

オブジェクト指向入門

管理工学科

篠沢佳久

円の円周と面積を求めるプログラム

```
import math
```

circle.py

```
r = 10
```

```
l = 2 * math.pi * r
```

```
s = r * r * math.pi
```

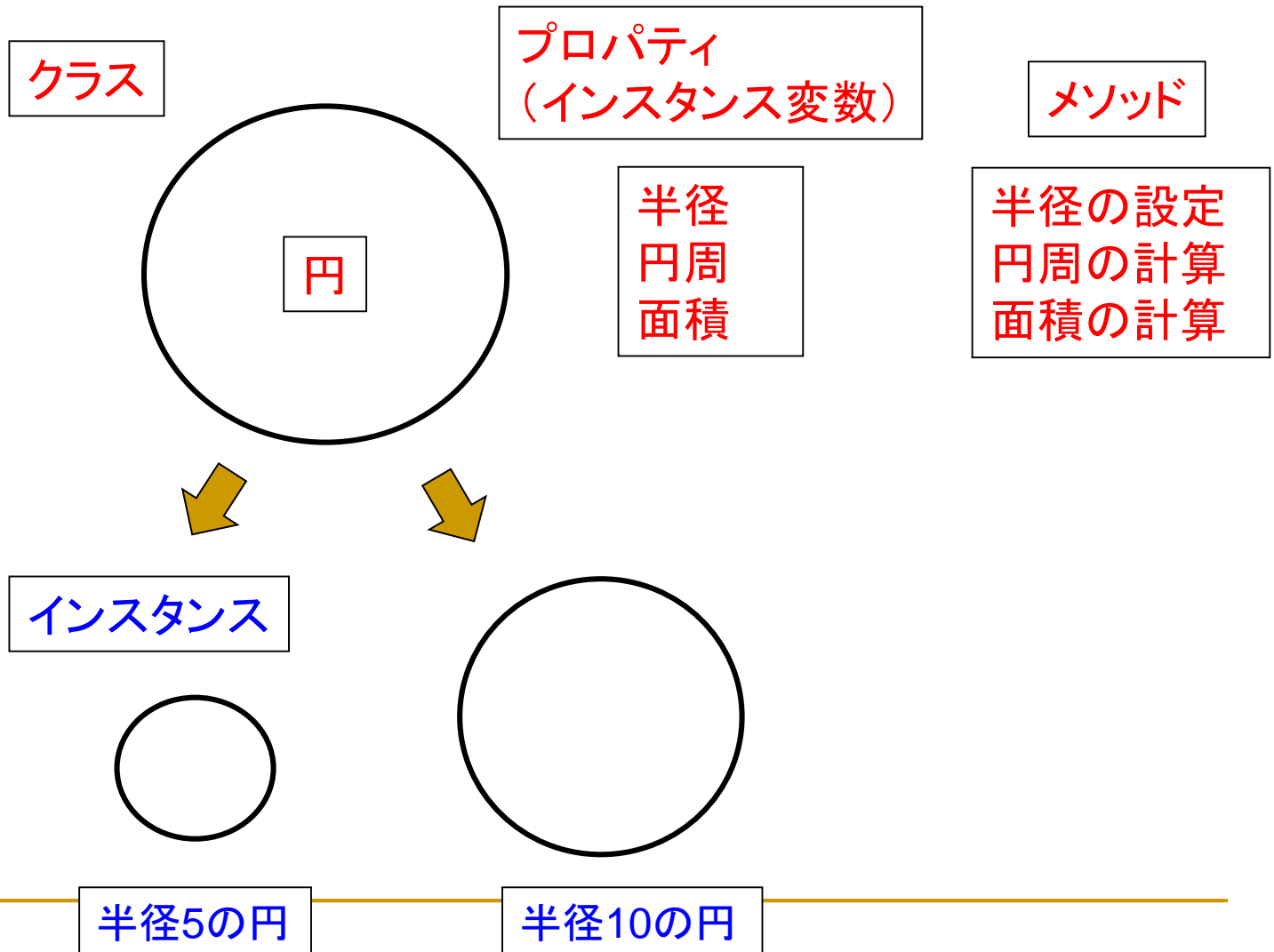
```
print( " 半径 {0:.2f}の円の円周は {1:.2f}, 面積は {2:.2f}です".format( r , l , s ) )
```

実行結果



The screenshot shows a Windows command prompt window with the title bar "C:\Windows\system32\cmd.exe". The command prompt shows the command "C:\Users\shino\Desktop>python circle.py" and the output "半径 10.00の円の円周は 62.83, 面積は 314.16です".

