Smalltalk勉強会@京都
2008年12月3日　キャンパスプラザ京都　6F第2講習室

テキストの「はじめに」の部分

（これまで・今が旬・これから）

青木 淳
atsushi@cc.kyoto-su.ac.jp
http://www.cc.kyoto-su.ac.jp/~atsushi/

京都産業大学コンピュータ理工学部
研究室： 1号館 4階 1402研究室
客員研究員： 浅岡 浩子 ・ 澤本 依里

バベルの塔

機械語　アセンブリ言語

ALGOL　FORTRAN　COBOL　SQL

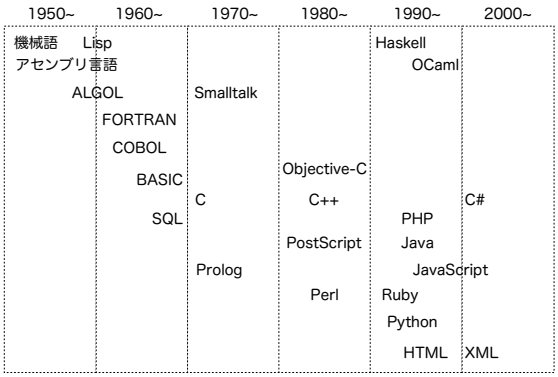
BASIC　C　C++　C#　Objective-C

Perl　Ruby　Java　JavaScript　Python　PHP

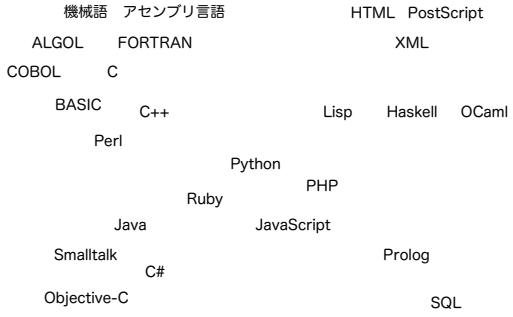
Lisp　Prolog　Smalltalk　Haskell　OCaml

PostScript　HTML　XML

バベルの塔

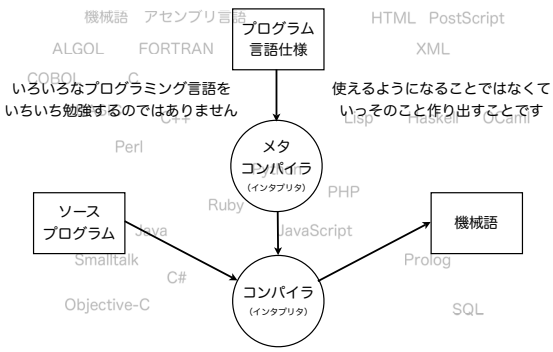


バベルの塔

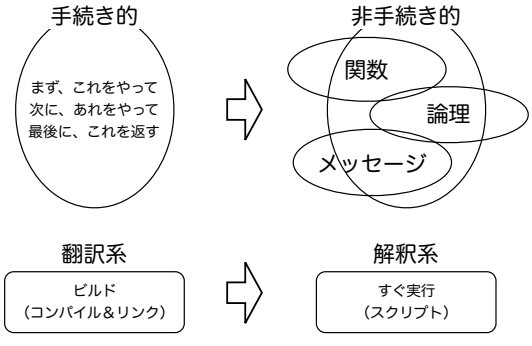


プログラミング言語を習得する早道

メタ系へ進みなさい！



プログラミング言語の変遷



プログラミング言語の変遷

データ型　　静的型チェック＆モジュール性

抽象データ型　　情報隠蔽＆カプセル化

クラスベースド　　総称性＆オーバーロード

オブジェクト指向　　インヘリタンス＆ポリモーフィズム

並列オブジェクト指向
分散オブジェクト指向　⇨　エージェント
形式オブジェクト指向　　強い型チェック＆契約（表明と確証）

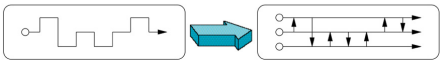
最近の傾向

インターネットやマルチメディアとキーワードは数知れないが…

➠ 命令から宣言へ

➠ 手続きからオブジェクトへ

➠ 逐次集中から並列分散へ



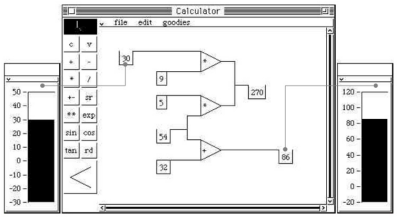
最近の傾向

摂氏と華氏の換算　86° F = 30° C * 9 / 5 + 32

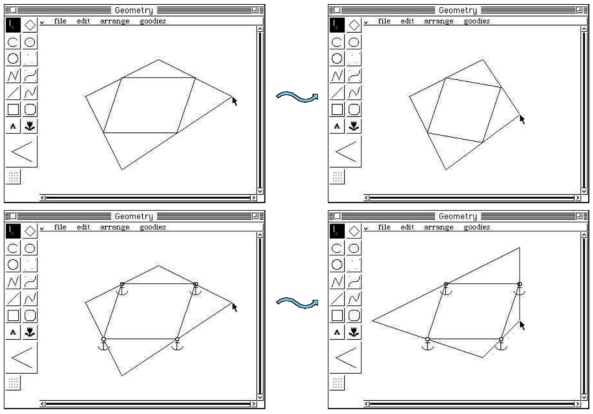
30
*
9
/
5
+
32
=

