

# ChatGPT のプログラミング演習での利用の可能性

北英彦\*1・中田大雅\*1  
Email: kita\_yh@yahoo.co.jp

\*1: 三重大学工学部

◎Key Words ChatGPT, プログラミング教育, プログラミング演習

## 1. はじめに

著者が ChatGPT<sup>(1)</sup>の存在を遅ればせながら意識したのは4月末の新聞の記事でレポートの作成,特に,コード生成ができると書いてあったためである<sup>(2)</sup>。プログラミング演習を担当する教員として学生が ChatGPT で何ができるのか,授業としてどう対応したらよいかを速やかに検討する必要を感じた。本発表では,短時間で調べた結果ではあるが ChatGPT のプログラミング教育での利用可能性の調査結果と試行錯誤しながらの授業の運用状況を報告する。また,授業でのプログラミングの授業における ChatGPT の暫定的な利用法を提案する。

## 2. ChatGPT

ChatGPT は OpenAI 社が開発した大規模言語モデル GPT-3.5 として用いた無料で利用できるチャット(会話)のサービスである。「言語モデル」とは文章の並び方に確率を割り当てるものであり情報を大規模に収集することでもっともらしい文章を生成できる。

「ChatGPT とは何か」を ChatGPT に質問した結果のプログラミングに関する部分の抜粋を示す。

- 情報提供: 一般的な知識に関する質問に答えることができる。2021年9月までの情報である。(考察: プログラミングに関しては影響が少ない)
- 質問応答: プログラミング, 数学, 科学的な概念などに関する質問に対しても回答を提供する。(考察: プログラミングを重視している, 数学の入試問題は解けない)
- 対話: 具体的なトピックについての会話, また, 質問への追加の質問や指示に対しても応答する。(考察: 対話的に生成するコードの修正が可能である)

OpenAI 社は GPT-3 をベースとして自然言語による要求を解析し対応するコードを生成する `codex` を開発した。`codex` はソフトウェア開発用プラットフォーム `GitHub`<sup>(3)</sup> に保管されている大量のコードを学習に用いている。逆に, `codex` は `GitHub Copilot`<sup>(4)</sup> においてコーディング作業を支援するために利用されている。推測ではあるが ChatGPT は `codex` の成果を利用しているものと思われる。

試してみたところ, Google の Bard, Microsoft の Bing AI でもコード生成は可能であり演習で作成するような簡単なコードについては生成されるコードにほとんど差はない。ChatGPT は生成したコードの説明を丁寧にしているのが特徴であり, 現時点では ChatGPT が先行していることもあり, 授業での利用は ChatGPT がよさそうである。

## 3. 三重大学の対応

東京大学では4月3日の時点で ChatGPT などの利用に関する文書を公開しており<sup>(5)</sup>, 他大学でも同様な表明をしている。使用を禁止することは事実上不可能であるので, 誤りを含むこと・そのままの利用が自分の学習にならないことを注意するものが多いようである。三重大学でも遅ればせながら学外には非公開だが5月30日に ChatGPT などの利用に関して以下の留意事項を学生・教員に周知している。

- 参考文献に類するものとして利用を明記すること
- 授業によっては利用を禁止することがある
- 不正行為として懲戒処分の対象となることがある
- そのままの利用は自分の学習にならない
- 誤りが混ざっていることがある, 自分で確認すること, 参考程度にとどめること
- 著作権侵害として処罰の対象になるおそれがある
- 個人情報の漏洩につながるおそれがある

