簡単なJavaプログラム

オブジェクト指向プログラミング特論

2016年度

只木進一：工学系研究科

Javaの特徴

- オブジェクト指向プログラミング言語
  - すべてクラスとして記述する
  - ファイル名そのものがクラス名になる
  - 文法はC++とほぼ同じ
- プラットフォーム依存が無い
  - JavaVMが動けば、どのOSでも動作
  - Windows、iOS、UNIX、Linuxなど
  - Jarファイルを複写して、他のOSで実行可能
- 多数のライブラリが言語とともに配布
  - GUI、FileIO、Thread、DBなどなど

実習

- SimpleSampleプロジェクトの生成
- firstSampleパッケージの生成
- NoClass.javaの作成
- バブルソートの作成と実行

「構築」または「ビルド」

- コンパイル
  - Program.java →Program.class
- jarファイルへ
  - *.class →プロジェクト名.jar
  - 必要なclassファイルをまとめる
- プロジェクトフォルダを見てみよう

- 保存時の自動コンパイル
- 整合性の確保のために
  - 「消去してビルド」
  - .classファイルを全て消して、再コンパイルとjar作成

実行

- NetBeansの中からの実行
  - プロジェクトのデフォルトの開始クラスから
  - main()のあるクラスを指定

実行：2

- コマンドラインから
- java -cp クラスパス クラス名 オプション
  - クラスパス：jarファイル
  - クラス名：main()メソッドのあるクラス名
- java -jar jarファイル
  - mainが定義されている場合

用語の導入 1

- クラス Class
  - データとその操作法(method)が組になったもの
  - 類型
  - 型に相当
- インスタンス Instance
  - クラスの実体化したもの
  - 型に対応する変数に相当

クラスを使わない例

- メインのクラスしか無い例題
  - NoClass.java
- プログラムの開始は
  - public static void main(String[] args)
    - argsはコマンドライン引数(後述)
  - mainに処理の詳細を書かないこと
    - コンストラクタを呼び、実行する

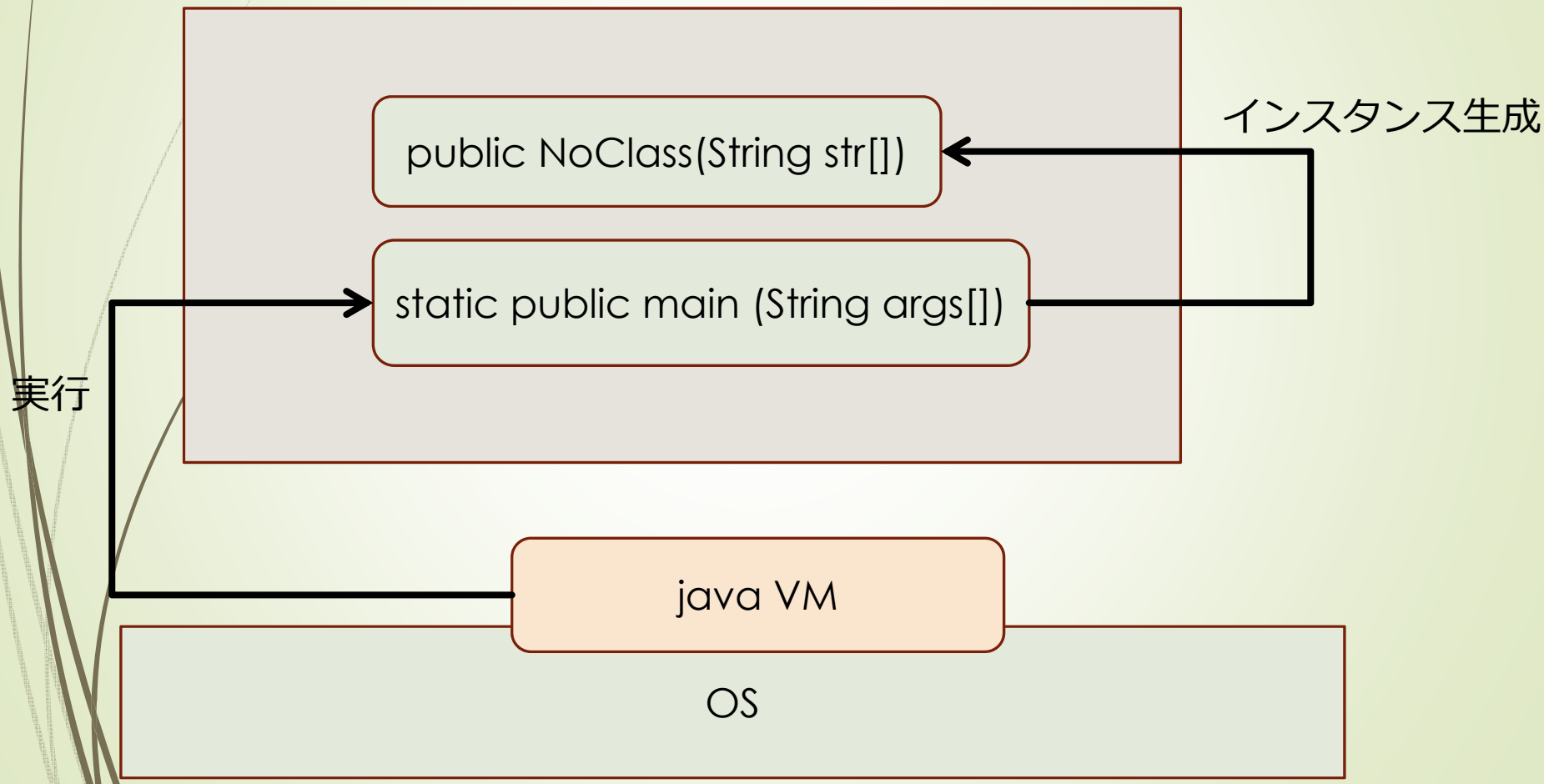
- コンストラクタ(constructor)
  - クラス名と同じメソッド
  - ここがinstance生成の場所
  - インスタンスを初期化
- this
  - 自身のインスタンス