摂氏で30度が，華氏の何度になるのかを，電卓で計算してみよう。まず，30と入れて，掛け算のボタンを押し，次に，9を入れて，割り算のボタンを押し，それから，5を入れて，足し算のボタンを押し，その次に，32を入れて，最後にイコールのボタンを押して，華氏86度を求める。これが命令的なプログラミングスタイルそのものである。まさに「まず，これをやって」「次に，あれをやって」「最後に，これを返す」と指示することになる。

宣言的なプログラミングスタイルは，右図に示すように，計算過程や関係構造を，そのまま明示する。図中のオペアンプのような三角形は演算子を表し，頭が開いた箱は変数を表し，頭が閉じた箱は定数を表している。摂氏30度を華氏に換算するためには，最上部の変数の箱の中に30と入力する。すると，図中の演算の関係を充足するように，変数の箱の中の値が自動的に変化する。実際のプログラムは「こうでなければならない」「あれとそれはこうである」「これに則っていなければならない」と指示することになる。



最近の傾向



アラン・ケイ（Alan Kay）

1940年生まれ
パーソナルコンピュータの父
1971年〜1981年にXerox社パロアルト研究所（PARC）
画期的な「Dynabook」構想を提示
オブジェクト指向プログラミング言語「Smalltalk」の開発
1981年〜1984年にAtari社
1984年〜1996年にApple社でフェロー
1996年〜2001年にWalt Disney社
2002年〜2005年はHewlett-Packard社
子供たちとコンピュータとの関わりに注力

私はコンピュータに関する発明をするときに、コンピュータに関する知識はまったく使いません。生物学や音楽など関係ない分野の知識と類推を利用します。私がオブジェクト指向の考え方を考えついたときにも、普通の人がコンピュータについて考えるようなことを考えずに、生物がどのようにして複雑な構造を作るかを考えました。

すべての細胞は細胞から
(all cells come from cells)

ルドルフ・ウィルヒョウ

1821年生まれ
解剖学と病理学の大家
影響力のある言葉を多く残した人
たとえば「すべての細胞は細胞から」や「医療はすべて政治であり、政治とは大規模な医療に他ならない」など

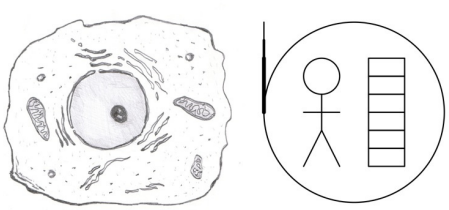
⇩

すべてのオブジェクトはオブジェクトから
(all objects come from objects)

