

プログラミング基礎 プログラミング基礎演習

全クラス合同ガイダンス

プログラミング基礎・同演習

- 総合情報学科 1年次 後期
 - 情報システム学系および数理情報学系必修科目
 - 担当者：マッキン、岸本、北風、海老原、平塚、早稲田
-
- 情報システム学系は月3,4限で実施
 - 数理情報学系は木4,5限で実施
 - 社会情報学系は数理情報クラス（木4,5限）を受ける
ただし履修者数制限がある

授業概要

- 「プログラミング基礎」「プログラミング基礎演習」とセットで、プログラミングの最も基本的な構造化プログラミングの理解を目指し、2年次以後で取り扱うプログラミング科目のための基礎作りを行う。プログラミング言語はJava言語を用いる。
- 到達目標: 構造化プログラミングを理解し、構造化プログラミングを用いたJavaプログラムを作成できることを目指す。

授業計画

1. 授業ガイダンス・Java&Linux概論
2. Linux実行・Javaコンパイル/実行
3. データ構造(1) 変数の型、変数の代入、キャスト
4. 演算子と式、標準入出力
5. 制御構造(1) if文、switch case文
6. 制御構造(2) for文、while文、do while文
7. 制御構造(3) 多重ループ、break文、continue文
8. 中間テスト
9. データ構造(2) 参照型(1) 文字列
10. データ構造(3) 参照型(2) 1次元配列
11. データ構造(4) 参照型(3) 多次元配列
12. メソッド(1) クラスメソッド、return文
13. メソッド(2) 引数
14. 構造化プログラミング
15. 総まとめと復習・授業評価アンケート

学年末定期試験あり

注意事項

- 科目URL: <http://www.edu.tuis.ac.jp/~mackin/program/>
- 準備学習・時間外学習: 毎回課される課題を必ず行うこと。
- テキスト: 「ITText Java基本プログラミング」(今城哲二編、オーム社)
- **全クラス共通**
- 参考図書: 「やさしいJava (第7版)」(高橋麻奈、ソフトバンク) (古い版も可)
- 他科目との関連: 2年次以降のプログラミング関連科目の基礎となる
- 履修条件: 「プログラミング基礎」「プログラミング基礎演習」とセットで履修すること
- 受講上の注意: 遅刻、欠席をしないこと。

成績評価

- プログラミング基礎：期末試験60%、小テスト20%、確認問題20%で評価する。
- プログラミング基礎演習：演習課題で評価する。
- 出席が2/3に満たない者は、欠席不可とする。遅刻および早退は欠席として扱う。
- セット履修のため、プログラミング基礎とプログラミング基礎演習のいずれかが不合格の場合、両方とも不合格となる。

授業の進め方

- この科目は演習を重視する。毎回の演習課題を完成することが重要
- 提出課題・復習含めて、毎週5時間程度の授業外学習課題が出る
- プログラミング言語は、自然言語（英語など）と同じく、どれだけ集中した時間をかけて、繰り返しを行ったかが重要

実習環境

- 授業ではデスクトップPCあるいはノートPCを利用する
- ノートPCもデスクトップPCと同じ環境が用意されている。授業外では、ノートPCで課題を行える。
- 詳しい方法は演習の時間に説明する

諦めないこと

- プロスポーツ選手は、説明だけ聞いて、すぐにそのスポーツのプロになったはずはない。誰よりも多く練習した人しかプロにはなれない。
- 野球ならば、プロになるまでにどれだけ素振りをしたのか、どれだけノックを受けたのか。1回素振りをしただけで、試合でホームランが打てるはずがない。
- どんな学習でも技術でも同じ。つまらないと思える基本をしっかりと身に付けることで、もっと難しい応用や実践にも対応できるようになる。
- つまり、1年次のプログラミングは素振り練習やノックと同じく、どれだけ繰り返すかがカギになる！

講義確認問題の予告

- 次の演習で、この講義の内容を理解したか確認問題を出す
- 講義のノートを取ること
- ノートを手で取ることによって記憶が強化される
- スクリーンを携帯で写メしても、携帯が賢くなるだけ

Java 言語
(Java Language)

プログラミングとは

プログラム：コンピュータを動作させる手順を書いたもの

(例：写真表示のプログラム)

