D-2

情報Ⅰのプログラミング教育に生成AIを利用することの是非

◆井手広康◆

愛知県立旭丘高等学校 教諭

k619154u@gmail.com

本研究の目的

- 令和4年11月にOpenAI社から**生成AI**であるChatGPTが公表されると，急速に世界中に普及し，教育の場での利活用も広まっている
- 文部科学省のガイドラインでは，「**プログラミングの授業において，児童生徒のアイディアを実現するためのプログラムの制作に活用する**」との記述がある
- プログラミング教育における生成AIの**活用事例**はまだまだ少なく，授業での生成AIの活用を躊躇している教員は多い（はず）

プログラミング教育に**生成AI**を活用し，その**効果・影響**について分析する

プログラミングの授業概要

対象生徒

年生 **１９９名**（井手が担当する５クラス）　※残り４クラスは別の教員が担当

実施科目

課題研究（情報）　※学校設定科目（情報  の代替科目）

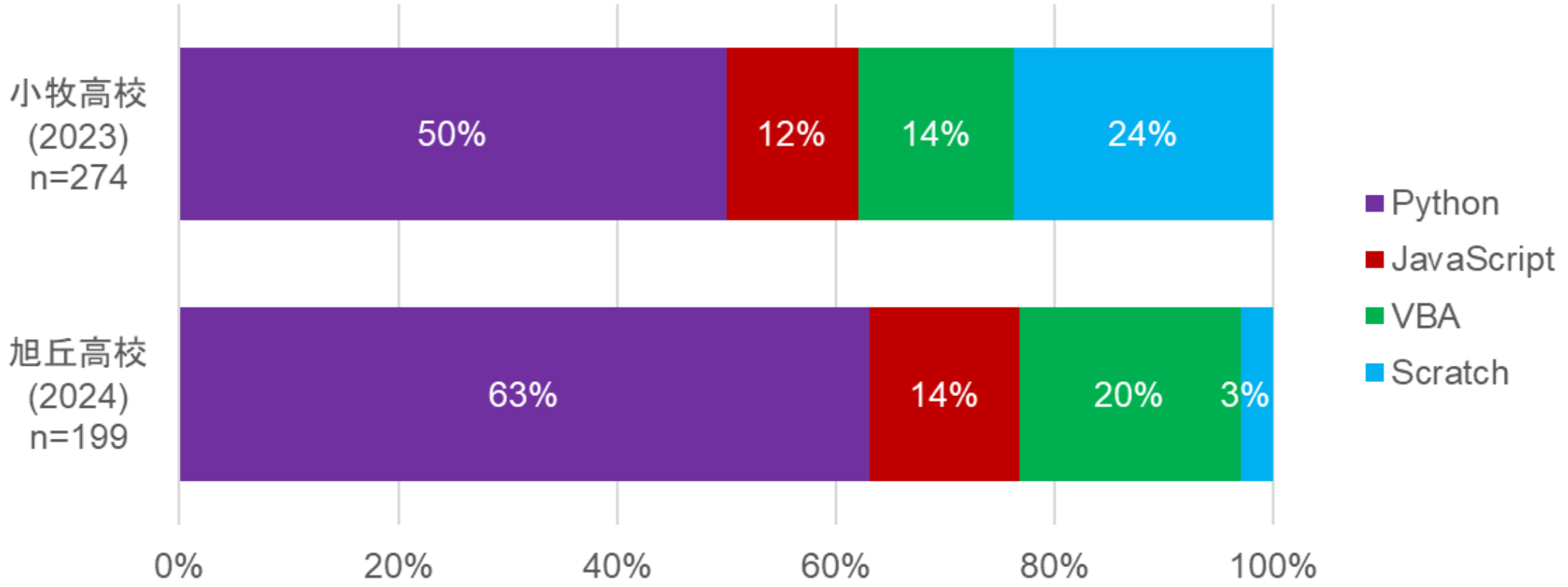
実施時期

２０２４年９月～２０２５年１月（**１単位**）

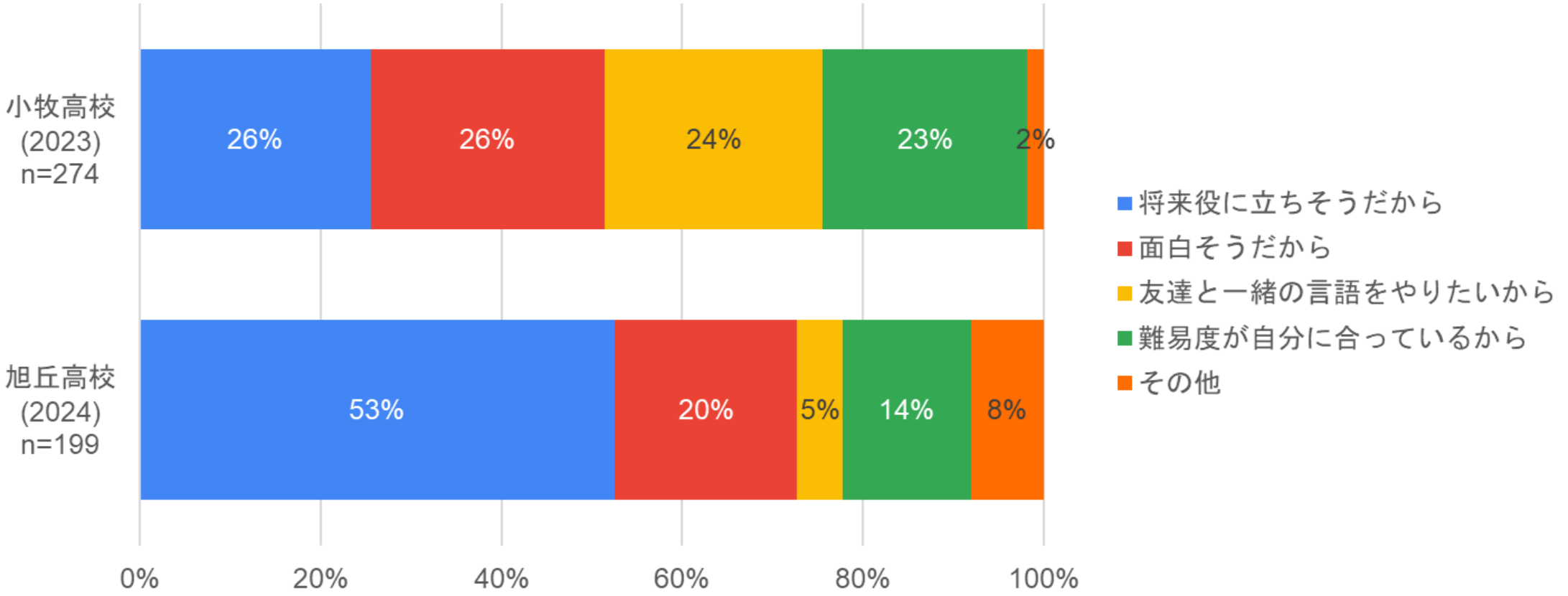
プログラミング言語

