プログラミング言語論

第1回

イントロダクション

担当：犬塚

講義内容と目的

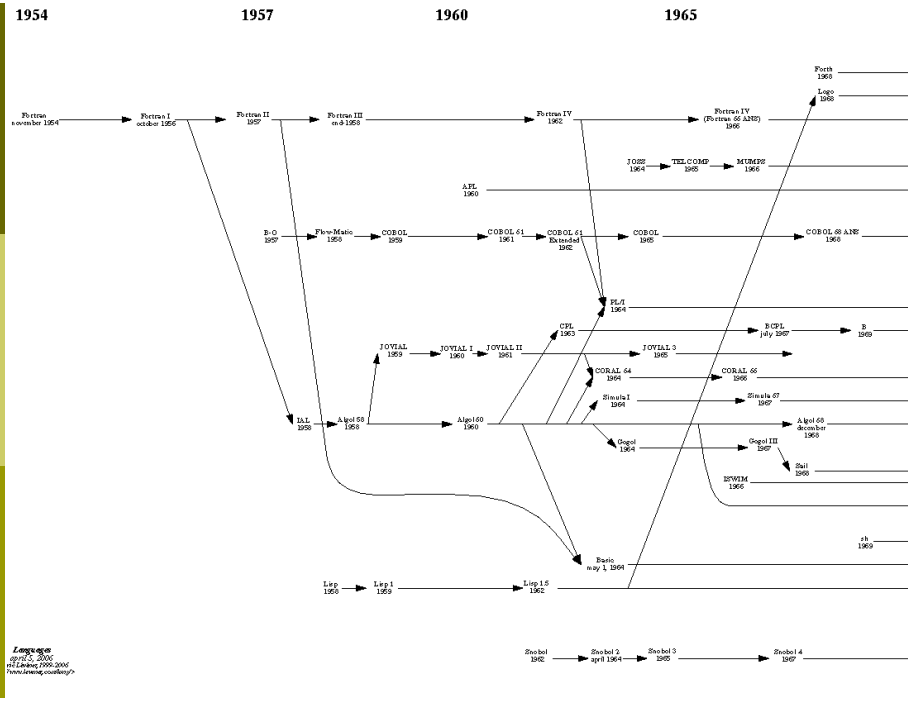
- プログラミング言語パラダイム
- プログラミング言語の諸要素
- プログラム理論
- プログラムの意味論