## 4. 授業での対応

今年度のプログラミング演習の授業における ChatGPT に関する試行錯誤での運用を時系列で紹介する。

- 5月9日: ChatGPT の利用に関する指導(初回)
  - ChatGPT を用いてもかまいません。(使用は禁止しない, 利用の推奨はしていない)
  - できること(コードの生成だけでなくデバッグで使えることを伝える)
    1. コードのデバッグ
    2. スタイルの修正(一部, 授業のコーディング規則と異なる)
    3. コードの生成(一部, 授業の運用と異なる)
  - プログラミングの知識を理解してプログラムの記述ができればこの科目に合格はできない。(安易な使用への警告)
- 5月9日: ChatGPT を用いた例の公開
  - デバッグの例とコード生成の例を公開した。どちらの例も誤りを含むこと, 授業の運用と異なる点を説明するものである。
  - Moodle 上で公開しただけなので全員が確認しているとは限らない。
- 5月16日: ChatGPT の利用状況: 95名中
  - 4月25日の演習でコードの作成に使用: 19名
  - 他の科目の宿題で使用: 22名
  - 東北大学の調査ではほぼ同時期に3割使用<sup>(6)</sup>
  - 考察: 初回の指導の前の時点で約2割の学生が ChatGPT を使用している。

- 5月30日：ChatGPTの使用の演習
  - ChatGPTにできることを理解させるために、全員にChatGPTの使用を授業として体験させる。
  - 課題：動作エラー・例外発生のある下記のメソッドをChatGPTを用いて修正すること、このメソッドの動作確認をするメインメソッドも記述すること

間違いの部分に下線（学生には非通知）

```
public static void reverse(int[] array) {
    for (int i = 0; i < array.length; i++) {
        array[i] = array[array.length - i - 1];
    }
}
```

- 課題の提出状況（出席者101名）
  1. 提出：77名
  2. ギブアップ・未提出：24名
- 考察：ChatGPTは妥当なコードを生成しておりほぼそのまま利用するだけで動作するはずなのだが、4分の1の学生は授業で用いている開発環境で動作するように加筆修正することができていない。プログラミング能力の低い学生（出席確認の小テストの点数でおおかかに確認）はChatGPTの生成したコードを利用できていない。ChatGPTの回答を利用するためには、ある程度のプログラミング能力が必要である。

- 5月30日：ChatGPTの利用の演習
  - 課題：与えられた配列の要素の値を反転した配列を生成するメソッドをChatGPTを用いて記述すること、与えられた配列の要素の値は変更しないこと、このメソッドの動作確認をするメインメソッドも記述すること
  - 課題の提出状況（出席者101名）
    1. 提出：55名
    2. ギブアップ・未提出：46名
  - 考察：上記の考察とはほぼ同じだが、未提出者の中にはプログラミング能力が低く時間不足で加筆修正を終了できていないものが多数含まれていると思われる。

- 6月6日：プログラムヘッダーのルールの変更
  - 大学のChatGPTの使用の有無の明記の方針に従いChatGPTの使用の有無を明記するように授業のルールを変更した。

```
/*
 * 課題番号：#NN
 * プログラムの概要：(処理の内容を簡潔に)
 * 作成者：
 * :
 * 著作権表示：(自分に著作権があることを意識
 *   させるために明示させる)
 * ChatGPTなどの利用：未使用, デバッグに使用,
 *   コードの生成に使用 (該当するものを残す)
 */
```

- 6月6日：三重大学からの留意事項の小テスト
  - 1日間隔で合計10回受験可能、周知のため
  - 初回の結果：出題前の予想通り、ほとんど学生が留意事項を読んでいないか覚えていない

- 6月6日：ChatGPTを強制的に使用させる小テスト
  - 1日間隔で合計10回受験可能、ChatGPTの回答が不十分であることを理解させるため
  - コード生成だとChatGPTの回答にある間違いを学生が見つめることが困難なので、解くのに少し工夫の必要な数学の問題でChatGPTを強制的に使わせる。試して見た結果では「少し工夫が必要な」の部分があるとChatGPTは正しく回答できないことが多い。

- 問題
  1. 以下の連立方程式のすべての解をChatGPTを用いて求めること
  2.  $2xy + y^2 - y = 0$ ,  $x^2 + 2xy - x = 0$
  3. 多肢選択の中からChatGPTが回答した解のみを選ぶこと

