令和6年9月11日

プログラミング（初級レベル）科目設計（案）

村田，嘉藤

# 1．プログラミング入門

目的：プログラミングの基礎概念を理解し，簡単なプログラムが作成できる．

内容

* データ型と変数
* 基本的な制御構文（if文、for文、while文）
* 基本的な入出力操作
* 関数の定義と使用
* 基本的なデータ構造（リスト、辞書）

特徴

* デバッグの基礎：エラーメッセージの読み方，基本的なデバッグ手法（プリントステートメントを使ったデバッグなど）を学ぶ．
* 簡単なコードレビュー：生成AIが書いたコードを読むことで，自分のコードとの違いを学び，コードの品質を向上させる方法を学ぶ．
* 生成AIの紹介：GitHub Copilotなどのツールを紹介し，どのように活用できるかを簡単に示す．ただし，これらのツールに頼りすぎないように注意を促す．

関連コース：プログラミング概論I（佐賀大・2年前期），プログラミング入門（熊本高専・2年前期）情報処理（熊本高専・2年後期）

■参考となる図書，コンテンツ

* 東京大学「python入門」（[PDF版](https://utokyo-ipp.github.io/IPP_textbook.pdf)，[HTML版](https://utokyo-ipp.github.io/IPP_textbook.pdf)，[Colob版](https://utokyo-ipp.github.io/IPP_textbook.pdf)）
* 新・明解Python入門 第2版　柴田 望洋：著者
* [ゼロからのPython入門講座](https://www.python.jp/train/index.html)
* プログラミング基礎I（熊本高専，2018，村田，C），プログラミング基礎II（熊本高専，2019，村田，C），プログラミング入門の一部（熊本高専2024，村田，Processing）,情報処理の一部（熊本高専2024，村田，Processing）

■利用コンテンツ：東京大学「python入門」（Colab版）

2020–2024, 東京大学 数理・情報教育研究センター (CC BY-NC-ND 4.0)

Google Colaboratory版を利用．生成AI Gemini搭載．Googleアカウントが必要．無料．

|  |  |
| --- | --- |
| 内容 | テキスト |
| データ型と変数，演算 | 1-1. 数値演算  1-2. 変数と関数の基礎  2-1. 文字列 |
| コンソール入出力 | 1-2. 変数と関数の基礎 |
| 基本的なデータ構造（リスト、辞書） | 2-2. リスト  3-1. 辞書 |
| 選択処理（if文） | 1-3. 論理・比較演算と条件分岐の基礎  2-3. 条件分岐 |
| 反復処理（for文、while文） | 3-2. 繰り返し  4-2. イテラブルとイテレーター |
| 制御構文の組み合わせ（if+if, for+for, if+for, for+if） | 3-2. 繰り返し |
| 基本的な入出力操作（ファイル入出力） | 4-1. ファイル入出力の基本  ▲CSVファイルの入出力 |
| 関数の定義と使用 | 3-3. 関数 |
| ライブラリの活用 | 5-3. NumPyライブラリ  7-1. pandasライブラリ  7-2. scikit-learnライブラリ  ▲Matplotlibライブラリ |
| テストとデバッグ | 1-4. テストとデバッグ |
| 上記以外 | 1-0. Colaboratory(Colab)の使い方 4-3. ディレクトリと木構造  5-1. モジュールの使い方  5-2. モジュールの作り方  6-1. 内包表現  6-2. 高階関数  6-3. クラス  ▲Jupyter Notebook の使い方  ▲セット(set)  ▲再帰  ▲簡単なデータの可視化  ▲Bokehライブラリ  ▲Pythonスクリプトとコマンドライン実行  ▲正規表現 |

■不足している教育コンテンツ

* 演習課題

# 2．アルゴリズムとデータ構造

目的：基本的なアルゴリズムとデータ構造を理解し，実装できる．

内容

* ソートアルゴリズム（バブルソート、挿入ソート、選択ソート）
* 検索アルゴリズム（線形探索、二分探索）
* 基本的データ構造（スタック、キュー、ツリー）

特徴

* 擬似コード：特定のプログラミング言語ではなく，擬似コードを用いてアルゴリズムのロジックを説明します．これにより，学生は概念を純粋に理解できます。
* 図解とフローチャート：アルゴリズムを視覚的に表現することで，言語の文法に囚われずに流れを理解できます．