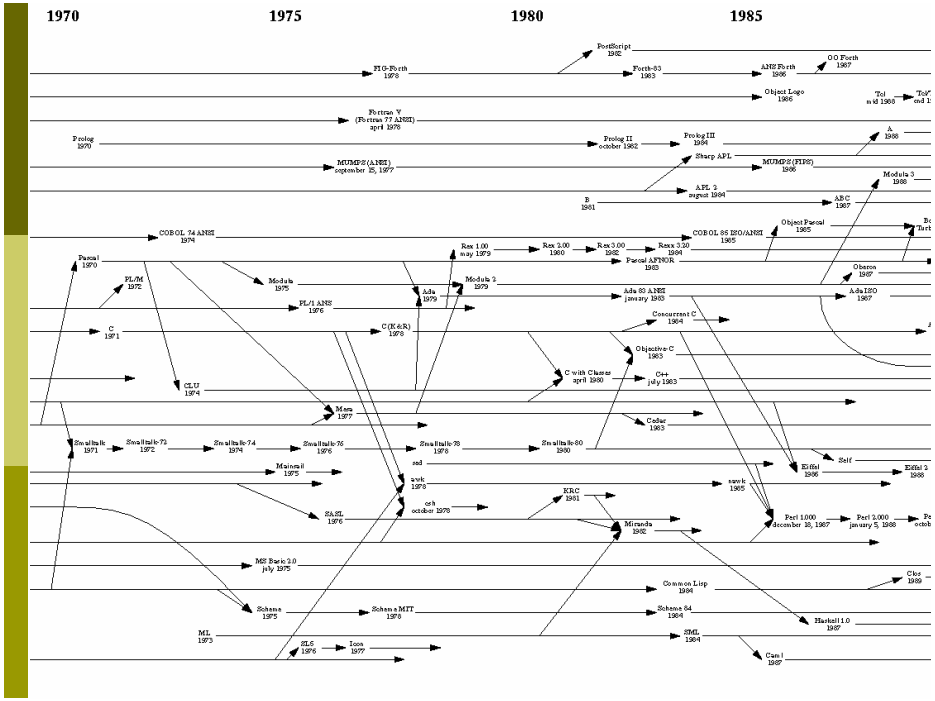
講義の目的

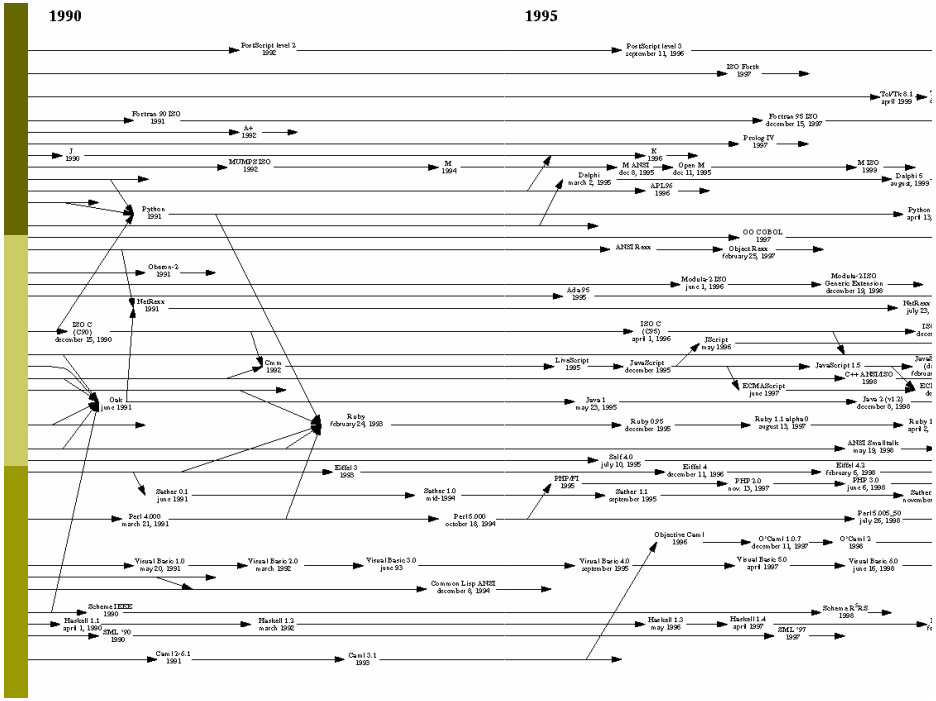
- プログラミング言語に関するアイデアを一通り習得すること。

初期のプログラミング言語

- 最初の言語＝機械語
  - 複雑な計算を書くのに大変時間がかかる
  - デバッグに多大な時間が取られる
- 数式を書くように記述して、自動的にプログラムを出す仕組みが欲しい
- 自動数式変換　＝　FORmula TRANslator
  - FORTRAN 1956







プログラミング言語

記述と抽象度のレベル
- 機械語
- アセンブリ言語
- 高級言語

実行の形式
- コンパイラ言語
- インタプリタ言語
- スクリプト言語

用途
- 科学技術計算用言語
- ビジネス用
- 人工知能用言語
- ネットワークプログラミング用
- 特殊用途用（シミュレーション用、データベース管理、…）

パラダイム

- プログラミング言語の分野だけでなく、広く使われている言葉。

  （本当は、もっと厳密に意味が決まっている）
  - 科学研究の1つのコミュニティ（分野）で共有されるアイデアの全体
  - その分野での、用語、考え方、成功事例等の全体。
  - パラダイムを共有していない集団は、話が通じない。
  - トーマス・クーン　科学革命（パラダイム転換）

プログラミング言語パラダイム

- プログラミングに関する考え方の総体
  （必ずしも、クーンの使い方とは一致していない）
- プログラミング言語パラダイム
  - 計算モデル
  - プログラミング言語に組み込まれる仕掛け

  場合によって
  - プログラミング作法
  - ライブラリ
  - 開発環境
  - そのプログラムを使った開発事例

プログラミング言語パラダイム

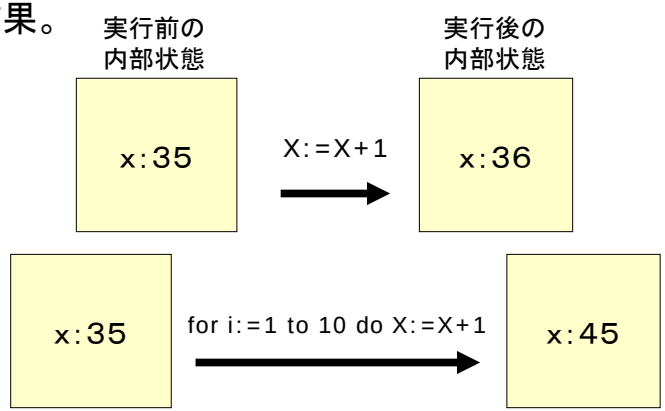
- 命令型（代入型，状態遷移型）言語パラダイム
- 関数型言語パラダイム
- 論理型言語パラダイム
- オブジェクト指向型言語パラダイム

