

Gittok Lecture Note

02 モデルとその形式表現

太田守重
2014

実世界で起きる諸現象を、どのように記述するか。
自然言語には多義性があるので、表現に曖昧さが残る。
そこで、誤解が起きないように、厳密に定義されたルールに従う
人工の言語（スキーマ言語）を使用することがある。
その言語の代表例としてUMLがある。
ではUMLでどのように諸現象のモデルを記述するか。

モデル (model)

もの (object) や現象がもつ重要な特性の系統的な記述をモデルという。科学的なモデルは物理的、視覚的、数学的または計算可能な形態をとり、科学理論を構築するときに使われることが多い。

A systematic description of an object or phenomenon that shares important characteristics with the object or phenomenon. Scientific models can be material, visual, mathematical, or computational and are often used in the construction of scientific theories.

[The American Heritage® Science Dictionary
Copyright © 2002. Published by Houghton Mifflin.]



メタモデルとその共有

メタモデル：モデル表現のためのルールを記述するモデル

メタモデルの共有ができていなければ、モデルの共有はできない。

共有には合意が必要

関係する者たちが合意したルール（メタモデル）を標準という

De jure or de facto

例えば、国際標準化機構（International Organization for Standardization）ISOなどの標準化組織が法律に裏付けされた（de jure）標準の審議を行い、国際標準を制定。日本を含むWTO加盟国は、原則としてISO標準を国内標準としても使用し、政府調達ではこれに準拠することになっている。W3C（World Wide Web Consortium）などの民間団体やソフトベンダーが推奨する標準は、多くの人々が認めれば事実上の（de facto）標準と見なされる。

論議領域 (Universe of discourse)

論議領域:複数の人にとって、興味の対象となる実世界の範囲.

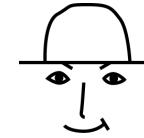
メタモデルを共有しても、論議領域を共有し、そこに含まれている物事をモデル化して共有しなければ、コミュニケーションは成立しない。例えば、クジラを想像しながらAさんが「動物」といったとき、Bさんは熊を想像したら、誤解が生まれるかもしれない。論議領域は共有するモデルに含まれる物事の範囲を決める。

論議領域の共有がなければ、コミュニケーションはできない。



Speak on the same universe of discourse!

地理空間情報のための標準(規格)



de jure 標準を規格という！

地理情報分野では、1994年に
例えば、国際標準化機構 (ISO: International organization of Standardization) が
専門委員会 TC211 を設置し、
電子的な地理情報交換のための
国際規格づくりを行っている。



The 35th Plenary and Working Group Meeting of ISO/TC 211,
Jeddah, Saudi Arabia, 8-12 December 2012

ISO/TC 211 による地理情報の国際規格

1. ISO 19100 シリーズ
2. デジタル／電子的地理情報の分野に関する標準化
3. 地球上の位置に直接ないし間接に関係づけられる対象物または現象に関する情報のために組織化された標準体系
4. 異なる利用者、システム、場所の間で地理情報を取得、処理、検索、表示、交換するための方法、ツール及びサービスの仕様化
5. 既存の、情報技術やデータに関する標準とのリンク
6. 地理データを使ったさまざまな応用システムの開発のための骨格の提供

<http://www.isotc211.org>

標準の記述には、誤解が起きないように、厳密に定義されたルールに従う人工の言語（スキーマ言語）を使用することがある。
その言語の代表例としてUMLがある。

スキーマ言語(UMLの場合)

UML: Unified Modeling Language

Grady Booch、James Rumbaugh、Ivar Jacobsonによって開発されたスキーマ言語.

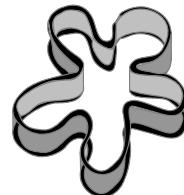
1997年11月にOMG (Object Modeling Group)によって標準として認定.

