オブジェクト指向論(Q)

オブジェクト指向概論(B1) オブジェクト指向(K1)

> 第2回講義 2023/4/17

> > 來村 徳信

今回の講義のトピックと流れ

- 復習 + a
 - ○オブジェクト指向とは?
 - ○オブジェクトとは?

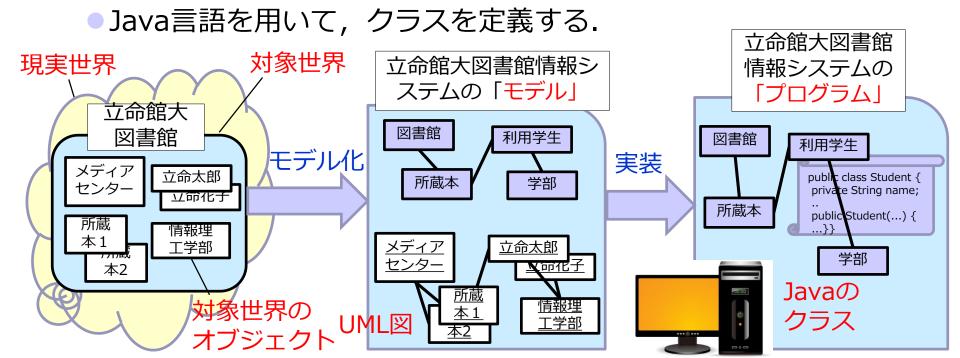


クラス図とオブジェクト図の基本

- クラスとインスタンス (詳細)
- ○オブジェクトの持つ性質
- ○(1) 属性と属性値
- ○(2) UMLのクラス図と オブジェクト図(=インスタンスの図)
- ○(3) UMLでの属性と属性値の表現
- ○(4) 関係の表現
 - クラス図の関連とオブジェクト図のリンクの関係
 - ●関連端名

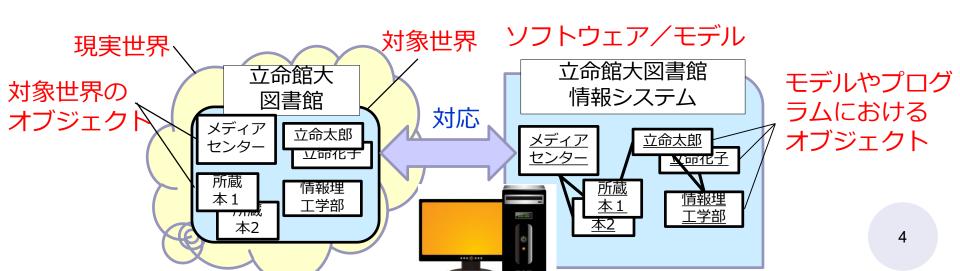
オブジェクト指向とは? (復習)

- ○オブジェクト指向モデリング
 - ソフトを作る前に、設計図として、対象世界のオブジェクトを 表すような「モデル」を記述する。
 - ●UMLの図(ダイアグラム)として記述する.
- ○オブジェクト指向プログラミング
 - モデルに基づいて、オブジェクトを表す部品の集まりとして、 「プログラム」を実装する。



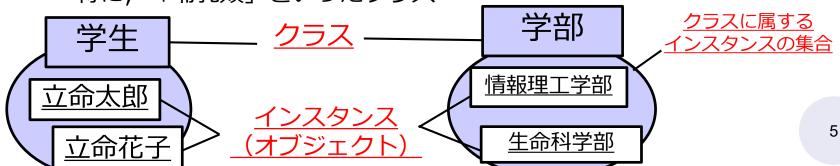
オブジェクト(object)とは

- <u>オブジェクト</u> ⇒ 対象世界に存在する「<u>モノ</u>」または「<u>コト</u>」
 - ●物理的に触れる「モノ」,例:本1(図書館にある本),立命太郎(学生)
 - ●物理的に触れない「モノ」,例:情報理工学部,立命館大図書館
 - ●物理的に触れない「コト」,例:立命太郎が本1を借出した,この講義
 - 「性質」を持つ、例:本1の題目は「やさしいJava」
- ○対象世界=ソフトウェアが対象としている「現実世界の範囲」
- ○モデルやソフトウェアのオブジェクト
 - モデルやプログラムで、対象世界のオブジェクトに対応する要素