写真データを読む

→画面表示できるデータに変換

→画面に表示する

コンピュータは**プログラムの通り**に動作する

プログラムが間違っているとコンピュータも**プログラムの通りに間違った動作をする**

プログラミング言語の誕生

- コンピュータの心臓部であるCPU(中央演算装置)では、非常に単純な命令(マシン語)しか理解できない
 - メモリからデータを読む・書き出す
 - 計算 (足し算、ビット演算、比較)
 - 特定の条件の時に、別の命令に制御を移す
- しかし、プログラムの手間がかかり、ミスを探すのが大変
- そこで、人の言葉に近い「プログラミング言語」が開発された

様々な高水準プログラミング言語

何千種類もプログラミング言語が生まれている

FORTRAN (1957)

LISP (1958)

COBOL (1959)

BASIC (1964)

LOGO (1967)

C (1972)

Java (1995)

自然言語とプログラミング言語

自然言語(日本語、英語など)

少し間違っているけども、適当に（コンテキストに合わせて）解釈される

(例：きのうはかさをわすれてしまった)

プログラミング言語

少し間違っただけでプログラム全体が動かない、あるいは、プログラムが途中で止まってしまう

厳密で曖昧さが無い

構造化プログラミング (structured programming)

- 3つの基本制御構造(逐次、分岐、反復)、および関数、ブロックを用いて、プログラムの制御および構造を分かりやすくした
- 階層構造も重要
- 2年次のオブジェクト指向プログラミングの基礎となる

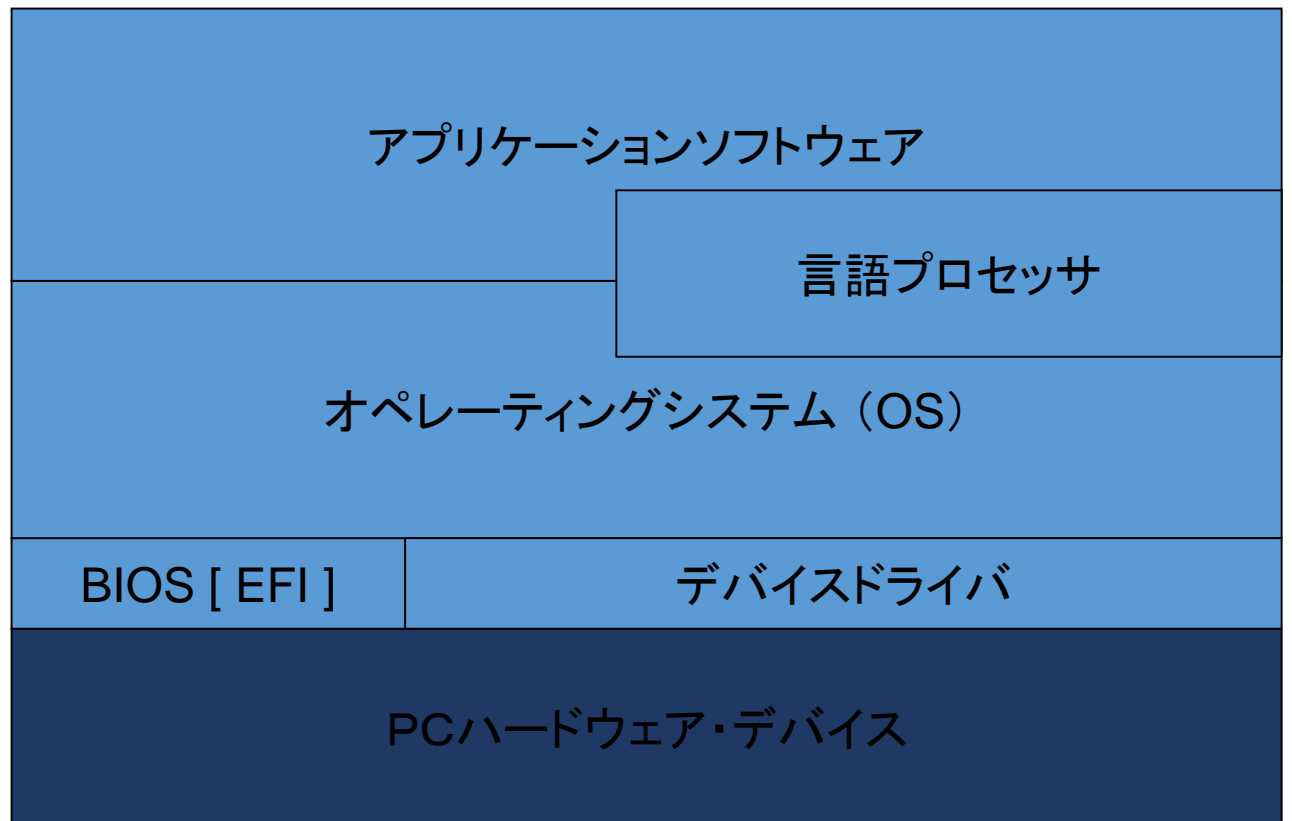
なぜJava?

