

オブジェクト指向

桑原 寛明
情報理工学部 情報システム学科

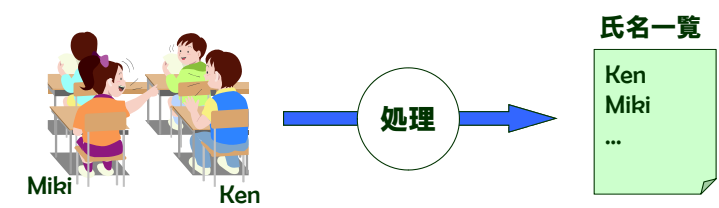
プロセス指向 vs. オブジェクト指向

■プロセス指向

- 計算機に何を実行させるかに注目
- 計算機の実行手順を命令で記述

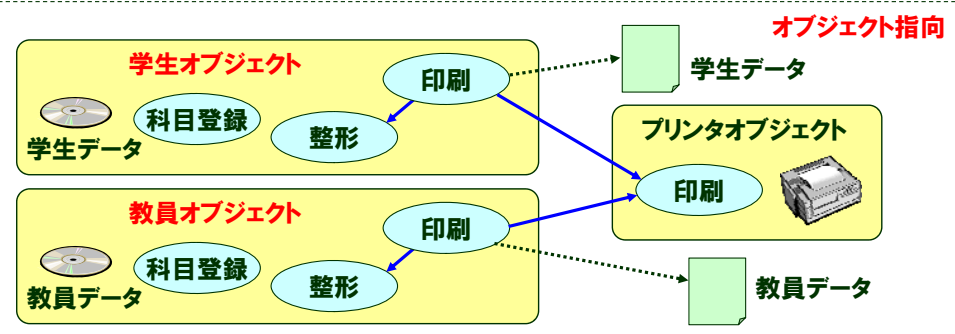
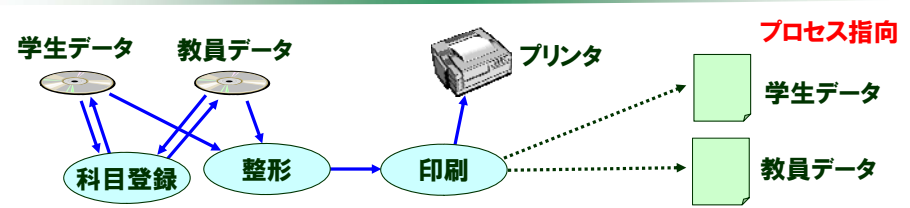
■オブジェクト指向

- オブジェクトの持つデータとオブジェクトに何を実行させるかに注目
- オブジェクトの実行手順を命令で記述
- 計算機はオブジェクトに実行を依頼



オブジェクト指向言語(2010年度)

プロセス指向 vs. オブジェクト指向の例



オブジェクト指向言語(2010年度)

オブジェクト指向(Object-orientation)

■実世界モデルをソフトウェアで直接的に表現する方法のひとつ

- オブジェクトによるモデリング
- 人間の認知方法にできるだけ近づけたモデル化
→ 人間にとって理解しやすい

オブジェクト指向のアプローチ データ中心のアプローチ

認知科学における概念

- 属性(attribute): 何からできているのか. 構造による認知
- 外延(extension): 何と似ているのか. 分類による認知
- 内包(intension): 何ができるのか. 機能による認知

