

Devoir Surveillé

Durée : 2h30

Tout document autorisé - Communication interdite

Eric Berthomier - eric.berthomier@free.fr



17 décembre 2014

Préambule

On considérera les saisies utilisateurs correctes.

Exercice 1 : entiers.py (1 point)

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 : jours.py (2 points)

Écrire un programme qui saisit 1 entier et qui donne son équivalent en jour de la semaine.

Exemple : 1 → Lundi

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

# Liste des jours
liste_jours = ["Dimanche", "Lundi", "Mardi", "Mercredi", "Jeudi", "Vendredi", "Samedi", "Dimanche"]

# Saisie du numéro de jour
num_jour = int(input ("Votre jour : "))

# Affichage du jour
print ("Jour ", num_jour, " : ", liste_jours[num_jour])
```


Exercice 3 : triangle_inverse.py (4 points)

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

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 chiffre
num_iteration = int(input ("Chiffre choisi : "))

# Double boucle itérative
for i in range (1,num_iteration+1) :

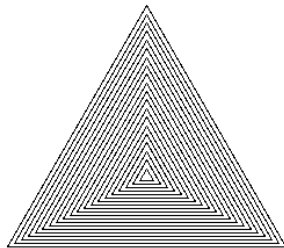
    # Affichage des blancs
    for j in range (num_iteration -i) :
        print (" ",end='')

    # Affichage du numéro
    for j in range (i) :
        print (i,end='')

    print ()
```

Exercice 4 : Triangles Russes (5 points)

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



Spécifications

- La fenêtre utilisée pour l’affichage sera de la taille suivante : 800x600px
- Le nombre de triangles est de 20
- La taille du côté du premier triangle est de 12 pixels
- La taille des triangles suivants augmente de 12 pixels à chaque fois (12, 24, 36, 48 ...)
- Pour conserver la géométrie, le point bas gauche du triangle doit se déplacer à chaque fois de 6 pixels sur l’axe des x et de 3 pixels sur l’axe des y .

Corrigé

```

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

import turtle

def triangle_equilateral (tortue, x, y, taille) :
    kame.up()
    kame.goto(x,y)
    kame.down()
    kame.forward (taille)
    kame.left(120)
    kame.forward(taille)
    kame.left(120)
    kame.forward(taille)
    kame.left(120)

#Dimensions de la fenetre
turtle.setup(800,600)

#Nom de la tortue
kame=turtle.Turtle()

x=-6
y=-3
cote=12

#20 Realisations de la figure

for i in range (20) :
    triangle_equilateral (kame, x, y, cote)

    # increment des dimensions
    x-=6
    y-=3
    cote+=12

#sortie du logiciel par un clic
turtle.exitonclick()

```

Exercice 5 : Moteur de portail électrique : portail.py (10 points)

Spécifications

- La taille de la barrière sera modélisée par la valeur : `lg_portail = 210`
- Le moteur connaît 2 vitesses : `+1 / +5`

Travail demandé

1. Écrire une fonction qui initialise la position du portail.

Cette fonction recevra 1 paramètre :

- ouvert : la position du portail est fixée à 0
- ferme : la position du portail est fixée à `lg_portail`
- autre que "ouvert" ou "ferme" : la position du portail 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 portail (pour ne pas démolir le mur).

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

2. Écrire une fonction avance qui avance le portail en fonction de la vitesse. Cette fonction recevra 3 paramètres : position courante, nombre d'impulsions et vitesse. Elle retournera la nouvelle position du portail.


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 avance, écrire une fonction recule qui recule le portail en fonction de la vitesse.

4. Écrire deux fonctions `fermer_portail()` et `ouvrir_portail()` qui respectivement ferme et ouvre le portail **le plus rapidement possible** : le moteur réduira sa vitesse quand il arrive en début de course ou en fin de course.
5. Tester vos fonctions au travers d'un programme. Vous afficherez à chaque fois la nouvelle position du portail.