簡単な説明

- C++と共通な部分
  - 式の表記、forやwhile、ifやswitch
  - メソッドの書き方
- C++と違う部分
  - pointerが無い
  - クラスインスタンスは全て参照
  - headerファイルが無い
  - デストラクタが無い
    - 自動ガベージコレクション
  - 配列もクラスオブジェクト
  - 文字列はStringというクラス

- package
  - クラスをグループ化・階層化
  - fieldへのアクセス制限で有効
    - 何も指定しないと、同一package内に対してpublic、他のpackageに対してprivate
- static宣言
  - クラスに属するメソッドやフィールド
    - インスタンスを作らなくても存在する
    - インスタンスを複数作っても、一つしか無い
- import
  - ライブラリの名前空間を導入

実行の仕組み


public NoClass(String str[])
インスタンス生成
static public main (String args[])
実行
java VM
OS

オブジェクト指向プログラミング

- モノ（オブジェクト）
  - 固有の属性を持つ
- モノの挙動・操作
  - 固有の属性を変更する
  - 固有の属性に基づいて反応する
- モノの類型→クラス
- 具体化したモノ→クラスインスタンス

オブジェクト指向の例 ショッピング

- 消費者が来店する
- 消費者がカートを持つ
- カートに商品を入れる
- レジで精算する

- 名詞をクラスとして考える

- 消費者のクラス
  - 支払い情報など
  - 使っているカート
- カートのクラス
  - 商品が入る
- 商品のクラス
  - 価格など

名詞・述語とクラス・メソッド

- 対象とするアプリケーションに現れる名詞⇒クラス
  - その操作や動作⇒メソッド
- メソッドの例
  - 消費者の行動
  - カートの操作

関数 (functions)

- 通常はクラスと結びつかない
- 変数を与えると、値が返る
- 関数の利用者の知らない影響（副作用）が無いように

- javaでは、static メソッドとして定義

サブルーチン (subroutines)

- 通常はクラスと結びつかない
- 変数に与えたものを加工する
- 関数の利用者の知らない影響（副作用）が無いように
  - 出力など、分かりやすいように
- javaでは、static メソッドとして定義

メソッド (methods)

- 通常は、インスタンスに付随
- メソッドの変数 (fields) に対して
  - 値を設定：setters
  - 値を取得：getters
- 処理を行って、結果としてfieldsを変更することが多い
- 名前の付け方が重要