- 初回の結果：ほとんどの学生が(1/3, 1/3)をChatGPTに回答させることができていない

- 6月9日～：連立方程式の問題の実施で理解したこと（学生の解答例）

- 二度同じ質問をすると違う答えを出してきた。
- 3分の1がなかなか出ないため、分数の計算が苦手なのかもしれない。
- 答えから逆算するように命令したら「前回の回答が正しくないと気づきました。」とだけ言い、逆算の過程を見せずにもう一度解き直し始めた。しかし計算速度が著しく落ちていた。
- 計算過程を見ると、因数分解はできていたが、代入ができていなかった。
- 回答を求めても必ずあっているとは限らず、参考までにするのが賢明である。
- たまに的外れの回答を示すため気を付けて情報を取捨選択しなければならない。
- 人間と会話しているような文章で解答を得ることができる。(重要)
- 何通りかある答えだったため、二段階に質問しなければならない。(重要)

- 6月13日：ChatGPTの可能性・限界に関する多肢選択子の出題（抜粋、正解に☆、不正解に★）

- ドメイン知識がないので必要に応じて質問に付け加える必要がある☆
- 常に最適なコードを生成する★
- 大量の情報に基づく妥当なものなのでそのままコピー&ペーストして提出してよい★
- 生成したコードが課題で要求されている要件を満たしているかを確認する必要がある☆
- 生成したコードをコンパイル・実行して動作確認をする必要がある☆
- 同じ内容の質問を何度もしてはならない★
- 授業で定めているコーディング規約を満たしているか確認する必要がある☆
- デバッグに使用してはならない★
- 生成したコードが課題で要求されていない処理をしていないか確認する必要がある☆
- 生成したコードの動作を説明できるようにならない★

- 6月20日：質問の仕方の公開
  - 処理の流れのアドバイスを求める場合の質問の仕方（プロンプティングの仕方）を公開した。
  - 「具体的なコードを示さないでください。」を質問の最初と念のため最後に書く
  - 課題文に書かれていない前提の情報も質問文に書く、例：配列を使って自然数の集合を表す、集合の要素は昇順に格納されているとする
- 6月20日：上記の利用の可能性の小テスト
  - 問題（ChatGPT への質問）：具体的なコードを示さずに処理の流れを解説してください。二つの自然数の集合の差集合を求める。（省略：詳細条件）具体的なコードは示さないでください（念を押しておく）。
  - ChatGPT からの回答（抜粋）：
    1. 二つの自然数の集合を昇順に格納された配列 A と配列 B とする。
    2. 差集合 A-B を格納するための空の配列 diff を作成する。
    3. インデックス変数 i と j をそれぞれ 0 で初期化します。これらの変数は配列 A と配列 B の要素にアクセスするために使用する。
    4. i が A の最後の要素のインデックスを超えるまで以下の手順を繰り返す。（不正確、学生には非通知）
      - A[i] と B[j] が等しい場合、j を 1 つ進めて次の要素を比較する。（不正確、学生には非通知）
      - A[i] が B[j] よりも小さい場合、A[i] を diff に追加して i を 1 つ進める。
      - A[i] が B[j] よりも大きい場合、j を 1 つ進めて次の要素を比較する。
    5. ループの終了後、A に要素が残っている場合は、残りの要素を diff に追加する
  - 学生の解答（コーディング前の見込み）81名
    1. 確実に記述できる：6名
    2. ほぼ確実に記述できる：14名
    3. おそらく記述できる：24名
    4. 五分五分：19名
    5. おそらく記述できない：9名
    6. ほぼ記述できない：6名
    7. 確実に記述できない：3名
  - 問題（ChatGPT への追加質問）：差集合
  - 問題（ChatGPT からの追加回答）（抜粋、例あり）：差集合を日本語で表現すると、「A の要素であり、B の要素ではないものからなる集合」となる。
  - 学生の解答（コーディング前の見込み）81名
    1. 確実に記述できる：11名
    2. ほぼ確実に記述できる：22名
    3. おそらく記述できる：22名
    4. 五分五分：14名
    5. おそらく記述できない：8名