「**Python，JavaScript，VBA，Scratch**，その他」から各自で選択する

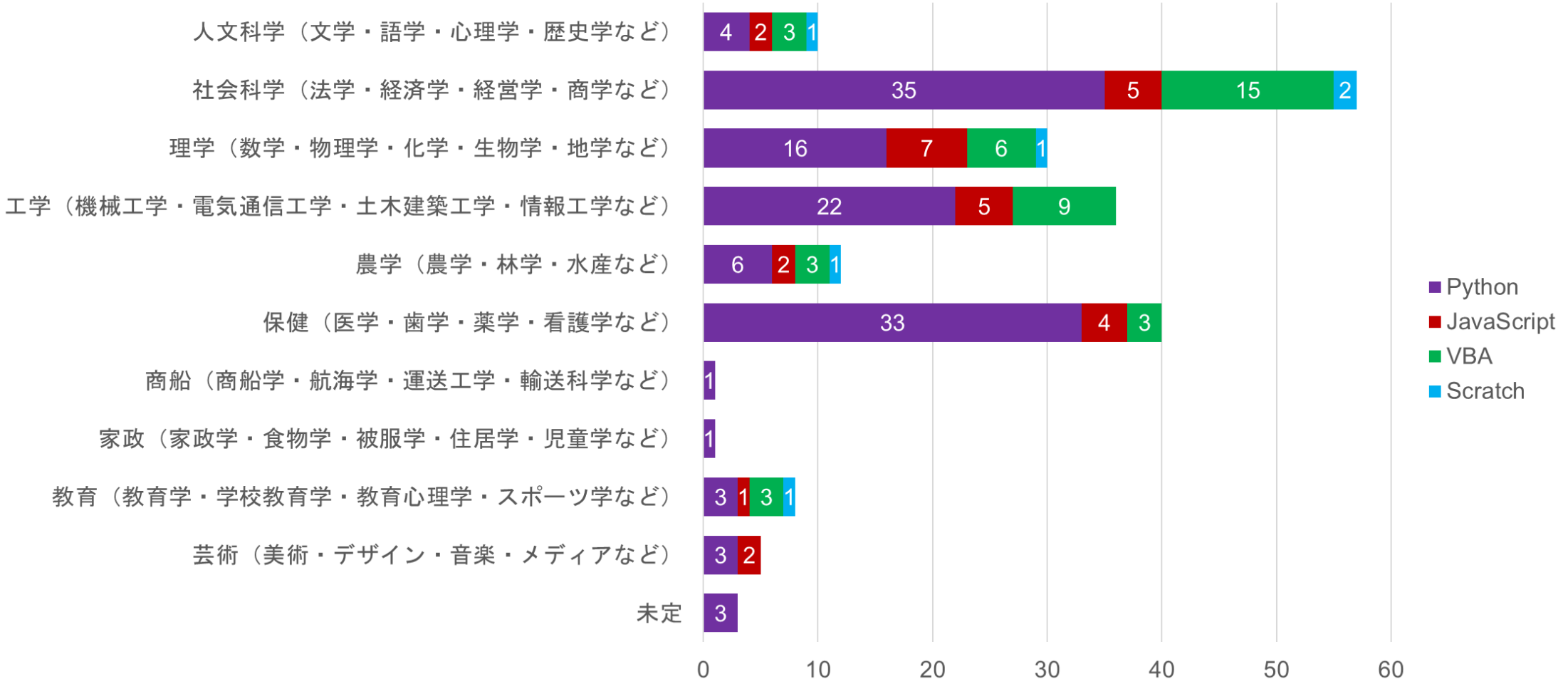
生徒が選択したプログラミング言語の内訳



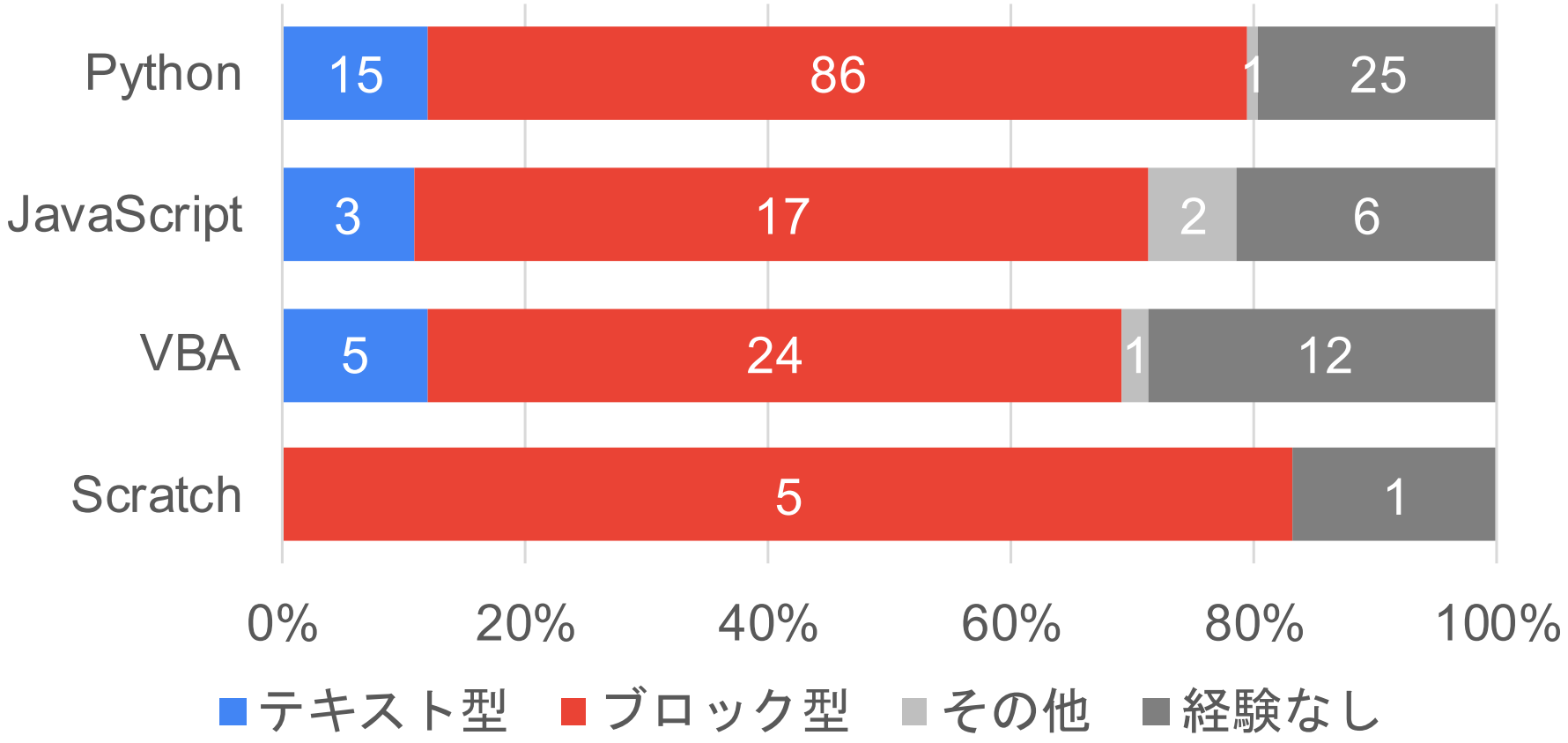
プログラミング言語を選択した主な理由



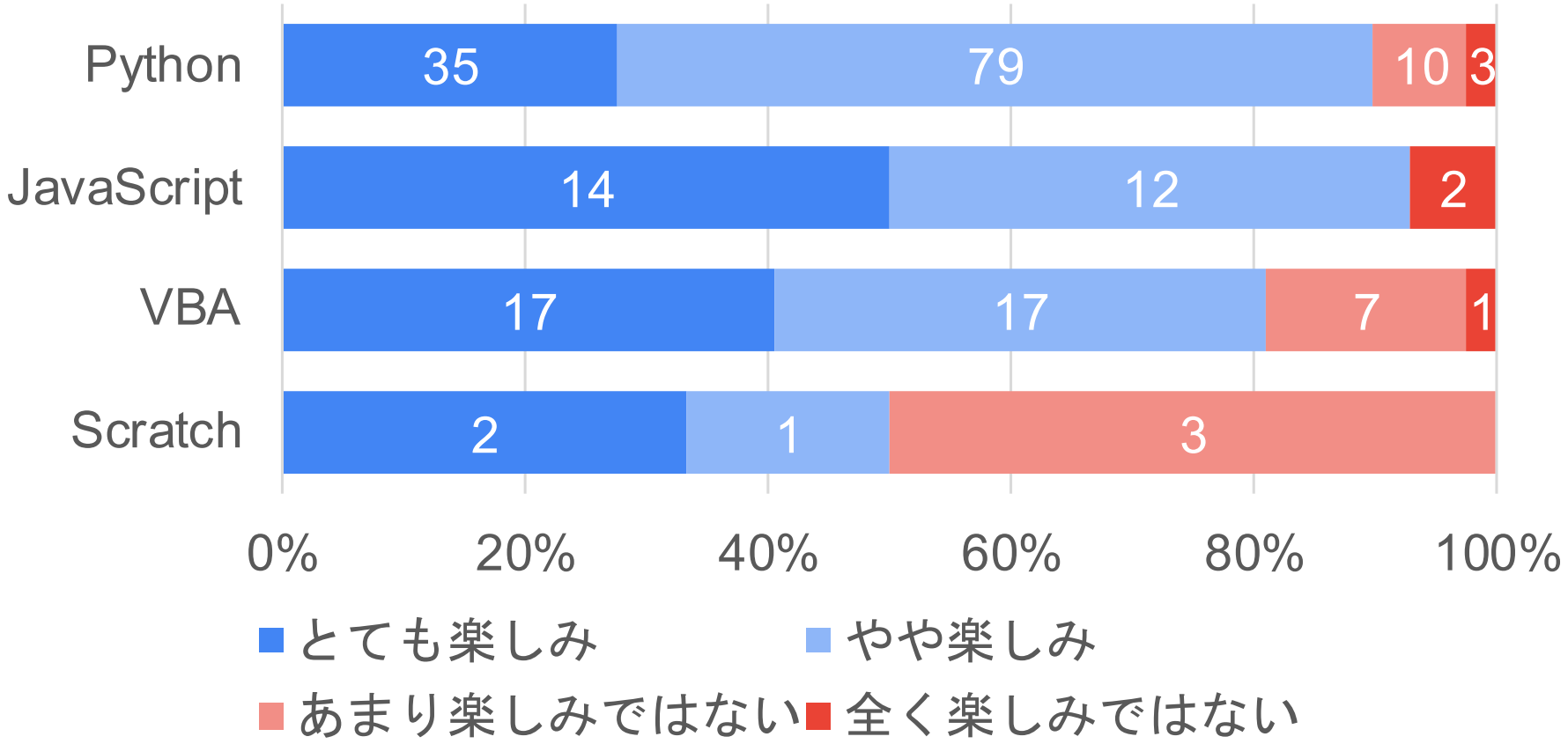
プログラミング言語と進路希望



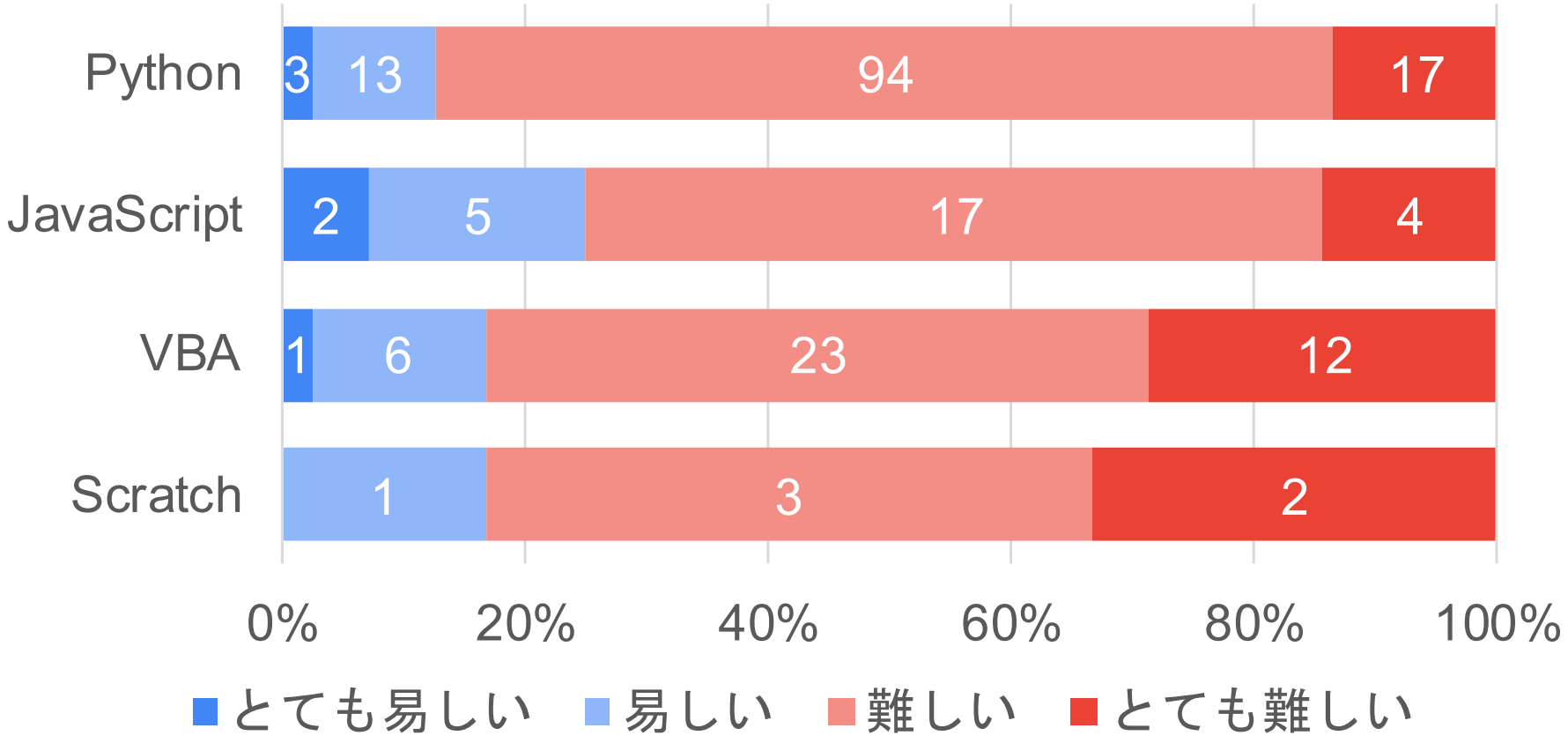
中学校でのプログラミング経験



プログラミングに対する楽しさ



プログラミングに対する難しさ



プログラミングの授業内容

【前半】プログラミングの基礎	
	メッセージ（画面出力）
	変数
	条件分岐（if文）
	繰り返し（for文）
	繰り返し（while文）
	配列／リスト（基礎）
	配列／リスト（応用）
	関数

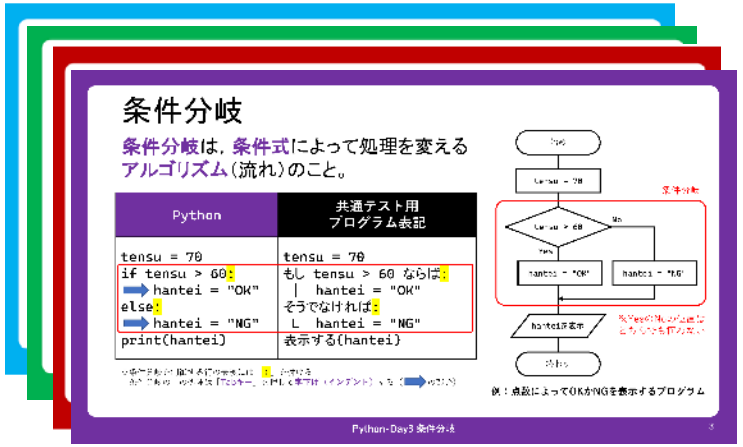
【後半】プログラム制作と発表会	
	プログラム制作①
	プログラム制作②
	プログラム制作③
	**中間レビュー会**
	プログラムの評価・改善
	**プログラム発表会**

