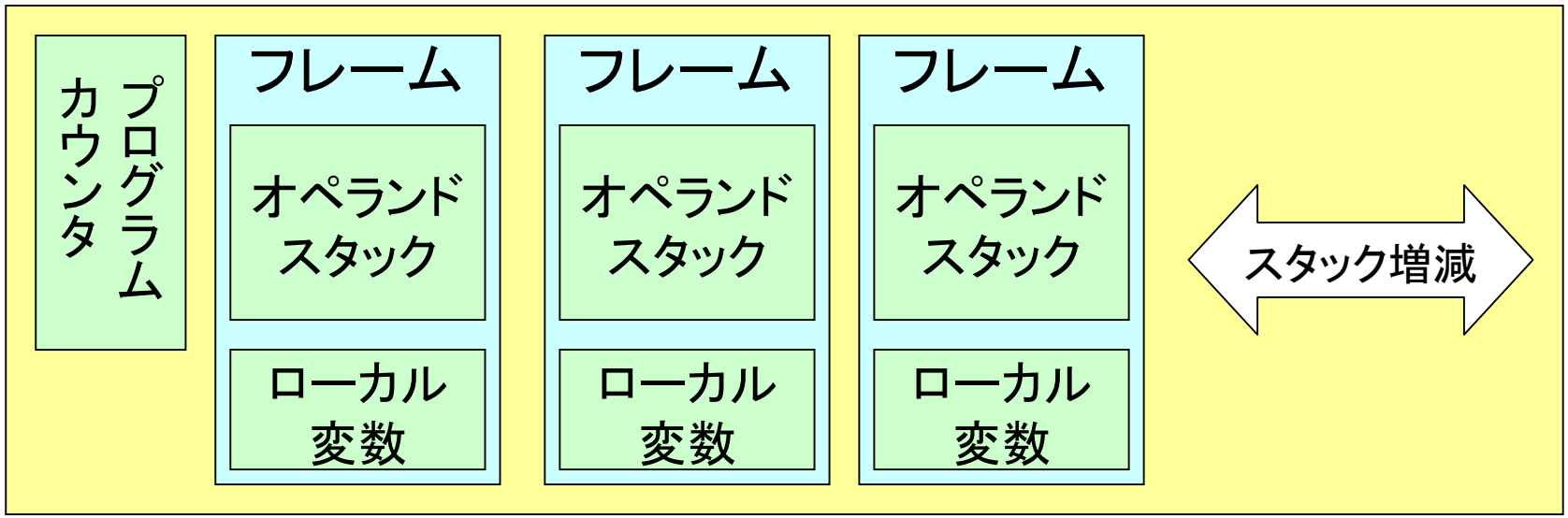
スタック操作の例



逆ポーランド記法とは

- 演算子を後に書く式の記述様式.
  - 例: (a+b)*c は a b + c * となる.
- 前述のスタックを使うと，すごく簡単に計算を実現できる.
- 関数(メソッド)呼び出しも広義には演算呼び出しと等価なので，Javaスタック内での内部表現として使われる.
- 詳細はリンクを参照.

Javaスタック内の構造



- 「フレーム」という要素のスタック.
- フレームは，1回のメソッド呼び出しに対応.
- フレーム内の計算のためにも，スタック（オペランドスタック）が利用されている.
- 詳細は「実行時の構造」の回にて.

例えば教科書 p.20の図

Java Byteコード

- JVMが解釈可能なマシン語
- クラスファイル内に入っているのもコレ
- Java Class File Format に準拠したデータ列
  - 別にファイルに入っている必然性はない.
- 主に以下の部分からなる.
  - 定数表(コンスタントプール)
  - 命令列(インストラクション)
  - 変数定義

