

【SWEST/ESS共同企画セッション】  
組込みに使うプログラミング  
言語について語ろう！

2019/9/5 21:00-22:30

アイシン精機株式会社

ソフトウェア技術部

間瀬 順一

# はじめに

- 組み込みソフトウェア開発では、長く C が使われています。
- 一方、車載ECUのソフトウェア開発では、MBD(MATLAB/Simulink)も普及期に来たと考えられています。
- 組み込みソフトウェア開発で使われるには、言語としてどんな特性が有利/不利なのかみんな語りましょう。
- パネルディスカッションの形式をとります。

# お願い

- プログラミング言語について議論を行いますが、主観的な批判、事実に基づかない主張の繰り返しとなる泥仕合になることは避けたいと考えております。
- 客観的・科学的な議論を進めていきたいと希望します。
- プログラミング言語の策定に関わっている方、言語を推進するコミュニティの方々へのリスペクトを忘れないように議論を進めていきましょう。

# パネルディスカッションのお題

## 最初のお題

Next Cはあるのか？

## 2番目のお題

Python が流行っている理由について

## 3番目のお題

ある言語が広く使われるためには、どんな特性が必要か？  
どんな活動が有効か？

# ポジショントーク(間瀬順一)

- 大学の専攻は数学
- 1998年～2002年 電気系のソフトウェア開発会社に勤務
  - ネットワーク管理アプリケーション開発に従事
- 2002年12月～ アイシン精機(株)に転職
  - 商用車向けオートマチックトランスミッション制御ソフトウェア開発に従事
- 2007年4月～2016年3月 アイシン・コムクルーズ(株)に出向
  - 引き続きAT制御ソフトウェア開発に従事
  - ソフトウェア技術者育成の仕組みづくりにも参画
- 2016年4月～2018年9月 APTJ(株)に出向
  - AUTOSAR Classic Platform 開発に従事
- 2018年10月～ アイシン精機(株) ソフトウェア技術部で勤務
  - ソフト基盤戦略主査を拝命

# Cの個人的経験

- 1988年 – 1991年
  - コンピュータの開発を行っている会社でアルバイトをしていた。
  - 働いている会社が移植した Unix 環境を使っていた。
  - その環境で、Cを使っていた。  
当時は、Cの規格化が進行しており、ANSI規格は1989年、ISO規格は1990年に成立した。規格準拠のコンパイラに移行するのは、少し時間がかかった印象。
  - 個人で所持しているパソコンのプログラミングは、アセンブラが中心  
(当時、コンパイラは有償で、学生が買うには高かった。)
- **アセンブラと比較すれば、Cの生産性は非常に高かった。**
  - ご存じのように、Unix環境が出自のCであったが、パソコンや組み込みソフトウェア開発でも使われるようになった。

# OODの個人的経験

- 1998年 – 2002年
  - 電気系のソフトウェア開発会社に勤務
  - Windows NT上でGUIを開発
  - プログラミング言語は、C++（MFCを利用）
  - 少し前の世代では、WindowsのGUIをCで書いていた。
    - GUIをCで書くのは、正直に言ってしんどいと思えた。
    - オブジェクト指向自体は、分野を問わず使える概念であるが、ウィンドウシステム上の画面上の部品に対して、メッセージでやりとりしながら、動作させる、という考え方がぴったりはまった。
    - オブジェクト指向にとって、ウインドウシステムのプログラミングは、キラーアプリとなったと理解
    - Javaも初期では、クライアントで動作するAppletが書けることをアピールしていた。
- オブジェクト指向も、GUIに限らず、広く使われるようになった。

