Web 付録　第２章　Python プログラム

プログラム１　公共財自発的供給メカニズム(VCM)

以下のプログラムは，公共財自発的供給メカニズムに従って，公共財の供給水準などを求めるプログラムです。

最初にプレーヤー数や各プレーヤーの初期保有を設定します。

次に，各プレーヤーの公共財への貢献額をリスト x に設定します。ここでは**例 2.3** に従って，プレーヤー1 が初期保有から 5 単位，プレーヤー2 が 4 単位，プレーヤー3 が 0 単位，プレーヤー4 が 8 単位，それぞれ公共財に貢献した場合に設定しています。その後，プレーヤーの貢献額を収めたリスト x を，合計を求める関数 sum に代入して公共財供給水準 G を決定し，それを画面表示します。

それから，各プレーヤーの利得を順番に計算し，それを画面表示します。利得は，初期保有から貢献額を引き，そこに公共財から得られる便益を足して得られます。この例では 1 単位の公共財供給から得られる限界便益の値を 0.6 に設定しています。

なお，Python のプログラムではプレーヤー番号が 0 から開始するので，画面表示する際には 1 から開始するようにするため，i+1 としていることに注意してください。

```
# Voluntary contribution mechanism
print('公共財自発的供給メカニズム')
print()

# プレーヤーの数：
N = 4

# 各プレーヤーの初期保有：
w = [10, 10, 10, 10]

# 各プレーヤーの貢献額：
x = [5, 4, 0, 8]
for i in range(N):
    print('プレーヤー',i+1,'の貢献額 = ',x[i])
print()

# 公共財供給水準：
G = sum(x)
```

```
print('公共財供給水準 = ',G)
print()

# 公共財からの限界便益：
a = 0.6

# 各プレーヤーの利得：
payoff = [0]*N
for i in range(N):
    payoff[i] = w[i]-x[i]+a*G
    print('プレーヤー',i+1,'の利得 = ',payoff[i])
```

このプログラムを実行すると，以下の結果が画面に出力されます。**例 2.3** と同じ結果になっていることを確かめてみてください。

```
公共財自発的供給メカニズム

プレーヤー 1 の貢献額 =  5
プレーヤー 2 の貢献額 =  4
プレーヤー 3 の貢献額 =  0
プレーヤー 4 の貢献額 =  8

公共財供給水準 =  17

プレーヤー 1 の利得 =  15.2
プレーヤー 2 の利得 =  16.2
プレーヤー 3 の利得 =  20.2
プレーヤー 4 の利得 =  12.2
```

プログラム 2　リンダール・メカニズム

以下のプログラムは，リンダール・メカニズムに従って，公共財の供給水準などを求めるプログラムです。

最初にプレーヤー数を設定します。

次に，各プレーヤーが G の水準の公共財から得る便益 v は，本文の例のとおり 2 次関数とし，各プレーヤーにとってその係数を左から順にリスト a に設定します。ここでは**例 2.5** に従って，各プレーヤーともに係数の値は 1 に設定しています。

それから，プレーヤーの便益関数の係数を収めたリスト a を，合計を求める関数 sum に代入して公共財供給水準 G を決定し，それを画面表示します。また，各プレーヤーの費用負担額をリスト q に設定し，それを画面表示します。

それから，各プレーヤーの利得を順番に計算し，それを画面表示します。利得は，公共財の便益から費用負担額を引いた値になります。

なお，Python のプログラムではプレーヤー番号が 0 から開始するので，画面表示する際には 1 から開始するようにするため，i+1 としていることに注意してください。

```
# Lindhal mechanism
print('リンダール・メカニズム')
print()

# プレーヤーの数：
N = 2

# 公共財からの便益：
# v = G-0.5*a[i]*G**2
# 便益関数のパラメータ：
a = [1,1]

# 公共財供給水準：
G = (N-1)/sum(a)
print('公共財供給水準 = ',G)
print()

# 費用負担割合
q = [0]*N
for i in range(N):
    q[i]=1-G*a[i]
    print('プレーヤー',i+1,'の費用負担割合 = ',q[i])

# 各プレーヤーの利得
payoff = [0]*N
for i in range(N):
    payoff[i] = G-0.5*a[i]*G**2-q[i]*G
    print('プレーヤー',i+1,'の利得 = ',payoff[i])
```

このプログラムを実行すると，以下の結果が画面に出力されます。**例 2.5** と同じ結果になっていることを確かめてみてください。

```
リンダール・メカニズム

公共財供給水準 =  0.5

プレーヤー 1 の費用負担割合 =  0.5
プレーヤー 2 の費用負担割合 =  0.5
プレーヤー 1 の利得 =  0.125
プレーヤー 2 の利得 =  0.125
```