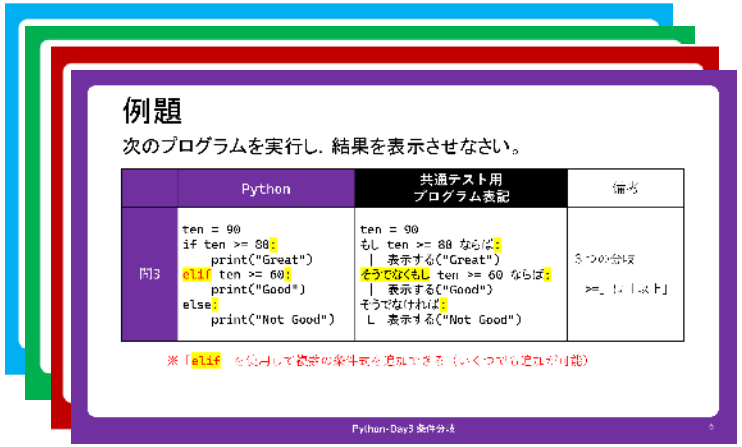
【前半】の基本的な授業の流れ

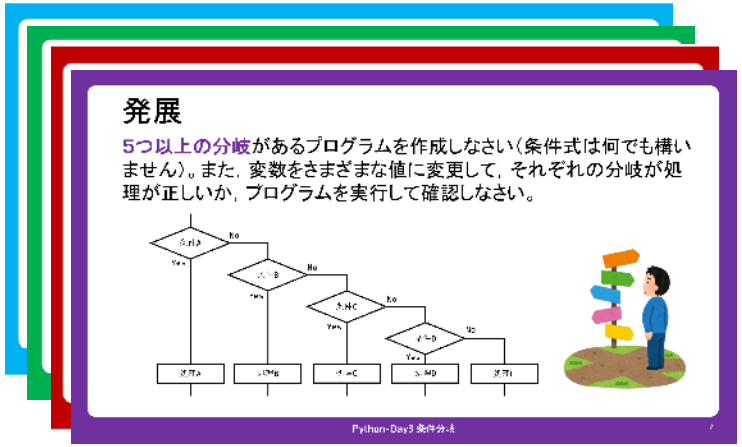
Step 一 **全体説明**（約10分）

Step 二 **例題演習**（約20分）

Step 三 **発展課題**（約20分）

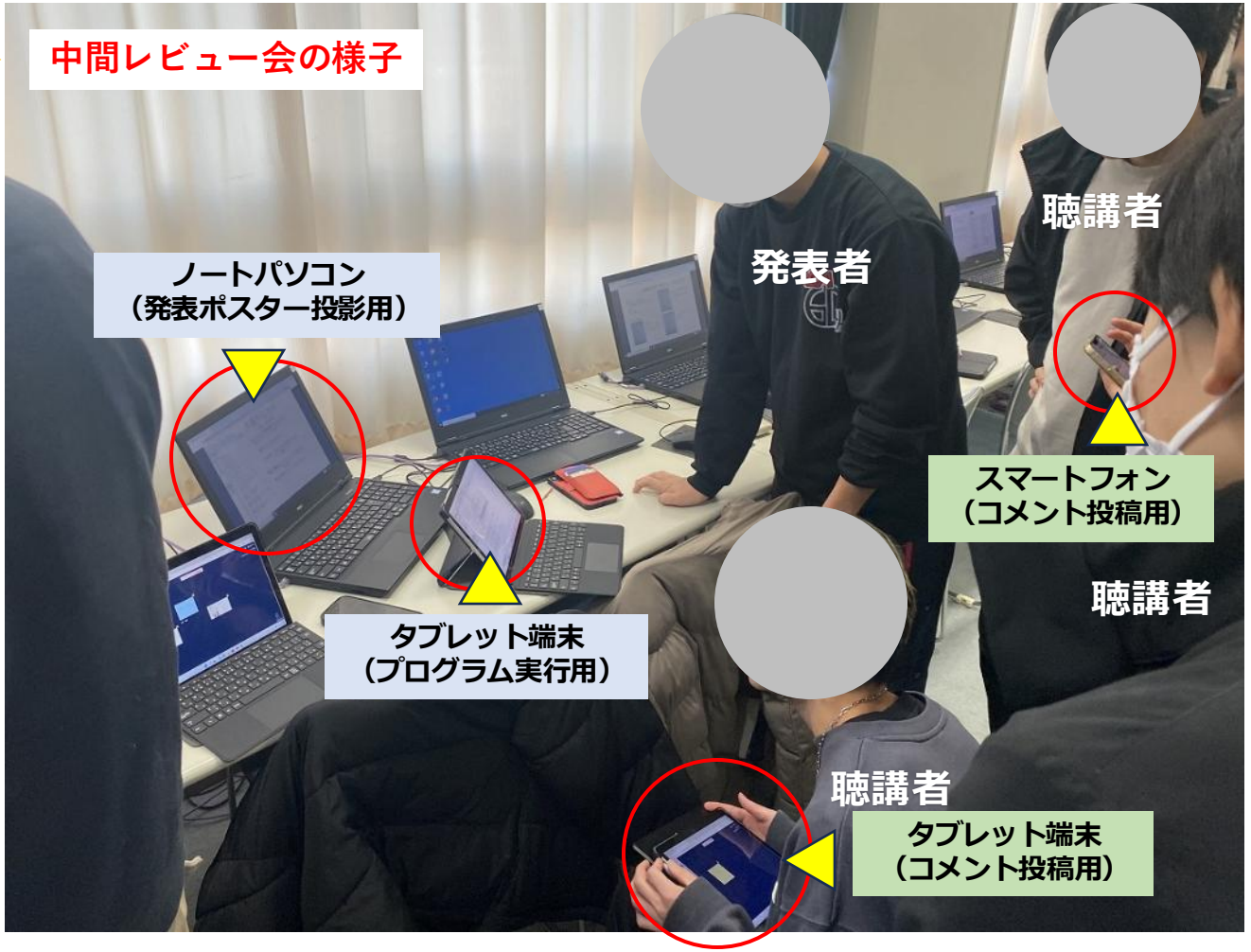


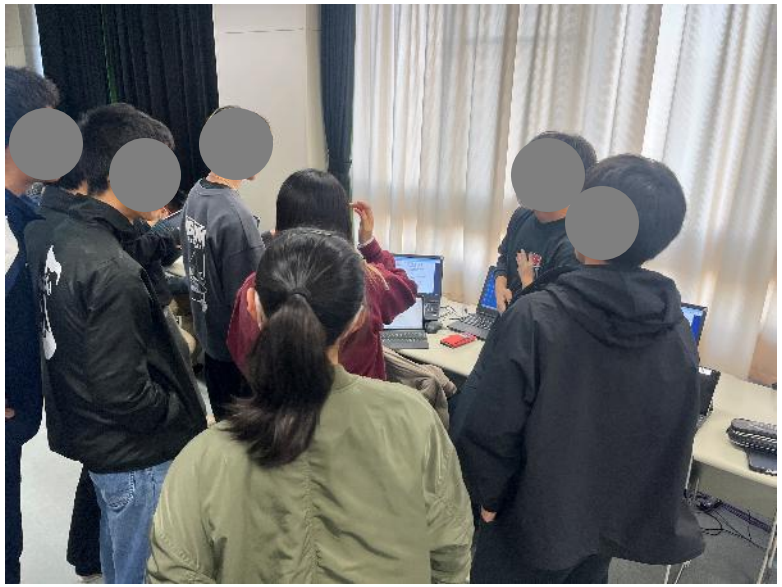


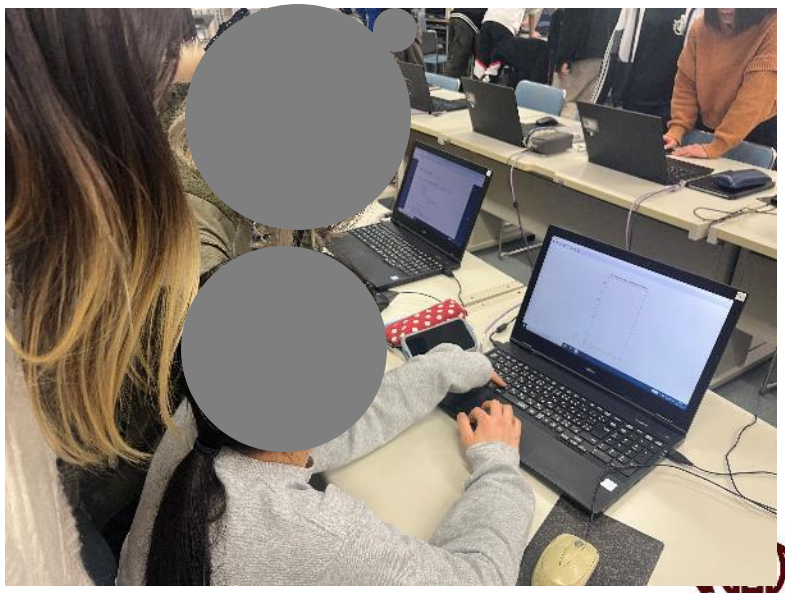


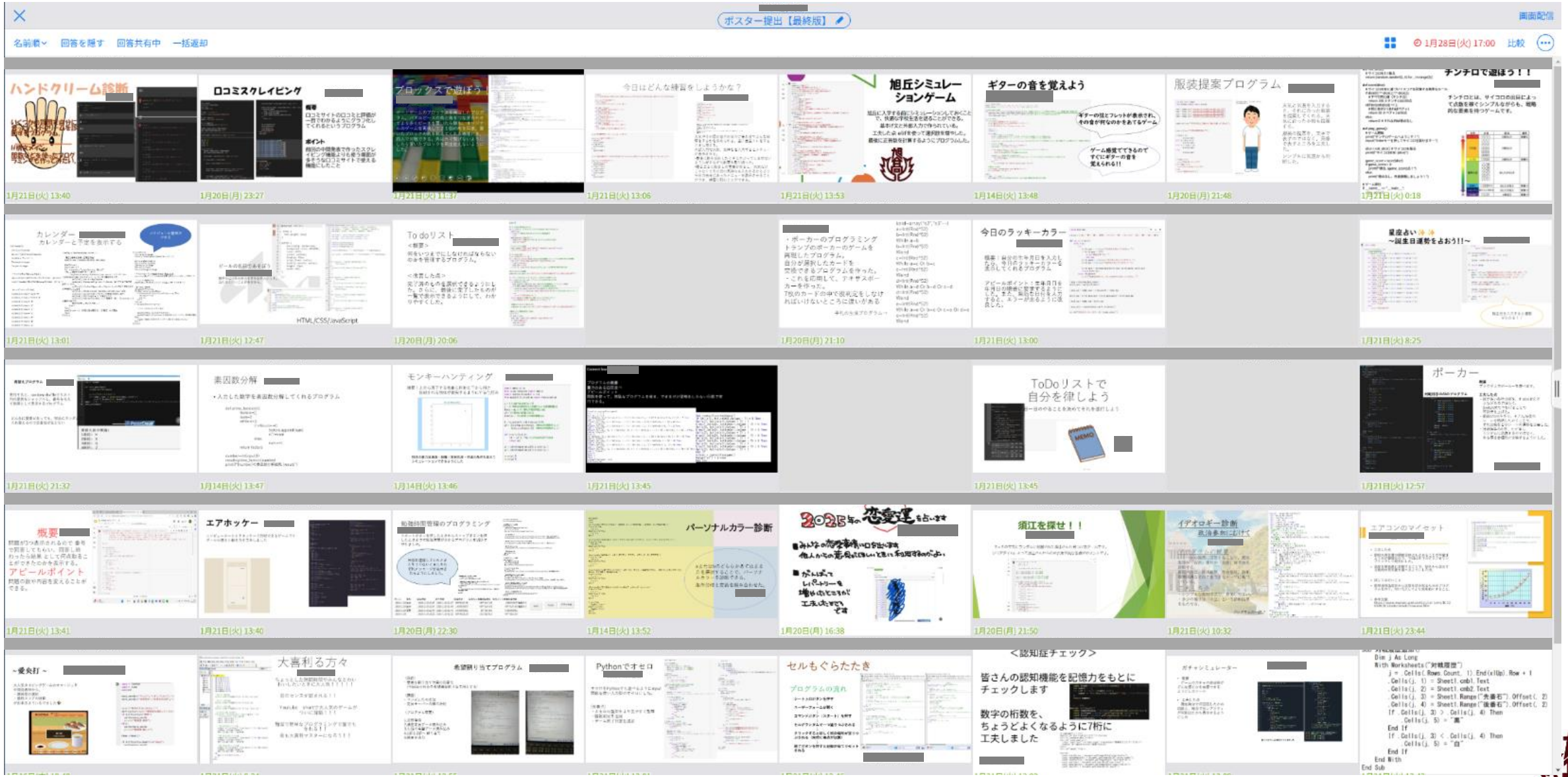
中間レビュー会の様子

授業回		授業内容
1	前半・8回	メッセージ
2		変数
3		条件分岐
4		繰り返し（for文）
5		繰り返し（while文）
6		配列／リスト（基礎）
7		配列／リスト（応用）
8		関数
9	後半・6回	プログラム制作①
10		プログラム制作②
11		プログラム制作③
12		中間レビュー会
13		プログラムの評価・改善
14		プログラム発表会








ポスター提出【最終版】
画面配信
名前順
回答を隠す
回答共有中
一括返却
1月28日(火) 17:00
比較
ハンドクリーム診断
口コミスクレイピング
ブロックで遊ぼう
今日はどんな練習をしようかな？
旭丘シミュレーションゲーム
ギターの音を覚えよう
服装提案プログラム
チンチロで遊ぼう！！
カレンダー
To doリスト
HTML/CSS/JavaScript
今日のラッキーカラー
星座占い
素因数分解
モンキーハンティング
ToDoリストで自分を律しよう
ポーカー
概要
アピールポイント
エアホッケー
パーソナルカラー診断
2025年の恋愛運を占います
須江を探せ！！
イデオロギー診断
エアコンのマイセット
大喜利る方々
希望振り当てプログラム
Pythonでオセロ
セルもぐらたたき
<認知症チェック>
皆さんの認知機能を記憶力をもとにチェックします
数字の桁数を、ちょうどよくなるように7桁に工夫しました
ガチャシミュレーター

早く家に帰りたい！

家に帰るためのバスが3駅から出ているので
そのうちどのバスが待ち時間が少ないのか
最適なバスを表示してくれるプログラム
きっと需要は私にしかない

15:55
鳴子北16:42
相生山17:09
神　沢16:29
神沢16:29が最速！

←こんな感じ

急　行

苦労点(聞いてほしい)：音源が見つからず、挙句の果てに名古屋駅の放送
音源を高蔵寺・神宮前・金山・西春・須ヶ口・知立駅放送から錬成しました。

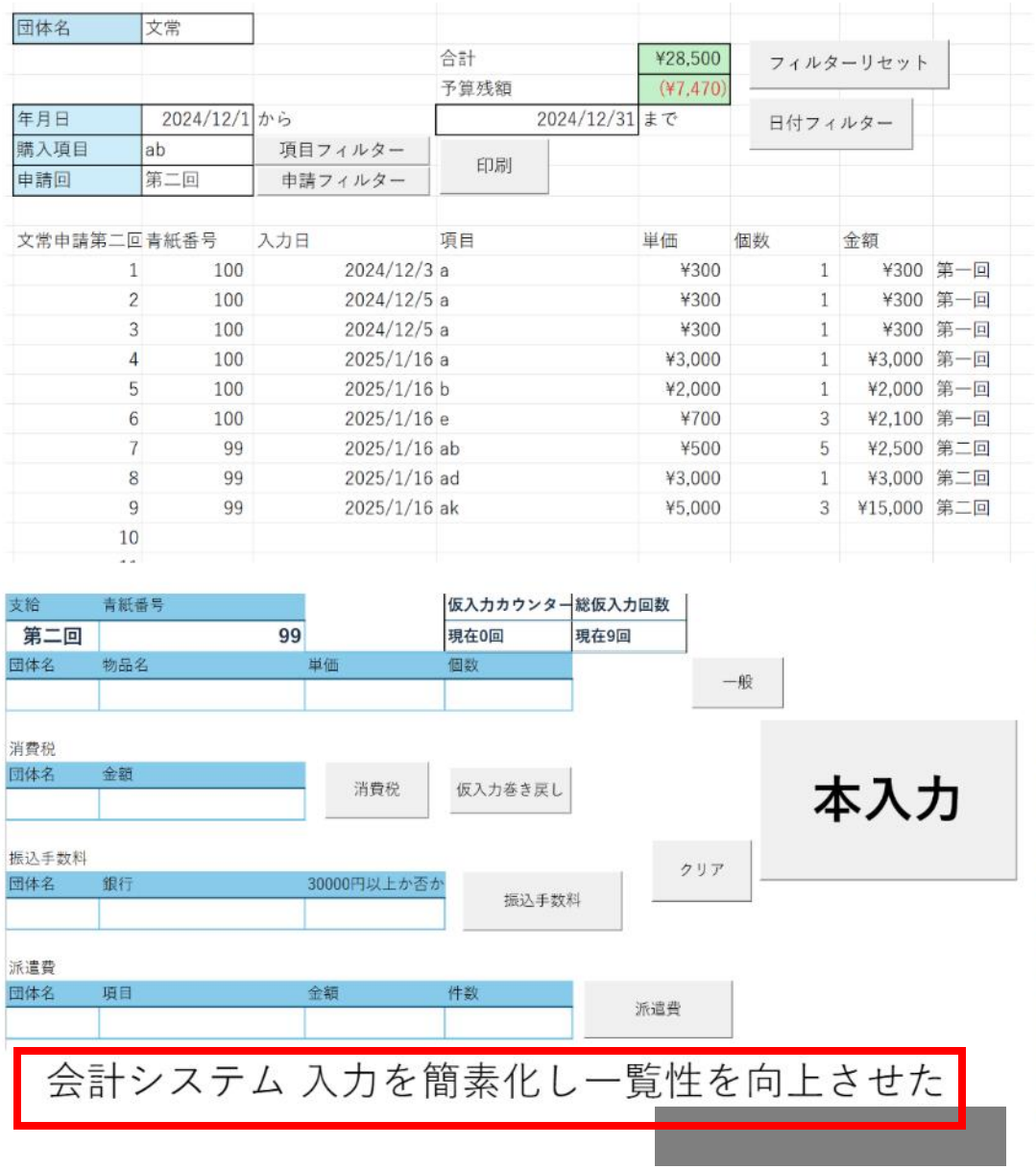
まもなく シン・鉄道発車標総合表示システム(ME-R7-3)

1 号車

JR民の方、長らくお待たせしました
祝　JRモードサービス提供開始！！

```
Sub ShapeCopy6()

    Dim shape As shape
    Dim area1 As Range
    Dim area2 As Range
    Dim area3 As Range

    Worksheets("メイン").Activate

    Set area1 = Range("B7:V20")

    For Each shape In ActiveSheet.Shapes
        If Not Intersect(shape.TopLeftCell, area1) Is Nothing Then
            shape.Delete
        End If
    Next shape
(略)
Worksheets("メイン").Select
    For Each myShape3 In Worksheets("コピー先 (2)").Shapes
        myShape3.Copy
        Worksheets("メイン").Paste
        Selection.Top = myShape3.Top
        Selection.Left = myShape3.Left
    Next

    Worksheets("メイン").Activate
    Dim myShape4 As Variant
    Worksheets("メイン").Select
    For Each myShape4 In Worksheets("コピー先 (4)").Shapes
        myShape4.Copy
        Worksheets("メイン").Paste
        Selection.Top = myShape4.Top
        Selection.Left = myShape4.Left
    Next

End Sub
```

仕組みの概要(名鉄→JRにするとき)
①モニターの中にある名鉄用の図形をすべて選択し消去する。
②右のコードで他のシートに張り付けてあるJR用の図形をすべて選択し、指定の位置に貼り付ける。



会計システム 入力を簡素化し一覧性を向上させた

```
End If
GoTo Theend
Error:
MsgBox "フィルター項目を入力してください"
Theend:

End Sub
Sub 申請フィルター()

If Sheets("検索システム").Range("C7").Value = 0 Then
GoTo Error
Else
Dim a
a = Sheets("検索システム").Range("C7")
Dim y