クラスとインスタンス

- インスタンス(instance) ⇒ 個物
 - 現実世界に存在する<u>個別</u>のモノ・コト. <u>実例</u>. <u>具体的</u>
 - オブジェクト ≒ インスタンス
 - 単にオブジェクトといった場合、インスタンスを指す(ことが多い)
 - ◆ 特にUML図の「オブジェクト図」はインスタンスの図を表す。
- クラス(class)
 - 「<u>共通の性質</u>」を持つインスタンスの「<u>集合</u>」に対応する.
 - あるインスタンス e_1 はあるクラス C に「 $\overline{\mathbf{g}}$ する」という. 「 $e_1 \in C$ 」
 - 集合論的には、インスタンスは「<u>集合の要素</u>」
 - 「<u>型</u>(type)」≒「種類(kind)」≒「概念」を表す.
 - <u>抽象的</u>:実例ではない. 物理的には触れない.
 - 特に,「哺乳類」といったクラス



クラスとして捉える意義

- 一般的に:
 - 個物をグループ化して、まとめて性質や処理を書きたい.
- グループ化(集合化)の意義
 - 人間の認知において: (本講義のメインテーマではないが)
 - ●一般的な概念(例:哺乳類)とはそもそもそういうもの.
 -)例:「学生」という概念は「教育機関に所属して,学んでいる人間」と いうグループに概ね対応する.
 - ○モデリングにおいて: (本講義の前半部)
 - 対象世界の個物をどのようにグループ化するかをモデルとして 明示化することで、対象世界の理解を共有できる(前回講義)
 - ●集合の全要素に対する「決まり事(制約)」を書ける(今回講義から)
 - プログラミングにおいて:一度でまとめて書ける(本講義の後半部)
 - あるインスタンス集合に適用可能なようにプログラムを書ける.
 - プログラムの記述量が減る(労力の削減)
 - 特に,上位一下位クラスの階層があると,上位クラスで記述しておけば継承 されるため,記述量が減る ⁶
 - → オブジェクト指向プログラミングの動機のひとつ

クラスとインスタンスの比較(冒頭部分)

	クラス	インスタンス
なにを表す?	型・種類・概念	個物・実例
抽象度は?	抽象的	具体的
集合論的には?	インスタンスの 集合	集合の要素
どんな <mark>性質</mark> ?	共通な性質	個別な性質
属性としては?	属性名と値のタイプ	属性値
クラス図・オブ ジェクト図は?	インスタンスの <mark>制約</mark> (約束事)	ある時間点の状態
関係は?	関連=リンクの集 合に関する <mark>制約</mark>	リンク=ある時間点で 成り立っている関係

オブジェクトの持つ性質の種類



<u>属性</u>(attribute)

- ○オブジェクトの性質
- ○例:学生の属性:氏名,学生証番号,性別,入学年度,...
- 処理(operation)
 - ○オブジェクトが行うアクション(行動/行為/処理)
 - ○例:利用者は「本の返却」というアクションを行う
- 関係(関連: association)
 - ○複数のオブジェクト間の関係(関係をもつという性質)
 - ○例:「学生」と「学部」の間には「所属」という関係がある
 - ●学生は学部に「所属する」という関係をもつ性質がある (≒そのような関係を持つものである)

属性と属性値

- 属性(attribute)
 - ○オブジェクト自体が備えている性質
 - ○例:「学生」の属性:氏名,学生証番号,性別...
- オブジェクトは「<u>属性</u>」と「<u>属性値</u>」を持つ
 - ○オブジェクト(O)-属性(A)-属性値(V)
 - OAV表現と呼ばれる
 - ●例:ある学生である「立命太郎」の性別は「男性」である
 - オブジェクト=「立命太郎」,属性=「性別」,属性値=「男性」
 - ■属性 と 属性値⇒ プログラムにおける<u>変数(名)</u> と <u>変数の値</u>に対応
 - ○属性と属性値を混同しないこと!
 - ●日常言語では混同しがち.
 - 例:立命太郎の身長は180cm
 - -> 立命太郎の「<u>身長 属性</u>」の「<u>値</u>」は180cmである