オブジェクト指向



```
import math
```

circle-1.py

クラスの定義

```
class Circle:
```

```
# コンストラクター
```

```
def __init__( self ):
```

```
    self.r = 0
```

r: 半径

```
    self.l = 0
```

l: 円周

```
    self.s = 0
```

s: 面積

インスタンス変数の初期化
__init__メソッド

インスタンス変数
self.変数名

```
# 半径の設定
```

```
def setRadius( self , r ):
```

```
    self.r = r
```

半径を設定するメソッド

self.r → インスタンス変数
r → 引数の値

```
# 円周の計算
```

```
def calc_L( self ):
```

```
    self.l = 2 * self.r * math.pi
```

円周を計算するメソッド

```
# 面積の計算
```

```
def calc_S( self ):
```

```
    self.s = self.r * self.r * math.pi
```

面積を計算するメソッド

インスタンスによる計算①

コンストラクター

```
en = Circle()
```

インスタンスの作成
(コンストラクター)

インスタンス名 = クラス名()
__init__() が実行

半径の設定

```
en.setRadius( 5 )
```

メソッドの呼び出し
インスタンス名.メソッド()

```
en = Circle()
```

```
en.r = 0  
en.l = 0  
en.s = 0
```

円周の計算

```
en.calc_L()
```

```
en.setRadius( 5 )
```

```
en.r = 5
```

面積の計算

```
en.calc_S()
```

インスタンス変数へのアクセス*
インスタンス名. インスタンス変数

インスタンス変数

```
print( " 半径 {0:.2f}の円の円周は {1:.2f}, 面積は {2:.2f}です".format( en.r , en.l , en.s ) )
```

*オブジェクト指向では、メソッド以外でインスタンス変数に直接アクセスしない方がよいとされています(カプセル化)

インスタンスによる計算②

コンストラクター

```
en1 = Circle()
```

インスタンスの作成
(コンストラクター)

半径の設定

```
en1.setRadius( 10 )
```

```
en1.setRadius( 10 )
```



```
en1.r = 10
```

円周の計算

```
en1.calc_L()
```

メソッドの呼び出し

面積の計算

```
en1.calc_S()
```

```
print( " 半径 {0:.2f}の円の円周は {1:.2f}, 面積は {2:.2f}です".format( en1.r , en1.l , en1.s ) )
```

クラスの定義

class クラス名:

クラス変数 = 初期値

クラス変数
クラス共通の変数
全てのインスタンスで同じ値

def __init__(self):

コンストラクター

self.インスタンス変数名 = 初期値

インスタンス変数
インスタンスごとに
異なる値

def メソッド名(self , 引数):