計算モデル

- 計算するとはどういうことであるかの説明。
- 計算モデルが変わるとプログラミング言語の設計も変わる。
- 計算モデルはプログラミング言語パラダイムの最重要かつ主要な構成要素

計算モデル1　状態モデル

- 計算とは、計算機の内部状態を変えてゆくもの。
- 計算が終わったときの計算機の内部状態が計算の結果。

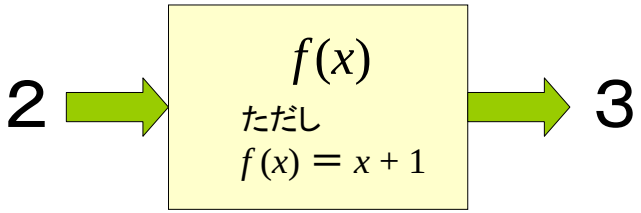


状態モデルに基づくプログラミング言語

- 命令型言語、代入型言語
  - FORTRAN(1954, 1957)　John W. Backus
  - ALGOL(1958)
  - PASCAL(1970)　Niklaus Wirth
  - C(1972)
- 状態を変えるための命令手順書の形式
- 変数と変数への代入
- 変数の値に関する仕組み＝データ型
- 命令を構造化する仕組み＝構造化プログラミング
- 命令を機能別にまとめる仕組み＝手続き

計算モデル2　関数モデル

- 計算とは、関数を適用することである。



関数モデルに基づくプログラミング言語

- 関数型言語
  - Lisp(1958)　John McCarthy
  - FP(1978)
  - Scheme
  - ML(1974)
  - Common Lisp
- プログラムは関数の定義、関数の適用
- ラムダ計算、再帰的定義、関数抽象
- リストとリスト操作関数
- ガーベージコレクション

計算モデル3　証明モデル

- 計算とは、ある条件式を満足するものが存在することを証明することである。
- プログラムとは条件を定義すること。

$\exists x\, P(x)$ を証明せよ。

- 論理プログラミング＝導出原理　Prolog(1972)
- 構成的論理＝条件を満たすものの存在を証明するため、それが何なのかを実際に示せ。

計算モデル4　ものモデル

- 計算とは、ものの相互作用である。
- ものにはその性質、状態、機能がある。
- もの＝オブジェクトが互いに働きかけることで、その状態を変化させ、計算が進む。

- オブジェクトのクラス、クラス継承
  関連概念：抽象データ型、カプセル化
- Smalltalk、C＋＋、Java

その他の計算モデル

- プロセスに基づくモデル
  - 平行プログラミング
- 分散アルゴリズム
- ネットワークプログラミング
- テキスト処理言語
- データベース管理言語

プログラミング言語の諸要素

文法	BNF
制御構造	構造化プログラミング
変数、値、データ型	型代数、オブジェクト指向 抽象データ型
サブルーチンと呼出し	クラス、継承、カプセル化
実行	部分計算、ネットワーク対応 検証、アサーション
入出力	イベント駆動

プログラム理論

- プログラムの正しさ
  正当性と停止性
- 2つのプログラムが等しいということ
  - 入力と出力の関係が等しいこと
  - プログラムの計算の仕方も含めて等しいこと
- プログラムによって計算できること、できないこと
  - 数学的に定義できる関数
  - プログラミング言語によって表現できる関数
  - 停止が飛翔できるプログラムで表現できる関数

プログラムの意味論

- プログラミング言語は言語である。
- 言語は何かを表現するためのもの。
- 表現しているものがその文の意味。

- プログラムは何を表現しているのか。
- プログラムは、計算機に何を要請しているのか。
- プログラムが正しいということは、何を期待しているのか

プログラミング言語意味論

- 操作的意味論
  - プログラムは状態をどのように変化させていくのか。
- 公理的意味論
  - プログラムの実行によってどんな性質が満たされるのか。
- 表示的意味論
  - プログラムが意味するものを、具体的に対応付ける。

参考書

- Robert W Sebesta, Concepts of Programming Languages, Addison Wesley, 1996-2005.
- Peter Van Roy and Self Haridi, Concepts, Techniques, and Models of Computer Programming, MIT Press, 2005.
- ラビ・セシィ(神林靖 訳), プログラミング言語の概念と構造(新装版), ピアソン・エデュケーション, 2002.
- Bill Kinnerslev,Collected Information On About 2500 Computer Languages, Past and Present. http://people.ku.edu/~nkinners/LangList/Extras/langlist.htm
- R.バード(土居範久 訳), プログラム理論入門, 培風館, 1981.