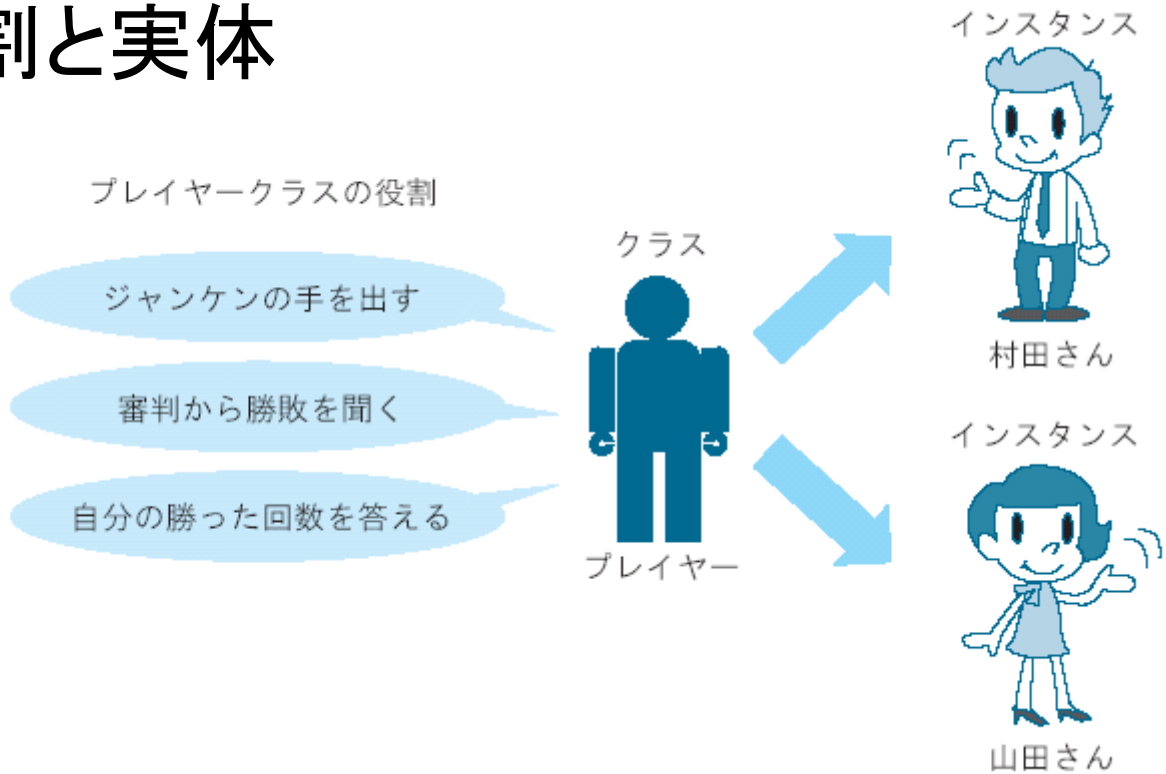
復習1 3章を中心に

海谷

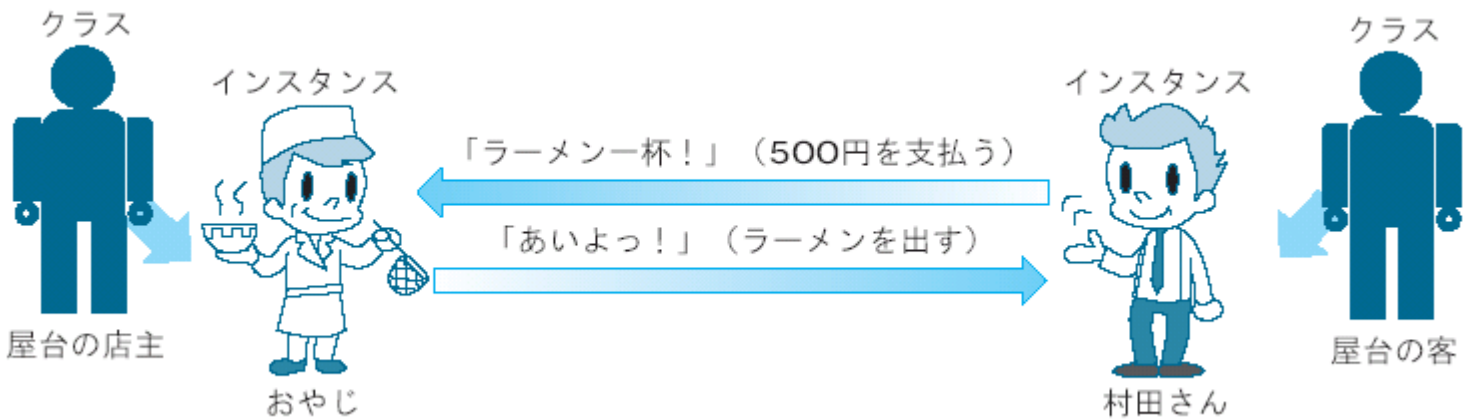
クラス/インスタンス

- p.52
- 役割と実体



メッセージパッシング

- p.54あたり
- Javaの場合，メッセージ送信は
  - 送り先のメソッド呼び出しとなる.
  - 返事は返り値となる.
  - 下記の場合，「おやじ.注文(500円)」に対して，返り値として「ラーメン」が返る.



操作(メソッド)のネーミング

- p.57にあるように，「操作」とは「メッセージをきっかにした，オブジェクトの振る舞い」
- Javaの場合，他のオブジェクトが操作を起動するという色彩が強い.
- よって，他のオブジェクトが「主語」となるような「述語/動詞」をメソッド名にするのがよい.
- 例/反例 at p.64 選手のメソッド
  - showHand() よりむしろ，getHand()がよい.
    - 審判が選手の手を得る
  - notifyResult()
    - 審判が結果(Result)を知らせる(notify).
  - getWinCount()
    - 審判が勝ち点(WinCount)を得る(get).

public, private そして何も無し

- p.68 アクセス修飾子
- public フィールド，メソッドが他のクラスに基づくオブジェクト(インスタンス)からアクセス可能.
- private 不能
- 何もなし．現時点ではpublicと同じ.

カプセル化 (p.68～)

- 通常，Javaではフィールドはprivateとする.
- 代わりにフィールドを変更しうるメソッドを(外部に)提供する.
- 理由
  - フィールド(値)の更新を管理，制限するため.
    - 例えば年齢データに負の数が入らないようにする.
  - 実際のデータ管理方法を外部に隠蔽するため.
    - 例えば数列を配列として保存するか，リスト構造として保存するかの影響範囲をクラス内に限定できる.

コンストラクタについて (p.70～)

- オブジェクトを生成する際の「 new Hoge()」は，コンストラクタという特殊なメソッドが呼び出されている.
- コンストラクタは特に自分で定義しなくても，引数無しのものが使えるようになっている.
- 通常のメソッド同様，引数があるものを定義できる.
  - 例 p.72 Player(String name)

オーバーロード overload

- 名前が同じだが引数や返り値の異なるメソッドを同じクラス内に定義できる.
  - C言語ではNGなのはご存知の通り.
- 同じ機能を担うメソッドに同じ名前を使えるので超便利.
- 標準API中にも多数例が見られる.
- コンストラクタには適用されている.

- 以降に出てくる「オーバーライド」とは異なる.