クラスと属性

- クラスは「共通」な性質(例:属性)を持つ.
- ○インスタンスは「個別」の性質を持つ(例:属性値)
- ○というより...,「共通な性質を見いだしたものがクラス」

クラス 学生 氏名 学生証番号 性別

立命太郎

氏名 = 立命太郎 学生証番号 = 2600170001 性別 = 男性

立命花子

氏名 = 立命花子 学生証番号 = 2600170002 性別 = 女性

属性と状態

- インスタンスは「状態」を持つ。
 - 属性値は時間とともに変化することがある(ことが多い)
 - 例:属性「回生」の属性値
 - ●プログラムの動作に伴って,変数の値が変化するのと同じ.
 - <u>ある時間点(瞬間)における状態</u>の値を表す
- 同じインスタンスなのか?
 - 学生証番号が同じであれば同じ学生、値が固有(ユニーク).変化しない、識別子(identifier)と呼ばれる



る

11

一般に「クラス」定義・記述とは

- ある特定のインスタンス集合が満たす,
 - 「共通な性質」に関する「条件」の記述である.
 - 集合論における「内包的定義」に対応する(↔「外延的定義」)
 - ●例:学生クラスの全インスタンスは、「(数値列で表現される) 学生証番号を持つ」という条件を満たす。
 - ○解釈1) そのような性質を持っているオブジェクトのみが「学生」であるという「定義」を表す.
 - ●必要十分条件である場合
 - ○解釈2) 学生であれば、必ずそのような性質をもっていなければならないという「制約」を表す。
 - ●必要条件.必要十分条件ではないため、インスタンスのクラス 同定はできないが、多くの場合はこちら.インスタンス

クラス

学生

氏名:文字列

-学生証番号:数值列

-回生:整数



クラス図とオブジェクト図の関係

● クラス図

クラス

- あるインスタンス集合に「共通する性質」を表す.
- インスタンスの属性や関連に関する「約束ごと(制約)」を表す

学生

-氏名:文字列

-学生証番号:数值列

-回生:整数

所属する▶

1..*

学部

- 名称:文字列

- オブジェクト図(=インスタンスの図)
 - ある瞬間のインスタンスの「
 「状態」を表す.

このインスタンス集合を 表す円は説明用で,実際 の UML 図では書かない.

立命太郎: 学生インスタ氏名=立命太郎ンス学生証番号=2600170001集合=回生=2

立命花子:学生

氏名=立命花子

学生証番号=2600170002

回生=3

所属する <u>学部1:学部</u> 名称=情報理工学部 学部2:学部

名称=生命科学部

所属する▶

13

UMLのクラス図:クラスの表現

一部を省略した表現の例: 例: 学生 <u>クラス名</u> 学生 ※省略不可 - 氏名:文字列 学生 - 学生証番号:数值列 属性 - 回生:整数 - 氏名:文字列 ※省略可 - 学生証番号:数値列 + 入学する() 回生:整数 操作 + 進級する() ※省略可 四角形で表す(本講義では紫色) 学生 • (1) <u>クラス名</u> + 入学する() そのクラスを指し示す名前 + 進級する()

- (2) <u>属性</u>
 - そのクラスの性質を表す属性名と属性値のタイプ(次スライド)
 - 一行がひとつの属性を表す、複数可能、省略も可能、
- (3) 操作
 - そのクラスのインスタンスが行える<mark>処理</mark>
 - ()内に操作の引数を書く(※詳しくはプログラミング回で)

UML: クラスの属性の表現

例:学生クラスの「氏名」属性

属性 **学生**- <u>氏名: 文字列</u>

- 属性の記述形式(の一部)
 - 「可視性」「名前」: 「タイプ」
- ●可視性
 - +:パブリック, ー:プライベート
 - 属性がクラスの外側からみえるかどうか
 - 詳しくはプログラミング回で扱う、モデリング回では省略してもよい

名前

• 属性を表現する文字列. 変数名に対応. (例:氏名)

● <u>タイプ</u>

- 属性の値を表すオブジェクトの種類やデータの型.
 - 例:文字列(String), 数値(Integer), 日付(Date), クラス名
- ▶ 変数の「型」に対応.