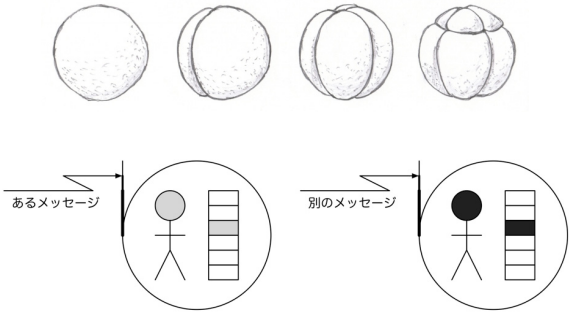
⇩

すべてはオブジェクト
(everything is object)

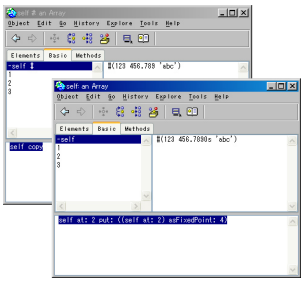
細胞とオブジェクト



複製と差異



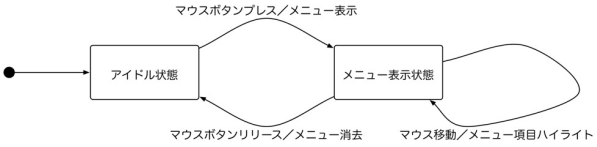
オブジェクトの能力



利用者からアクセス可能な要素であるオブジェクトは、利用者が観察したり、操作したりする際に、いつも意味のあるやり方で、自分（オブジェクト）自身を提示できる能力を持ちます。

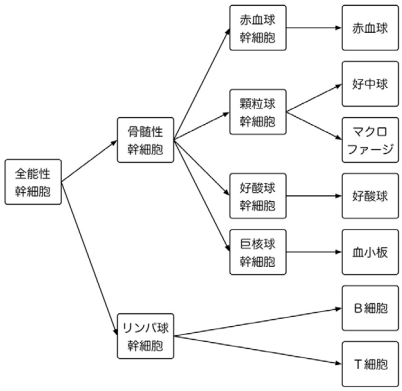
オートマトン（自動人形）

1940年代に神経生理学の研究者たちによって脳（神経回路網）の働きを数学的に説明するために開発された



ポップアップメニューの状態遷移：　G＝｛Q，X，Y，P，ε｝
有限個の内部状態の集合：　Q＝｛アイドル状態，メニュー表示状態｝
有限個の入力の集合：　X＝｛マウスボタンプレス，マウス移動，マウスボタンリリース｝
有限個の出力の集合：　Y＝｛メニュー表示，メニュー項目ハイライト，メニュー消去｝
有限個の状態遷移関数の集合：　P＝｛アイドル状態×マウスボタンプレス→メニュー表示，
メニュー表示状態×マウス移動→メニュー項目ハイライト，
メニュー表示状態×マウスボタンリリース→メニュー消去｝
初期状態：　ε＝アイドル状態

発生と文法



発生と文法

全能性幹細胞の発生（生成）文法：　G＝｛N，T，P，σ｝
有限個の非末端細胞の集合：　N＝｛全能性幹細胞，骨髄性幹細胞，リンパ球幹細胞，
赤血球幹細胞，顆粒球幹細胞，好酸球幹細胞，巨核球幹細胞｝
有限個の末端細胞の集合：　T＝｛赤血球，好中球，マクロファージ，
好酸球，血小板，Ｂ細胞，Ｔ細胞｝
有限個の生成規則の集合：　P＝｛全能性幹細胞→骨髄性幹細胞／リンパ球幹細胞，
骨髄性幹細胞→赤血球幹細胞／顆粒球幹細胞／
好酸球幹細胞／巨核球幹細胞，
リンパ球幹細胞→Ｂ細胞／Ｔ細胞，
赤血球幹細胞→赤血球，
顆粒球幹細胞→好中球／マクロファージ，
好酸球幹細胞→好酸球，
巨核球幹細胞→血小板｝
初期細胞：　σ＝全能性幹細胞

