

プログラミング言語 第六回

担当: 篠沢 佳久
櫻井 彰人

平成20年 5月26日

1

本日の内容

- 繰り返し(2)
- times, each
- 繰り返しについての練習問題
- 第二回レポート課題

2

回数の決まった繰り返し

```
times  
each
```

3

times①

- 同じ処理をn回繰り返したい
- n.times

```
n.times {  
  式  
}
```

式をn回繰り返す

4

回数がわかっている繰り返し①

- 10.times

```
10.times {print "やっほ～ "; puts "    Yee-ha! " }
```

```
10.times {  
  print( "やっほ～ " )  
  puts( "    Yee-ha! " )  
}
```

停止条件は書かなくてもよい
n.times で指定されたn回、式を繰り返す

5

loop{}で書く場合には

```
10.times {  
  print( "こんにちは¥n" )  
}
```



```
i = 0  
loop{  
  i = i+1  
  print( "こんにちは¥n" )  
  break if i == 10  
}
```

6

練習①

1から100までの整数のうち偶数のみを印字するプログラムを times を用いて書いて下さい

ヒント1: まず、1から100までの整数を印字するプログラムを書いてみましょう。右の通りです

```
100.times{ |i|
  print( i, "\n" )
}
```

ヒント2: では、「偶数のみ」にはどう対応しますか？右のようにします。

```
100.times{ |i|
  if ??? then
    print( i, "\n" )
  end
}
```

13

練習②

1から100までの整数の合計値を印字するプログラムを times を用いて書いて下さい

ヒント1: まず、1から100までの整数を印字するプログラムを書いてみましょう。右の通りです

```
100.times{ |i|
  print( i, "\n" )
}
```

ヒント2: では、「合計」はどう計算しますか？右のようにします。

```
?? = 0
100.times{ |i|
  total = ?? + ???
  # または total += ???
}
print( ????? )
```

14

each①

```
n.times{ |i|
  式
}
```

n回式を繰り返す
iには自動的に0からn-1が代入される

```
(n..m).each{ |i|
  式
}
```

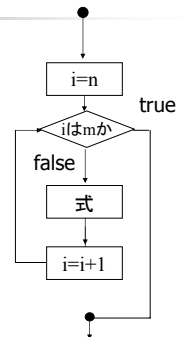
m-n回式を繰り返す
iには自動的にnからmが代入される

15

each②

```
(n..m).times{ |i|
  式
}
```

iには自動的にnからmが代入される
その結果、式は m-n回繰り返される



16

変数範囲が決まっている繰り返し①

10のべき乗を表示するプログラム

```
(0..9).each{ |i|
  print( 10 ** i, "\n" )
}
```

```
10.times{ |i|
  print( 10 ** i, "\n" )
}
```

17

変数範囲が決まっている繰り返し②

```
(5..10).each { |i|
  print( i, if i%2==0 then " は偶数" else " は奇数" end, "\n" )
}
```

```
(5..10).each { |i|
  print( i )
  if i%2==0 then
    print( " は偶数" )
  else
    print( " は奇数" )
  end
  print( "\n" )
}
```

```
s = 5
e = 10
(s..e).each { |i|
  print( i )
  if i%2==0 then
    print( " は偶数" )
  else
    print( " は奇数" )
  end
  print( "\n" )
}
```

5: は奇数
6: は偶数
7: は奇数
8: は偶数
9: は奇数
10: は偶数

18

変数範囲が決まっている繰り返し③

合計値を求めるプログラム

```
total=0
(10..100).each { |i|
  total += i
}
print( total )
```

```
n=gets.chomp.to_i
m=gets.chomp.to_i
total=0
(n..m).each { |i|
  total += i
}
print( n, "から", m, "までの合計は", total )
```

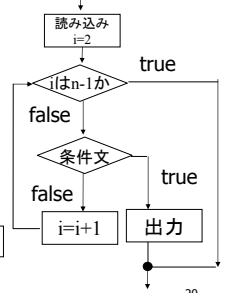
19

変数範囲が決まっている繰り返し④

何を調べているプログラムでしょう

```
n=gets.chomp.to_i
(2..n-1).each { |i|
  if n % i == 0 then
    print( "〇〇ではありません" )
    break
  end
}
```

break でループを抜けることも可能



20

変数範囲が決まっている繰り返し⑤

```
(-10..10).each{ |i|
  print( i, " ", i**2, "¥n" )
}
```

負の場合は？

```
(10..-10).each{ |i|
  print( i, " ", i**2, "¥n" )
}
```

(n..m).each
n>m の場合は？

21

刻み幅に小数値を使用したい場合

①

0から1まで0.1刻みで二乗の計算を行なう

```
11.times{ |i|
  print( i / 10.0, " ", (i/10.0)**2.0, "¥n" )
}
```

i は0から10まで1刻みの整数値

10.0で割る

22

刻み幅に小数値を使用したい場合

②

```
(0..10).each{ |i|
  print( i / 10.0, " ", (i/10.0)**2.0, "¥n" )
}
```

```
(0..100).each{ |i|
  x = i.to_f / 100
  print( x, " ", x**2.0, "¥n" )
}
```