オブジェクト図:インスタンスの表現

●例:学生クラスのインスタンス

インスタンスの名前

属するクラスの名前

※<mark>インスタンスの名前を</mark> <mark>省略した</mark>場合の表現

下線を引く 学生1:学生 - 氏名=立命太郎 - 学生証番号=2600170001 - 回生= 2 - 回生= 2

: 学生

- 氏名=立命太郎
- 学生証番号 = 2600170001
- 回生 = 2

・ 属性値 ・大課業では立名

- ▶ 四角形で表す(クラスと同じ.本講義では白色で区別している)
- インスタンスの名前(名前自体は本質的ではなく省略することが多い)
 - <u>下線を引く</u>(下線によって, インスタンスであることを表す)
 - 「:」の左にインスタンスの名前,右に<u>属するクラスの名前</u>を書く.
 - インスタンス名を省略して、「:クラス名」だけ書くことがある
- 属性と属性値
 - ▶ 「属性」<mark>=</mark>「属性値」.
 - 属するクラスで定義されている属性名とタイプである属性値であること.
- 操作はクラスと同じなので通常,省略される.
- ●ある時間点におけるインスタンスの状態を表す

UMLのクラス図とオブジェクト図

- クラス図
 - あるインスタンス集合に「共通する性質」を表す.
 - インスタンスの属性や関連に関する「約束ごと(制約)」を表す

学生

- 氏名: 文字列

-学生証番号:数值列

-回生:整数

属性名は同じ(一致させる).変数名に対応.

属性値はクラスで定義されたタイプ (例:文字列) であること、変数の

値の「型」(String)に対応.

- オブジェクト図(=/インス/タンスの図)
 - ある瞬間のインズタンスの「<u>状態</u>」を表す.

インスタ ンス 集合 = クラス 立命太郎 /: 学生

氏名=立命太郎

学生証番号=2600170001

回生=2

立命花子:学生

氏名=立命花子

学生証番号=2600170002

回生=3

このインスタンス集合を表す円は説明用で, 実際の UML 図では書かない.

クラスとインスタンスの比較(前半)

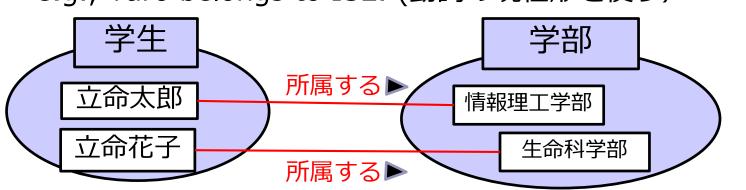
	クラス	インスタンス
なにを表す?	型·種類·概念	個物・実例
抽象度は?	抽象的	具体的
集合論的には?	個物の集合	集合の要素
どんな <mark>性質</mark> ?	共通な性質	個別な性質
属性としては?	属性名と 値のタイプ	属性値
クラス図・オブ ジェクト図は?	オブジェクト図の <mark>制約</mark> (約束事)	ある時間点の <mark>状態</mark>
関係は?	関連=リンクの集合に関する制約	リンク=ある時間点で 成り立っている関係

オブジェクトの性質(2)

- 属性(attribute)
 - ○オブジェクトの性質
 - ○例:学生の属性:氏名,学生証番号,性別,入学年度,...
- 処理(operation)
 - ○オブジェクトが行うアクション(行動/行為/処理)
 - ○例:利用者は「本の返却」というアクションを行う
 - <u>関係</u>(関連:association)
 - ○複数のオブジェクト間の関係(関係をもつという性質)
 - ○例:「学生」と「学部」の間には「所属」という関係がある
 - 学生は学部に「所属する」という関係をもつ性質がある (≒そのような関係を持つものである)

