

Devoir Surveillé

Durée : 2h30

Tout document autorisé - Communication interdite

Eric Berthomier - eric.berthomier@free.fr



15 décembre 2014

Préambule

On considèrera les saisies utilisateurs correctes.

Exercice 1 : entiers.py (2 points)

Écrire un programme qui demande la saisie de 2 entiers.

- Si le premier est plus petit que le second, afficher le message "Up".
- Si le second est plus petit que le premier, afficher le message "Down".
- Si les deux nombres sont égaux, afficher le message "Equal".

```
#!/usr/bin/python3
# -*- coding : UTF-8 -*-

# Saisie des 2 entiers
entier1 = int (input("Entier 1 : "))
entier2 = int (input("Entier 2 : "))

if ( entier1 > entier2 ) :
    print ("Up")
elif ( entier2 > entier1 ) :
    print ("Down")
else :
    print ("Equal")
```

Exercice 2 : mois.py (2 points)

Écrire un programme qui saisit 1 entier et qui donne son équivalent en mois de l'année.

Exemple : 1 → Janvier

```
#!/usr/bin/python3
# -*- coding : UTF-8 -*-

# Liste des mois
liste_mois = ["Janvier", "Février", "Mars", "Avril", "Mai", "Juin", "Juillet", "Aout", "Septembre", "Octobre", "Novembre", "Décembre"]

# Saisie du numéro de mois
num_mois = int(input ("Votre mois : "))

# Affichage du mois
print ("Mois ", num_mois, " : ", liste_mois[num_mois-1])
```

Exercice 3 : triangle.py (3 points)

Écrire un programme qui pour une valeur de i choisie au clavier, affiche i fois la valeur i sur la i ème ligne.

Pour $i=5$, on a par exemple l'affichage suivant :

```
1
22
333
4444
55555
```

Aide

👉 `print ("1",end='')` permet de ne pas sauter de ligne à la fin du print.

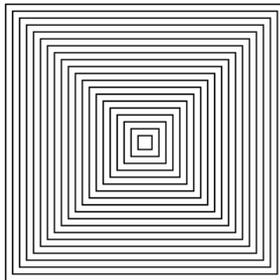
```
#!/usr/bin/python3
# -*- coding : UTF-8 -*-

# Saisie du numéro de mois
num_iteration = int(input ("Chiffre choisi : "))

# Double boucle itérative
for i in range (num_iteration) :
    for j in range (i+1) :
        print (j+1,end='')
    print ()
```

Exercice 4 : carres_russes.py (5 points)

À l'aide du module Turtle et en utilisant l'orienté objet de ce module, dessiner la figure ci-dessous.



0.1 Spécifications

- La fenêtre utilisée pour l'affichage sera de la taille suivante : 800x600px
- Le nombre de carrés est de 20
- La taille du côté du premier carré est de 10
- La taille des carrés suivants augmente de 10 à chaque fois (10, 20, 30, 40 ...)

```
#!/usr/bin/python3
# -*- coding : UTF-8 -*-

# Import de la librairie Turtle
import turtle

# Définition d'une fonction Carré
# Trace un carré de côté "cote" dont le coin bas gauche est en position (x,y)
def carre (tortue, cote, x, y) :

    # Déplacement de la tortue
    tortue.up()
    tortue.goto(x,y)
```

```

tortue.down()

# Tracé du carré
for i in range (4) :
    tortue.forward (cote)
    tortue.left (90)

# Définition de la fenêtre d'affichage
turtle.setup (800,600) # Taille du canevas
wn = turtle.Screen () # La fenêtre d'écran de Turtle
wn.title ("Black_Cube") # Définit un titre

angelo= turtle.Turtle() # Instance d'un objet "Turtle"
angelo.shape ("turtle") # Choix de la forme d'angelo
angelo.speed (10)

for i in range (20) :
    carre (angelo, 10 + i * 10, -i*5, -i*5)

# On montre notre oeuvre sans la tortue
angelo.hideturtle()

wn.exitonclick() # On attend un clic sur la croix

```

Exercice 5 : Volet roulant électrique : volet_roulant.py (10 points)

0.2 Spécifications

- La taille du volet sera modélisée par la valeur : `lg_tablier = 100`
- Le moteur connaît 2 vitesses : +1 / +5

0.3 Travail demandé

1. Écrire une fonction qui initialise la position du volet roulant.
 Cette fonction recevra 1 paramètre :
 - haut : la position du volet roulant est fixée à 0
 - bas : la position du volet roulant est fixée à `lg_tablier`
 - autre chose que "haut" ou "bas" : la position du volet roulant est fixée à -1 et un message d'erreur apparaît

Attention

 Pour les 2 questions suivantes, on prendra en compte le dépassement de fin de course du volet en le signalant avec un message.