'Debug.Print Application.WorksheetFunction.CountIf(Sheets("保存先").Range("D:D"), "*")
For y = 10 To Application.WorksheetFunction.CountIf(Sheets("保存先").Range("D:D"), "*") + 9
```

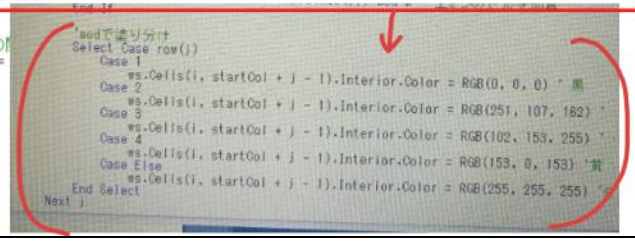
```
If Sheets("入力").Range("B1").Value = 0 Then
MsgBox "仮入力を行ってください"
Else
I = Sheets("入力").Range("C1")
Dim c
For c = N - M + I + 2 To N + I + 1
Sheets("保存先").Cells(c, 1).Value = Sheets("メモリ").Cells(c - I - 1, 1)
Sheets("保存先").Cells(c, 2).Value = I + 1
Sheets("保存先").Cells(c, 4).Value = Sheets("メモリ").Cells(c - I - 1, 2)
Sheets("保存先").Cells(c, 5).Value = Sheets("メモリ").Cells(c - I - 1, 3)
Sheets("保存先").Cells(c, 6).Value = Sheets("メモリ").Cells(c - I - 1, 7)
Sheets("保存先").Cells(c, 7).Value = Sheets("メモリ").Cells(c - I - 1, 4)
Sheets("保存先").Cells(c, 8).Value = Sheets("メモリ").Cells(c - I - 1, 5)
Sheets("保存先").Cells(c, 9).Value = Sheets("保存先").Cells(c, 7) * Sheets("保存先").Cells(c, 8)
Sheets("保存先").Cells(c, 10).Value = Date
Sheets("保存先").Cells(c, 10).Font.Color = RGB(255, 255, 255)
Sheets("保存先").Cells(c, 11).Value = Sheets("入力").Range("A3").Value
Sheets("保存先").Cells(c, 11).Font.Color = RGB(255, 255, 255)
Sheets("保存先").Cells(c, 3).Value = Application.WorksheetFunction.CountIf(Range(Sheets("保存先").Cells(2, 4), Sheets("保存先").Cells(c, 4)), Sheets("保存先").Cells(c, 4)) & Sheets("保存先").Cells(c, 4)
Sheets("保存先").Cells(c, 3).Font.Color = RGB(255, 255, 255)
Application.StatusBar = c & "/" & M + I + 1
Application.StatusBar = False
Next c
Sheets("保存先").Range(Sheets("保存先").Cells(N + I + 2, 1), Sheets("保存先").Cells(N + I + 2, 9)).Interior.Color = RGB(169, 169, 255)
Sheets("保存先").Cells(N + I + 2, 1).Value = Sheets("入力").Range("A3").Value
If Sheets("保存先").Cells(N + I + 1, 1).Value = "" Then
Sheets("保存先").Cells(N + I + 2, 2).Value = Sheets("保存先").Cells(N + I + 1, 4)
Else
Sheets("保存先").Cells(N + I + 2, 2).Value = Sheets("保存先").Cells(N + I + 1, 1)
End If
Sheets("保存先").Cells(N + I + 2, 3).Value = "入力日"
Sheets("保存先").Cells(N + I + 2, 4).Value = Date
Sheets("保存先").Cells(N + I + 2, 5).Value = "合計金額"
Sheets("保存先").Cells(N + I + 2, 6).Value = Application.WorksheetFunction.Sum(Sheets("保存先").Range(Sheets("保存先").Cells(N - M + I + 2, 9), Sheets("保存先").Cells(N + I + 1, 9)))
Sheets("保存先").Cells(N + I + 2, 7).Value = "予算残額"
Sheets("保存先").Cells(N + I + 2, 8).Value = Application.WorksheetFunction.VLookup(Sheets("保存先").Cells(N + I + 2, 2), Sheets("予算表").Range("A1:B300"), 2, False) - Application.WorksheetFunction.
If Sheets("保存先").Cells(N + I + 2, 8).Value < 0 Then
Sheets("保存先").Cells(N + I + 2, 8).Interior.Color = RGB(255, 0, 0)
Else
```