23

練習③

下記と同様な機能を持つプログラムをeach を用いて書きなさい

```
total = 1
10.times { |i|
  total *= (i+1)
}
print( total + "¥n" )
```

```
total = 0
10.times { |i|
  total += (i+5)
}
print( total + "¥n" )
```

24

練習③'

- $-\pi$ から π まで 0.1π 刻みで $\cos$ の値を求めるプログラムを `each` を用いて書きなさい。

25

擬似乱数①

- コンピュータが計算して作り出す乱数。本当の乱数ではないが、かなり本物に近い。
- ruby には組み込み関数として、`rand()` がある
 - `rand(n)` とすると 0 以上 n 未満の整数値が一樣ランダムに生成される

```
irb(main):090:0> rand()
=> 0.0422245532019152
irb(main):091:0> rand(10)
=> 7
irb(main):092:0> rand(100000)
=> 16339
```

26

擬似乱数②

```
30.times {
  if rand() > 0.5 then
    print( "1" )
  else
    print( "0" )
  end
}
```

```
30.times { print rand(2) }
```

実行する度に結果は異なります

111111001001110001110110000110=> 30

```
30.times {
  print( if rand() > 0.5 then "1" else "0" end )
}
```

27

擬似乱数③

```
s = 0
loop{
  s = rand(6)+1
  break if s == 6
  print( s , "¥n" )
}
```

1から6の整数を生成

どういう乱数でしょうか
 $(\text{rand}(10)+1)*100$

$(\text{rand}(10)+1)/10.0$

$(\text{rand}(5)+1)/10.0+0.5$

28

練習問題④

- 100未満の整数を n 個生成するプログラムを作成しなさい
- 次に以下のことができるようにプログラムを改良しなさい
- n 個の整数の合計値、平均値を求めなさい
- n 個の中で最小値、最大値を調べなさい

29

練習問題⑤

- 二つの整数 m と n ($m>0$, $n>0$)を読み込み m の n 乗を求めるプログラム `times` および `loop` を用いて書きなさい

30

練習問題

- 練習問題①から⑤を行ないなさい。
- 第五回の自習問題(特にloopの部分)をダウンロードして試してみてください。
- プログラムと実行結果をワープロに貼り付けて、keio.jp から提出して下さい。

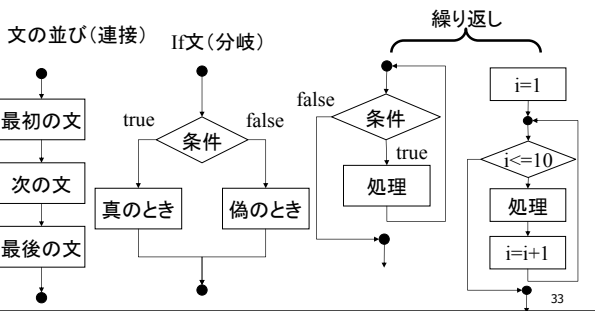
31

前回の復習

制御構造
繰り返し(1)

32

制御構造(復習)



33

無限の繰り返し

```

loop{
  式
}

loop{
  式
  break 条件式
}
次の式
  
```

式が永久に実行される
停止するために **break** を
用いる

条件式を満たした場合のみ
停止する(loopブロックの
次の式を実行する)

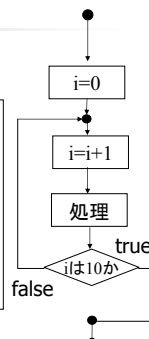
34

無限の繰り返し①

10回「こんにちは」を表示するプログラム

```

i = 0
loop{
  i = i+1
  print( i, "回目のこんにちは\n" )
  break if i == 10
}
  
```



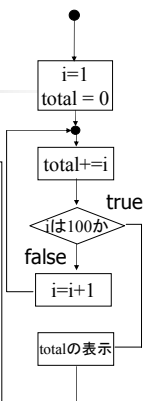
35

無限の繰り返し②

100以下の整数の和を求めるプログラム

```

i = 1
total = 0
loop{
  total += i
  break if i == 100
  i = i+1
}
print( "合計は", total )
  
```



36

times と loop の関係(1)

```
10.times { |i|  
  print( i , "回目のこんにちは¥n" )  
}
```



```
i = 0  
loop{  
  i = i+1  
  print( i , "回目のこんにちは¥n" )  
  break if i == 10  
}
```

37

times と loop の関係(2) (回数がわかっている繰り返し③)

10のべき乗を表示するプログラム

```
10.times{ |i|  
  print( 10 ** i , "¥n" )  
}
```

```
i = 0  
loop{  
  print( 10 ** i , "¥n" )  
  break if i == 10  
  i = i+1  
}
```