スキーマ:一定のルールに従って形式化したモデル

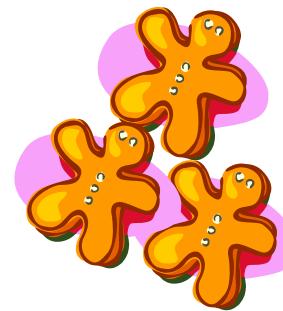
クラス:同じ性質をもつオブジェクトの型

オブジェクト:クラスに含まれるインスタンス(個物)

クラス(型)



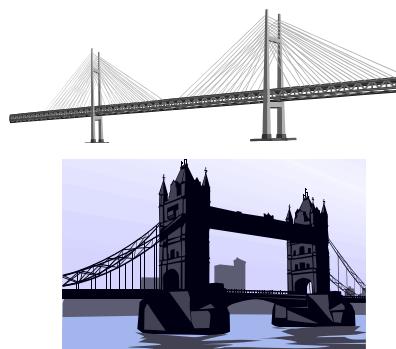
クラスはオブジェクトの性質を規定する



オブジェクト

クラス
橋

河川や海を横断
するための人工
構造物



横浜ベイブリッジ

タワーブリッジ

クラス定義の手順



概念化

私の自転車は、
通学用で、
車輪のサイズが24inch、
メーカーはBrightstone、
妹や母が使うときもある。

自転車
所有者:文字列
使用目的:文字列
サイズ:整数
メーカー:文字列
使用者[1..*]:文字列

仕様化:個物の記述の仕方を、誤解が
起きないように明示すること。

仕様化

自転車
所有者:String
使用目的:目的コード
サイズ:Integer
メーカー:メーカーコード
使用者[1..*]:String

概念化:複数の個別な事柄の共通点を取り上げ、対象となる事柄とは、こういうものだと説明すること

目的コード	メーカーコード
通学	Brightstone
通勤	PanaPana
遊び	RoadStar
スポーツ	

UMLクラス図

UMLでは、クラス図を使って、実世界の現象の抽象概念であるクラス及びクラス間の関係記述を行う。

クラスはその名称、属性及び操作などで定義する。

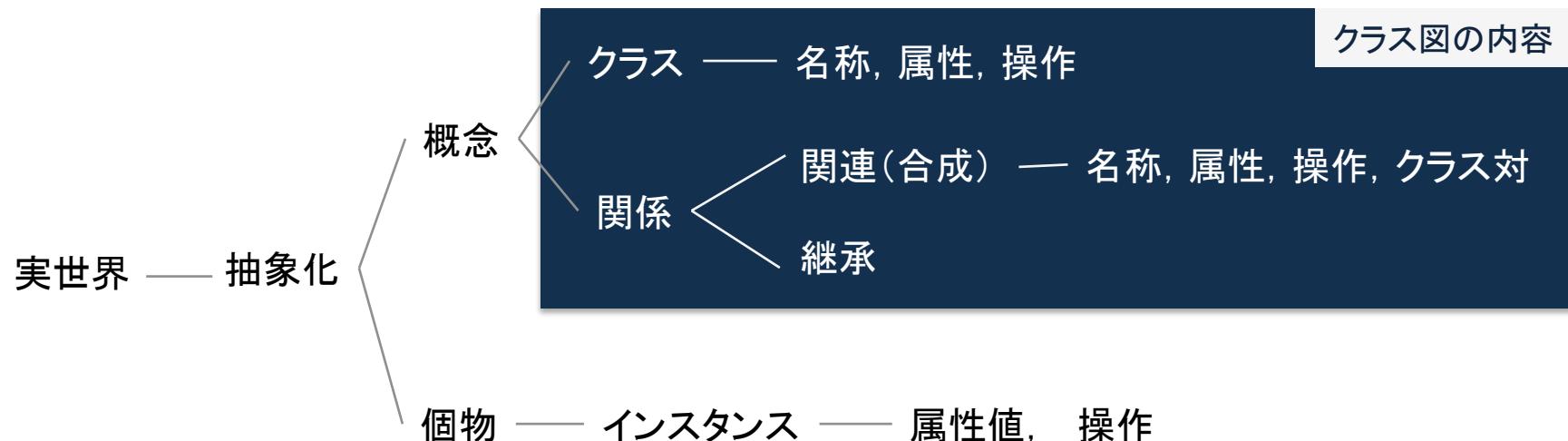
属性とは、そのクラス固有の性質のこと。

操作とは、属性の値を得るメソッドのこと。

クラス間の関係には、関連、特殊な関連である合成、そして継承がある。

関連(合成)は名称、属性、操作及びクラスの対などで定義する。

継承はより抽象度の高いクラスの属性や操作を受け継ぐことである。



クラス (class)