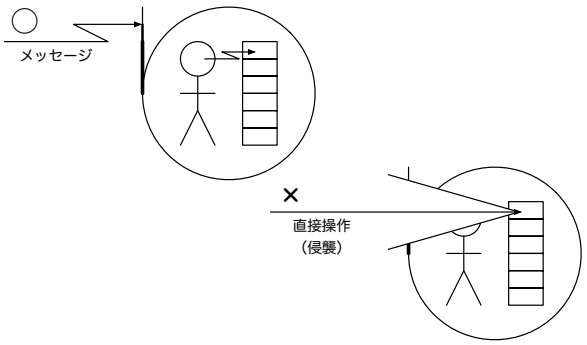
元来をたどることができ、どのような差異が生じて分かれたいったのかを物語る

⇩

インヘリタンス（継承）

オブジェクトの自律性

（自分のことは自分で決める）



オブジェクトと日本

（オブジェクト指向という訳語が悪かったかも…）

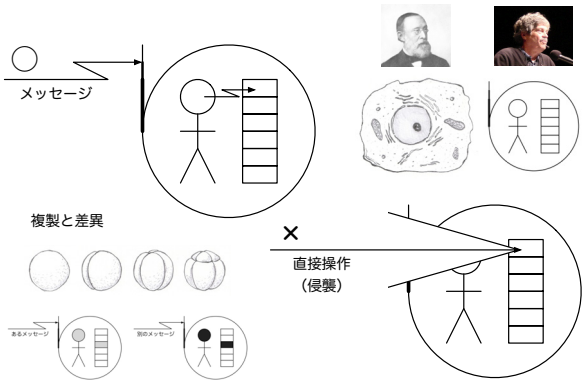
1970年代の中頃に提唱されながら、
1970年代の後半から1990年代の前半に渡って、
オブジェクトや抽象データ型の本質は日本の開発現場に浸透していない。
Liskov, B. H. and Zilles, S. N., "Programming with Abstract Data Type", Proceedings of ACM SIGPLAN Conference on Very High Level Languages, SIGPLAN Notices 9, 4, pp. 50-59, 1974.