機能(プロセス)中心のアプローチ

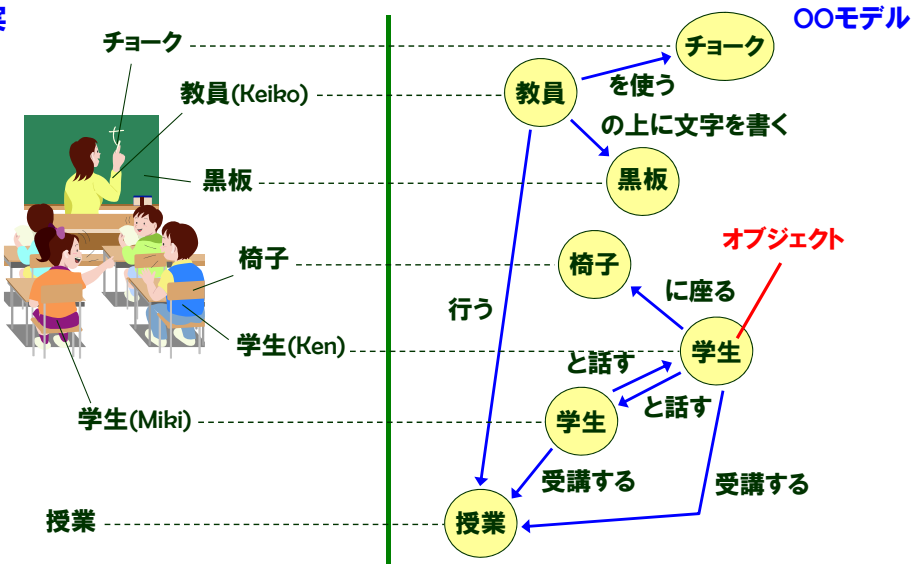
■オブジェクトを構成単位としてソフトウェアを構築する枠組み

- オブジェクト = モジュール

オブジェクト指向言語(2010年度)

オブジェクト指向によるモデリング

現実



オブジェクト指向言語(2010年度)

オブジェクト(Object)

■オブジェクト

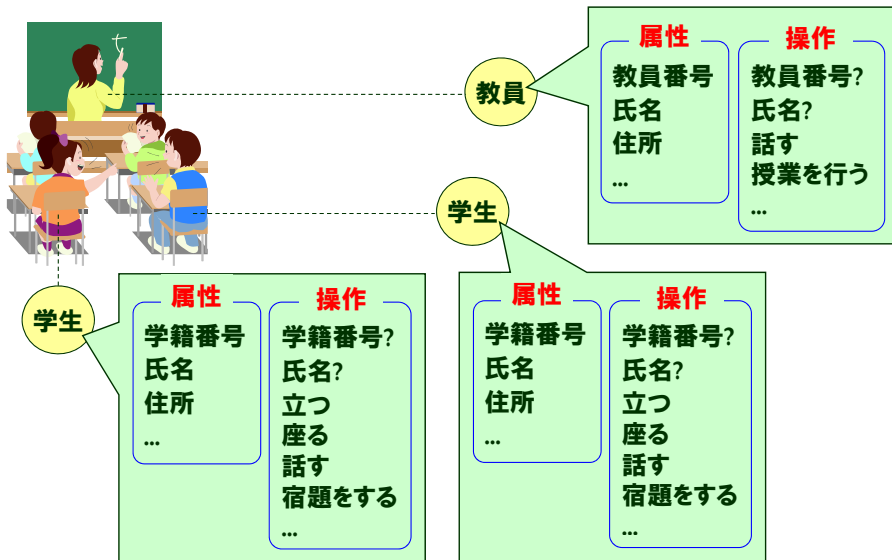
- 人間が認知できる具体的あるいは抽象的な「もの」
- 実世界の「もの」や「役割」などの事柄(thing)を抽象化した「もの」
- 物理的な「もの」、概念的な「もの」
- 境界と明確な責務(役割)を持ち、個別に識別可能な「もの」
- モデラーや開発するシステムに依存

■オブジェクトの持つ特性

- 状態**(state): オブジェクトの性質、オブジェクトの持つ情報
= **属性**(attribute), プロパティ(property)
- 振る舞い**(behavior): オブジェクトが実行できる動作
= **操作**(operation), メソッド(method)
- 識別性**(identity): オブジェクトを区別する手段

