プログラミング演習 第 14 回演習レポート

2064229 福場脩真

# 指定された課題番号と課題名

# 基本課題 3. 探索木の削除

# (2) フローチャートあるいは自然言語によるアルゴリズムの記述

# ダイアグラム 自動的に生成された説明

# (3) アルゴリズムが「正しいこと」である説明あるいは証明

削除したい登録番号を入力したとき、その番号の選手があった場合、ルートに近い選手でも正しく削除できた。また登録がない番号を入力しても、登録がない旨のメッセージを出力してプログラムが停止した。よって、このプログラムは正しい。

# (4) ソース・プログラムの説明

探索木の設計について。

各ノードは番号、名前、プロフィール、次のノードのポインタleft,rightが設定されている。また木構造そのものもクラスで定義し、挿入や削除の機能を実装している。

ノード削除をするTreeのdeleteメソッドとdel\_treeメソッドについて。

実行の際にはdeleteメソッドだけ呼び出され、deleteメソッドの中でdel\_treeメソッドを呼び出す。Deleteメソッドはすでにあるノードから引数numと同じものを探索し、存在すればそのノードを削除する。木のルートが削除される場合には新しいルートを作成する。Del\_treeメソッドは、ノードの最右節点を探索し、それを引数dsttの示す節点にコピーする。こうすることで、該当のノードを削除できるようになっている。

# (5) 考察

今回はアルゴリズム講義の資料を参考に、節点のクラスだけでなく、探索木そのもののクラスも定義して、その中でツリーの表示や挿入、削除の機能を実装した。この方法のメリットはいくつかあり、まず同じ構造を持つ木構造をいくつも生成できる。また機能をまとめられるので、コードの可読性も上がる。

改善点は、ルートの扱いである。今回実装した木構造では、インスタンス生成時に木のルートも定義されるようになっている。本来最初に挿入されるデータがルートになるようにしたほうがデータの扱いがしやすくなるメリットがあるが、ルートが入ることでshowメソッドを呼び出したときに余計な表示が出現したり、様々な障害が生じた。改善案は、ルートを無視したアルゴリズムを組むか、ツリーのインスタンス生成時先頭データをルートとして定義するかのどちらかであると考えられる。