Cox「ハードウェアは進歩している。
ソフトウェア もそれに続いている。
いちばん遅れているのが ヘッドウェアである。」

DeMarco「ソフトウェア開発上の問題の多くは、
技術的 というよりも社会学的である。」

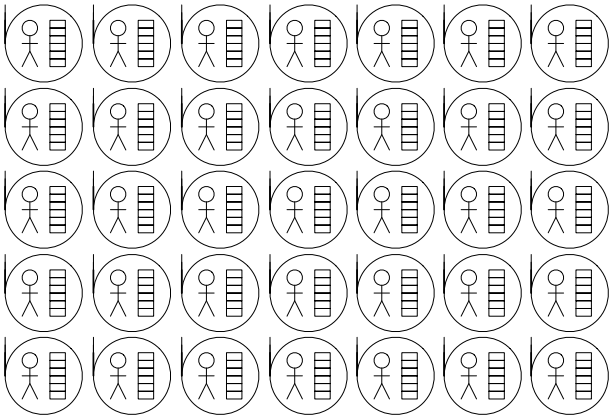
開発現場
の保守性

オブジェクト（対象）

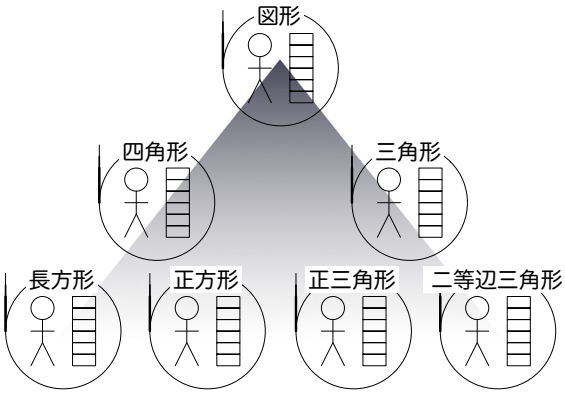


60兆個（自律性と分権性）

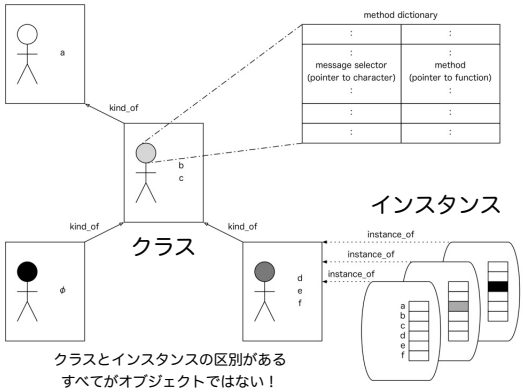
複製と重複



インヘリタンス（継承）



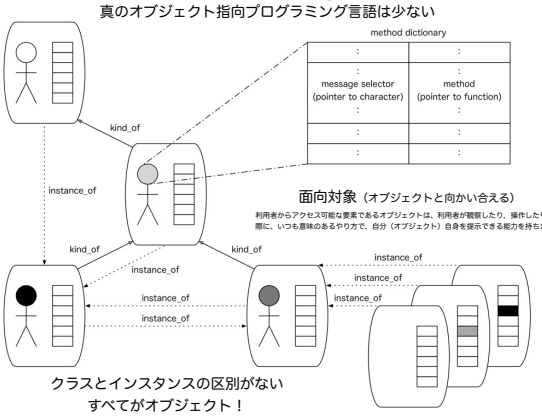
差異と節約（似非オブジェクト指向）



クラスとインスタンスの区別がある
すべてがオブジェクトではない！

オブジェクト指向

真のオブジェクト指向プログラミング言語は少ない



面向対象（オブジェクトと向かい合える）

利用者からアクセス可能な要素であるオブジェクトは、利用者が観察したり、操作したりする際に、いつも意味のあるやり方で、自分（オブジェクト）自身を提示できる能力を持ちます。

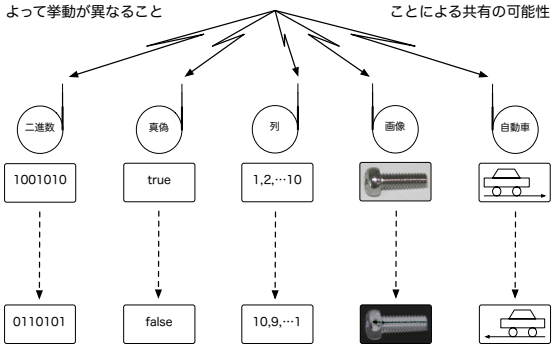
クラスとインスタンスの区別がない
すべてがオブジェクト！

ポリモーフィズム（多相）

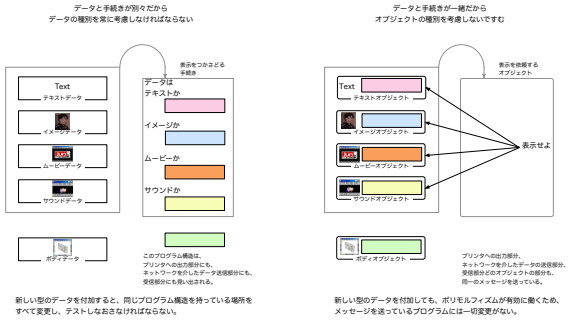
ポリモーフィズム＝同じ
メッセージでも受け手に
よって挙動が異なること

反転しなさい

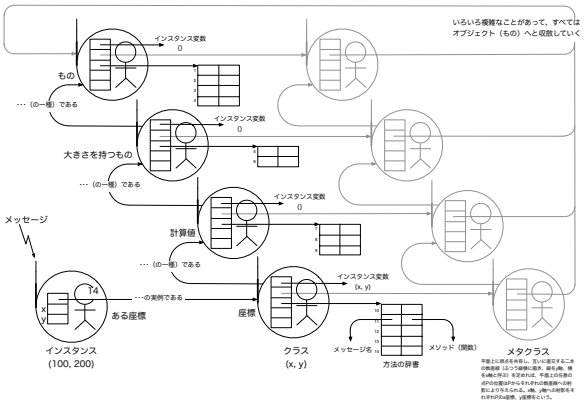
ひとつのメッセージから
多くの相（姿）が現れる
ことによる共有の可能性



手続き的からオブジェクト指向へ



オブジェクト指向のダイナミクス



まとめて

- オブジェクト（対象）
  - クラス（部類・抽象データ型）
  - インスタンス（実例・抽象データ）

問題解決手段が複数あることを示すのに有効
改変容易性

- 継承（インヘリタンス）

オブジェクトを分類し整理するのに有効
認知的経済性

- 多相（ポリモーフィズム）

他分野のオブジェクトを包括するのに有効
共有可能性