クラスは3つの欄で記述する。

名称: クラスを識別するための「名前」

属性: そのクラス固有の性質

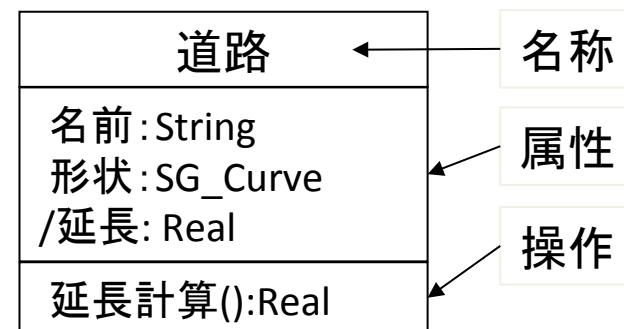
例: 量、質、状態、場所、時間、…

操作: 属性の値を求めるメソッド

例: 延長計算()

道路の形状を示す線データから、
道路の延長を計算し、値を戻す。

この例の場合は、属性「延長」に値が
入る。



/ : 別の場所で得た値が
入るという意味。このような属性は
派生 (derived) 属性という。

属性 (attribute)

属性: クラスがもつ固有の性質

- 属性名, 多重度及びデータ型で示す。

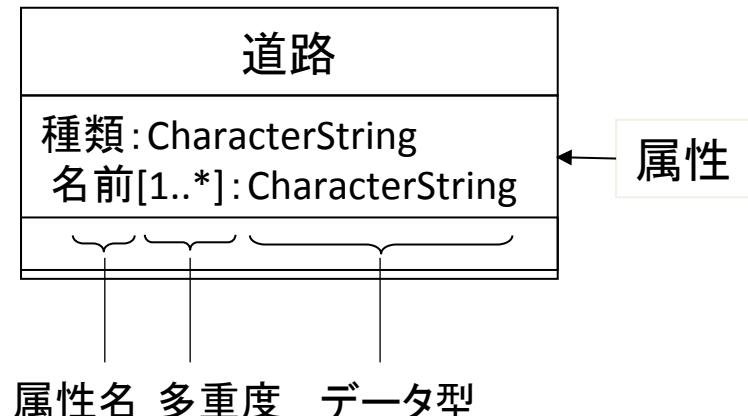
属性値が従うデータ型

- 単純データ型

例: CharacterString(文字列),
Integer(整数), Real(実数),
Bool(真偽値)

- 複合データ型

例: SG_Curve(曲線),
Coordinate2D(二次元座標)



多重度 (multiplicity)

- 同じ種類の属性が複数あるときに使う
- 属性名の後の[]の中に記入
 - [最小値...最大値]
- 多重度が1の場合は省略

【多重度の表現】	
クラスA	:0または1
属性名[0..1]:属性型	
クラスA	:1以上
属性名[1..*]:属性型	
(* は複数を表す。n と記述することもある) 「,(カンマ)」で区切ることで離散値を指定し、「..(2つのドット)」で区切ることで範囲を指定する。	

ただし, gittok では, 単数か複数かの区別しかできない.

操作 (operation)

操作

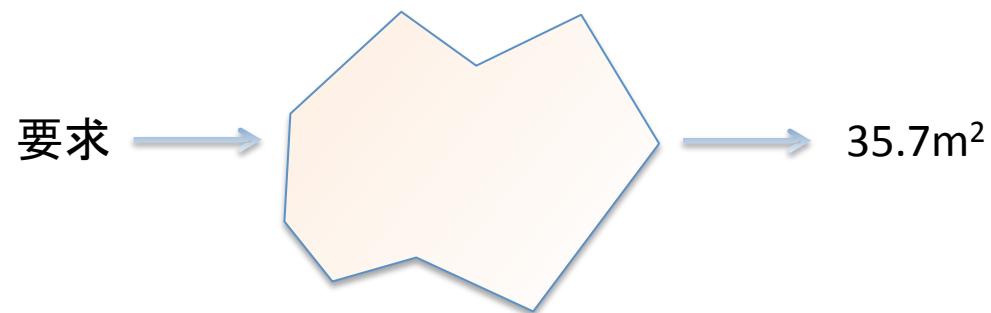
インスタンスが要求を受け取ると、特定の処理を行い、その時点の結果を要求元に返すメソッド。クラス図には、処理を行うプログラムは含まない。

操作の記法

操作名(文字列)

引数((属性名、属性型)の列)