メソッド

インスタンスによる処理

- コンストラクターの呼び出し

インスタンス名 = クラス名()

```
en = Circle()
```

- メソッドの呼び出し

インスタンス名.メソッド名()

```
en.setRadius(10)
```

```
en.setRadius(10)
```

- クラス変数へのアクセス

クラス名.クラス変数

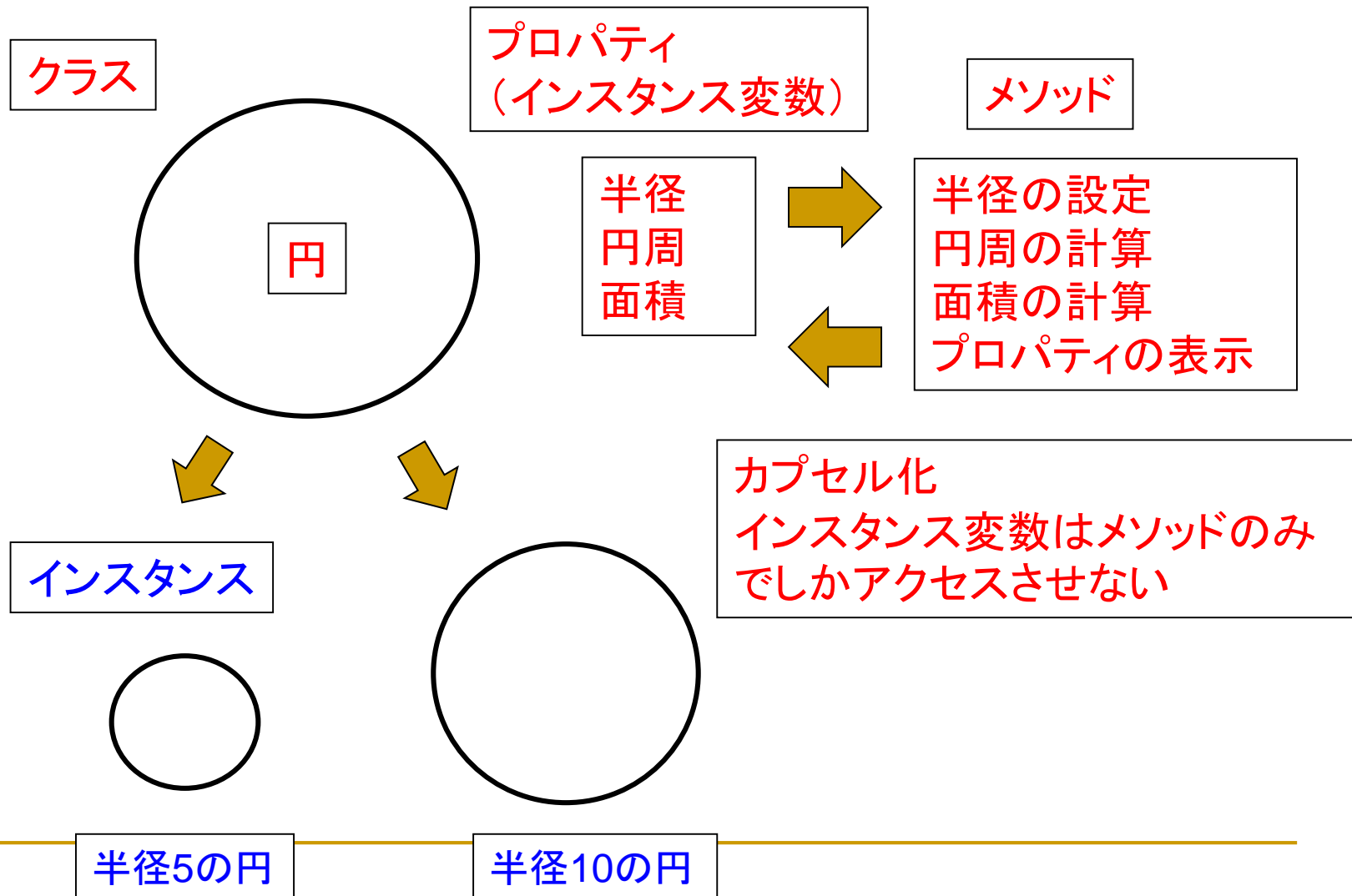
```
Circle.DEG
```

- インスタンス変数へのアクセス

インスタンス名.インスタンス変数

```
en.r = 10
```


カプセル化



```
import math
```

circle-2.py

クラスの定義

```
class Circle:
```

```
# コンストラクター
```

```
def __init__( self ):
```

```
    self.r = 0
```

r: 半径

```
    self.l = 0
```

l: 円周

```
    self.s = 0
```

s: 面積

```
# 半径の設定
```

```
def setRadius( self , r ):
```

```
    self.r = r
```

```
# 円周の計算
```

```
def calc_L( self ):
```

```
    self.l = 2 * self.r * math.pi
```

```
# 面積の計算
```

```
def calc_S( self ):
```

```
    self.s = self.r * self.r * math.pi
```

表示

表示用のメソッド(追加)

```
def Print( self ):
    print( " 半径 {0:.2f}の円の円周は {1:.2f}, 面積は {2:.2f}です".format( self.r ,
self.l , self.s ) )
```