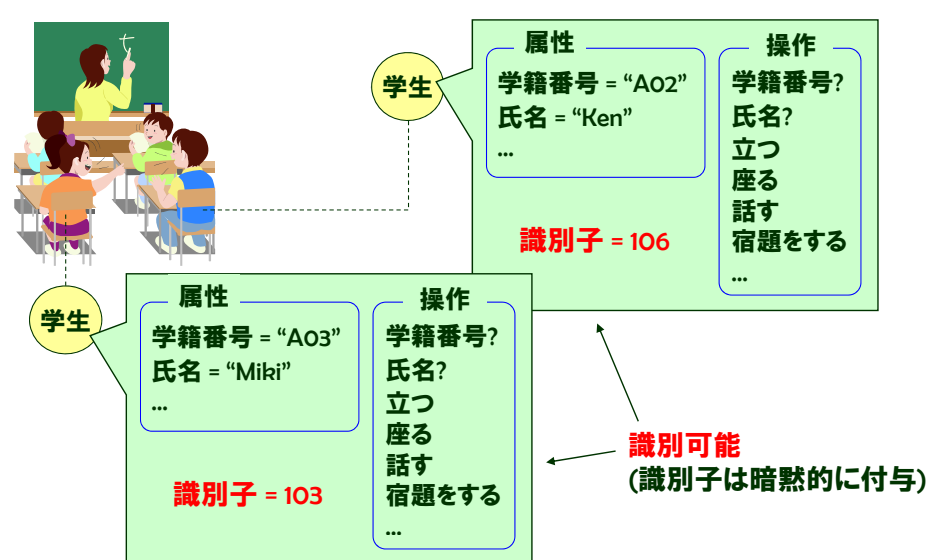
オブジェクト指向言語(2010年度)

属性と操作



オブジェクト指向言語(2010年度)

識別性



オブジェクト指向言語(2010年度)

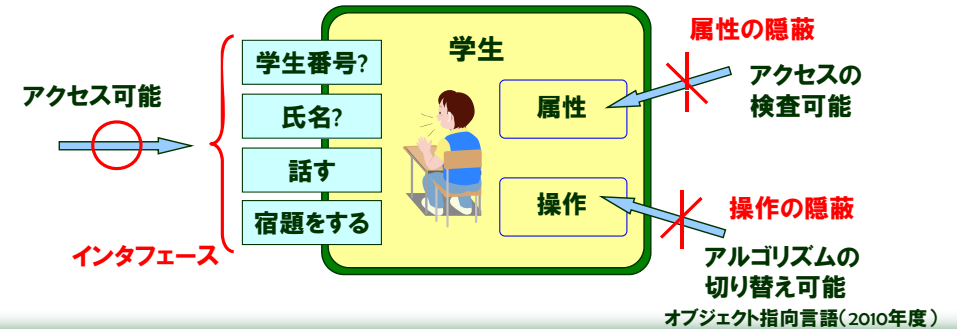
オブジェクト指向の基本概念(その1)

- カプセル化(encapsulation)
- メッセージパッシング(message passing)
- クラスとインスタンス(class & instance)
- 関連(association)

オブジェクト指向言語(2010年度)