38

times と loop の関係(3) (練習①でした)

下記と同様な機能を持つプログラムをtimes を用いて書きなさい

```
i = 1  
loop{  
  if i % 2 == 0 then  
    print( i , "¥n" )  
  end  
  break if i == 100  
  i = i+1  
}
```

1から100までの整数で偶数を調べる

39

times と loop の関係(4) (練習②でした)

下記と同様な機能を持つプログラムをtimes を用いて書きなさい

```
i = 1  
total = 0  
loop{  
  total += i  
  break if i == 100  
  i = i+1  
}  
print( "合計は" , total )
```

1から100までの整数の合計値を求める

40

第二回レポート課題

問題①②③

41

問題①

$$e^x = 1 + \frac{x}{1!} + \frac{x^2}{2!} + \frac{x^3}{3!} + \cdots + \frac{x^n}{n!} + \cdots$$

上記の近似式より e の値を求めるプログラムを書きなさい(x = 1)
n = 1000 として求めた場合、Math::E との値を比較しなさい

42

問題②

- 元金 n 円, 日歩 r で借りた場合, d 日目までの金利を出力するプログラムを書きなさい
- ただし, 7日以内は金利「0!」と出力するようにしなさい
- 元金 n 円, 日歩 r , 日数 d についてはキーボードから値を入力(標準入力)できるようにしなさい

43

問題③

■ Collatz-角谷の予想

- 自然数 n を選び,
 - 奇数ならば, 3倍して1をたす。
 - 偶数ならば, 2で割る。

これを繰り返すと, どんな n を選んでも, いつかは, 1になる

```
3, 10, 5, 16, 8, 4, 2, 1
4, 2, 1
5, 16, 8, 4, 2, 1
6, 3, 10, 5, 16, 8, 4, 2, 1
7, 22, 11, 34, 17, 52, 26, 13, 40, 20, 10, 5, 16, 8, 4, 2, 1
8, 4, 2, 1
9, 28, 14, 7, 22, 11, 34, 17, 52, 26, 13, 40, 20, 10, 5, 16, 8, 4, 2, 1
10, 5, 16, 8, 4, 2, 1
11, 34, 17, 52, 26, 13, 40, 20, 10, 5, 16, 8, 4, 2, 1
```

44

問題③

- 整数 n を標準入力で読み込み, 前頁の性質を確認するプログラムを作成しなさい

45

第2回レポート提出①

- 「keio.jp」の「教育支援システム」で提出
 - レポート(ワープロ)、プログラムファイルおよび実行結果を電子的に提出してください。
 - レポート本文
 - 先頭の一行は、「プログラミング言語第2回レポート」とし、以下に学籍番号・氏名を記してください。
 - それ以下に、課題内容、各プログラムの説明とプログラムおよび実行結果、そしてプログラムを書く上で工夫した点、悩んだ点(考察)を書いてください。
 - 最後に、感想を書いてください。

46

第2回レポート提出②

- 課題: 問題①②③
- 締め切り
 - 6/9(月) 10時半(講義開始前まで)

47

レポートの書き方①

プログラミング言語 第2回レポート
学籍番号 氏名

1. 課題内容

何を目的としたプログラムか
入力に対してどのような出力をするプログラムなのか

2. プログラムの説明

使用した変数の説明
問題を解く上でどのようにプログラミングしたのか
プログラムの各行の説明
(ただしプログラムが長い場合、一行ごとに説明する必要はない、まとまった部分ごとで説明すればよい)

48

レポートの書き方②

3. プログラムと実行結果

```
# 第二回レポート  
# 問題(1)  
  
x = 0  
loop {  
  y = x ** 2  
  print( "x=%d, y=%d\n", x, y )  
  break if x > 10  
  x += 1  
}
```

Rubyプログラム

```
C:\Ruby\bin> ruby sample.rb  
x=0, y=0  
x=1, y=1  
x=2, y=4  
x=3, y=9  
x=4, y=16  
x=5, y=25  
x=6, y=36  
x=7, y=49  
x=8, y=64  
x=9, y=81  
x=10, y=100  
C:\Ruby>
```

コマンドプロンプト上での実行結果

49

レポートの書き方③

- プログラムと実行結果
- プログラムは画面のキャプチャーはせず、本文中に挿入して下さい
(プログラムの動作確認のためです)

50

レポートの書き方④

4. 考察

目的とした出力が適切に得られているか
プログラムの作成上において工夫した点
プログラムの作成上において悩んだ点(分からなかった点)
自分がこの課題を作成する上で苦労した点、
他の人とは違うという点を主張して下さい

5. 感想

意見, 要望など

51

レポートを書く上での注意

- プログラム, レポートのコピーは厳禁
- 相談するのはかまわない
- 完成しなかった場合, どこまでできたのか, うまく動かなかったこと, 悩んだことを考察に記述して下さい

52