戻り値のデータ型



gitkok では戻り値は、そ
の操作を持つ地物などの
派生属性に入れられる。



getArea (shape:SG_Surface): area: Real

操作名

引数

戻り値が入る派生属性と
そのデータ型

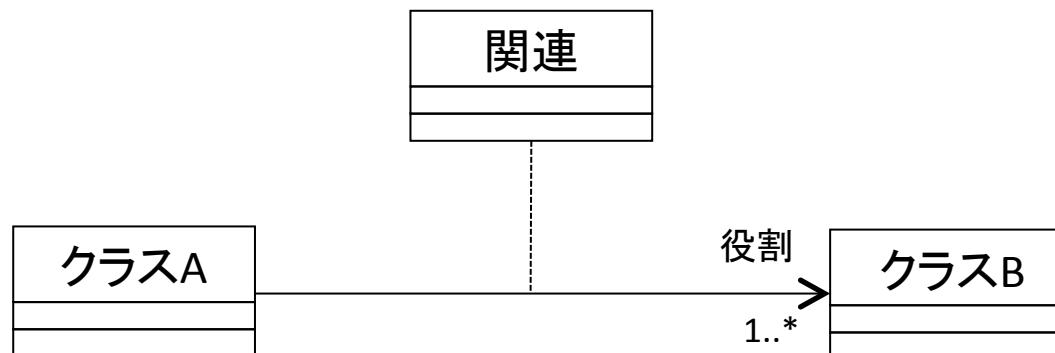
関連 (association)

関連とは、相手のクラスのもつ役割で結びつく対等な関係

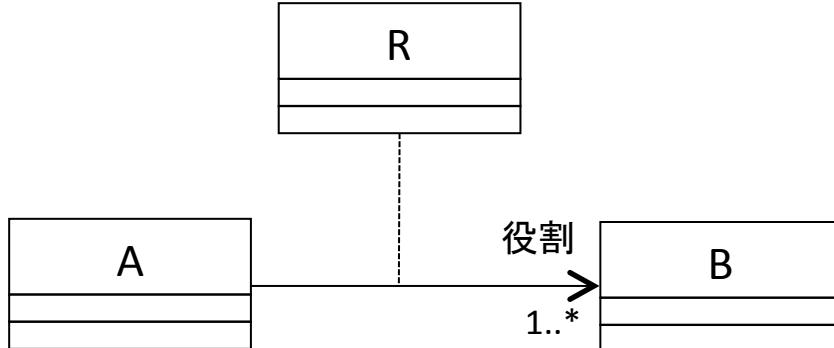
関連もクラスと同様に実世界の現象なので、名称、属性、操作をもつ。

関連先のクラスは、役割名をもつ（学校にとって、建物は校舎という役割をもつ）

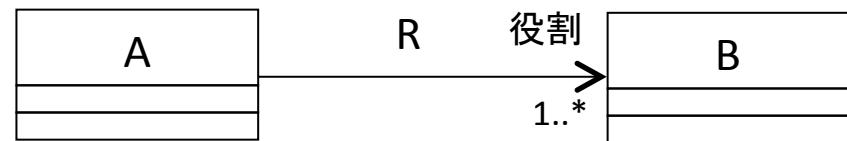
関連元クラスの1インスタンスに対して、関連先クラスのインスタンスが複数になることがある（学校の校舎は一棟とは限らない）