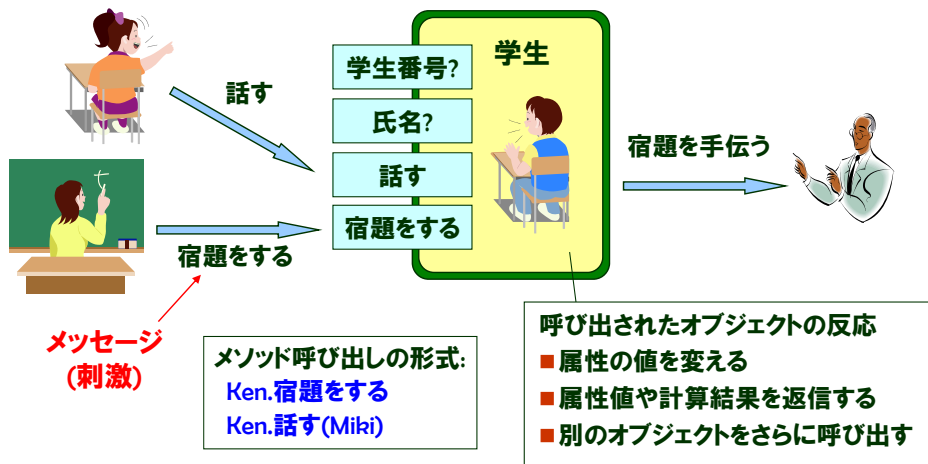
カプセル化

- 属性(データ)とその操作(手続き)をグループ化
 - オブジェクト内部をブラックボックス化
 - オブジェクトの中身を知る必要がない
- インタフェースと実装(属性と操作)の分離
 - 情報隠蔽(information hiding)
 - オブジェクトの中身を知ることができない



メッセージパッシング

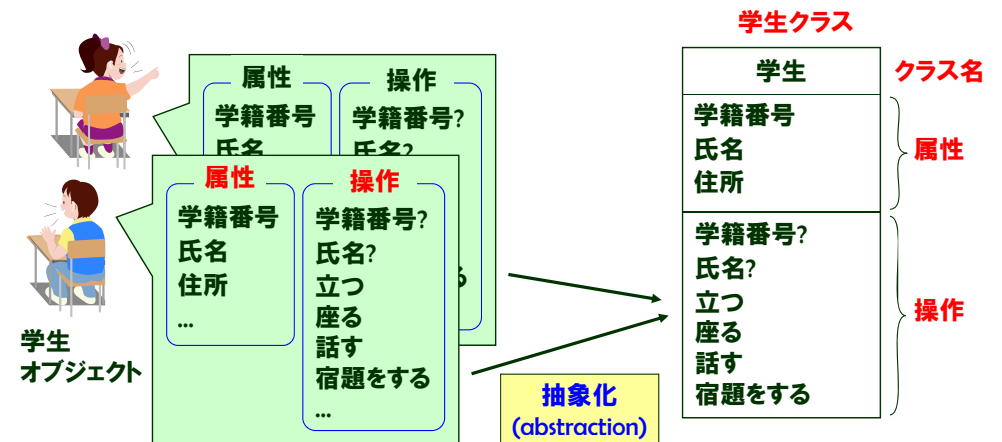
- オブジェクトに対して操作の実行を依頼する仕組み
 - オブジェクトに対するメソッド呼出し(method invocation)



オブジェクト指向言語(2010年度)

クラス

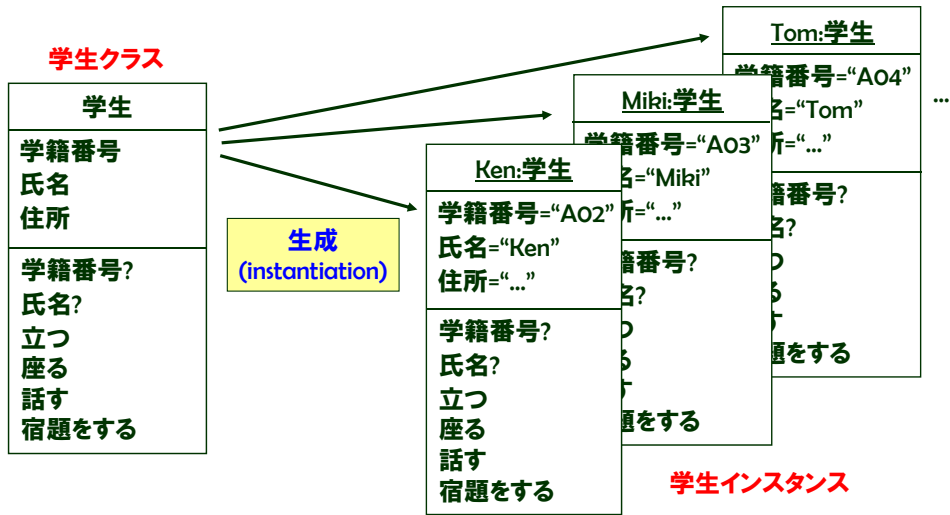
- 共通の属性と操作を持つオブジェクトを抽象化した雛形
- オブジェクトの設計図



