関数型とオブジェクト指向型言語の対比

NE24-0120B 西辻駿

NE24-0224J 佐藤周弥

関数型言語とは

- 関数を中心としたスタイル
- 純粋な関数型では参照透過性が保証される

研究目的

1. 利点、欠点

2. アプローチの選択

3. 概念の理解

研究してきたこと

オブジェクト指向について

- カプセル化
- 継承（抽象と実装）
- ポリモーフィズム

関数型について

- 参照透過性が高い
- 副作用が少ない（遅延評価）

コード例（ROT13）

Ruby

```ruby
def rot13ch(ch)
  if 'A' <= ch && ch <= 'M' || 'a' <= ch && ch <= 'm'
    ch = (ch.ord + 13).chr
  elsif 'N' <= ch && ch <= 'Z' || 'n' <= ch && ch <= 'z'
    (ch.ord - 13).chr
  else
    ch
  end
end

def rot13(x)
  result = []
  x.chars { |xs| result << rot13ch(xs) }
  result.join
end

hello13 = "Hello, World!"
puts hello13
puts rot13(hello13)
```

Haskell

```haskell
import Data.Char

rot13ch ch
    | 'A' <= ch && ch <= 'M' || 'a' <= ch && ch <= 'm' = chr $ ord ch + 13
    | 'N' <= ch && ch <= 'Z' || 'n' <= ch && ch <= 'z' = chr $ ord ch - 13
    | otherwise = ch

rot13 "" = ""
rot13 (x:xs) = rot13ch x : rot13 xs

main = do
    let hello13 = rot13 "Hello, World!"
    print hello13
    print $ rot13 hello13
```

最終目標

- 体系的に説明
- 実際に使えるコードを書く