パスカルの三角形の合同式での塗り分けによるフラクタル図形の作成(変更版)

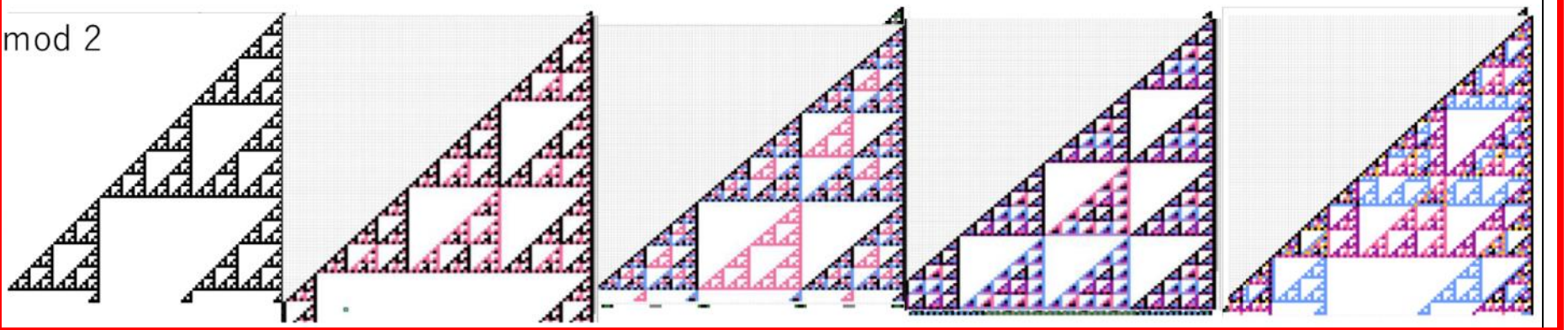
中間発表のときから色を変更

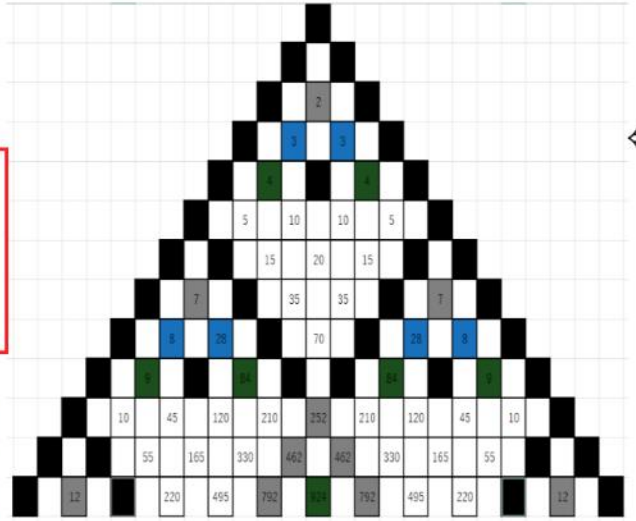
mod 5の場合

```
size = 100
p = 5 ' mod○

' シートをクリア
Dim ws As Worksheet
Set ws = ThisWorkbook.Sheets(4)
ws.Cells.Clear

' 配列を初期化
ReDim row(1 To size)
row(1) = 1 ' 最初の行の最初の値を1に設定

' 初期の開始列を設定（中央揃え）
startCol = size

' パスカルの三角形
For i = 1 To size
    Dim prevRow() As Integer
    prevRow = row ' 現在の行を前の行として保持

    ' 現在の行を計算
    For j = 1 To i
        If j = 1 Or j = i Then
            row(j) = 1 ' 両端は1
        Else
            row(j) = (prevRow(j - 1) + prevRow(j)) Mod p ' 上2つのセルを加算
        End If

        'modで塗り分け
        Select Case row(j)
            Case 1
                ws.Cells(i, startCol + j - 1).Interior.Color = RGB(0, 0, 0) ' 黒
            Case 2
                ws.Cells(i, startCol + j - 1).Interior.Color = RGB(128, 128, 128) ' 灰
            Case 3
                ws.Cells(i, startCol + j - 1).Interior.Color = RGB(25, 107, 36)
            Case 4
                ws.Cells(i, startCol + j - 1).Interior.Color = RGB(77, 147, 217)
            Case Else
                ws.Cells(i, startCol + j - 1).Interior.Color = RGB(255, 255, 255) '白
        End Select
    Next j

    ' 次の行の
    startCol =
Next i
End Sub
```



mod 2





【合同式】

$7 \equiv 4 \pmod 3$

⇔ 7 と 4 は 3 で割ったときの余りが等しい

【手順】

①パスカルの三角形を描く

②ｘで割ったとき(mod ｘ)の余りごとに色分け

←手打ちで描いたもの

発表　最終版
「素数の分布の視覚化」

＜概要＞
エラトステネスのふるいの考え方を用いて効率的に素数の配列を作り、その要素と要素が何番目かをすべて表示する。座標平面は縦軸は素数の大きさ、横軸は何番目の素数かを表す。

＜プログラムの要約＞
hanni=〇〇まで調べるとき、
(これを自由に変えてグラフの表示範囲を変えられる)
配列primesを定義し、最初の素数2を含めておく。
3から調べるhanniまでの素数を配列primesに含めていく。
hantei=1を代入して、√iまでのprimesの要素で割っていく。
3から調べるhanniまでの素数を配列primesに含めていく。
素数でないと確定すればその時点でbreakをし、次のiを調べる。
以上の操作により、配列を用意したら後はx番目の素数をyとして、プロット
＜グラフを修飾・表示するプログラム＞

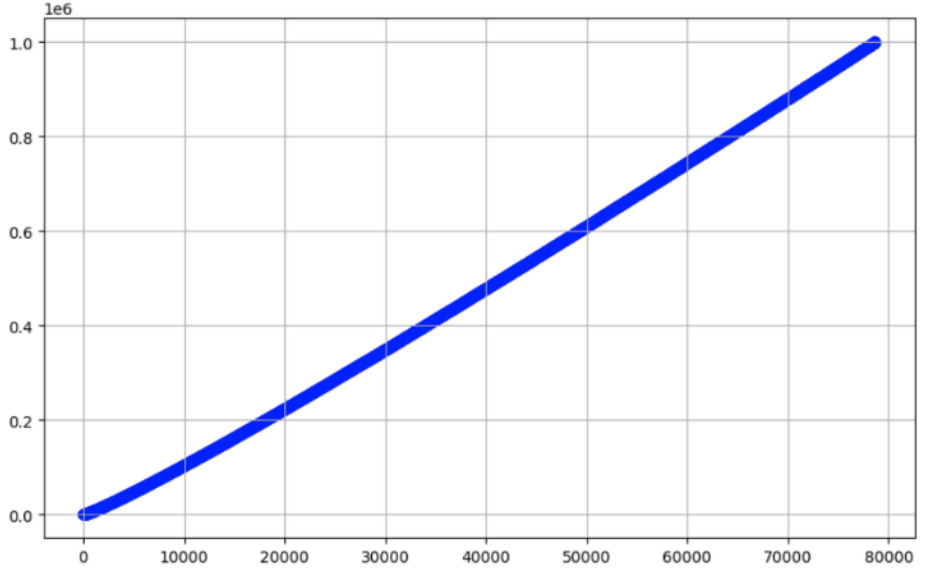
＜工夫＞
グーグルコラボを使ってpythonの機能を拡張して座標平面を使えるようにし、プログラムを組んだ。
素数判定のプログラムでは√nまでの自然数で順に割るようにし、無駄な計算を極限まで省くことで処理を軽くした。
エラトステネスのふるいの手法を繰り返しで順次利用していき、配列に素数を追加する。中間発表のプログラムに比べて大きく効率が向上している。

＜改善点＞
hanniを1000万にすると100万にくらべられないほどの時間がかかる。
(1分54秒)処理時間が早まったとはいえ、大きな数になるとその影響はわからないほど小さくなってしまう。Hanniの値によってふるいの大きさを最適化するプログラムが最も早いと考えられる。(セグメントふるい)