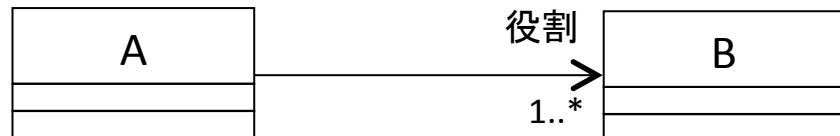
関連型の省略表記



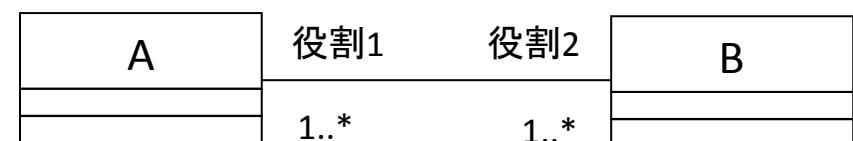
関連に属性・操作がなければ、
以下のように表記してよい。



関連名を省略して表記することもある。



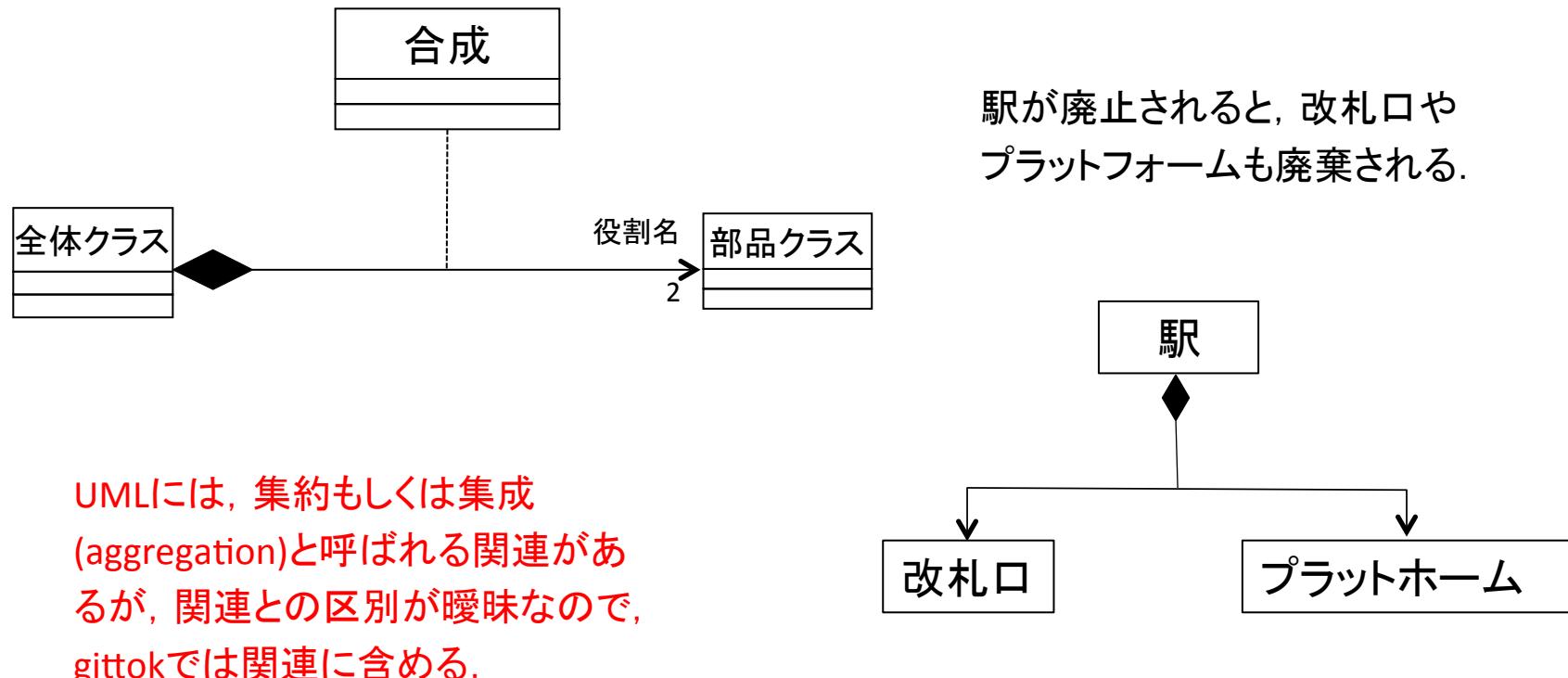
関連が双方向のときは、以下の
ような表記も許される。



合成 (composition)