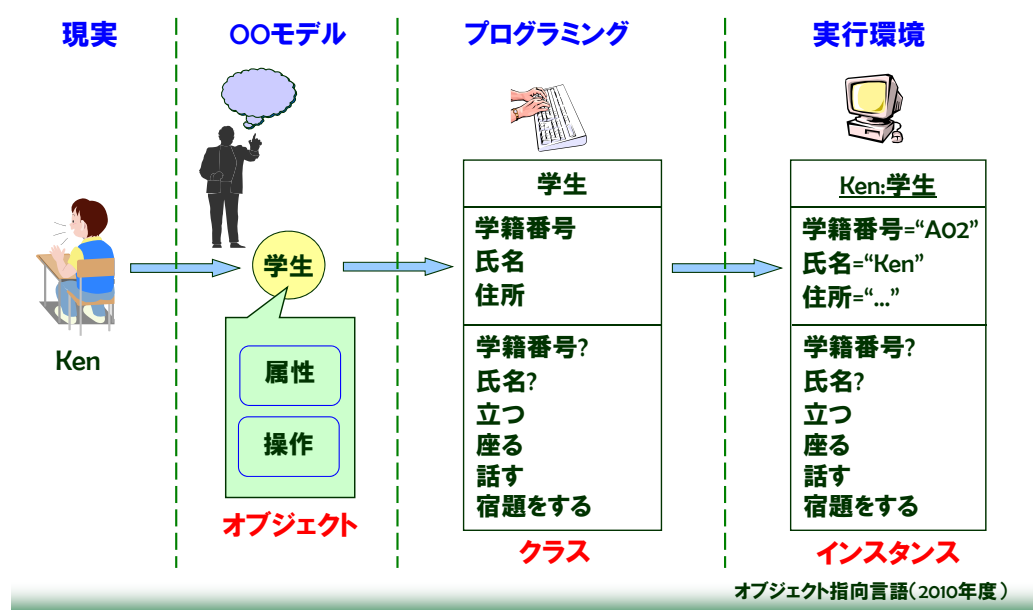
オブジェクト指向言語(2010年度)

インスタンス

■クラスから生成されたオブジェクト



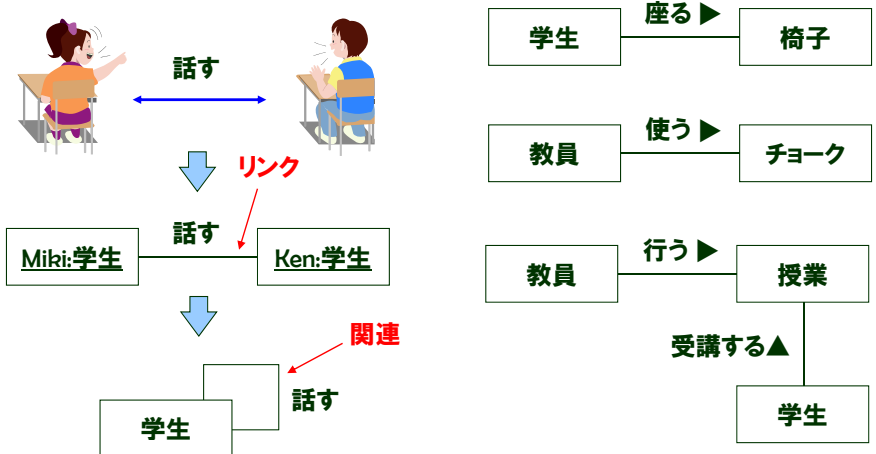
オブジェクト, クラス, インスタンス



関連

■クラス間の関係

□インスタンス間の関係 = リンク(link)