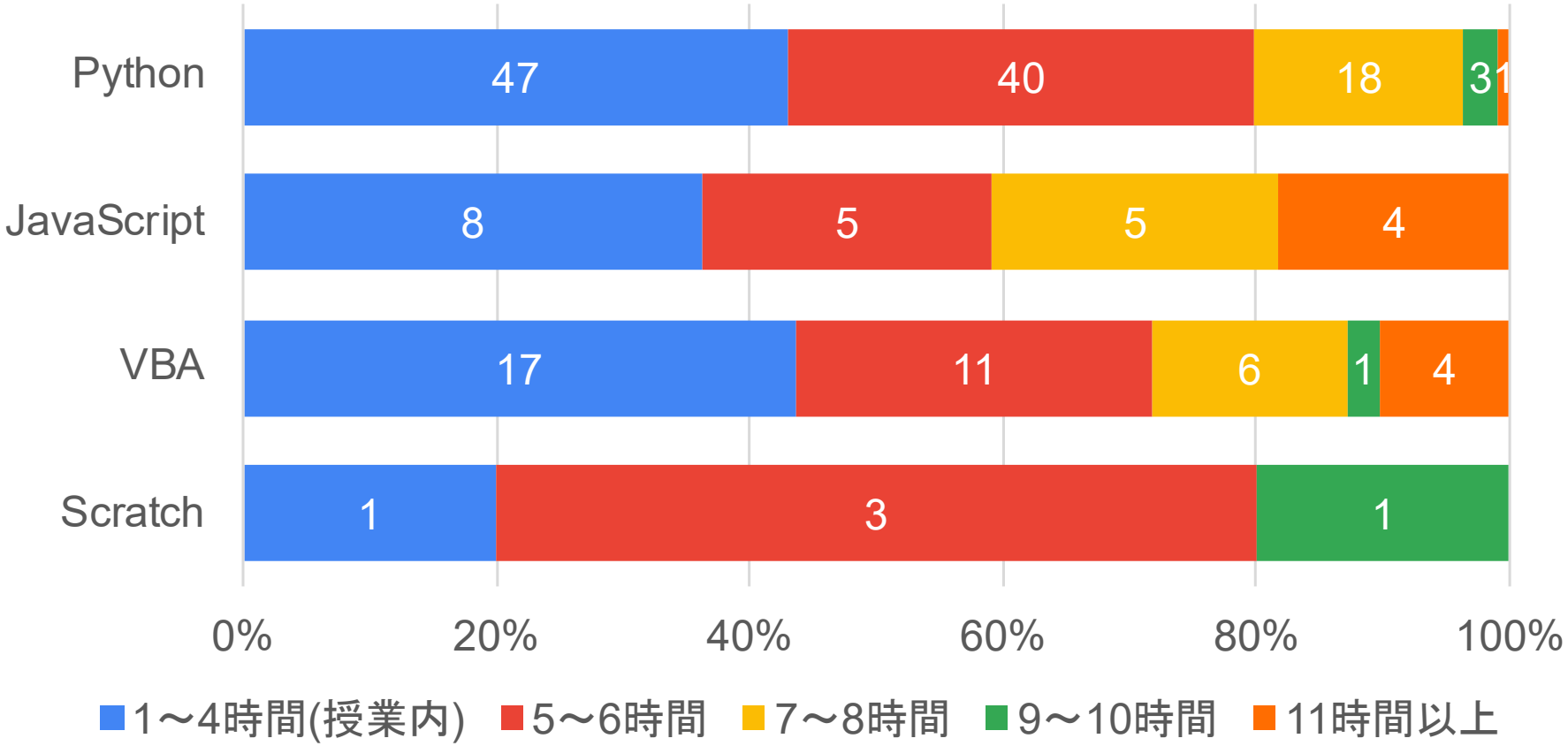
＜プログラム＞（グーグルコラボのpython）

```
import matplotlib.pyplot as plt
hanni=10**6
primes=[2]
for i in range(3,hanni+1):
 hantei=1
for j in range(1,len(primes)):
  if i%(primes[j-1])==0:
    hantei=0
    break
  if primes[j-1]>=i**(1/2):
    break
~~~~~~~~~~~~~~~~~~~~~~~~~~
if hantei==1:
 primes.append(i)
x=[]
y=[]
for i in range(1,len(primes)+1):
y.append(primes[i-1])
x.append(i)plt.figure(figsize=(10,6))
plt.scatter(x,y,color='blue',label='prime')
plt.grid(True)
plt.show
```

＜実行結果＞
＊100万までの素数で調べると、改良前(中間発表参照)が13秒かかったのに対し、改良後(上に示したプログラム)は7秒で終了し、中間発表での改善点を克服できた。
以下はhanni=10**6=100万での出力である。

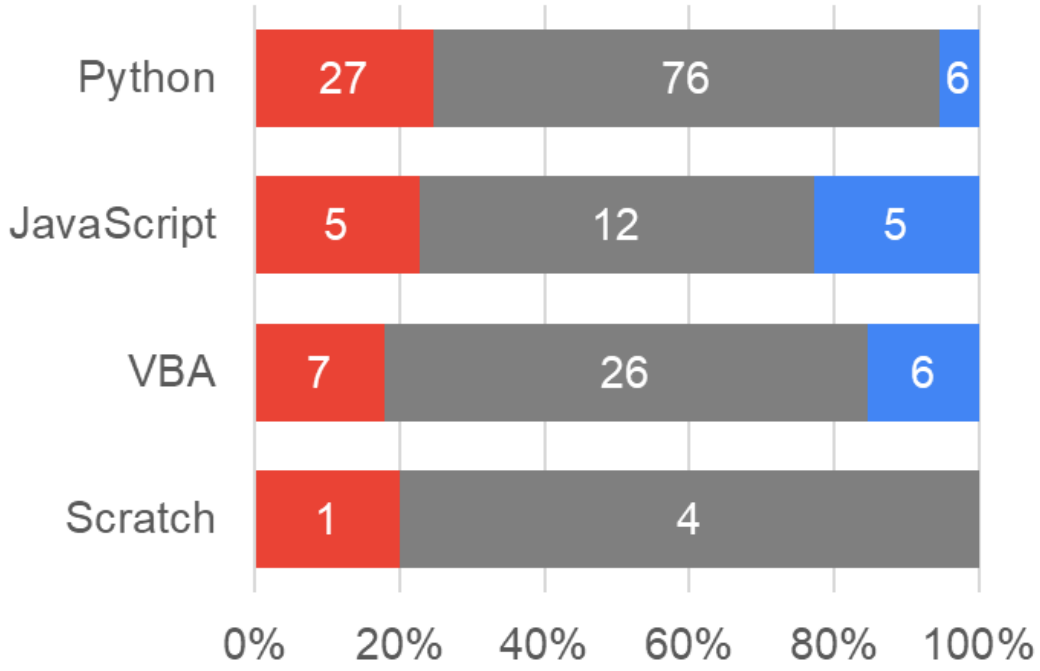