合成は、強い関連。全体を示すインスタンスが削除されると相手（部品）も削除される。

記法：全体を表すクラスの側に「黒いダイヤモンド」をつける

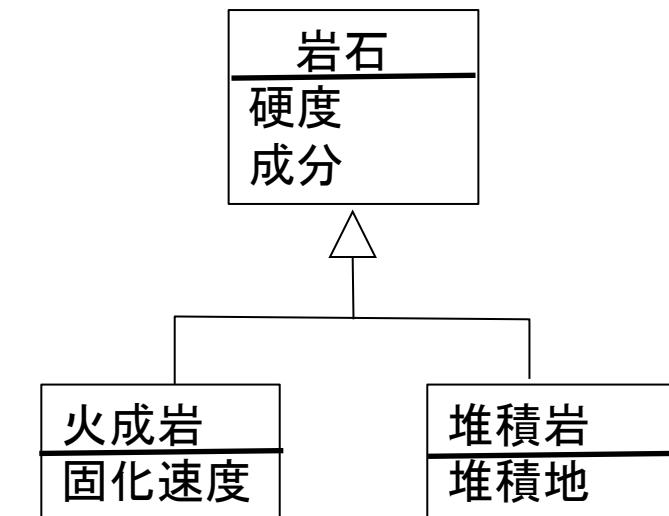
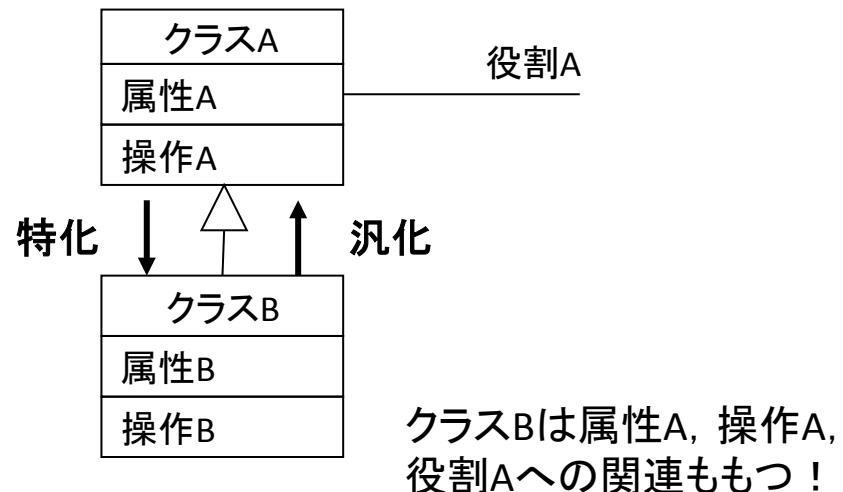


継承 (inheritance)

継承 とは、より抽象度の高いクラスと、その下位のクラスの関係
上位の属性、操作、及び関連を下位が受け継ぐ
is a の関係ともいう。

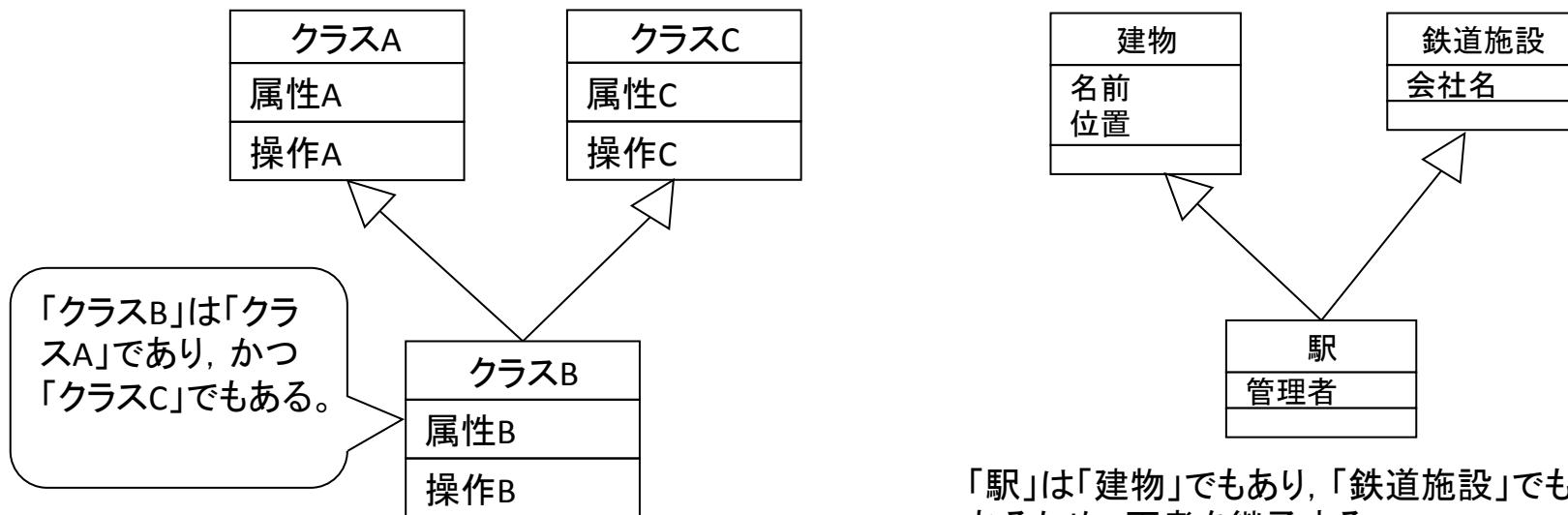
継承の記法

上位クラスと下位クラスを線で結び、上位クラス側に「白い三角」を付ける



多重継承 (multiple inheritance)