- 現在システム開発の主流言語
 - 特にサーバサイドで多く利用されている
- 本格的なオブジェクト指向言語
 - 大規模なソフトウェア開発で広く普及
- 様々な環境で動作(UNIX(Linux), Windows, Macintosh, Android, ...)
- Cに似せてあるため、C/C++への移行も容易
- 2年次以降のプログラミングの基礎となる

Linuxとは

基本ソフトウェア

- ソフトウェアは大きく分けてハードウェアに近い基本ソフトウェアと、ユーザに近い応用ソフトウェアに分けることができる
- 基本ソフトウェアには、
 - BIOS [EFI]
 - OS
 - デバイスドライバ
 - 言語プロセッサ
(ライブラリ)などがある



応用ソフトウェア
基本ソフトウェア

OS

- オペレーティングシステム (Operating System)
- 以下の機能を提供する
 - 管理機能
 - プロセス管理・メモリ管理・ファイル管理・周辺機器管理
 - ネットワーク機能
 - ユーザインタフェース提供
 - A P I (開発用ライブラリ) 提供

OSの歴史

- 初期のコンピュータはOSはなかった
- 汎用機
 - GM OS & GM-NAA I/O (1955)
 - IBM System/360 (1966) ⇒ IBM z/OS (2001)
- マイクロコンピュータ用
 - CP/M (1977) ⇒ 86-DOS (1980) ⇒ MS-DOS (1981) ⇒ Windows 1.0 (1985) ⇒ Windows 3.0 (1990)
 - OS/2 (1987) ⇒ WindowsNT(1993)

OSの種類

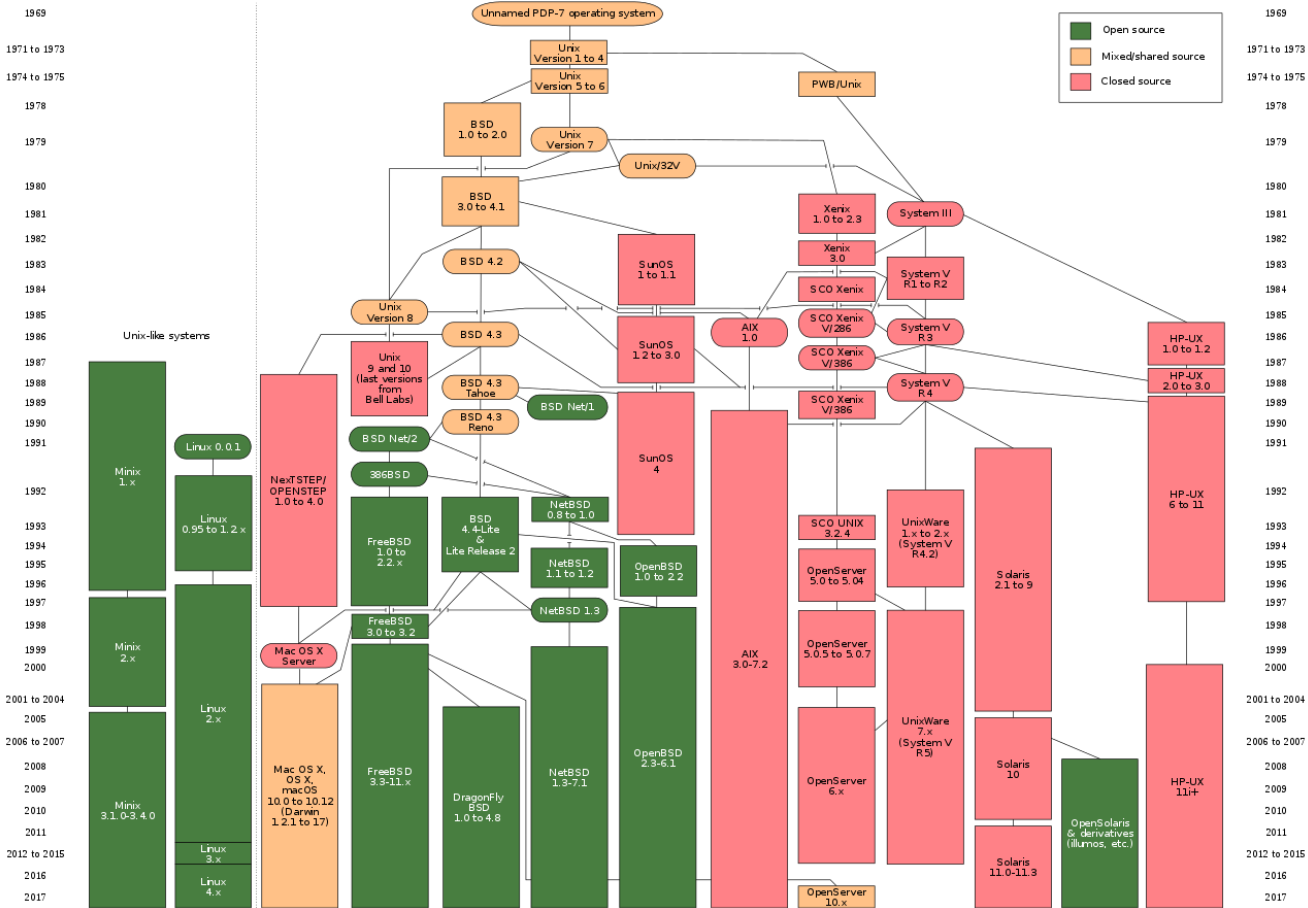
- PCで動作するOSも様々存在する
 - Microsoft系
 - MS-DOS, Windows3.1, Windows95, Windows98, WindowsME
 - WindowsNT, Windows2000, WindowsXP, Windows Vista, Windows7, Windows8, Windows10
 - UNIX系
 - Linux, FreeBSD, MacOS, Solaris, BeOS
 - その他
 - CP/M, OS/2, OS-9, BTRON (TRON)

