

# JavaとOOP復習

## モデリングとシミュレーション

### 特論

1

2019年度

只木進一

# C++からJavaへ

- ▶ C++はCの構造体をクラスに拡張
  - ▶ Cのオブジェクト指向を追加
- ▶ Javaはオブジェクト指向を最初から目指している
- ▶ 代入、繰り返し、条件分岐はC++と同じ
  - ▶ java独自の拡張あり

# javaとクラス

- ▶ javaでは全てがクラス内部に
  - ▶ クラス外に、変数や関数を置くことは出来ない
  - ▶ 全ての関数は、クラスに結び付いたメソッド
  - ▶ mainメソッドを持つクラスから開始

# クラスとインスタンス

## ▶ class

- ▶ field (変数)とmethodが組合わされた定義されている類型

## ▶ instance

- ▶ classに実際に値の入ったモノ

## ▶ 注：型と変数

# 変数渡し

- ▶ 原始型変数は値渡し
  - ▶ 原始型 : int double char boolean
- ▶ クラスインスタンスは参照渡し
- ▶ 原始型には対応するクラスが存在
  - ▶ Integer Double Character Boolean
- ▶ ポインタは無い

# 一つのファイルに一つのクラス

- 一つのファイルに一つのクラスが原則
- ファイル名とクラスが一致

# static fieldとstatic method

- ▶ staticを指定すると、そのfieldやmethodはクラスそのものに結び付く
  - ▶ インスタンスではない
  - ▶ インスタンスが生成されなくても利用できる
- ▶ mainメソッドはstaticになっている

- ▶ staticメソッドから使えるのは
  - ▶ static メソッド
  - ▶ static フィールド
  - ▶ クラスインスタンス

# static methodの使い方

- ➡ 数学関数
  - ➡ 引数だけから、結果を返す
- ➡ サブルーチン
  - ➡ 引数だけから、処理を実行
- ➡ クラスの他の部分に影響しない
- ➡ さらに、static private methodを利用してもよい。

- 数学関数やサブルーチンなどの汎用メソッド群だけからなるクラス
- インスタンスを作らないのが普通
- コンストラクタをprivateにして、インスタンス生成を抑制する

# static fieldの使い方

- ▶ クラスに付随する
- ▶ インスタンスを生成すると共通の変数となる
- ▶ 定数として利用する

# javaの利点

- 豊富なライブラリが言語とともに配布される
- プラットフォーム独立
  - VM上で動作
- 上位互換
  - 一度書けば、ずっと動かせる

# javaの利点

## クラスの継承と抽象クラス

- ▶ 一つのクラスしか継承できない
  - ▶ Interfaceは複数継承可能
- ▶ Abstract Class
  - ▶ クラスの原型
- ▶ Interface
  - ▶ 他のクラスからの見え方の類型

# javaの利点

## class template

- ▶ 対象クラスを一般化できる
  - ▶ リスト、スタックなどのライブラリ
- ▶ 対象に依存しない操作を定義できる

## その他

- ▶ 例外処理
- ▶ 拡張されたforループ
- ▶ ラムダ式
- ▶ ダイヤモンド記法