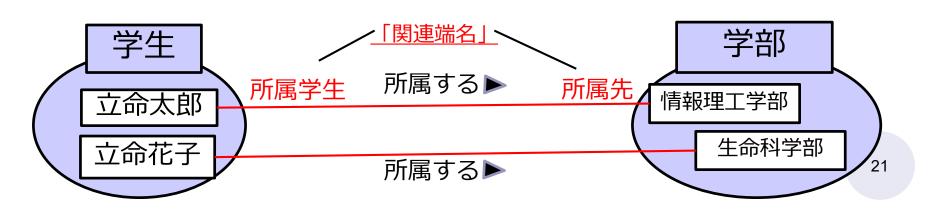
インスタンス間のリンク(1):リンク名

- 複数のインスタンス間の関係を表す
 - ○ある時点で,成り立っている関係を表す.
 - ●例:「立命太郎」と「情報理工学部」の間には「所属する」という関係が成り立っている、という状態を表している.
 - ●注:これはインスタンスがクラスに「属する」とは異なる
 - ○オブジェクト図ではインスタンス間を結ぶ線(リンク)で表す
 - ●関係の種類(型)を「リンク名」で表す(cf. 後述の関連名)
 - 方向性を表す矢印(►)を付けることが多い(なくてもよい)
 - ○リンク名を動詞にして,読み下す場合もある(が多い)
 - ●例:「立命太郎」は「情報理工学部」に「所属している」
 - ●e.g., Taro belongs to ISE. (動詞の現在形を使う)



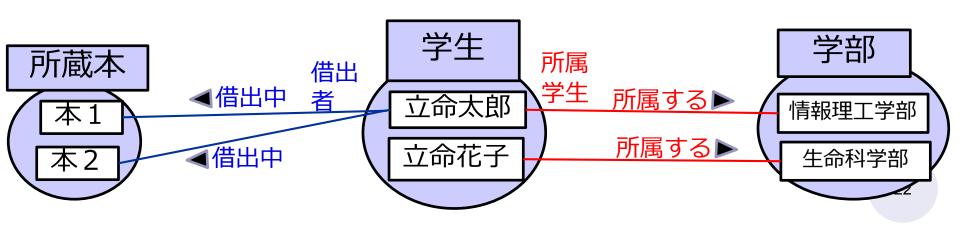
インスタンス間のリンク(2):関連端名

- オブジェクトは関係のもとで特定の役割を果たす
 - ○「<mark>関連端名</mark>」と呼ぶ(UMLでは)
 - ●例1:「立命太郎」は「情報理工学部」の「所属学生」である.
 - これは「情報理工学部」からみたときの「立命太郎」の役割である.
 - ●例2:「情報理工学部」は「立命太郎」の「所属先」である.
 - これは「立命太郎」からみたときの「情報理工学部」の役割である.
 - ●両者は同じ「所属する」という関係のもとで成り立つ役割である
 - 関連端名が関連名として使われる場合も多い(後述)が、本来は別.
 - クラスレベルで定義できる(後述)
 - ○一般的に「□ール概念」と呼ばれる重要な概念である.



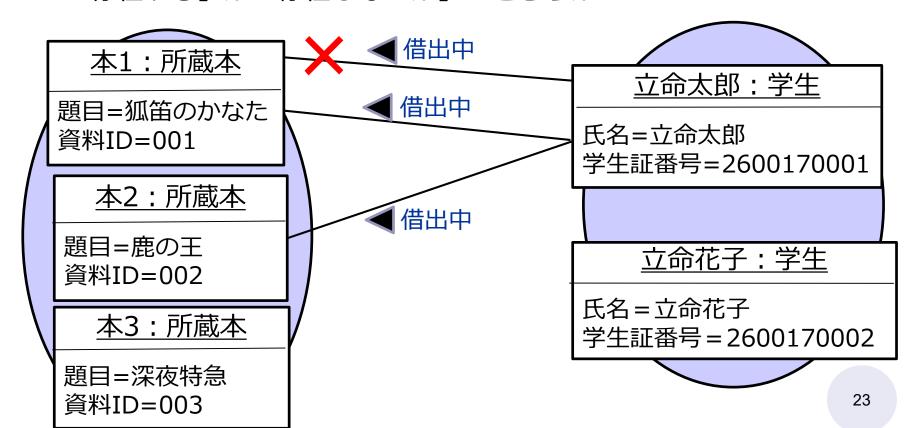
インスタンス間のリンク(3):複数の関係

- 1つのインスタンスが複数の関係を持つ場合がある
 - ○例:図書館の本の「借出中」関係を考えてみよう
 - ●下記の例:立命太郎は「本1」と「本2」を借りている. 立命花子は一冊も借りていない.
 - ○例:1つの学生インスタンスは:
 - ●「借出中」関係を複数,異なる「所蔵本」との間に持てる.
 - それとは別に、異なる種類の「所属する」関係も持てる.
 - 1つの学生インスタンスは、必ず、1つの学部インスタンスとの間に、 ただ1つの「所属する」関係を持つ. (「多重度」次週)
 - ●関連端名は異なる(「借出中」関係のもとでは「借出者」)