UNIX系(Unix-like) OS

UNIX

- マルチタスク・マルチユーザ機能を持つOS
- 1969年AT&Tベル研究所で開発
- C言語で書かれている（初期はアセンブリ言語）
- 長い歴史の中でいくつかの系譜ができた
 - BSD(Berkeley Software Distribution)系
 - System V系
 - 独自（MINIX, Linux）
- 近年、PCで動作するPC-Unixが普及している
 - Linux, FreeBSD, MacOS, Solaris, BeOS, ...
- iOSやAndroidもUNIX系

UNIX系譜



By Eraserhead1, Infinity0, Sav_vas - Levenez Unix History Diagram, Information on the history of IBM's AIX on ibm.com, (CC BY-SA 3.0) <https://commons.wikimedia.org/w/index.php?curid=1801948>

Linux

- 1991 Linus Torvalds によりPC/AT互換機用に開発されたUnix-like OS
- 最初はMINIXをベースに開発された
- フリーかつオープンソース
- GNUライセンス
- 近年はビジネスユースも広がっている
- Linuxには多くのディストリビューション(配布元)がある
 - Debian系, Red Hat系, Slackware系, 等

なぜLinux?

- Java言語はマルチプラットフォームなので、様々なOSで動く
- UNIX(Linux)はネットワークサーバで広く利用され、特にサーバ側のシステム開発で必須
- 情報システム開発・ネットワーク構築においては、UNIX(Linux)の知識は必須
- UNIX(Linux)を使いこなすことで、今後のシステム系・ネットワーク系授業に役立つ
- Linuxは世界で最も普及しているOS
(AndroidもLinuxベース)

確認事項（全体）

- JPort履修登録で「プログラミング基礎」「プログラミング基礎演習」両方が履修登録されているか確認する
- 次週は、ノートPCを持ってくること

確認事項（情報システム学系）

4限は各クラスに分かれて実施

- J19003-J19094 aクラス 102教室（マッキン）
- J19095-J19189 bクラス 323教室（岸本）
- J19191-J19266 cクラス 203教室（北風）
- J19267-J19353 dクラス 204教室（海老原）
- J19354-J19427 eクラス 221教室（平塚）
- J19428-J19487 fクラス 322教室（早稲田）
- 再履修者は、クラスを伝えるので前に来ること

確認事項（数理情報学系・社会情報学系）

数理情報学系・社会情報学系学生は木4-5限のクラスを受講する

- 社会情報学系の学生には履修者制限(10名)がある
- 社会情報学系の学生は前にくること

Java スコーラ (課外講座) の紹介