

提出日：2024/6/26

プログラミング演習 第11回演習レポート

担当教員：杉本 千佳先生

所属：理工学部 数物・電子情報系学科
電子情報システム EP

学年・クラス：2年 Fe1

学籍番号：2364092

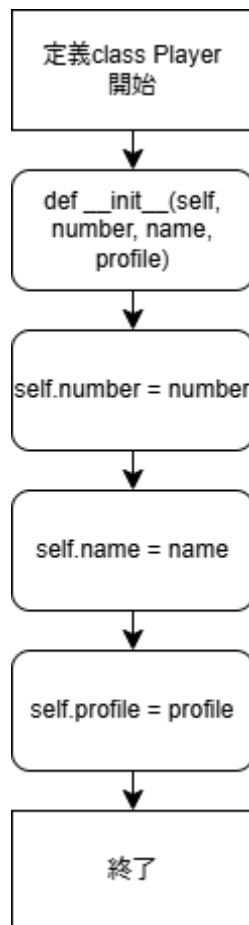
氏名：熊田 真歩

(1) 課題番号：基本課題 3

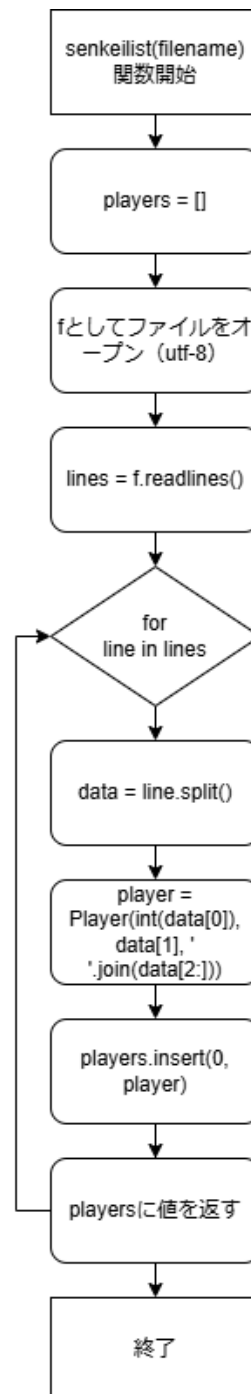
課題名：線形リストの削除

(2) プログラムのフローチャート

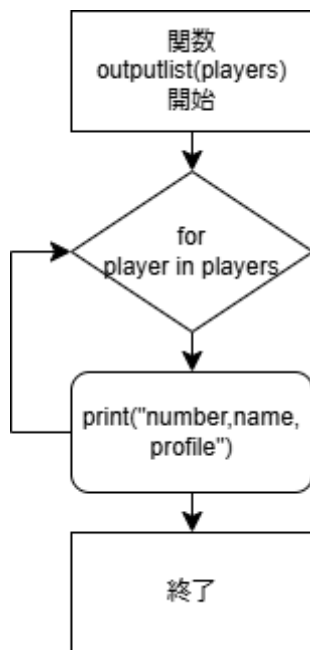
・クラス class Player



