基礎プログラミングII

第4回　再帰処理

再帰

何かの中身にそれ自身が現れること

再帰的定義

何かの定義にそれ自身が登場するもの

再帰的定義の例：年齢

- 定義１：生まれた年月日から経過した年数
- 定義２：１年前の**年齢**に１を足したもの

再帰的定義

再帰を利用したプログラミング

例題：自然数に対する階乗

定義１：再帰的でない場合の定義

Nの階乗とは

1からNまでのすべての自然数を掛けたもの

再帰を利用したプログラミング

例題：自然数に対する階乗

定義２：再帰的定義

Nの階乗とは

N-1の階乗とNを掛けたものである。ただし、1の階乗は1である。

Nの階乗を求めるメソッド（定義１）

```
def factorial (n)
    ans = 1
    i = 1
    while i <= n
        ans *= i
        i += 1
    end
    ans
end
```

Nの階乗を求めるメソッド（定義２）

```
def factorial (n)
    if n <= 1
        1
    else
        n * factorial (n-1)
    end
end
```

再帰的メソッドの注意点

- 問題を分割できるとき、分割部分を解く考え方と全体を解く考え方が同じとき、再帰を使える。
- 問題分割を繰り返し、これ以上問題が分割できない場合に結果そのものを返す条件ブロックを入れる（**脱出条件**）。
- 分割した問題を再帰的に同じメソッドを呼んで得られた結果を合成したものを最後の結果として返す。

クイックソートの手順

大きな箱Xに任意個のデータが入っているとする。

1. Xの中のデータが１個以下なら並べ換え完了。そうでなければ次に進む。
2. Xの中から、任意のデータを１個抜き取り、その値をkとする。
3. kを抜き取ったあとの中のデータを順次見ていき、kより小さかったら左側に置く、そうでなければ右側に置くことをデータがなくなるまで繰り返す。
4. 左側に寄せたものをクイックソートする。
5. 右側に寄せたものをクイックソートする。
6. 左右のかたまりの中間にkを置いて並べ換え完了。

クイックソートを行うメソッドqsortの設計

- qsortメソッドは引数として配列を受け取る。
- 左側の箱、右側の箱としてそれぞれ、配列left、配列rightを利用する。
- left、rightに対して再帰的にqsortする。
- 最後にleft、真中の値、rightを並べて完了。

クイックソートを行うメソッドqsortの設計

```
def qsort (data)
    if データが１個以下なら
        dataそのものを返して並べ換え完了
    else
        left (左側配列)を新規作成
        right (右側配列)を新規作成
        変数dataからデータを1個取り出す
        これをkとする
        残ったdata全てに対して以下を繰り返す
            if kより小さかったら
                leftに追加
            else
                rightに追加
            end
        繰り返し終わり
        「leftをqsortしたもの」とkと「rightをqsortしたもの」をその順に結合したものを結果として並べ換え完了
    end
end
```

まとめ

- 再帰とは、何かの中身にそれ自身が現れること。
- 再帰処理を用いると複雑な問題を簡単にして解くことができる場合がある。