- 6. ほぼ記述できない：1名
- 7. 確実に記述できない：3名
- 考察：ChatGPT が示した処理の流れの中の用語の意味が理解できれば約 7 割の学生がコードが作成できると回答している。
- 6月20日：生成したコードを利用する場合の指示
  - コードそのものを質問するのは最後の手段とすること
  - ChatGPT が生成したコードの動作を説明できるようにすること
  - 類似の課題に対して自分でコードが記述できるようにすること

## 5. 授業での対応に関する考察

- 授業で ChatGPT に関して紹介する前に ChatGPT を使用したことのある学生が約 2 割いた。授業で強制的に使用の経験をさせたので履修者の全員が利用できることを認識しているはずである。
- ChatGPT の回答に誤りがあったり回答の内容が不十分だったりすることを、ChatGPT の使用例・小テスト・演習で繰り返し体験させたので、ChatGPT の回答をそのままは利用できないことは理解しているはずである。ただし、演習中のコード作成途中の状況やギブアップしたコードを見ると、プログラミング能力の低い学生の中には自分で加筆修正ができずにそのまま利用している学生がいた。
- ChatGPT の回答を利用して課題の要求通りに動作するコードを完成させることができない学生がいる。ChatGPT の回答を利用する場合でも、ある程度のプログラミング能力が必要である。処理の流れを ChatGPT から提供されても処理の流れの中の 1 段階ずつをコードに表現できない学生がいる。Java の各機能の理解が不十分であるか、または、どの機能を用いたらよいか、または、それらをどのように利用できるかの発想できないことが理由として考えられる。

## 6. 数学の入試問題を用いた調査

プログラミングから外れるが ChatGPT が「数学」に対しても回答できると回答していたので、三重大学の昨年度の数学の入試問題で試してみた。比較的簡単である小問題集（問題 1）の中の問題で試してみた。

- 質問（小問の最初の部分）  
 $\cos x = a$  として、 $\cos 3x$  を  $a$  で表せ。
- chatGPT の回答（最後の部分のみ）  
 $\cos(3x) = (a)^3 - 3a * (\sin(x))^2$  (5月初旬)  
 $\cos(3x) = 4 * a^3 - 3 * a$  (6月末)
- 質問の仕方の変更  
 $\cos 3x$  を  $\cos x$  で表せ。
- $\cos(3x) = 4 * (\cos(x))^3 - 3 * \cos(x)$

5 月末の時点では、受験生なら当然気が付くはずの  $(\sin(x))^2$  が  $\cos x$  で表せることに気が付いていない。ChatGPT は同じ日に同じ質問をしても回答が異なることがある。期待する回答が得られるまで質問の表現を変えたりして再質問する必要がある。ただし、数学の問題の場合は時間をかけても正解できないことが多々あった。

## 7. ChatGPT のコード作成能力

ChatGPT のコード生成能力を調べるために演習で用いた課題のコードを ChatGPT に生成させてみた。また、デバッグで利用できるかどうかを調べてみた。プログラミング演習を担当する教員として妥当なコードを生成することができ、また、デバッグに利用できることが分かった。以下に具体的な例を示す。

- 課題 1: 「Hello, World.」と表示するプログラムを Java で作成すること

- 課題 1 に対して ChatGPT の生成したコード  

```
public class HelloWorld {  
    public static void main(String[] args) {  
        System.out.println("Hello, World.");  
    }  
}
```

- 評価 1: 妥当なコード、ただし、授業の開発環境ではクラス Main しか記述できないので構文エラーとなり実行できない

- 課題 2: 以下の Java のコードを書きなさい

- 三角形の 3 辺の長さをひとつずつ 1 行で入力させる、適切な入力プロンプトを表示すること、入力プロンプトと同じ行で辺の長さをひとつ入力させる、3 行で入力させる
- 「辺の長さが 3, 4, 5 の三角形の種別を調べます。」のように表示する。

