

# プロジェクト演習課題 4 作業内容の報告

学籍番号 745086

氏名 松浦 康大

グループ 07

今回、我々のグループは repeat 文の追加を目標として作業を行なった。各自担当箇所から、repeat 文を追加する際に変更が必要になるファイルを探し、変更を加えた。

結果、実装まではいかなかったが自分が行った作業内容を報告する。

私がソースコード読解を担当した

私は、担当ファイルの `execute.c` を変更した。具体的な作業内容を以下に記す。

```
static ex_condition execRepeat(const repeatnode *rep)
{
    ex_condition r = ex_normal;
    do {
        evaluate(rep->expr);
        if (stack[sp++] == 1) break; //条件が真なら break
        r = execStatements(rep->body);
    }while (r == ex_normal);
    return (r == ex_return) ? ex_return : ex_normal;
}
```

下記のソースコードを追加した。execRepeat()関数は repeat 文を実行する関数である。execWhile()関数の if 文の条件を 0 から 1 に変えている。これは、while 文の終了条件が、条件式を満たさなかった場合(0)であるのに対し、repeat 文の終了条件が、条件式を満たした場合である為である。また、while 文とは違い、until の後に条件式が来る為、それを stack に push しておく必要がある。

```
while (stp) {
    switch (stp->kind) {
        .....

        case node_repeat:
            r = execRepeat((const repeatnode *)stp);
            if (r != ex_normal) return r;
            break;

        .....
    }
```

次に `execStatements()`関数の `case` 文に `repeat` 文のための追加を行った。ここは他の文と同様で変更する必要はないと考える。

まだコンパイルが通っただけで、サンプルプログラムが思うように通っていないため、改良が必要である。自分の担当箇所が `execute.c` で実行部分であるため、今の所ここまでたどり着いておらず、このプログラムで実際に動くかはわからない。問題があるとすれば、`while` 文と `repeat` 文の条件式の終了条件の式の位置が異なるため、今のままでは動かないかもしれない点である。`repeat` 文の終了条件は `until` の後にくるため、もしうまくいかなかった場合、実際にサンプルプログラムの動作を追いながら、条件を `stack` に突っ込めるように変更を加えようと思う。