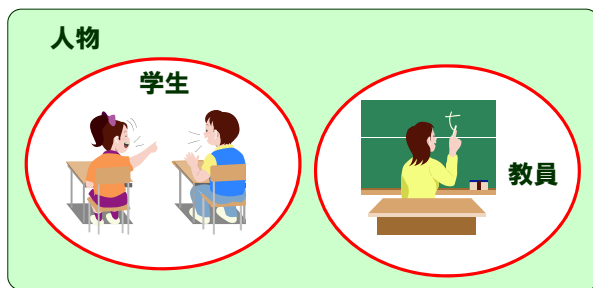
オブジェクト指向の基本概念(その2)

- 継承(inheritance)
- 集約(aggregation)
- 多相性/多態性(polymorphism)
 - 動的束縛(dynamic binding)

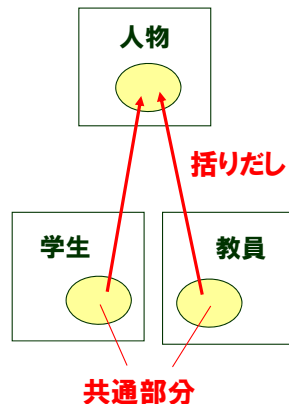
継承(1/3)

■複数のクラスにおける共通の性質を共有させる仕組み

- is-a関係
- 既存クラスの属性と操作の引継ぎ(と洗練)

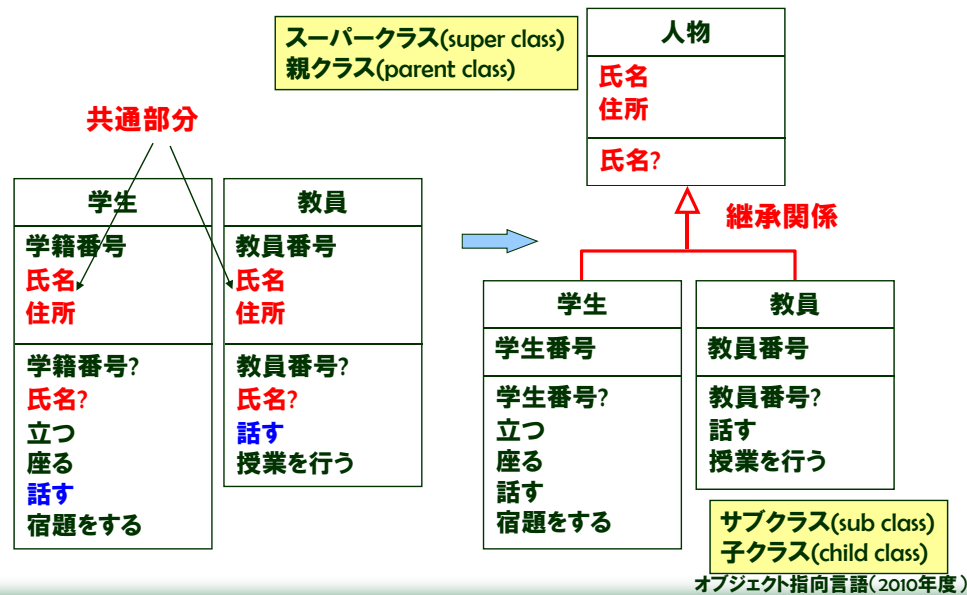


学生 is a 人物 (学生は人物である)
 教員 is a 人物 (教員は人物である)