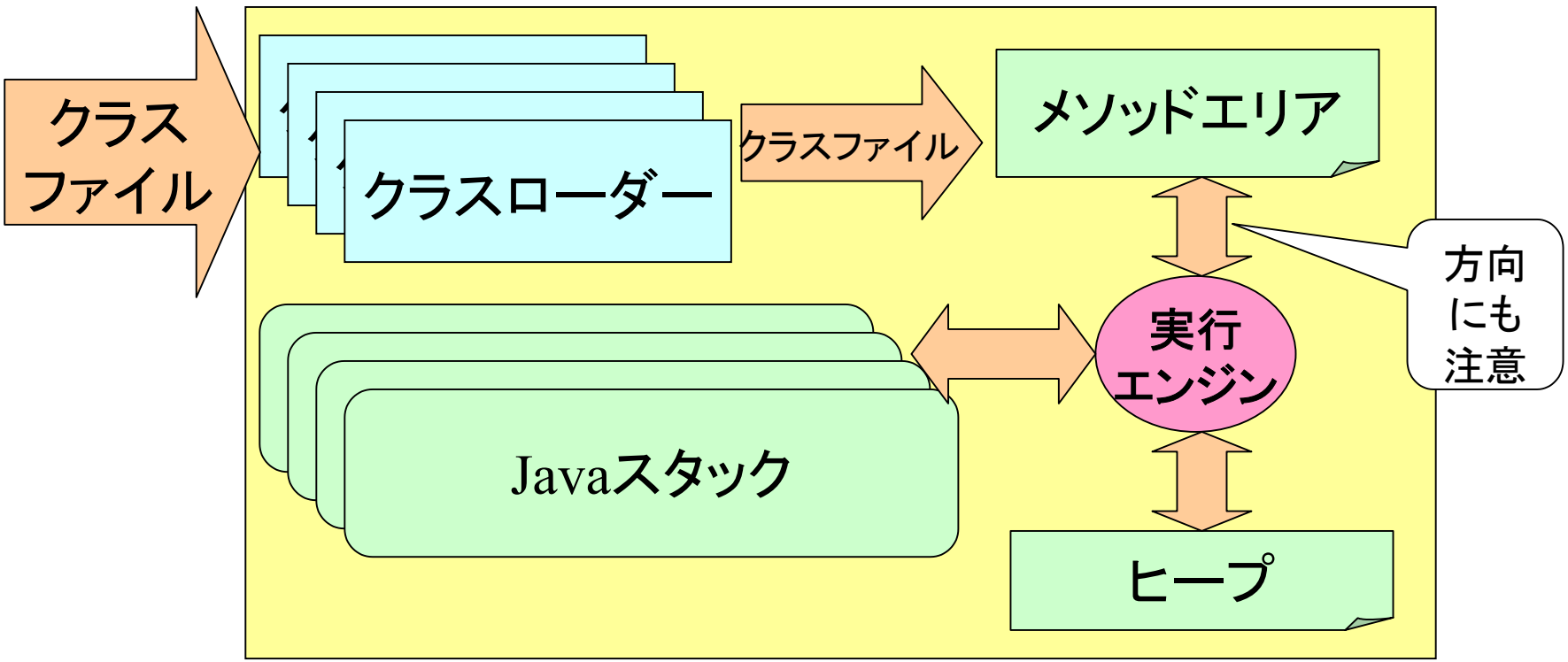
Byteコードのインストラクション

- Byteコード内の命令文の最小単位
- 教科書p.219以降に個々の詳細説明あり.
  - 例えば, p.317のiaddなど.
- 基本的には逆ポーランド記法を前提とした計算を行う.

以上，それぞれ説明しました

クラスファイルの
内容チェック

クラスデータを保存



各実行スレッドのローカルデータ
(実行経過)を保存

インスタンスデータ
を保存

* 原著および教科書p.15をベースに書いた. リンク有

実習: クラスロードを観察

- `java -verbose` クラス名
  - でロードするクラスの名(と出生)を確認できます.
- Hello World程度でも，沢山のクラスをロードしています. (jdk1.2.2 linuxで178個)
- 同じクラスが二度利用されても，二度はロードされてません. (重要)
- 詳細はリンク参照.