テキスト処理 第4回 (2006-05-16)
mapのレポート説明

田中哲
産業技術総合研究所
情報技術研究部門
akr@isc.senshu-u.ac.jp
http://staff.aist.go.jp/tanaka-akira/textprocess/

レポート

- Array#map と同様な動作をする map を実装し、動作させ、解説せよ

```
def map(ary)
  ここを埋める
end
```

- 正しく定義すると以下のように動く

```
p map([1,2,3]) {|v| v * 2 }        #=> [2, 4, 6]
p map([]) {|v| v + 1 }             #=> []
p map([2,3,4]) {|v| v * v }    #=> [4, 9, 16]
```

- Array#map を使って ary.map {|v| yield v } などというのは控えるように

レポート (つづき)

- 足りないメソッドはリファレンスマニュアルで探すこと http://www.ruby-lang.org/ja/man/
  (おそらく Array#length が必要になる)
- IT's class で提出すること
- ✍切は次回の授業が始まるまで
  2006-05-16 16:20

mapの実装

- コード例

```
def map(ary)
  i = 0
  r = []                          # 空配列は毎回生成
  while i < ary.length
    r[i] = yield(ary[i])          # 配列は自動拡張
    i += 1
  end
  r                               # 最後の式が返り値
end
```

Array#each による実装

- コード例

```
def map(ary)
  r = []
  ary.each {|v|       # 要素を先頭から繰り返し
    r << yield(v)
  }
  r
end
```

- Array#<< は配列の最後に要素を追加する

再帰による実装

- コード例

```
def map(ary)
  if ary.empty?
    []
  else
    [yield(ary[0])] + map(ary[1..-1]) {|v| yield v }
  end
end
```

- このような不要な再帰は良くない
- この例は O(n²) になっている

いろいろな実装案

- Array#length で長さを得て while でループ
- Array#each
- Array#each_with_index
- Array#reverse_each で逆順 (推奨できない)
- 再帰 (map には推奨できない)
- r << v で追加
- r += [v] で追加 (推奨できない)
- etc.