



## 第2回

プログラムの入力, コンパイル, 実行  
(1.1~1.3)

AJ科 宮川 治

# 概要1

- プログラムとは、コンピュータに対する命令(処理)を記述したものです。大まかに、以下の2形式に分けられます。
  - ソースコード:人間が理解できるもの
  - バイトコード:マシーン(機械)が理解できるもの
- コンパイル
  - ソースコードをバイトコードに翻訳すること。

# 概要2

- コンパイル言語 ⇔ 動的プログラミング言語
  - コンパイル言語: ソースコードからバイトコードに翻訳し、バイトコードを実行する
  - 動的プログラミング言語: ソースコードを実行時にバイトコードに翻訳
- 汎用プログラミング言語 ⇔ ドメイン固有言語
  - 汎用プログラミング言語: 特定の用途に特化しない
  - ドメイン固有言語: 特定の用途向け

⇔: 対義語

# 概要3

## ■ 特殊記号

- シングルクォート( ' )、バッククォート( ` )
- ダブルクォート( " )、ハイフン( - )
- アンダーバー( \_ )、パイプ( | )
- 小なり( < )、大なり( > )
- セミコロン( ; )、ドット( . )

# 進め方の注意1(再)

- ゼロから・最初から(スクラッチで)プログラミングできること。
- 汎用プログラミング言語でのプログラム作成に関して、実際にプログラムを作成しながら学びます。
  - 授業中に、エラーを経験することで授業外でも自身で対応できる力を伸ばします。
    - コンパイル・エラー(ソースコード)
    - 実行時のエラー(バイトコード)

## 進め方の注意2(再)

- プログラム作成のための環境を自身のPCに構築します。
  - プログラム作成に最低限必要な物はエディタとJDK(Java Development Kit)です。
- 本授業では、プログラム(ソースコード)の体裁に関するしても学びます。体裁とはプログラムの外見になります。体裁が整っていないプログラムは、文法的な間違いがなくても理解しづらいものになります。

## 進め方の注意3(再)

- タッチ・タイピングは各次授業の予習として練習してください。苦手でも練習することで身につきます。
- コピー・アンド・ペースト(コピペ)は悪き習慣になるため行いません。
- プログラミングでコピペが楽だと思う時は他の選択肢を学ぶ必要があります。

## 進め方の注意4(再)

- 本授業ではプログラム作成の手順を学び、自らプログラムを作成する能力を高めます。
  - 最終的なプログラム(ソースコード)を写すことではありません。
  - 作成手順を学び、プログラムの構造を理解することが必要となります。
- 教授者の指示に従ってスモール・ステップで学ぶことが重要になります。



# 括弧

- 言語（ソースコードを含む）の記述の中で、その一部を一对の括弧で囲むことにより、その中と外とを区切ります。
  - 波括弧対{ }
  - 丸括弧対( )
  - 角括弧対[ ]
- 入れ子
  - 括弧対の中に新たな括弧対を入れた記述（構造）のこと。

# [プログラムの作成]

- プログラム(ソースコード)を記述する作業をコーディングと言います。
- コーディング規約とは、ソースコードの記述で守るべきルールのことです。

# 設問1

- 今までのプログラミング経験を質問する。一番知っている言語はの単数選択
- その言語は、コンパイル言語／動的プログラミング言語か
- その言語は汎用プログラミング言語／ドメイン固有言語か
- コマンドプロンプトの出力結果を貼り付けること(md, cd)。dir コマンドの出力

## 設問2

- 括弧に関して質問します。それぞれの読み方を選択してください。
- 特殊記号に関して質問します。それぞれの読み方を選択してください。
- コマンドプロンプトの出力結果を貼り付けること(java)。dirコマンド
- プログラムを入力方法に従ったスナップショット