プログラム制作にかけた時間

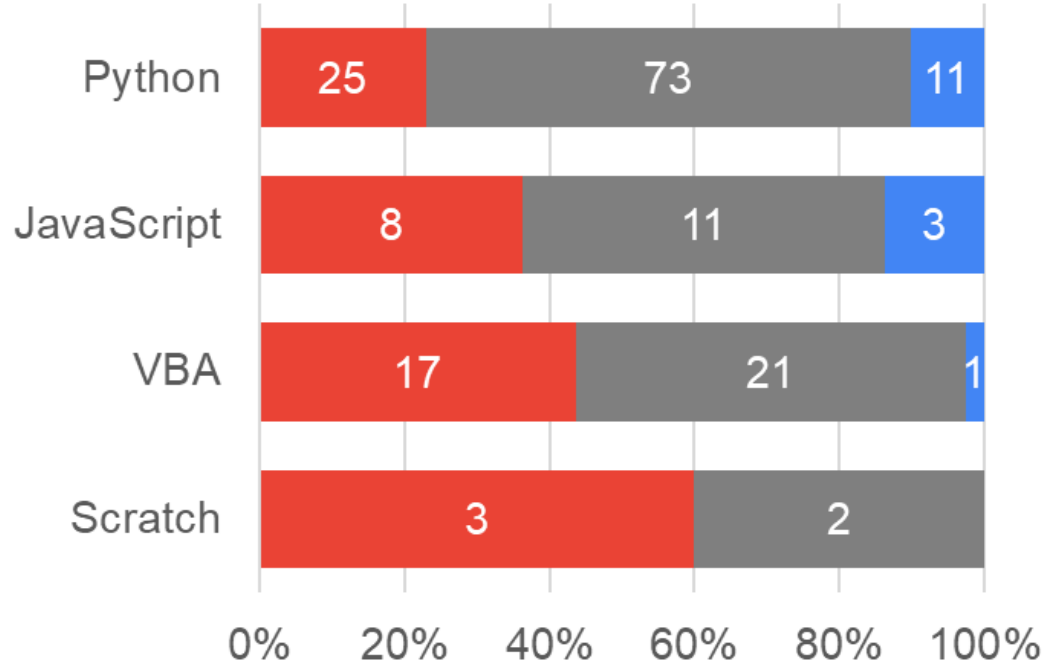


プログラミングの授業回数について

【前半】プログラミングの基礎（8回）



【後半】プログラム制作と発表会（6回）

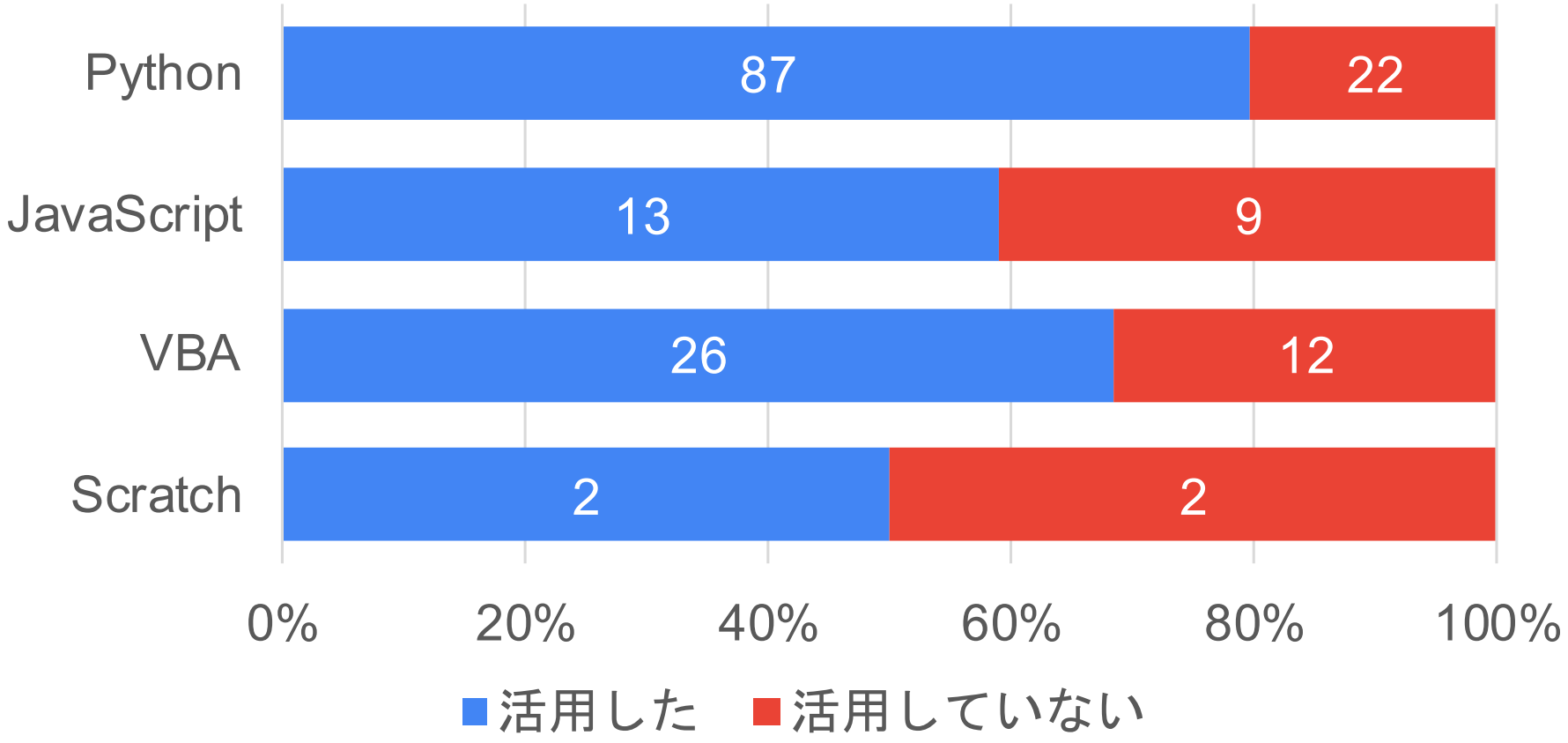


■もっと多い方がいい ■ちょうどいい ■もっと少ない方がいい

プログラミングにおける生成AIの利用

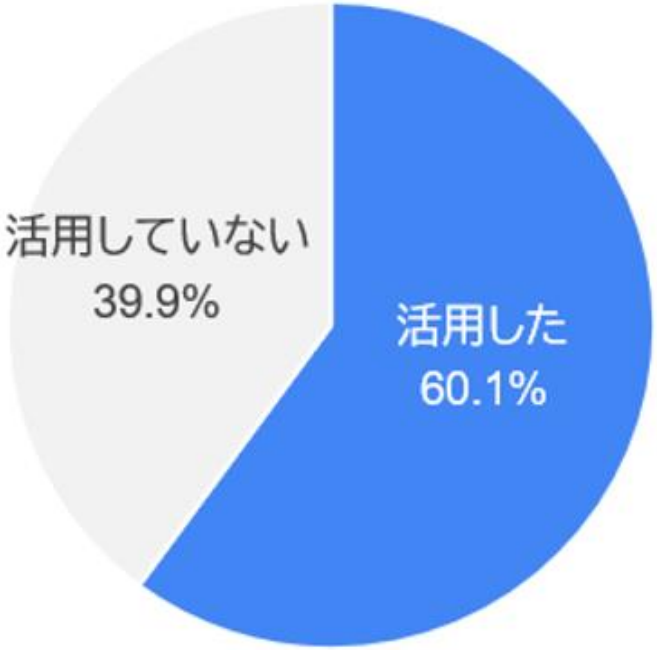
- 【後半】のはじめに生成AIの活用について説明
  -  **アイデアの提案**
  -  **プログラムの生成**
  -  **プログラムの修正（デバッグ）**
- 生成AIの使用は任意
- 生徒全員，保護者からの同意書を提出済み

プログラム制作で生成AIを活用したか

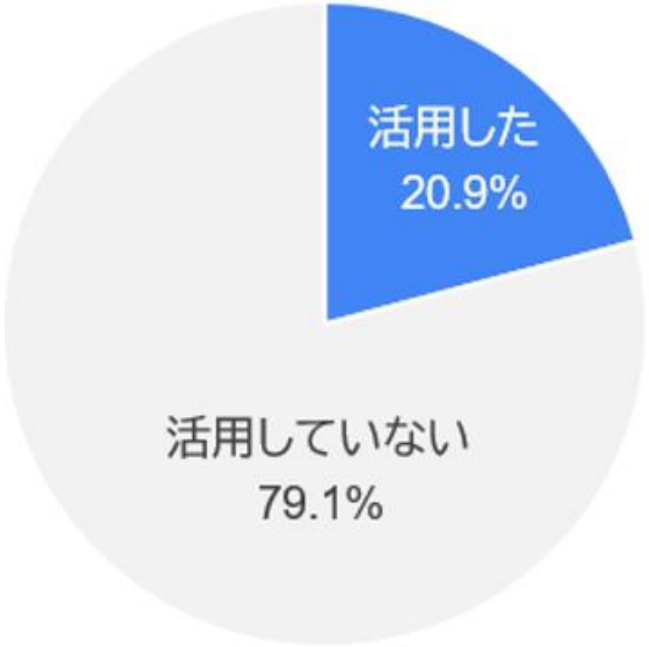


生成AI活用の活用場面（複数回答可）

(b) ①アイデアの提案

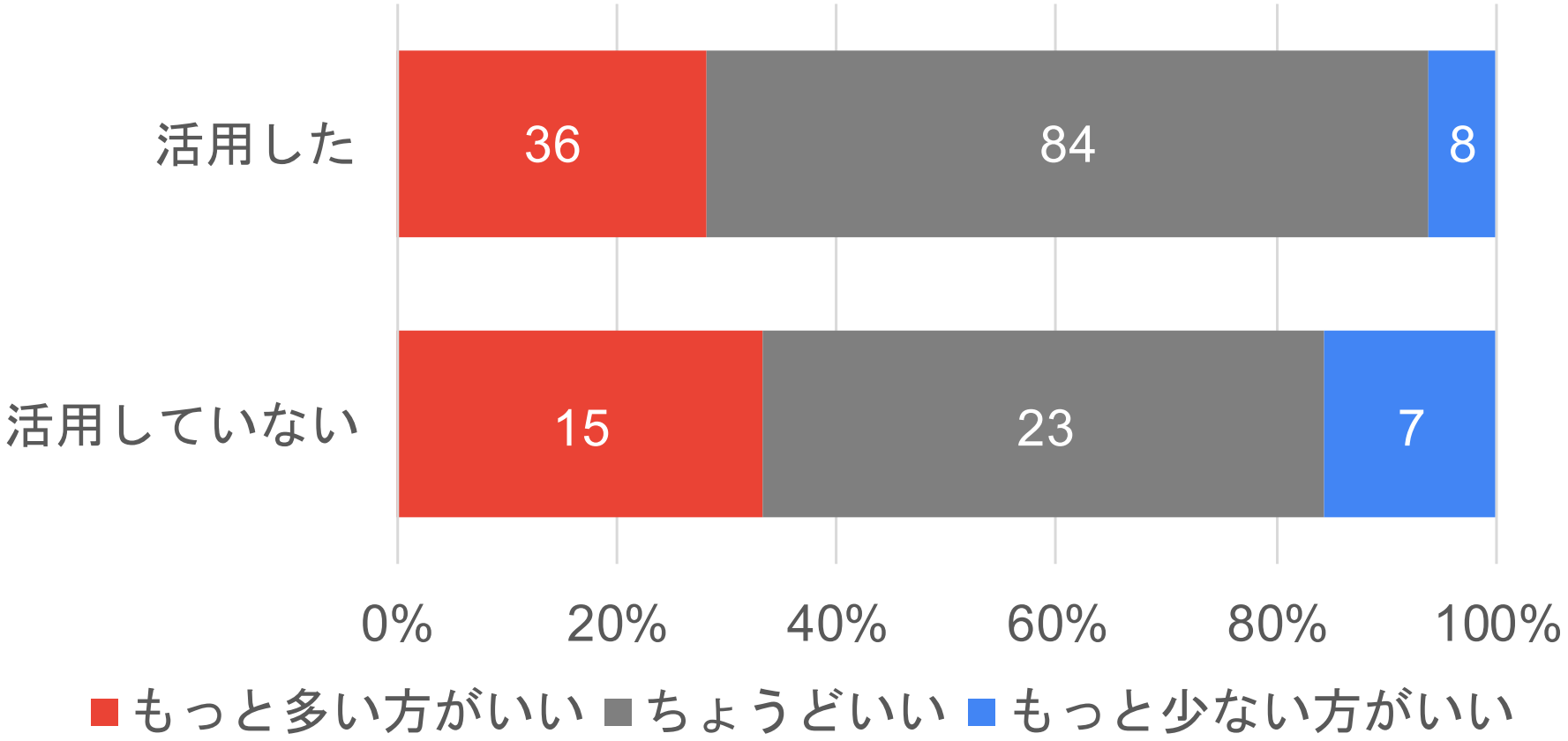


(c) ②プログラムの生成

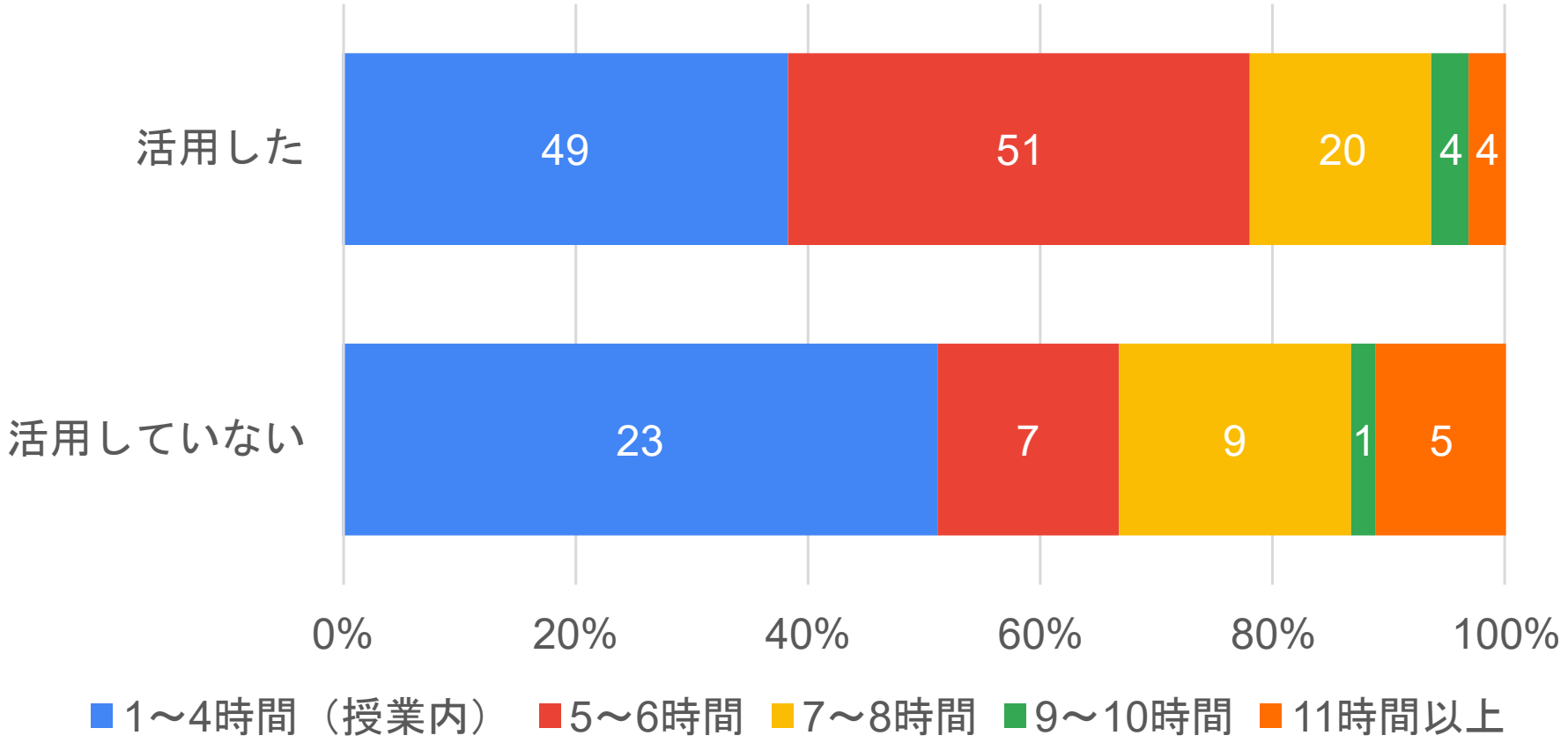


(d) ③プログラムの修正

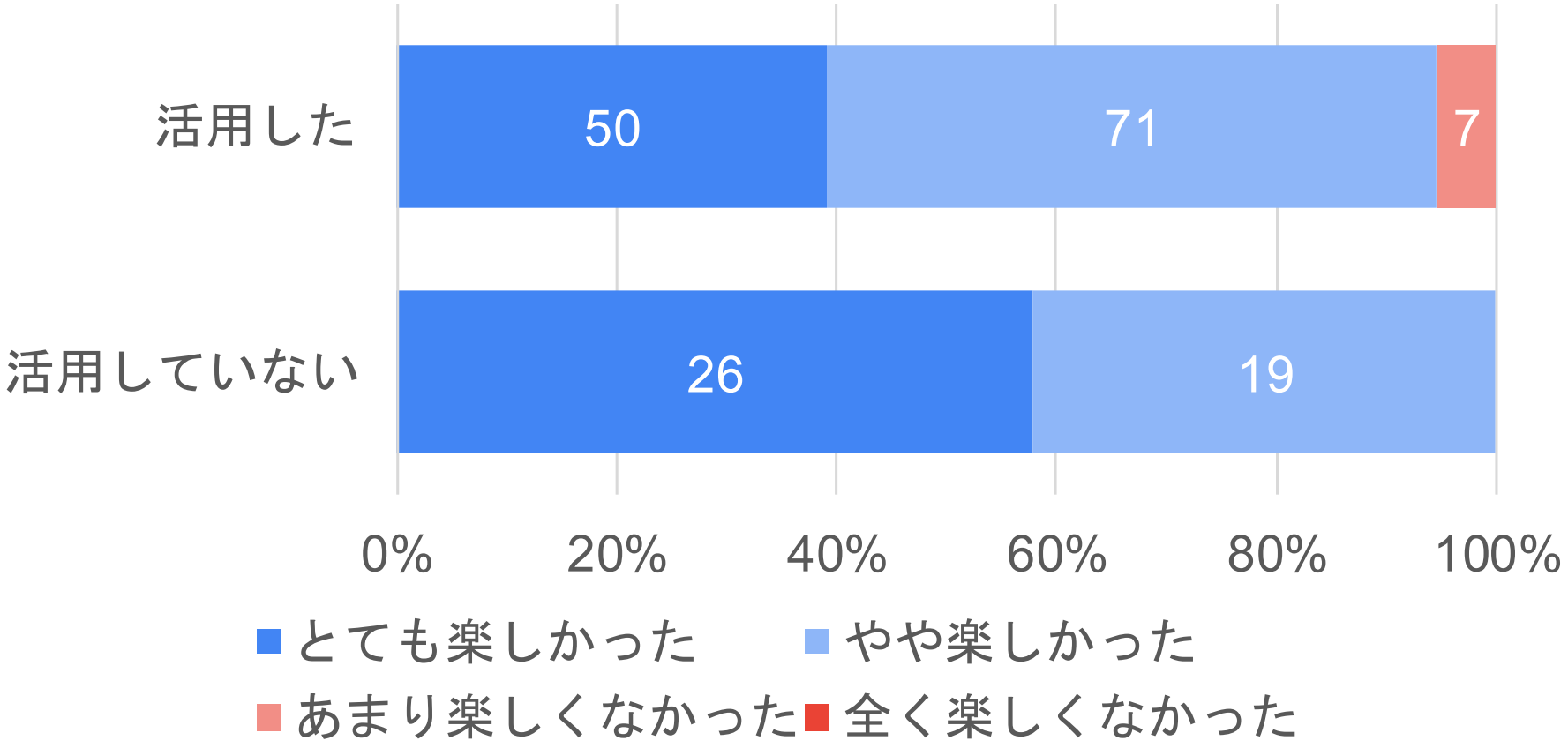
生成AIの活用×授業回数