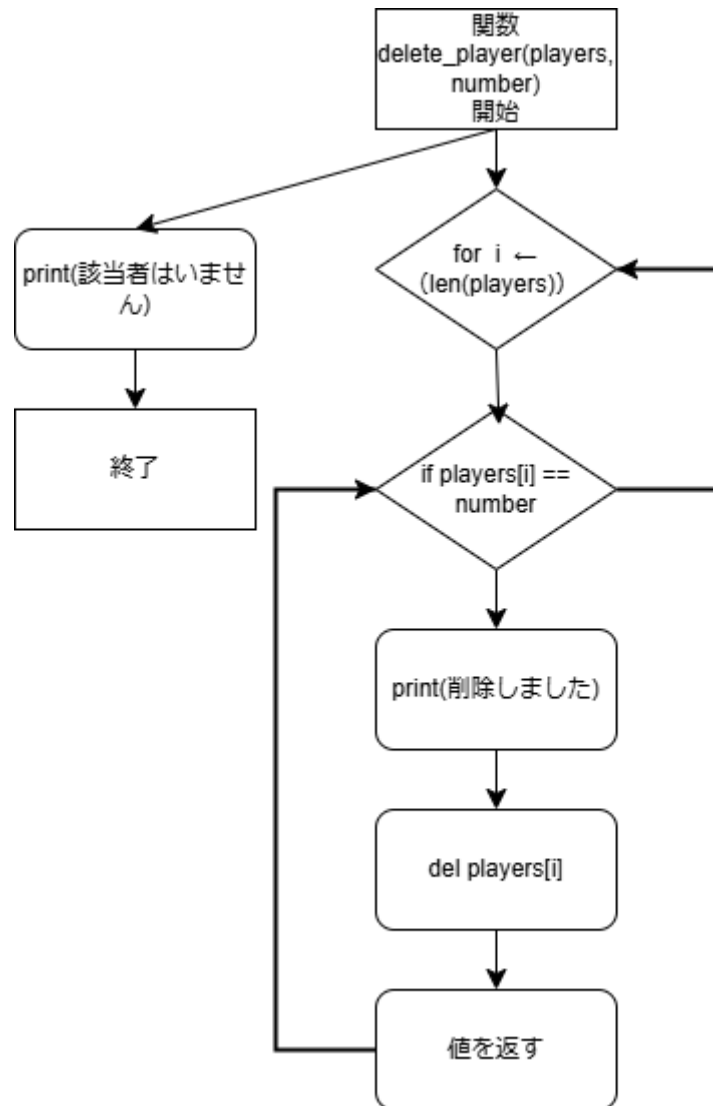
・ senkeilist(filename)関数について



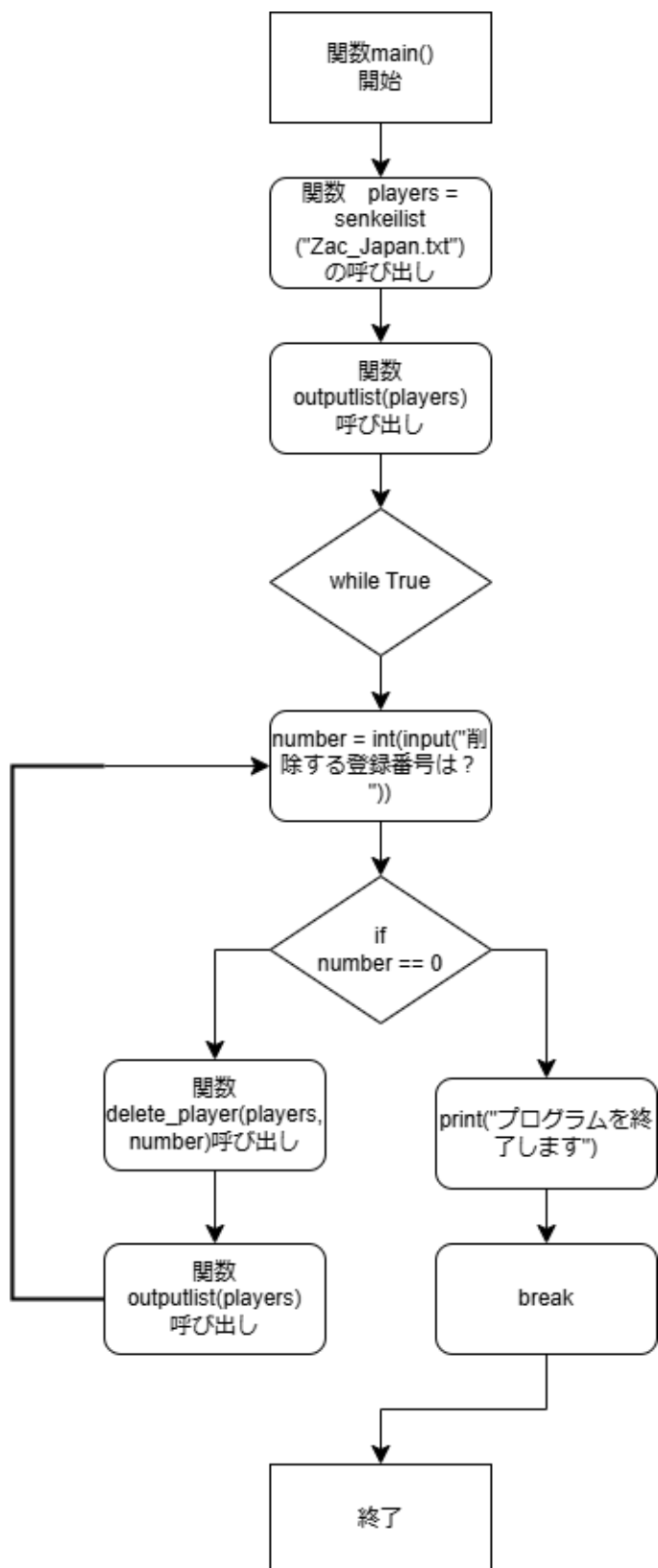
・関数 outputlist(players)について



・関数 delete_player(players, number)について



- ・ main()関数について



(3) アルゴリズムが「正しいこと」である説明

本プログラムではファイル内のデータについて、線形ファイルを用いて表示させるプログラムである基本課題 1 を利用することで、キーボードから選手の登録番号を入力するとそのデータを除いた残りのデータを表示するということを実現した。