関連コース：データ構造とアルゴリズム（佐賀大・2年前期），プログラミング入門（熊本高専・2年前期）情報処理（熊本高専・2年後期）

■参考となるコンテンツ

* データ構造とアルゴリズム資料（佐賀大，2019，掛下先生）
* アルゴリズム論　浅野哲夫，和田幸一，増澤利光：共著
* [VisuAlgo(様々なアルゴリズムを ブラウザ上で視覚的に再現してくれるサイト)](https://visualgo.net/en)  
  参考記事：<https://softantenna.com/blog/visu-algo/>

■作成済みの教育コンテンツ

■不足している教育コンテンツ

* 各動画コンテンツ
* 動画コンテンツ作成ソフトの利用
  + Storm Maker 　https://suite.logosware.com/storm-maker/

# 3．プログラミングを通じた問題解決

目的：問題解決の手法を学び，プログラミングを通じて具体的な問題を解決する能力を養う．

内容

* 問題の分解とアルゴリズム設計
* 擬似コードとフローチャート
* 簡単なプロジェクトを通じた実践

特徴

* 問題の分解：大きな問題を小さな部分に分割し，それぞれを解決する方法を教える．
* アルゴリズム設計：具体的なアルゴリズムを設計し，それを異なる言語で実装する練習を通じて、アルゴリズム思考を養う．
* 関連コース：プログラミン演習I（佐賀大・2年前期），応用情報処理I（熊本高専・3年前期）

■参考となるコンテンツ

* ICPC大学対抗プログラミングコンテスト過去問（ProblemA程度）
* JOI２次予選過去問題（１次予選は簡単，本選は難しい）
* パソコン甲子園過去問
* プログラミング基礎I（熊本高専，2018，村田，C）
  + 2つの抵抗と電圧，接続方法を入力して，電流を求める．
  + 実数を入力して，実数に応じた＊を表示させる．＊の表示にはprint(“\*”)のみが利用可能．
* プログラミング基礎II（熊本高専，2019，村田，C）
  + 3×3に並んだ〇×について，盤面を表示し，〇が3つ並んだラインの数を求める．
  + 10個の整数が保存されている．順位を入力して，その順位の整数を表示させる．