多重継承:複数の上位クラスからの継承



gittokでは、多重継承は認めない。

「駅」は「建物」でもあり、「鉄道施設」でもあるため、両者を継承する。
その結果、「駅」は以下の属性を持つ。
名前
位置 } 建物から継承
会社名
管理者 } 鉄道施設から継承
駅に定義

抽象と具象 (abstract and concrete)

抽象クラス

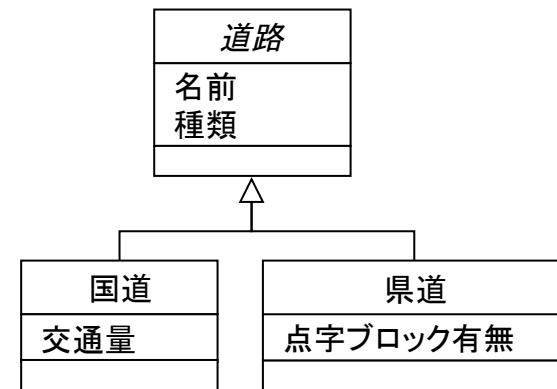
抽象的な概念を示し、インスタンスをもたないクラス

抽象クラスの親クラスは、抽象クラス

クラス名を斜体にして示す

具象クラス

インスタンスをもつことができるクラス



抽象クラス：

道路

具象クラス：

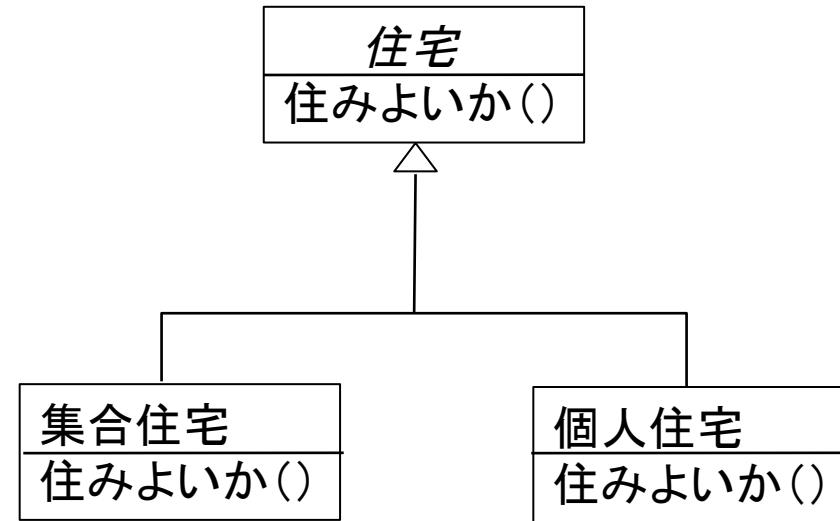
車道, 歩道

抽象クラスである道路は直接
インスタンス化されず、道路
を継承する車道及び歩道が
インスタンス化される。

多態性 (Polymorphism)

上位クラスが、下位クラスと同じ名前の操作をもつとき、
下位クラスのインスタンスは下位側の操作を実行すること。
この場合、操作は継承されない。

多態性の例



住宅の評価項目
コミュニティ?
プライバシー?

住宅の評価項目
周辺の環境?
庭?

文書化 (documentation)

クラス図の要素(クラス、属性、操作、関係)の定義は文章で示す。

建物
名前
種類

建物:屋根と壁で区切られた人工構造物

名前:他の建物と区別するために与える識別情報。

種類:建物の特徴を示す名前

クラス図は通常、専用のソフトウェアツールで作成する。ツールは必ずクラス図の要素の定義を記入できるようにしている。

まとめ

モデルの記述が自然言語だと、表現にあいまいさが残る。

スキーマ言語を使って、モデルの形式表現を行えば、あいまいさが軽減する
スキーマ言語の一つとしてUMLがある。

UMLでは、同じ性質をもつものの集まりをクラスとし、属性、操作、他のクラスとの
継承及び関連でスキーマを表現する。

gittokでは、UMLの規則を若干単純化した規則を設けている。

でも、クラス図の要素の定義は、文章で補う。