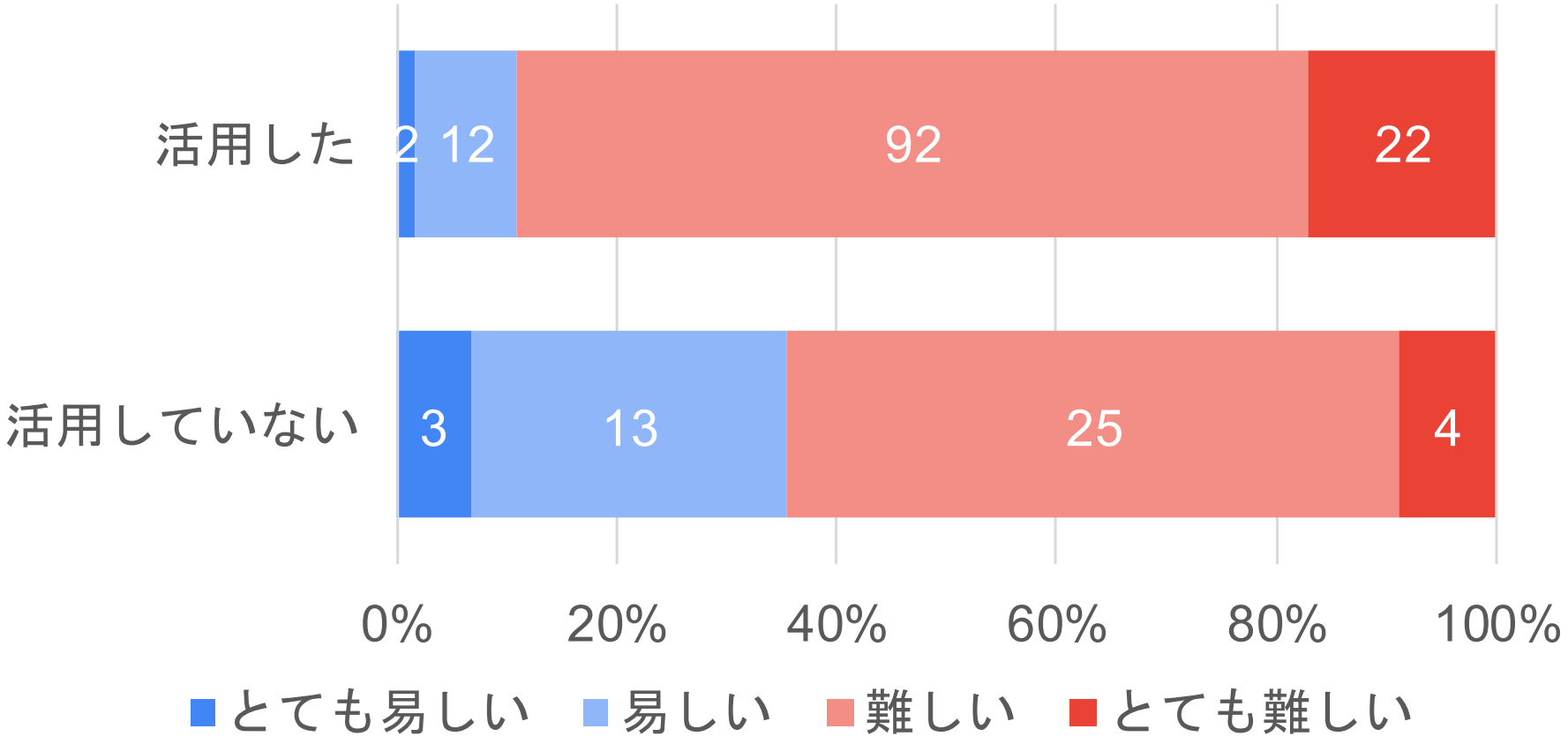
生成AIの活用×プログラム制作の時間



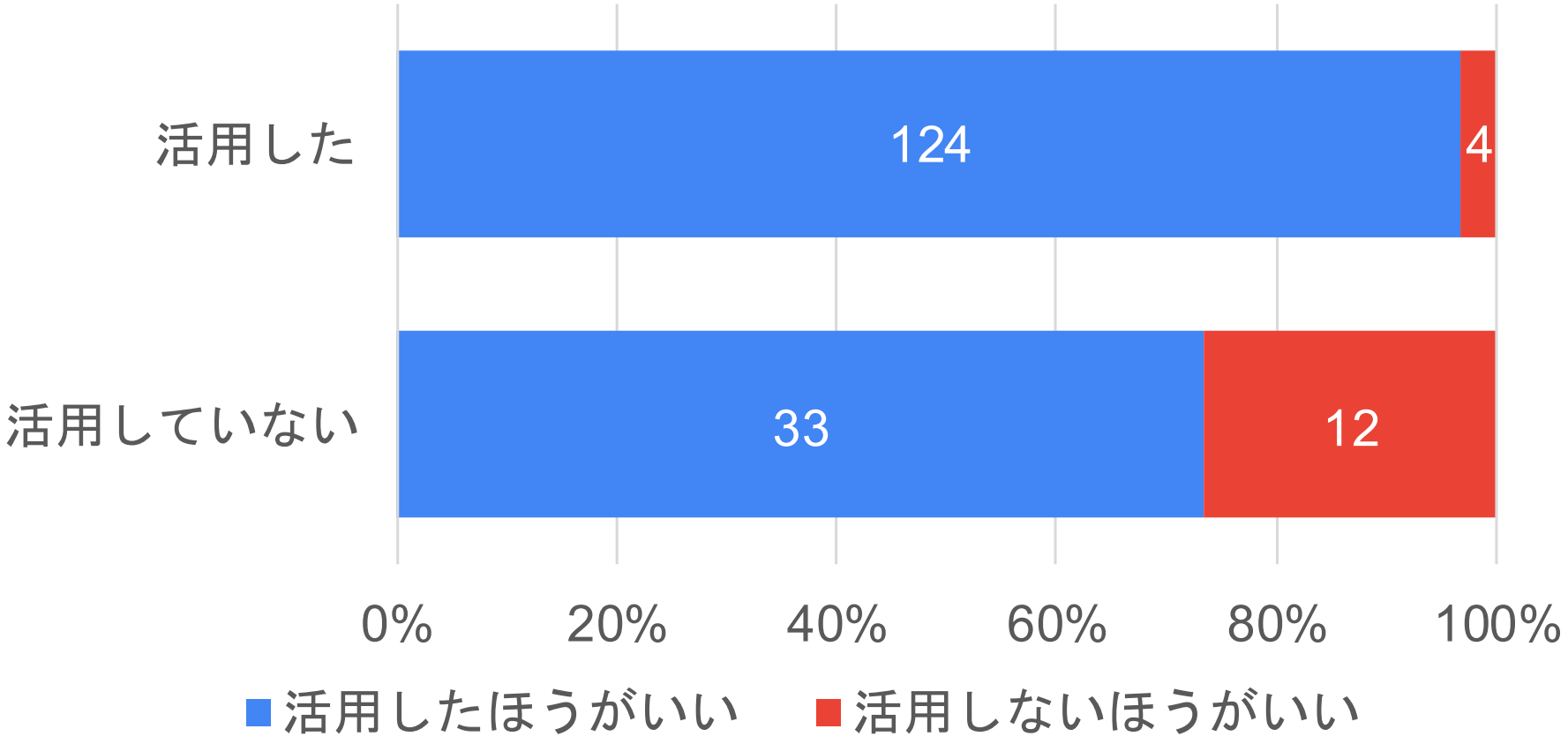
生成AIの活用×プログラミングの楽しさ



生成AIの活用×プログラミングの難しさ



生成AIの活用×生成AIの活用の是非



生成AIに関する自由記述

- プログラミングの配列など独特の考え方はかなり練習しないと自分の作品作りには使いにくいと感じました。その一方で、**チャットGPTなどを使って自分のアイデアを大まかにプログラムしてもらい、細かな修正を自分でする**ことで最初は作るのが難しいだろうと思っていた作品でも完成させることができて達成感を感じました。
- プログラムを自分で作るにあたって、ChatGPTをはじめて使用してみたのですが、**完璧ではないコードの提案も多く、試行錯誤しました。**だんだん、**ChatGPTに修正をお願いするよりも、自分で座標とか角度とか変えたほうが早いということに気づいて**、数学の円の半径とか中心の知識と関連してるな、とか思いながら修正していて、楽しかったです。

個人的なまとめ

- 生成AIの活用によって**自己解決**できる生徒が増えた印象
- 「生成AIを使用した」≠「良いプログラム」
- 生成AIの使用の**判断**は生徒に委ねたい（強要はしない）
- 生成AIは生徒の**思考のサポート**の位置づけ
- 生成AIはプログラミング以外でも**有効なツール**となる