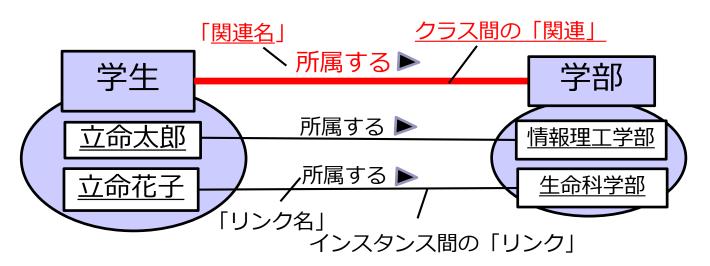
インスタンス間のリンク(4):ある関係

- ある時間点において成り立っている関係を表す.
 - 2つのインスタンスの間に、同じ種類のリンクを 2本以上、引くことはできない。
 - 2つのインスタンスを特定したとき,ある特定の種類のリンクは「存在する」か「存在しないか」のどちらか.



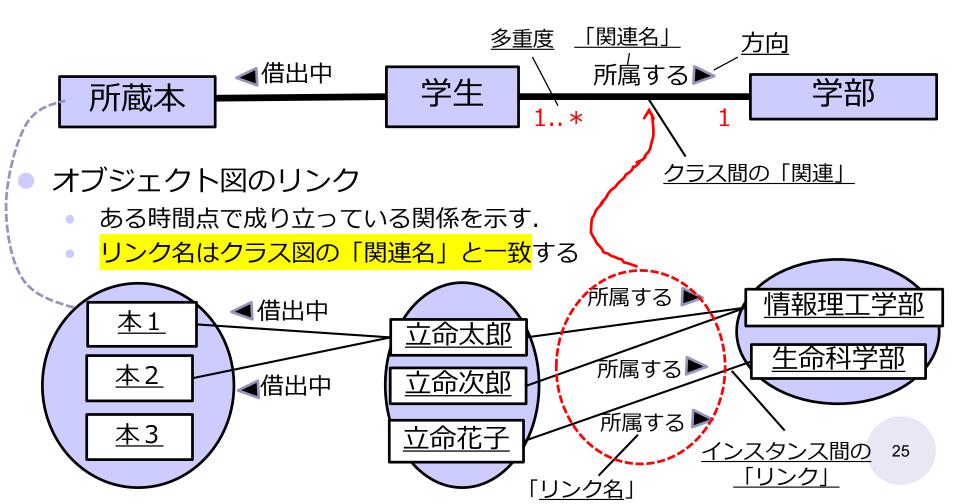
クラス間の「関連」

- クラスに所属するインスタンスが<u>持ちうる</u>関係を クラス間の「関連」と呼ぶ
 - クラスの共通な性質の一種
 - 「その関係をもちうる」という共通な性質を持つ
 - クラスの間の線(関連)で表す. 関連名等を付ける(後述).
 - インスタンス間の「リンク」の種類(cf. クラス)を表す. リンク名は関連名と一致させる.
 - リンクの「集合」に対応する(次スライド.cf.クラス)
 - 「必ず持つ」場合や,「持つこともある」場合など様々
 - → 関連の「多重度」(次週)



クラス図の関連と オブジェクト図のリンク

- クラス図の関連
 - クラスのインスタンス間に成り立ちうる「リンク」の種類と制約
 - リンクの「<u>集合</u>」:含まれるリンクのパターンを<u>多重度</u>(次週)で制約する



