



第1回
ガイダンス

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

進め方の注意1

- 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ばす。
 - 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

- プログラム作成のためのソフトウェアのインストールを自身の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)の体裁に関する事も学びます。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

- 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

- 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。

【(アッパー)キャメルケース】

- (アッパー)キャメルケースとは、複合語をひと綴りとして、二つ目以降の要素語の最初を大文字で書き表すこと。複合語の要素数が一つでも下記のルールに従うものとする。
 - 一つ目の要素語の先頭が大文字：アッパーキャメルケース
 - 一つ目の要素語の先頭が小文字：キャメルケース

コーディング規約1

- ファイル名に関してはアッパーキャメルケースに従う。ただし.java拡張子を除く。また、数字の区切り文字にはアンダーバー(_)を使用する。
 - **アッパーキャメルケース**: 複合語の先頭を、大文字で書き始める。

コーディング規約2

- プログラム中のスペース(空白)は区切り文字として使用される。ただし、全角スペースを除く。
- プログラム中の複合語はスペースではなく、以下の3種類が主に使用される。
 - アッパーキャメルケース: 複合語の先頭を、大文字で書き始める。
 - キャメルケース: 複合語の先頭を、小文字で書き始める。
 - スネークケース: アンダーバー(`_`)を区切記号として要素語をつなげる。

文法

- メソッド中では、命令文の終わりにセミコロン(;)が必要である。
- 文字列のリテラル
 - 例: "Hello World"

プログラムの分割

- エントリーポイント: プログラムを実行するうえで、プログラムやサブルーチン(メソッド)の実行を開始する場所のこと
 - mainから実行

コーディング規約3

■ クラスの宣言

- クラス名はアッパーキャメルケースを使用する
 - アッパーキャメルケース: 複合語の先頭を、大文字で書き始める。

■ 振る舞いの(メソッド)の宣言

- 振る舞い名はキャメルケースを使用する。
 - キャメルケース: 複合語の先頭を、小文字で書き始める。単にキャメルケースと言えば先頭は小文字。
- **クラス図にアンダーライン**がある場合はstatic

オブジェクト指向

- 複雑さを隠蔽する技術
- 変数とメソッドのセット(組み合わせ)を基本要素にしてプログラムを作る考え方
- 基本要素のことをオブジェクト(インスタンス)と言う

オブジェクト(インスタンス)の設計図であるクラス図の注意点

■ クラス図

○ 一段目

- クラス名: アッパーキャメルケース

○ 二段目

- この授業で学ぶ
- 状態の変数(インスタンス変数)名

○ 三段目

- 振る舞い(メソッド)名: キャメルケース
- アンダーラインを引かない