

GO FOR IT -コードで世界は変えられる-
問題1 回答

栗花悠輔

平成 24 年 2 月 12 日

1.1 プログラムのコード (ファイル名 : life_clock.rb)

```
#!/usr/bin/ruby -Ku
# -*- coding:utf-8 -*-

=begin
one line to give the program's name and an idea of what it does.

Copyright (C) 2012 Yusuke Kurihana

This program is free software; you can redistribute it and/or modify
it under the terms of the GNU General Public License as published by the
Free Software Foundation; either version 2 of the License,
or (at your option) any later version.

This program is distributed in the hope that it will be useful,
but WITHOUT ANY WARRANTY; without even the implied warranty of
MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE.
See the GNU General Public License for more details.

You should have received a copy of the GNU General Public License
along with this program; if not, write to the Free Software Foundation,
Inc., 59 Temple Place - Suite 330, Boston, MA 02111-1307, USA.
=end

# life_clock.rb は 一生を 24 時間に例えた時の今日の時間を表示するプログラムである

# 秒 (float 型) で与えられる時間を [時間, 分, 秒] に直す
# === Args
# 浮動小数点数で与えられる秒
# === Return
# 時間、分、秒の 3 要素からなる配列
def sec2hour(s)
  hours = 0
  minites = 0

  while s >= 60.0
    if minites >= 60
      hours += 1
      minites = 0
    end
    s -= 60.0
    minites += 1
  end
end
```

```

end

return [hours, minites, s]
end

# 得られた現在時刻を文字列に整形し出力する
# === Args
# _obj_ :: 得られた現在時刻
def life_clock_to_string(obj)
  h = obj[0]
  m = obj[1]
  s = obj[2]

  printf "life clock: %d:%2d:%5.5f\n", h, m, s
end

# a年 b月 c日に生まれ n年生きる
# === Args
# _i_ :: 年
# _j_ :: 月
# _k_ :: 日
# _m_ :: 寿命
def life_clock(i, j, k, m)
  a = i
  b = j
  c = k
  n = m

  birth_time = Time.local(a, b, c).to_f
  death_time = Time.local(a + n + 1, b, c).to_f - 1.0
  now = Time.now.to_f
  a_sec_of_life = (death_time - birth_time) / (24 * 60 * 60).to_f

  life_clock_to_string(sec2hour((now - birth_time) / a_sec_of_life))
end

# 不正な ARGV かどうかをチェックする
# === Return
# _true_ :: 不正な ARGV 引数
# _false_ :: 正しい ARGV 引数
def invalid_argv?
  ret = false
  month = ARGV[0].to_i

```

```

day = ARGV[1].to_i

unless 1 <= month || month <= 12
  ret = true
end

case month
when 4, 6, 9, 11
  ret = !(1 <= day || day <= 30)
when 2
  ret = !(1 <= day || day <= 29)
else
  ret = !(1 <= day || day <= 31)
end

return ret
end

# 設問 人生の時計 を行う主たる関数
def main
  if ARGV.length != 2 || invalid_argv?
    puts "invalid arguments"
    puts "usage: ruby #{ $0 } your_birthmonth your_birthday"
    exit(1)
  end

  puts Time.now

  puts "(i)"
  puts "n = 80"
  1990.upto(2000) do |i|
    print "year #{i}: "
    life_clock(i, 9, 28, 80)
  end

  puts "(ii)"
  puts "n = 200"
  1900.upto(2000) do |i|
    print "year #{i}: "
    life_clock(i, 9, 28, 200)
  end
end
end

```

main

1.2 出力

私の生年月日を入力した際の動作を示す

```
$ ruby life_clock.rb 9 28
2012-02-12 21:48:44 +0900
(i)
n = 80
year 1990: life clock: 6:20:2.20804
year 1991: life clock: 6: 2:15.52778
year 1992: life clock: 5:44:27.39648
year 1993: life clock: 5:26:41.45090
year 1994: life clock: 5: 8:55.50532
year 1995: life clock: 4:51:8.96927
year 1996: life clock: 4:33:20.69376
year 1997: life clock: 4:15:34.74818
year 1998: life clock: 3:57:48.80260
year 1999: life clock: 3:40:2.41076
year 2000: life clock: 3:22:13.99104
(ii)
n = 200
year 1900: life clock: 13:17:55.66575
year 1901: life clock: 13:10:46.10190
year 1902: life clock: 13: 3:36.53806
year 1903: life clock: 12:56:26.33965
year 1904: life clock: 12:49:16.23349
year 1905: life clock: 12:42:6.66964
year 1906: life clock: 12:34:57.10580
year 1907: life clock: 12:27:46.93081
year 1908: life clock: 12:20:36.80123
year 1909: life clock: 12:13:27.23738
.
(中略)
.
year 1991: life clock: 2:25:59.34516
year 1992: life clock: 2:18:48.72374
year 1993: life clock: 2:11:39.15990
year 1994: life clock: 2: 4:29.59605
year 1995: life clock: 1:57:19.93632
year 1996: life clock: 1:50:9.29148
year 1997: life clock: 1:42:59.72764
year 1998: life clock: 1:35:50.16379
year 1999: life clock: 1:28:40.59995
year 2000: life clock: 1:21:29.92583
```

1.3 アルゴリズムの説明

主な処理は `life_clock` 関数で行なっている .

現在時刻と死ぬ予定の時刻、現在時刻を UNIX 時刻 (浮動小数点数) に変換する . そしてその数を一日の秒数で割ることで、一生を 24 時間に例えた時の 1 秒に相当する時間を得る事ができる . それを用いて生きた時間 (今から生まれた時間を引いた差分) を割ることで、1 日に例えた時の現在時刻を算出する

1.4 その他

1.4.1 実行方法

コマンドラインから

```
ruby life_clock.rb 3 8
```

1.4.2 実行環境

ruby1.8.7、1.9.3 では動作を確認した

1.4.3 ライセンス

本プログラムは GPL ライセンスに準拠している