ここで、すべての入力に対してプログラムは停止するという停止性について議論する。ここで、前提条件として選手の登録番号は整数値とし、少数値や分数値は考慮しないこととする。以下に登録番号に負の数を入力した時の結果を示す。

```
22 吉田麻也 DF_1988/08/24_189cm_81kg_サウサンプトン
15 今野泰幸 DF_1983/01/25_178cm_73kg_ガンバ大阪
2 内田篤人 DF_1988/03/27_176cm_67kg_シャルケ
4 本田圭佑 MF_1986/06/13_182cm_74kg_A C ミラン
1 川島永嗣 GK_1983/03/20_185cm_82kg_スタンダー
9 岡崎慎司 FW_1986/04/16_174cm_76kg_マインツ
7 遠藤保仁 MF_1980/01/28_178cm_75kg_ガンバ大阪
13 大久保嘉人 FW_1982/06/09_170cm_73kg_川崎 F
5 長友佑都 DF_1986/09/12_170cm_68kg_インテル
17 長谷部誠 MF_1984/01/18_180cm_72kg_ニュルンベルク
10 香川真司 FW_1989/03/17_172cm_63kg_マンチェスター U
削除する登録番号は? -2
---該当者はいません---
```

このように該当者はいませんというメッセージの表示と共に次の登録番号が聞かれた。また登録番号が 0 の場合は以下のように停止性があることが分かる。

```
10 香川真司 FW_1989/03/17_172cm_63kg_マンチェスター U
削除する登録番号は? 0
プログラムを終了します
```

すなわち上記で述べた条件の範囲内であればアルゴリズムは全ての入力に対してプログラムは正しく停止する。

次にすべての入力に対して「正しい答え」を出力することについて議論する。ただしこの場合も登録番号は上記で記した条件とする。このアルゴリズムの「正しい答え」とは存在する登録番号を入力したときはそれを削除し、存在しない登録番号を入力したときには該当者はいませんというメッセージを表示し、再び登録番号を聞くというものである。また登録番号に 0 を入力した時のみプログラムは終了する。登録番号が 0 の時、また該当番号が存在しないとき（負の数として）に関しては上の二つの例により「正しい答え」が存在することが示されている。よって以下では登録番号が存在する時の結果を表示する。よって、すべての入力に対して「正しい答え」を出力することが分かる。

```
削除する登録番号は? 10
---削除しました---
22 吉田麻也 DF_1988/08/24_189cm_81kg_サウサンプトン
15 今野泰幸 DF_1983/01/25_178cm_73kg_ガンバ大阪
2 内田篤人 DF_1988/03/27_176cm_67kg_シャルケ
4 本田圭佑 MF_1986/06/13_182cm_74kg_A C ミラン
1 川島永嗣 GK_1983/03/20_185cm_82kg_スタンダー
9 岡崎慎司 FW_1986/04/16_174cm_76kg_マインツ
7 遠藤保仁 MF_1980/01/28_178cm_75kg_ガンバ大阪
13 大久保嘉人 FW_1982/06/09_170cm_73kg_川崎 F
5 長友佑都 DF_1986/09/12_170cm_68kg_インテル
17 長谷部誠 MF_1984/01/18_180cm_72kg_ニュルンベルク
削除する登録番号は? 0
```