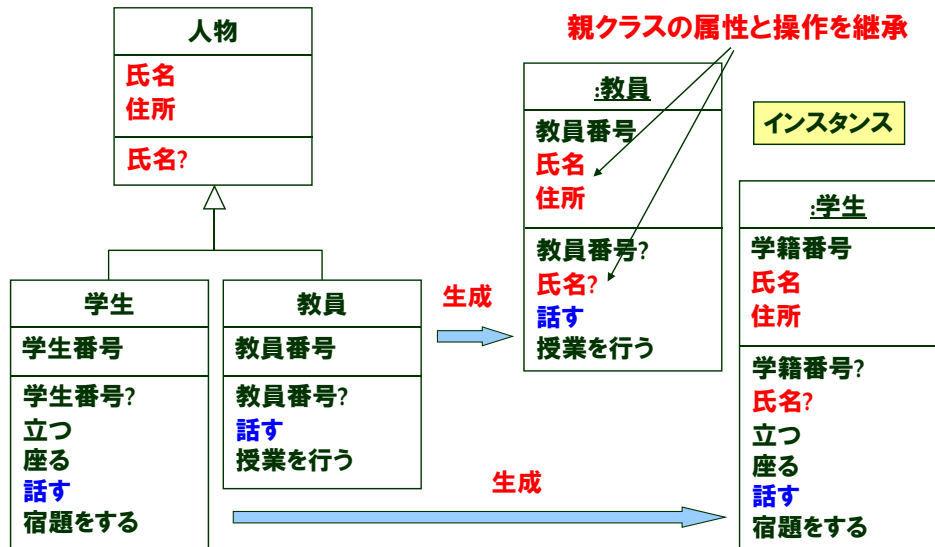
オブジェクト指向言語(2010年度)

継承(2/3)



オブジェクト指向言語(2010年度)

継承(3/3)

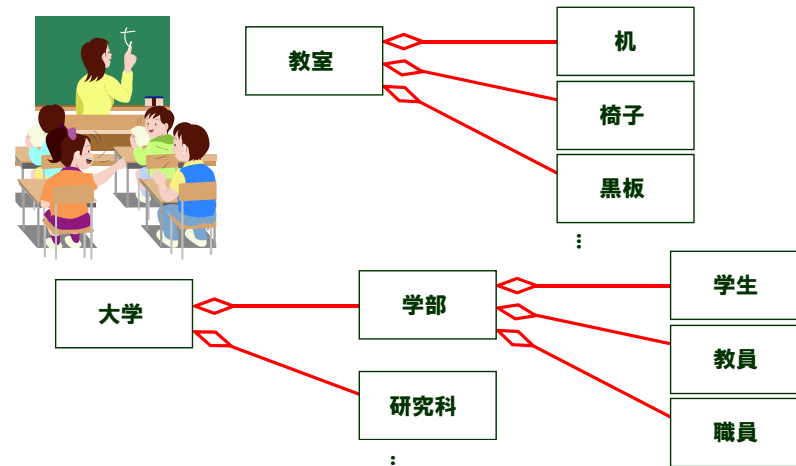


オブジェクト指向言語(2010年度)

集約

■複数のオブジェクトをひとまとめにして扱う仕組み

- part-of関係, has-a関係

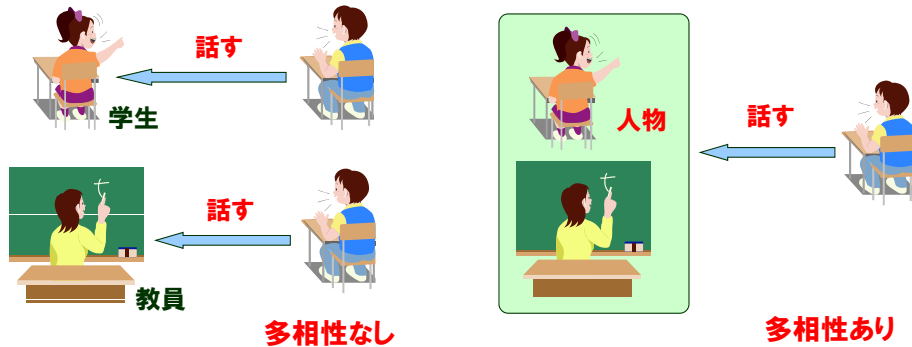


オブジェクト指向言語(2010年度)