- 課題 2 に対して ChatGPT の生成したコード: 省略

- 評価: ほぼ妥当なコード、下記のように不十分な部分および課題で要求されていない余分な実装がある

- 入力プロンプトの表示が不適切である、「辺の長さ」のみ表示しており何の辺の長さがプログラムの利用者にわからない、「三角形の辺の長さ」と何の辺の長さか明示すべきである
- 辺の長さの型が不適切である。表示例「辺の長さが 3, 4, 5 の三角形の種別を調べます。」を見ると辺の長さは整数であることが分かる、課題の文章全体をみて型を推測することができていない
- 課題で要求している『「辺の長さが 3, 4, 5 の三角形の種別を調べます。」のように表示する.』がコーディングされていない
- この課題は少しずつ三角形の種別を判別させるコードに近づけていくいくつかの課題の最初の課題であるので、ある意味不自然ではあるのだが「辺の長さが 3, 4, 5 の三角形の種別を調べます。」までで止めてある、これに対して「正三角形」「二等辺三角形」「不等辺三角形」の判別という要求された以上のことがコーディングされている、三角形の種別の判別をさせる方が一般的であるので言語モデルに蓄積してあるコードから三角形の種別までコードを生成していると思われる

- デバッグに用いた場合: ほぼ完璧、「構文エラーを修正してください」という文がなく構文エラー・スタイル違反があるコードのみを入れても適切にデバッグ・修正がされる、メソッド名などから処理内容が分かる場合には動作エラーもデバッグされる

## 8. 授業での ChatGPT の利用の提案

プログラミングの授業における ChatGPT の暫定的な利用法を提案する。「ほぼ」の部分は学生がコードを理解し適切に修正する必要がある。

- ChatGPT ができることを伝える。
  - コードのデバッグ (利用に問題はない、ほぼ完璧でデバッグできる)
  - スタイルの修正 (利用に問題はない、ほぼ完璧にスタイルを修正できる)
  - 処理の流れのアドバイス (処理の流れはまずは自分で考えてみることに、処理の流れを考えることができることはプログラミング能力の一部)
  - コードの断片の生成 (処理の流れの各段階を見てもコードを書けない場合にはコード全体ではなくその部分のみの書き方を質問すること)
  - コード全体の生成 (コードの断片を組み合わせても課題で要求する動作を実現できない場合に用いる、最初からコード全体を生成させないこと、コード全体を生成させた場合は生成されたコードの動作を理解できるようにすること)
- ChatGPT を利用するときの注意事項を周知する
  - 間違いを含む場合がある
  - そのままの利用は自分の学習にならない
  - 個人情報の流失がないように気を付ける
  - ChatGPT を利用した場合は明記する
- ChatGPT を用いた演習を 1 回のみ実施する
- ChatGPT の回答に誤りがある可能性があることを小テストなどで実感させる

以上は、従来の授業のやり方の中で ChatGPT をどのように利用するかを暫定的に提案したものである。これだけでは不十分で、ChatGPT などの生成 AI の出現により、授業の内容や授業のやり方、そのものを根本的に見直す必要があるかもしれない。

## 9. おわりに

ChatGPT のプログラミング教育での利用可能性の調査結果と試行錯誤しながらの授業の運用状況を報告した。また、プログラミングの授業における ChatGPT の暫定的な利用法を提案した。原稿の提出時点では前期の授業の中間の時点であるので、発表のときには前期終了時の状況を含めて報告する予定である。

## 参考文献

- (1) ChatGPT, <https://openai.com/chatgpt> (参照: 2023 年 6 月 28 日)
- (2) “AI 時代「もう後戻りできない」”, 朝日新聞, 2023 年 4 月 21 日, 朝刊, 2 ページ, 東京本社
- (3) GitHub, <https://github.co.jp/> (参照: 2023 年 6 月)
- (4) GitHub Copilot, <https://docs.github.com/ja/copilot/getting-started-with-github-copilot> (参照: 2023 年 6 月)
- (5) “生成系 AI(ChatGPT, BingAI, Bard, Midjourney, Stable Diffusion 等)について”, 東京大学, <https://utelecon.adm.u-tokyo.ac.jp/docs/20230403-generative-ai> (参照: 2023 年 6 月)
- (6) “対話型 AI 大学生の 3 割使用”, 朝日新聞, 2023 年 6 月 8 日, 朝刊, 2 ページ, 東京本社