Décomposition de la monnaie [ss05] - Exercice

Karine Zampieri, Stéphane Rivière

Unisciel algoprog Université Haute-Alsace

Version 16 mai 2018

Table des matières

Python - Décomposition de la monnaie (Solution)

Mots-Clés Algorithmes paramétrés ■
Requis Structures de base, Structures conditionnelles ■
Difficulté • ○ ○

Objectif
Cet exercice reprend l'exercice @[Décomposition de la monnaie] en réalisant une décomposition en procédures et fonctions et traite le cas des coupures nulles.
(image : google/images)

1 Décomposition de la monnaie / pgmonnaie

1.1 Décomposition du problème

L'exercice @[Décomposition de la monnaie] décompose une somme d'argent (en euros) en son équivalent minimal en billets de 100 €, 50 €, 10 €, et de pièces de 2 € et 1 €.

Programme initial

```python
def PGMonnaie1():
    somme = int(input("Somme a decomposer? "))
    b100 = somme // 100
    r100 = somme % 100
    print("==> {:4d} billet(s) de 100 euros, reste {:d}".format(b100, r100))
    b50 = r100 // 50
    r50 = r100 % 50
    print("==> {:4d} billet(s) de  50 euros, reste {:d}".format(b50, r50))
    b10 = r50 // 10
    r10 = r50 % 10
    print("==> {:4d} billet(s) de  10 euros, reste {:d}".format(b10, r10))
    p2 = r10 // 2
    r2 = r10 % 2
    print("==> {:4d} pieces(s) de   2 euros, reste {:d}".format(p2, r2))
    p1 = r2
    print("==> {:4d} pieces(s) de   1 euro".format(p1))

PGMonnaie1()
```

Exemple d'exécution
(Avec cet algorithme)

```
Somme à décomposer? 1254
==>  12 billet(s) de 100 euros, reste 54
==>   1 billet(s) de  50 euros, reste 4
==>   0 billet(s) de  10 euros, reste 4
==>   2 piece(s) de 2 euros, reste 0
==>   0 piece(s) de 1 euro
```

Proposez une décomposition en procédures et fonctions en écrivant les profils de ces dernières.

Aide simple
Constatez que trois lignes se reproduisent.

Solution simple
Les trois lignes qui se reproduisent ont la forme suivante : Une procédure `calcAfficher(somme,valeur,nc,` permettra de faire les calculs et l'affichage.

1.2 Procédures et Programme

Écrivez une procédure `calcAfficher(somme,valeur,nc,rt,typec)` qui, pour une somme d'argent `somme` (entier) et une valeur de coupures `valeur` (entier), calcule le nombre de coupures dans `c` (entier), le reste à décomposer dans `rt` (entier) et affiche le résultat suivant (où `[x]` désigne le contenu de `x`) :

```
==> [nc] [typec] de [valeur] euros, reste [rt]
```

Le type de la coupure est défini par le caractère `typec` qui vaut `'b'` (billet) ou `'p'` (pièce). Dans le cas où `nc` est nul, n'affichez pas le texte.

Solution Paramètres

Entrants : `somme`, `valeur`, `typec`

Sortants : `nc`, `rt`

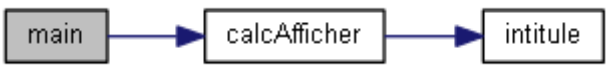
Validez votre procédure avec la solution.

Solution Python @[pgmonnaie.py]

```python
def calcAfficher(somme, valeur, typec):
    """  Calcul et affichage de la coupure

     :param somme: somme à décomposer
     :param valeur: valeur de la coupure
     :param typec: type de la coupure
     :return: le tuple (nombre de coupures, reste)
    """
    nc = somme // valeur
    rt = somme % valeur
    intitule = ("billet" if typec == "b" else "piece")
    if nc != 0:
        print("==> ", nc, " ", intitule, "(s) de ", valeur, " euros, reste ", rt, sep="")
    return (nc, rt)
```

Ré-écrivez un script en utilisant la procédure `calcAfficher`.



Aide simple

Pour calculer le nombre de pièces de 1 euro, ajoutez une variable `r1` pour l'appel.

Testez. Exemple d'exécution :

```
Somme à décomposer? 1254
==>  12 billet(s) de 100 euros, reste 54
==>   1 billet(s) de  50 euros, reste 4
==>   2 piece(s) de 2 euros, reste 0
```

Validez votre script avec la solution.

Solution Python @[pgmonnaie.py]

```python
def PGMonnaie():
    somme = int(input("Somme a decomposer? "))

    b100, r100 = calcAfficher(somme, 100, "b")
    b50, r50 = calcAfficher(r100, 50, "b")
    b10, r10 = calcAfficher(r50, 10, "b")
    p2, r2 = calcAfficher(r10, 2, "p")
    p1, r1 = calcAfficher(r2, 1, "p")

    rs = 100 * b100 + 50 * b50 + 10 * b10 + 2 * p2 + 1 * p1
    print("==> ", rs, sep="")
```

2 Références générales

Comprend [Rohaut-JV1 :c4 :xm] ■