(4) ソース・プログラムの説明

```
class Player:                                #クラスの定義
    def __init__(self, number, name, profile): #クラスの型を(登録番号、名前、
                                                プロファイル)と設定
        self.number = number                 #登録番号のクラス設定
        self.name = name                     #名前のクラス設定
        self.profile = profile               #プロフィールのクラス設定

def senkeilist(filename):                    # 線形ファイルを作成する関数
    players = []                             #選手の情報を格納するファイル
    with open(filename, 'r', encoding='utf-8') as f: #ファイルを utf-8 仕様で
                                                読み取り専用で開く
        lines = f.readlines()                #変数 lines にファイル内容
                                                を一行ずつ格納
        for line in lines:                  #lines のそれぞれの行に
                                                ついて
            data = line.split()              #変数データに 1 行を分割し
                                                た文字列を格納
            player = Player(int(data[0]), data[1], ' '.join(data[2:]))
                                                #変数 player に lines 一行ずつのクラスを格納
            players.insert(0, player)         #変数 players に上で定義したクラスを挿入
    return players                           #players に値を返す

def outputlist(players):                    #ファイル内を出力する関数
    for player in players:                  #クラス player 中の players に対して
        print(f"{player.number:2d} {player.name} {player.profile}")
                                                #データを出力

def delete_player(players, number):         #指定された選手を削除する関数
    for i in range(len(players)):           #ファイルに含まれている players の数だけ
                                                実行
        if players[i].number == number:    #その登録番号が指定された番号と一致す
                                                る時
            print("---削除しました---")      #削除しましたというメッセージの出力
            del players[i]                  #players のリストから削除する
    return                                  #戻り値なしで関数終了
```

```

print("---該当者はいません---")          #指定された番号が登録番号にない時該当者
                                         はいませんのメッセージを出力

def main():                               #main()関数
    players = senkeilist("Zac_Japan.txt") #ファイル名を指定して線形リスト関数を
                                         呼び出す
    outputlist(players)                  #outputlist 関数を呼び出してリスト内
                                         の内容を表示

    while True:                           #条件が正しい時は継続
        number = int(input("削除する登録番号は？ ")) #削除する登録番号を変数
                                                         number に格納
        if number == 0:                       #number の値が 0 の時
            print("プログラムを終了します")      #プログラム終了のメッセージを
                                                         出力
            break                               #繰り返しのループから抜ける

        delete_player(players, number)         #delete_player 関数を呼
                                                         び出して指定番号の内容を削除
        outputlist(players)                   #outputlist 関数を呼び出して
                                                         ファイル内を出力

```

(5) 考察

初めに一番奥のデータ (NULL 側) が削除できるように工夫した点について、本プログラムではファイルのデータ数が少なくプログラムの実行時間はどのように記述しても多くはないと考え、NULL 側からデータを削除するという関数の適応を行わなかった。すなわち、NULL 側のデータに関しても前から順に判定を行い削除するという作りである。本プログラムのようにデータ数が少ないデータの場合はこの作業は厭わない場合が多いが、データ数が多くなった時にこれは改善すべきである。具体的には `len(players)` とリストの長さを変数とする関数を用いて末尾のインデックスを `delete_player` 関数に適応するというものである。このプログラムを追加することによって NULL 側のデータを直接削除することが可能となる。

また、一番手前のデータを削除する場合について、これは `senkeilist` 関数を用いてクラスファイル `Player` を作成し更にリスト `players` にデータを格納しているがこれに `insert(0,`

player)を用いている。これにより一番手前のデータを削除することが可能となっている。よって、この作業はたとえデータ数が多いとしても非常に短い時間で実行できていると考えられる。

次に、全部のデータを削除するために工夫した点について、これは delete_player 関数で指定したどの登録番号のデータでも削除できるようにすることで実現した。すなわち、全部のデータを容易に削除することができる。

最後に採用したアルゴリズムが課題を解くのにあたって適していたかどうか、改善の余地はないのかなどは議論する。ここで、本プログラムで採用したアルゴリズムが課題を解くにあたって適していたかどうかについて考察する。ファイル内容をクラスを用いて表示させることは、本プログラムを実現するにあたって不可欠であったと考える。これは、登録番号を入力して該当選手を削除するという作りであったため、単に文字としてのテキストファイルでは実現できないからである。よって、クラスを用いることでどのデータが登録番号であるのかを識別し、入力した登録番号との一致を実現できた。本プログラムは大まかに言うに存在する登録番号を入力したときはそれを削除し、存在しない登録番号を入力したときには該当者はいませんというメッセージを表示し、再び登録番号を聞くというものである。また登録番号に 0 を入力した時のみプログラムは終了する、というものである。ここで、アルゴリズムが正しいことである説明でも述べた通り、入力する登録番号は、正負は問われないが常に整数値である必要性がある。すなわち、聞かれた登録番号に少数値を代入するとプログラムはエラーとなる。この問題は while True を使い、登録番号に整数値が入力されるまでメッセージを表示しもう一度登録番号の入力を促されるというアルゴリズムの採用を行うとより汎用性が高く、良いプログラムになったと考えた。input 関数を用いたときに int 型指定をしたがこの指定を外してもより汎用性が高くなったと考えた。

(6) 感想

本プログラムは実際に 1 から作っているとクラスを使おうというアイデアが思い浮かばないような気がする。しかし、基本問題 1、2 を解きクラスを用いると良いと気づいたので基本問題 3 でも同様にクラスを用いたアルゴリズムを考察することができた。このように、あるテキストファイルから情報の選抜を行いたいときにはクラスを用いることで単なるテキストデータをプログラムの判定素材として用いることができるというのは画期的であると思った。こういったプログラムをうまく活用することで実生活において、時間が大幅に短縮できたら素晴らしいのではないかな。