Précisions

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

Pour les faire, regarder les fonctions `ouvrir_portail` et `fermer_portail`

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

lg_portail = 210
debug=1

def initialise_position_portail (pos) :
    if (pos == "ouvert") :
        return 0
    elif (pos == "ferme") :
        return lg_portail
    else :
        print ("=====")
        print ("Erreur Initialisation")
        print ("=====")
        return (-1)

def avance (position_courante, nbre_impulsions, vitesse) :
    if (debug) :
        print ("-> Avance : ", nbre_impulsions, " * ", vitesse)

    if ( position_courante + nbre_impulsions * vitesse > lg_portail ) :
        print ("=====")
        print ("Sécurité Fermeture Portail")
        print ("=====")
        return (lg_portail)
    else :
        return (position_courante + nbre_impulsions * vitesse)

def recule (position_courante, nbre_impulsions, vitesse) :
    if (debug) :
        print ("-> Recule : ", nbre_impulsions, " * ", vitesse)

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

def fermer_portail (position_courante) :
    ecart = lg_portail - 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 = avance (position_courante, pas_gde_vitesse , 5)
    position_courante = avance (position_courante, pas_petite_vitesse , 1)

    return (position_courante)

def ouvrir_portail (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 = recule (position_courante, pas_gde_vitesse , 5)
position_courante = recule (position_courante, pas_petite_vitesse , 1)
return (position_courante)

# 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 ("\nPosition actuelle du portail : ",position, " / ", lg_portail, "\n")
    print ("1 - Avance - Vitesse lente")
    print ("2 - Avance - Vitesse rapide")
    print ("-")
    print ("3 - Recule - Vitesse lente")
    print ("4 - Recule - Vitesse rapide")
    print ("-")
    print ("5 - Ouverture portail")
    print ("6 - Fermeture portail")
    print ("-")
    print ("0 - Sortir")
    print ("")

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

choix=1
position=initialise_position_portail ("ferme")

while (choix) :

    choix=menu()

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

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

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

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

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

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

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

```

6. Afin de protéger les données liées au portail de toute erreur de programmation, on désire transformer ce programme en Objet.

Construisez les différentes **entêtes** des fonctions qui composeraient la classe CPortail 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 Portail () :

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

```

```

self.lg_portail = 210
self.debug=debug
return

def initialise_position_portail (self,pos) :
    if (pos == "ouvert") :
        self.position = 0
    elif (pos == "fermer") :
        self.position = self.lg_portail
    else :
        print ("=====")
        print ("Erreur Initialisation")
        print ("=====")
        return (-1)

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

    if ( self.position + nbre_impulsions * vitesse > self.lg_portail ) :
        print ("=====")
        print ("Sécurité Fermeture Portail")
        print ("=====")
        self.position = self.lg_portail
    else :
        self.position += nbre_impulsions * vitesse

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

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

def fermer_portail (self) :
    ecart = self.lg_portail - 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.avance (pas_gde_vitesse , 5)
    self.avance (pas_petite_vitesse , 1)

def ouvrir_portail (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.recule (pas_gde_vitesse , 5)
    self.recule (pas_petite_vitesse , 1)

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

```

```

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

import CPortail

portail=CPortail.Portail(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 - Avance - Vitesse lente")
    print ("2 - Avance - Vitesse rapide")
    print ("-")
    print ("3 - Recule - Vitesse lente")
    print ("4 - Recule - Vitesse rapide")
    print ("-")
    print ("5 - Ouverture portail")
    print ("6 - Fermeture portail")
    print ("-")
    print ("0 - Sortir")
    print ("")

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

choix=1
portail.initialise_position_portail ("ouvert")

while (choix) :

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

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

    if (choix == 1) :
        portail.avance(impuls, 1)

    if (choix == 2) :
        portail.avance(impuls, 5)

    if (choix == 3) :
        portail.recule(impuls, 1)

    if (choix == 4) :
        portail.recule(impuls, 5)

    if (choix == 5) :
        portail.ouvrir_portail ()

    if (choix == 6) :
        portail.fermer_portail ()

```

7. On souhaite adapter des capteurs de présence au moteur du portail, quelle fonctionnalité de l'objet allez vous utiliser ?
L'Héritage (Dérivation). On dérivera la classe Portail pour y ajouter les fonctionnalités liées aux capteurs.