

Java入門

モデリングとシミュレーション

2019年度

1

オブジェクト指向

Object Oriented Programming

- 対象をモノ(object)と、その運動・変化として捉える
- 対象システムを記述
 - 名詞をクラスとして捉える
 - 動詞をオブジェクトの操作、動作として捉える

クラス(class)とは

- ▶ クラス：モノを抽象化したもの
 - ▶ 普通名詞相当
 - ▶ 属性：数値や文字列など
 - ▶ メソッド：操作

メソッド(method)とは

- ▶ モノ(クラス)の動き
 - ▶ モノの操作
 - ▶ モノの変化
- ▶ メソッドとして記述
- ▶ **名詞に付随した述語**

クラスインスタンス(instance) とは

- クラスの具体化したもの
 - クラス⇔クラスインスタンス
 - 一般名詞⇔固有名詞
- instance : a particular example or case of something

クラスに属さない一般化された 述語

- ▶ 副作用を起こさないモノ
- ▶ 数学的関数
 - ▶ 引数の値に応じて値を返す
- ▶ サブルーチン
 - ▶ 引数の値を加工する
- ▶ 述語を集めたクラス(ライブラリ)のスタティックメソッド

クラスとインスタンスの例

- ▶ 学生を学籍番号と氏名で登録する
 - ▶ 学生というクラス
 - ▶ 属性としての学籍番号と氏名
- ▶ 具体的な学生がクラスインスタンス
 - ▶ 具体的に学籍番号と氏名が設定される

StudentSampleクラス

```
package studentSample0;

public class Student {
    public final String name;//氏名
    public final int studentID;//学籍番号

    //コンストラクタ、クラスインスタンスを生成する
    public Student (String name, int StudentID){
        //thisはコンストラクタで生成された「この」インスタンス
        this.name = name;
        this.studentID = StudentID;
    }
}
```


main()から起動

```
package studentSample0;
public class Main {
    //ここから実行が始まる
    public static void main(String[] args) {
        // Studentクラスインスタンスの生成
        //ここでは、クラスインスタンスの配列を生成
        Student students[] = {new Student("Aoyama",1),
                               new Student("Asou",2) };
        for (int i = 0; i < students.length; i++) {
            System.out.println(students[i].name); }
    }
}
```

Javaの特徴

- 文法はほぼC++と同じ
 - ポインタが無い
- すべてクラスで記述する
 - クラスインスタンスは参照型
- mainメソッドを持つクラスから起動
- OS非依存
 - “Write once, run anywhere”

- ▶ mainメソッドを有するクラスを起動する
- ▶ mainメソッドの役割
 - ▶ クラスインスタンスを生成して起動する
 - ▶ ここに、**処理の本体を書かないこと**
 - ▶ 修飾子staticの意味を考える

良いプログラムに向けて

- 一貫した名前の付け方：後述
- 適切なモジュール、クラスへの分割
- メソッドは短く
 - 一つのメソッドに一つの機能

Javaの命名規則

- ▶ 読みやすいプログラムを書くための重要な要素
 - ▶ 標準的な命名規則に従う
- ▶ クラス名は大文字で始める
- ▶ 変数名、インスタンス名は小文字で始める
 - ▶ 例外：定数は大文字
- ▶ パッケージ名は小文字で始める

Javaの命名規則 2

- ▶ クラスのインスタンスは、クラス名の最初を小文字にすると分かりやすい
 - ▶ 例 : Student student;
- ▶ Camel記法
 - ▶ 複数の英単語を結ぶ名称の場合、単語の最初を大文字としてスペースを入れずにつなぐ
 - ▶ 例 : StudentRecord

Javaの命名規則 3

setterとgetter

- クラスフィールドへのアクセスメソッドの標準的命名規則
- 値の設定
 - setフィールド名
- 値の取得
 - getフィールド名
- Netbeansの機能に従う