L'apparition d'un tel message sera considéré comme une erreur de programmation.

2. Écrire une fonction bas qui baisse le volet roulant en fonction de la vitesse. Cette fonction recevra 3 paramètres : position courante, nombre d'impulsions et vitesse. Elle retournera la nouvelle position du volet roulant.
 Exemple :
 - 5 impulsions à la vitesse 1 = descente de 5
 - 3 impulsions à la vitesse 5 = descente de 15
3. En vous inspirant de la fonction bas, écrire une fonction haut qui remonte le volet roulant en fonction de la vitesse.
4. Écrire deux fonctions fermer() et ouvert() qui respectivement ferme et ouvre le volet **le plus rapidement possible** : le moteur réduira sa vitesse quand il arrive en haut ou en bas.
5. Tester vos fonctions au travers d'un programme. Vous afficherez à chaque fois la nouvelle position du volet.

Précisions

 Une gestion plus fine des mises en sécurité devrait être réalisée pour les fonctions "Haut" et "Bas" mais ne rentre pas dans le cadre de cette correction.

Pour les faire, regarder les fonctions ouvrir_volet et fermer_volet

```

#!/usr/bin/python3
# -*- coding : UTF-8 -*-

lg_tablier = 100
debug=1

def initialise_position_volet (pos) :
    if (pos == "haut") :
        return 0
    elif (pos == "bas") :
        return lg_tablier
    else :
        print ("=====")
        print ("Erreur Initialisation")
        print ("=====")
        return (-1)

def bas (position_courante, nbre_impulsions, vitesse) :
    if (debug) :
        print ("-> Bas : ", nbre_impulsions, " * ", vitesse)

    if ( position_courante + nbre_impulsions * vitesse > lg_tablier ) :
        print ("=====")
        print ("Sécurité volet Bas")
        print ("=====")
        return (lg_tablier)
    else :
        return (position_courante + nbre_impulsions * vitesse)

def haut (position_courante, nbre_impulsions, vitesse) :
    if (debug) :
        print ("-> Haut : ", nbre_impulsions, " * ", vitesse)

    if ( position_courante - nbre_impulsions * vitesse < 0 ) :
        print ("=====")
        print ("Sécurité volet Haut")
        print ("=====")
        return (0)
    else :
        return (position_courante - nbre_impulsions * vitesse)

def fermer_volet (position_courante) :
    ecart = lg_tablier - position_courante
    pas_gde_vitesse = int (ecart / 5)
    pas_petite_vitesse = ecart % 5

    if (debug) :
        print ("Ecart : ",ecart)
        print ("Pas grande vitesse : ",pas_gde_vitesse )
        print ("Pas petite vitesse : ",pas_petite_vitesse )

    position_courante = bas (position_courante, pas_gde_vitesse , 5)
    position_courante = bas (position_courante, pas_petite_vitesse , 1)

    return (position_courante)

def ouvrir_volet (position_courante) :
    pas_gde_vitesse = int (position_courante / 5)
    pas_petite_vitesse = position_courante % 5

    if (debug) :
        print ("Ecart : ",position)
        print ("Pas grande vitesse : ",pas_gde_vitesse )
        print ("Pas petite vitesse : ",pas_petite_vitesse )

    position_courante = haut (position_courante, pas_gde_vitesse , 5)
    position_courante = haut (position_courante, pas_petite_vitesse , 1)
    return (position_courante)

# Demande des instructions à l'utilisateurs
# L'initialisation ne se faisant qu'une seule fois, elle n'apparaît pas dans le menu
# 0 pour sortir
def menu () :

```

```

print ("\nPosition actuelle du volet : ",position, " / ", lg_tablier, "\n")
print ("1 - Haut - Vitesse lente")
print ("2 - Haut - Vitesse rapide")
print ("-")
print ("3 - Bas - Vitesse lente")
print ("4 - Bas Vitesse rapide")
print ("-")
print ("5 - Ouverture volet")
print ("6 - Fermeture volet")
print ("-")
print ("0 - Sortir")
print ("")

choix = int(input ("Votre choix : "))
return (choix)

choix=1
position=initialise_position_volet ("haut")

while (choix) :

    choix=menu()

    if ((choix >=1 ) and (choix <=4)) :
        impuls = int(input ("Nombre d'impulsions ? "))

    if (choix == 1) :
        position = haut (position, impuls, 1)

    if (choix == 2) :
        position = haut (position, impuls, 5)

    if (choix == 3) :
        position = bas (position, impuls, 1)

    if (choix == 4) :
        position = bas (position, impuls, 5)

    if (choix == 5) :
        position = ouvrir_volet (position)

    if (choix == 6) :
        position = fermer_volet (position)

```

6. Afin de protéger les données liées au volet de toute erreur de programmation, on désire transformer ce programme en Objet. Construisez les différentes entêtes des fonctions qui composeraient la classe CVoletRoulant et écrire la méthode (fonction) `__init__`. Expliquez vos choix...