多相性

- ひとつのオブジェクト(インスタンス)が複数のクラスに属することを可能とする仕組み

Miki(学生)とKeikoは人物クラスのインスタンスとみなせる



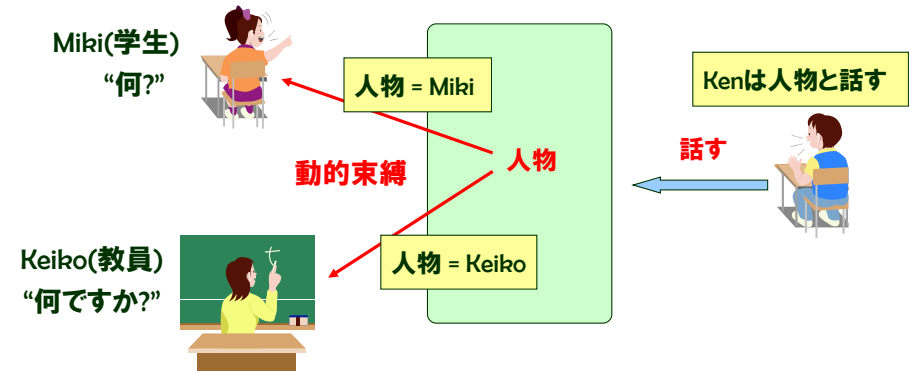
Ken(学生)はMiki(学生)と話す
Ken(学生)はKeiko(教員)と話す

Ken(学生)はMiki(人物)と話す
Ken(学生)はKeiko(人物)と話す

オブジェクト指向言語(2010年度)

動的束縛

- 応答するオブジェクトは受信側で実行時に決定
 - 呼び出し側は、呼び出され側を特定しなくてよい



オブジェクト指向言語(2010年度)

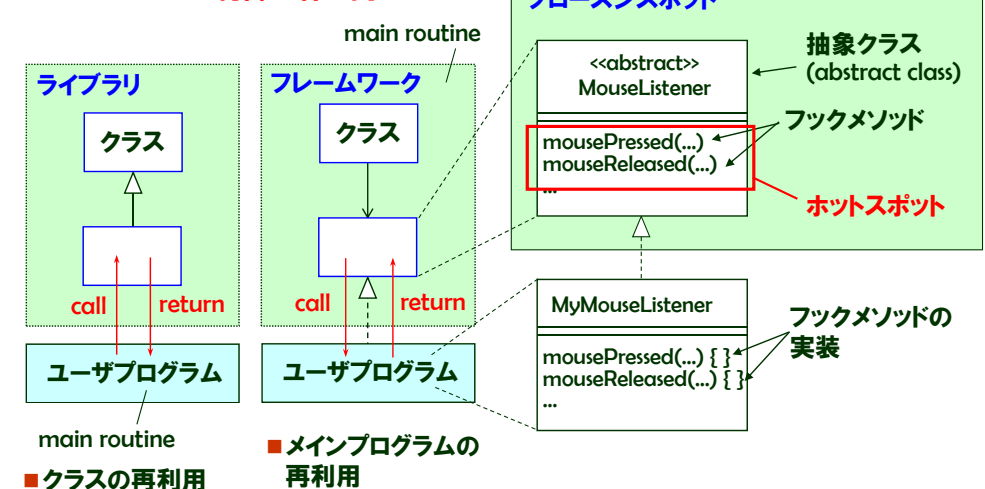
オブジェクト指向フレームワーク(OO framework)

- 再利用されるクラスとそのインスタンス間の相互作用を内包する設計(コードの再利用 + 設計の再利用)
- 開発者がカスタマイズ(拡張)できる半完成アプリケーション
 - Java AWT, Java Swing, MFC, .NET, IBM San Francisco, MacApp, Carbon, Cocoa, InterViews, ET++, ...
- アルゴリズム(固定部分)とその内容(可変部分)を分離
 - フローズンスポット(frozen spot): ドメインに対して標準的(固定)な部分, テンプレートメソッド(template method)で実装
 - ホットスポット(hot spot): 特定の要求に対応する柔軟(可変)な部分, フックメソッド(hook method)で実装

オブジェクト指向言語(2010年度)

制御の逆転(Inversion of control)

フレームワークが制御主体を内包



オブジェクト指向言語(2010年度)