UMLクラス図における「関連」の表現

例:「所属する」関連 関連名 <u>方向</u> 学生 所属する 所属先 学部

名前:文字列 1..* /

ソース側の多重度

ターゲット側の多重度

名称:文字列

- 両クラスのインスタンス間になりたちうる「<mark>関係</mark>」を表す
 - 関連名の「方向」: ソース側 ▶ ターゲット側 方向がある場合,関連名は動詞を使うこともある(特に英語の場合).
 〈ソース側〉が〈ターゲット側〉を/に~する(例: 学生が学部に所属する) この ▶ は読み方を表現しているだけ.
 - 「<u>関連端名</u>」
 - その関係のもとでの相手先オブジェクトの呼び名. 関連ごとに異なる
 - その関係のもとで担う役割(「□ール」)を表す(前述)
 - 「<u>多重度</u>」(multiplicity) 【次週】
 - ある1つのインスタンスが、その関係を「いくつの」相手側 インスタンスと持てるかを表す。

クラス図における関連の向き

例:「所属する」関連

方向 関連名

所属される

学生

氏名:文字列

所属学生

1...*

___ ターゲット側の多重度 所属先

^{馬先} | 学部

名称:文字列

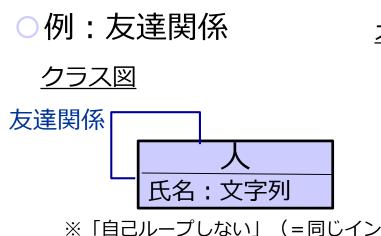
ソース側の多重度

- 関連は基本的に両方向的.
 - 「方向」は「関連名」の読み方を表現しているだけ」
 - 逆向きの関連名を付けることもできる.
 - 英語の動詞で表現していた場合,受け身形にすれば,逆向きになる.
 - 異なる関連名をつけることもできる(例:「収容する」)
 - 関連端名や多重度も変わらない。
 - 後述する「誘導可能性」を表す矢印は、実装レベルで意味を持つ。
- 関連名と関連端名は省略可能
 - どちらかだけで表現されることも多い。
 - 「関連端名」は相手側からみたときの「呼び名」を表す
 - その向きの関連名として使われることが多い 【重要】
 - 例:学生がソース側で学部がターゲット側な関連名:「所属先」

27

同じクラスのインスタンス間の関連

- 関連=インスタンスの間の関係
 - ○同じクラスに所属するインスタンス間にも関連がありえる
 - クラス図では、ひとつのクラスにループするように関連を書く
 - オブジェクト図では, ひとつのインスタンスでループするよう に記述されるわけではない.
 - 友達関係は、ひとつのインスタンスでループしない
 - 異なるインスタンス間でも必ずしもループするわけではない。
 - クラス図の関連は、オブジェクト図のリンクの「集合」だから



ない)という制約の記述が必要.

人クラスの オブジェクト図 インスタンス集合 氏名=立命太郎 友達関係 友達関係 、2:人 入3:人 氏名=立命花子 氏名=立命かな スタンスの間には友達関係は成り立た 28

クラスとインスタンスの比較(まとめ)

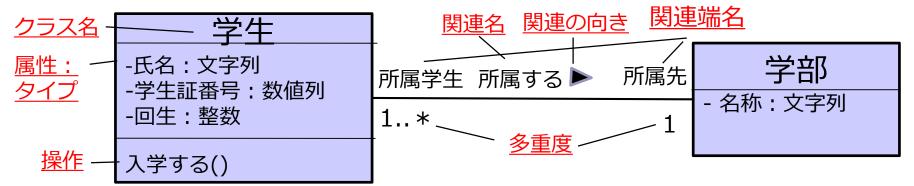
	クラス	インスタンス (オブジェクト図)
なにを表す?	型・種類・概念	個物・実例
抽象度は?	抽象的	具体的
集合論的には?	個物の集合	集合の要素
どんな <mark>性質</mark> ?	共通な性質	個別な性質
属性としては?	属性 名 と 値のタイプ	属性値
クラス図・オブ ジェクト図は?	インスタンスの <mark>制約</mark> (約束事)	ある時間点の <mark>状態</mark>
関係は?	関連=リンクの集合に関する制約	リンク=ある時間点で 成り立っている関係

まとめ: クラス図とオブジェクト図

• クラス図

回牛=3

○ インスタンスの性質に関する「制約」を表す.



- オブジェクト図(=インスタンスの図)
 - ある瞬間のインスタンスの「状態」を表す. クラス図の制約に従う.