Précisions

 Il n'était pas demandé de réaliser le programme. Celui-ci est présenté pour information et valeur pédagogique.

```

#!/usr/bin/python3
# -*- coding : UTF-8 -*-

class VoletRoulant () :

    # On ne connaît pas l'état du volet donc -1
    def __init__(self,debug) :
        self.position=-1
        self.lg_tablier = 100
        self.debug=debug
        return

    def initialise_position_volet (self,pos) :
        if (pos == "haut") :
            self.position = 0
        elif (pos == "bas") :
            self.position = self.lg_tablier
        else :
            print ("=====")
            print ("Erreur Initialisation")

```

```

        print ("=====")
        return (-1)

def bas (self, nbre_impulsions, vitesse) :
    if (self.debug) :
        print ("-> Bas : ", nbre_impulsions, " * ", vitesse)

    if ( self.position + nbre_impulsions * vitesse > self.lg_tablier ) :
        print ("=====")
        print ("Sécurité volet Bas")
        print ("=====")
        self.position = self.lg_tablier
    else :
        self.position += nbre_impulsions * vitesse

def haut (self, nbre_impulsions, vitesse) :
    if (self.debug) :
        print ("-> Haut : ", nbre_impulsions, " * ", vitesse)

    if ( self.position - nbre_impulsions * vitesse < 0 ) :
        print ("=====")
        print ("Sécurité volet Haut")
        print ("=====")
        self.position = 0
    else :
        self.position -= nbre_impulsions * vitesse

def fermer_volet (self) :
    ecart = self.lg_tablier - self.position
    pas_gde_vitesse = int (ecart / 5)
    pas_petite_vitesse = ecart % 5

    if (self.debug) :
        print ("Ecart : ",ecart)
        print ("Pas grande vitesse : ",pas_gde_vitesse )
        print ("Pas petite vitesse : ",pas_petite_vitesse )

    self.bas (pas_gde_vitesse , 5)
    self.bas (pas_petite_vitesse , 1)

def ouvrir_volet (self) :
    pas_gde_vitesse = int (self.position / 5)
    pas_petite_vitesse = self.position % 5

    if (self.debug) :
        print ("Ecart : ",self.position )
        print ("Pas grande vitesse : ",pas_gde_vitesse )
        print ("Pas petite vitesse : ",pas_petite_vitesse )

    self.haut (pas_gde_vitesse , 5)
    self.haut (pas_petite_vitesse , 1)

def affiche_position (self) :
    print ("Position actuelle : ", self.position, " / ", self.lg_tablier)

```

```

#!/usr/bin/python3
# -*- coding : UTF-8 -*-

import CVoletRoulant

voletroulant=CVoletRoulant.VoletRoulant(1)

# Demande des instructions à l'utilisateur
# L'initialisation ne se faisant qu'une seule fois, elle n'apparaît pas dans le menu
# 0 pour sortir
def menu () :
    print ("1 - Haut - Vitesse lente")
    print ("2 - Haut - Vitesse rapide")
    print ("-")
    print ("3 - Bas - Vitesse lente")
    print ("4 - Bas Vitesse rapide")
    print ("-")
    print ("5 - Ouverture volet")
    print ("6 - Fermeture volet")

```

```

print ("-")
print ("0 - Sortir")
print ("")

choix = int(input ("Votre choix : "))
return (choix)

choix=1
voletroulant.initialise_position_volet ("haut")

while (choix) :

    voletroulant.affiche_position ()
    print ()
    choix=menu()

    if ((choix >=1 ) and (choix <=4)) :
        impuls = int(input ("Nombre d'impulsions ? "))

    if (choix == 1) :
        voletroulant.haut(impuls, 1)

    if (choix == 2) :
        voletroulant.haut(impuls, 5)

    if (choix == 3) :
        voletroulant.bas(impuls, 1)

    if (choix == 4) :
        voletroulant.bas(impuls, 5)

    if (choix == 5) :
        voletroulant.ouvrir_volet ()

    if (choix == 6) :
        voletroulant.fermer_volet ()

```

7. On souhaite adapter des capteurs thermiques aux volets, quelle fonctionnalité de l'objet allez vous utiliser ?
 L'héritage (Dérivation). On dérivera la classe VoletRoulant pour y ajouter les fonctionnalités liées aux capteurs.