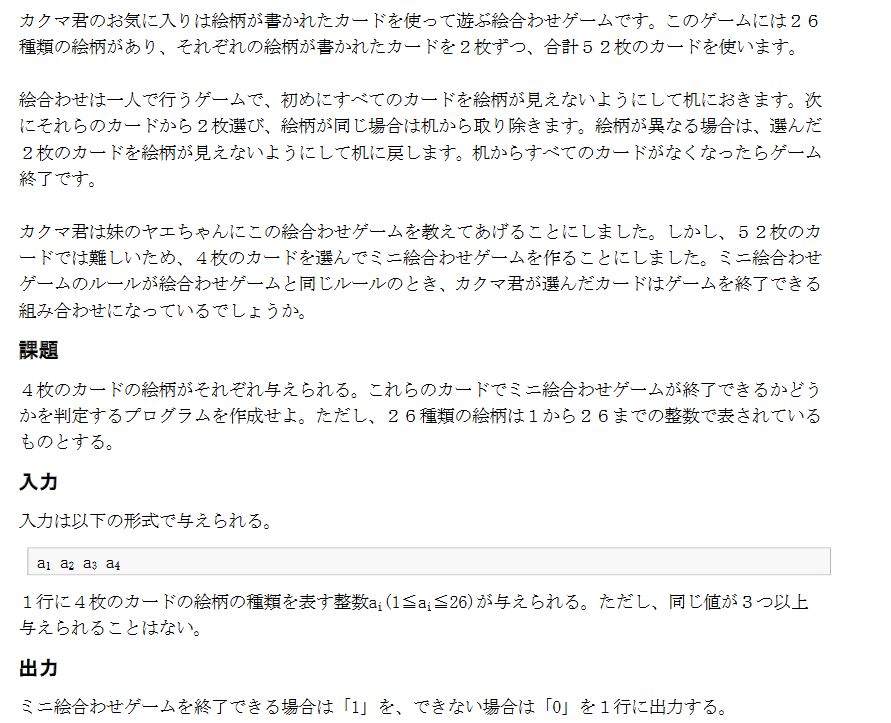
■作成済みの教育コンテンツ：

■不足している教育コンテンツ

* 開発環境の構築（python+VSCord+GitHub）（動画）
* 参考：
  + [Github Copilotとは？使い方や料金、VScodeへの導入方法を解説 | AI総合研究所 (ai-souken.com)](https://www.ai-souken.com/article/github-copilot-overview)
  + [GitHub CopilotをVSCodeに導入して開発効率をグン上げする](https://maasaablog.com/blog/klob7evqu/)  
    　　　　（上記サイトのYouTube: https://www.youtube.com/watch?v=0E5isjlv\_KQ）
  + [VSCode ではじめる GitHub Copilot 活用術 #githubcopilot - Qiita](https://qiita.com/RyoWakabayashi/items/1207128e88669c76bf5f)
  + [python japan:VSCode開発環境のインストール](https://www.python.jp/python_vscode/windows/setup/index.html)
* 生成AIの活用（コード生成，自動補完機能，エラー修正）（動画）

参考：

* + [Visual Studio CodeとGitHub Copilotでコーディング効率を革新！AIを駆使した開発ガイド](https://sqripts.com/2024/01/25/88348/)
  + [VSCodeでGitHub Copilotを効率的に使用する方法](https://kakaku-techblog.com/entry/vscode-github-copilot)
* 例題を通した問題分解とアルゴリズム設計方法，疑似コードとフローチャートを利用．（2問程度）（動画）
* 演習課題（10問程度）
  + 難易度を易しい，普通，難しいに分類し，それぞれ3～4問を準備する．
    - パソコン甲子園の問題は，問題ごとに配点（2点から12点）付きなので，難易度が付しやすい
  + 例（2021 年 9 月実施・パソコン甲子園予選問題　3点）



# 5．全体的な教育方針

初級レベルのプログラミング教育では，基本的なプログラミングスキルと問題解決能力に重点を置き，デバッグやコードレビュー（コーディングスタイル，可読性，再利用性，汎用性）の基礎を教えることが重要である．また，新しいツールや技術に触れる機会を設けることで，学生の興味を引き出し，現代のプログラミング環境に適応できるようにすることも大切である．

## 5.1 基本的なプログラミングスキル（プログラミング入門）

* 概念の説明は従来のプログラミング教育と同様とする．
* 演習は，Google Colaboratoryを活用する．

## 5.2 デバッグやコードレビューの基礎（プログラミングを通じた問題解決）

* デバッグは，GitHub Copilotのチャット機能を利用する．
* 生成されたコードを確認させる．
  + 不明な点は，GitHub Copilotを利用する．

## 5.3. 新しいツールや技術の活用（プログラミングを通じた問題解決）

* GitHub Copilot Individualの申請

参考：

* + [【GitHub Docs】学生として GitHub Education に応募する](https://docs.github.com/ja/education/explore-the-benefits-of-teaching-and-learning-with-github-education/github-education-for-students/apply-to-github-education-as-a-student)
  + [【GitHub】学生申請をして無料でGitHub Copilotを使う #Swift - Qiita](https://qiita.com/SNQ-2001/items/796dc5e794ac3f57a945)
* Visual Studio CodeとGitHub Copilotの連携

参考

* + [VSCode ではじめる GitHub Copilot 活用術 #githubcopilot - Qiita](https://qiita.com/RyoWakabayashi/items/1207128e88669c76bf5f)
  + [GitHub Copilot導入後、初めて使う時。(豊富な使用例付き) #githubcopilot - Qiita](https://qiita.com/masakinihirota/items/0e58a6b921e4420a2882)
  + [Github Copilotとは？使い方や料金、VScodeへの導入方法を解説 | AI総合研究所 (ai-souken.com)](https://www.ai-souken.com/article/github-copilot-overview)

## 5.4 まとめ

生成AIを活用したプログラミングでは，生成AIが提案するコードや解説を利用した基本概念やアルゴリズムの理解，またアルゴリズムの応用に変化すると考えられる．従来のプログラミング教育においては，実施のコーディング演習を行っていたが，生成AIを活用することで細かい文法やエラー修正に費やす時間を減らし、基礎概念の理解、アルゴリズムの応用が可能である．