コンストラクター

```
en = Circle()
```

半径の設定

```
en.setRadius( 20 )
```

インスタンス変数にアクセスしていない

円周の計算

```
en.calc_L()
```

面積の計算

```
en.calc_S()
```


表示

```
en.Print()
```

表示用のメソッドの呼び出し

実行結果

circle-1.py の実行結果



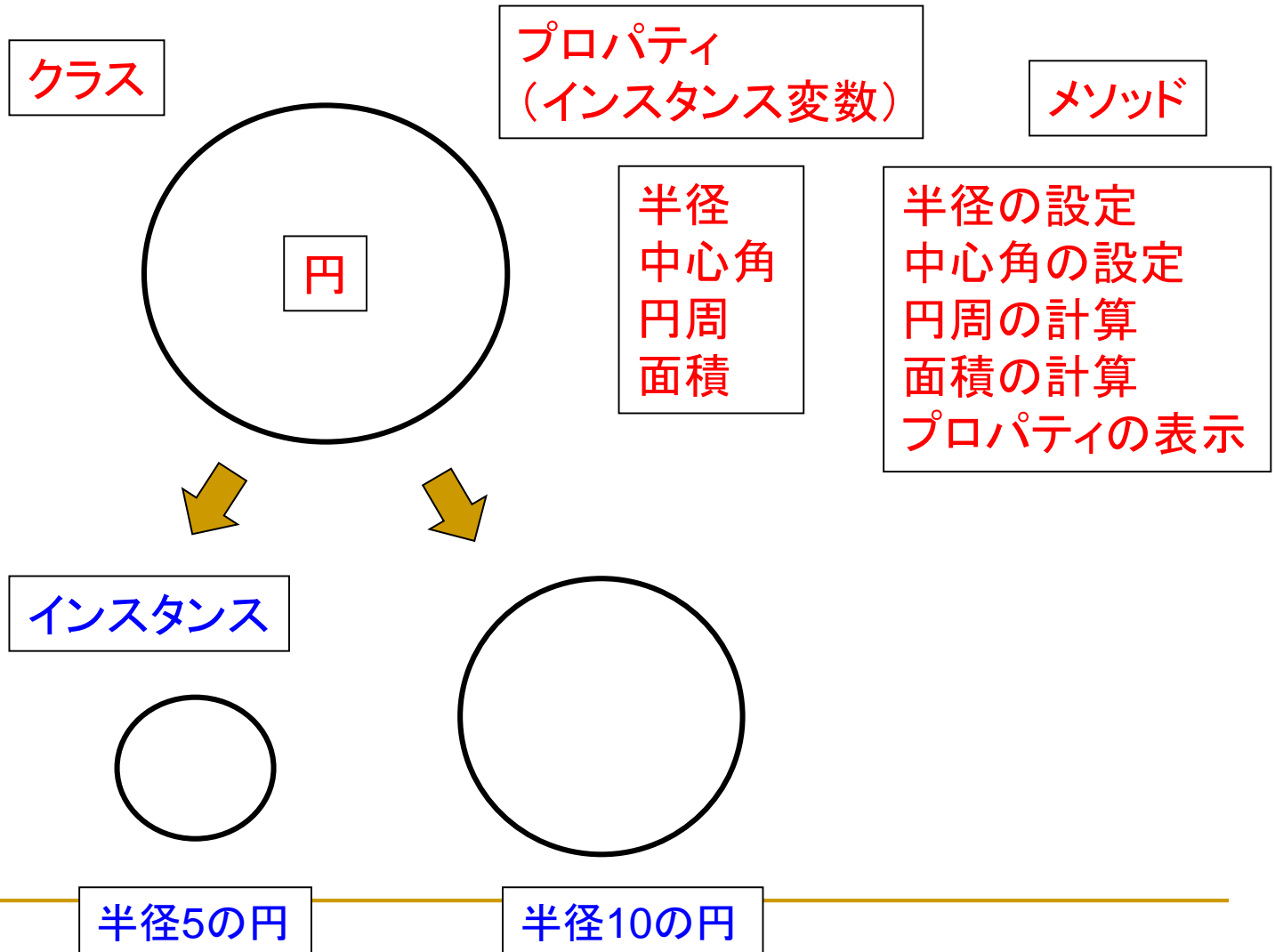
```
C:\Windows\system32\cmd.exe
C:\Users\shino\Desktop>python circle-1.py
半径 5.00の円の円周は 31.42, 面積は 78.54です
半径 10.00の円の円周は 62.83, 面積は 314.16です
```

circle-2.py の実行結果



```
C:\Windows\system32\cmd.exe
C:\Users\shino\Desktop>python circle-2.py
半径 20.00の円の円周は 125.66, 面積は 1256.64です
```

扇形への拡張



クラスの定義 (circle-3.py)

```
import math
```

```
class Circle:
```

```
# クラス変数
```

```
DEG = 360
```

クラス変数
クラス共通の変数
全てのインスタンスで同じ値

```
# コンストラクター
```

```
def __init__( self ):
```

```
    self.r = 0
```

```
    self.l = 0
```

```
    self.s = 0
```

```
    self.d = Circle.DEG
```

r: 半径

l: 円周

s: 面積

d: 中心角

Circle.DEG
クラス変数

```
# 半径の設定
```

```
def setRadius( self , r ):
```

```
    self.r = r
```

中心角の設定

```
def setDegree( self , d ):
    self.d = d
```

中心角の設定用のメソッド(追加)

円周の計算

```
def calc_L( self ):
    self.l = 2 * self.r * math.pi * ( self.d / Circle.DEG )
```

Circle.DEG
クラス変数

面積の計算

```
def calc_S( self ):
    self.s = self.r * self.r * math.pi * ( self.d / Circle.DEG )
```

表示

```
def Print( self ):
    if self.d == Circle.DEG:
        print( " 半径 {0:.2f}の円の円周は {1:.2f}, 面積は {2:.2f}です".
              format( self.r , self.l , self.s ) )
```

円の場合

else:

```
    print( " 半径 {0:.2f} 中心角 {1:.2f}の扇形の円弧の長さは {2:.2f}, 面積は
           {3:.2f}です".format( self.r , self.d, self.l , self.s ) )
```

扇形の場合

circle-3.py

コンストラクター

en = Circle()

半径の設定

en.setRadius(20)

中心角の設定

en.setDegree(180)

円周の計算

en.calc_L()

面積の計算

en.calc_S()

表示

en.Print()

>python circle-3.py

半径 20.00 中心角 180.00の扇形の円弧の長さは 62.83, 面積は 628.32です