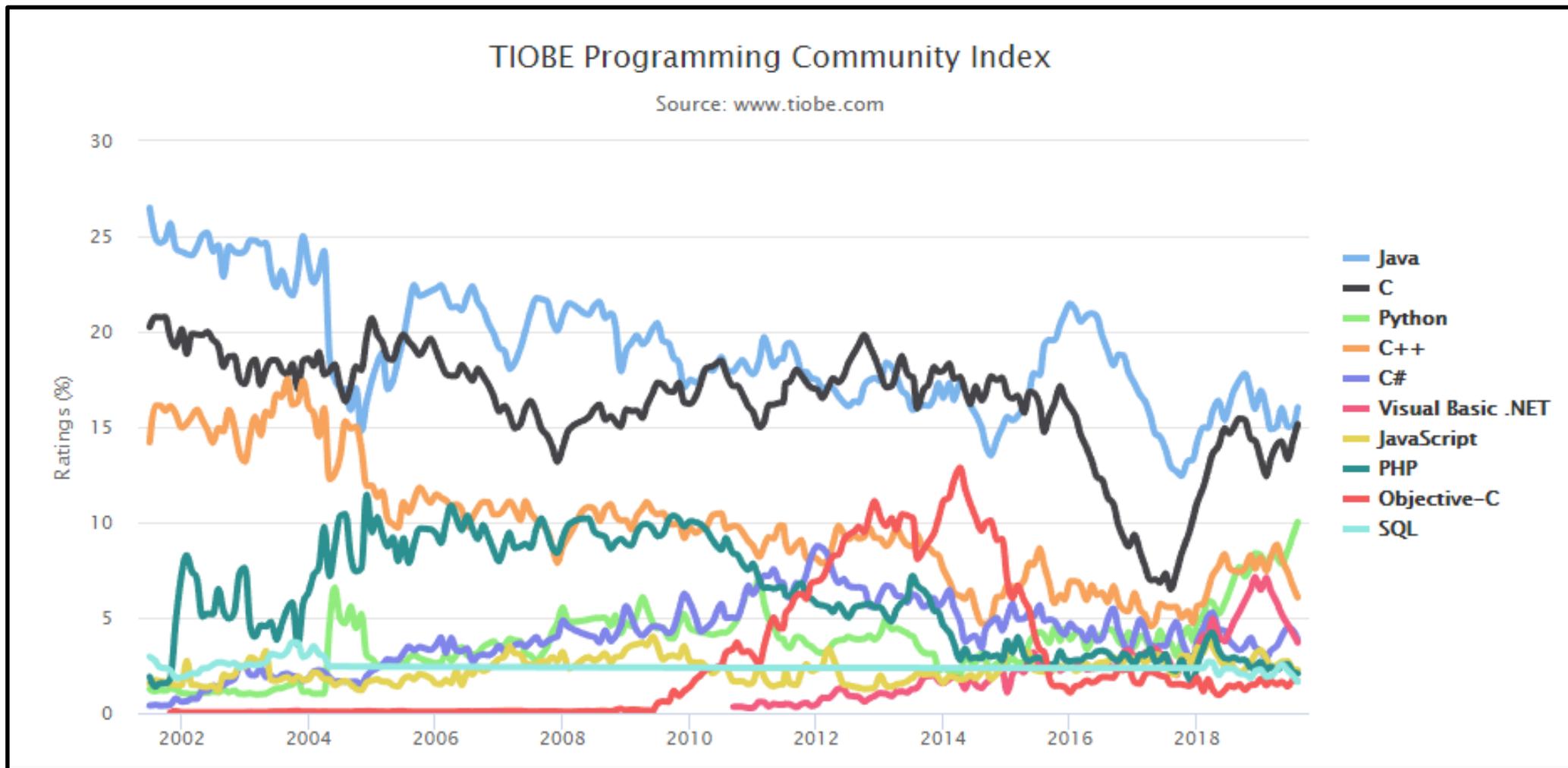
# パネラーのみなさま

イーソル株式会社	竹内	陽児様
東京工業大学	渡部	卓雄先生
九州工業大学	田中	和明先生
北九市立大学	山崎	進先生

一人5分程度のポジショントークをお願いします。

# 言語の変遷を振り返る

TIOBE Indexより



# 少し昔の書籍より

- パソコン言語学
- アスキー出版局 1984年 石田晴久監修
- 当時のパソコンで使えるプログラミング言語の紹介
- 紹介しているプログラミング言語（章立て順）
  - COBOL
  - FORTRAN
  - Pascal
  - C
  - Ada
  - LOGO
  - Prolog
  - LISP
  - FORTH

# 組込み機器のソフトウェア開発

- 組込み機器のソフトウェア開発の特性
  - 動的なリソースの取得/解放を避けたい場合がある
    - ガーベージコレクション（GC）の動作が心配
    - GCのアルゴリズムによっては、リアルタイム制約と両立できるか？
  - 省リソースが求められる場合がある。
    - 半導体の進歩によって、マイコンの性能は上がってかつ安くなっているが、ROM 数十Kbyte、RAM 数Kbyteの製品があるのも事実

# 最初のお題

- リアルタイム制約が厳しい製品、省リソースが求められる製品では、いまだにCが使われている。
- Next Cはあるのか？あるとすればどんな言語か？
- 環境の変化によって、求められる言語特性が変わるか？  
(それは、どんな環境の変化か？)

# Pythonに元気がある

- 2020年春試験から基本情報技術者試験のプログラミング言語の対象になる。
    - 「C、Java、Python、アセンブラ言語、表計算ソフト」から選択
    - Pythonの採用理由は  
「適用範囲の拡大と利用の増加、機械学習やディープラーニングに関わる主要なOSSでの採用の広がり等により」  
<https://www.ipa.go.jp/about/press/20190124.html>
  - Udemy (<https://www.udemy.com/>) のコース数と選択人数
    - Java 812コース 5,174,223人が選択
    - C 227コース 992,624人が選択
    - Python 1239コース 9,891,160人が選択
- ※2019/9/1 23:00調査

## 2番目のお題

- Pythonは、組込まれるソフトウェアの開発に使われているわけではないが、周辺環境整備への利用なども含めて利用が急に拡大した。
- その理由は何か？
  - 確かに「機械学習やディープラーニングに関わる」ソフトウェア環境の採用が多かった。そこから活用が広がったのか？
  - 書きやすい言語との評判はよく聞くが、同じくらい書きやすい言語は他に無いわけではない。Pythonは何が良かったのか？
- このお題の議論は、（最後となる）3番目のお題につなげることを意図しています。

# 3番目のお題

- ある言語が広く使われるためには、どんな特性が必要か？  
どんな活動が有効か？
  - キラーアプリが開発できるか？  
(キラーアプリのヒント)
    - 並列動作を効率よく処理できる、もしくは簡単に書ける。
    - 高い信頼性が容易に実現できる。
  - 個人で行える活動は何か？
  - コミュニティで行える活動は何か？
    - 言語に関する教育のコースウェアが充実していると良い
    - 言語自体のアップデートは必要だが、頻繁すぎると使いづらくなる可能性も
- 応援したい言語について、アイデアを語ってください。

# 最後に（クロージング）

- ここまでの議論をまとめたいと思います。
- 90分という限られた時間ではありましたが、パネルディスカッションへの参加ありがとうございました。
- まだ議論が足りないと感じた方は、今日でも明日でも、声をおかけください。