



BENEMERITA UNIVERSIDAD AUTONOMA DE PUEBLA

David Alvarez Vargas

Programación Paralela Y Concurrente

clase archivo

Python

Profesor: Adolfo Aguilar Rico

Clase archivo

Realizar un programa que contenga la clase Archivo, con los métodos

- I. Cuenta vocales
- II. Cuenta consonantes
- III. Cuenta signos de puntuación
- IV. Cuenta espacios
- V. Cuenta palabras
- VI. Cuenta líneas
- VII. Mayúsculas
- VIII. Minúsculas
- IX. Copia archivo
- X. Convierte a mayúsculas
- XI. Convierte a minúsculas
- XII. Muestra en hexadecimal

Desarrollo

```
import shutil, os

from sys import exit

class Archivo:
    def __init__(self, nombre):
        try:
            self.f = open(nombre, 'r')
            self.nombre = nombre
        except:
            print("No se puede abrir el archivo", nombre)
            exit()

    def muestra(self):
        i=1
        for linea in self.f:
            print( "{:3}{}".format(i, linea), end="")
            i+=1
        self.f.seek(0)
```

Primero creamos la clase principal, en este caso como trabajamos con un archivo, a la clase la nombramos como **class Archivo**, ahí es en donde se encontrará el texto para que se lleve a cabo la realización del programa.

La función **def muestra(self)** mostrará el texto que tiene nuestro archivo.

```

def cuentaVocales(self):
    def vocales(s):
        contador = 0
        for i in range(len(s)):
            if s[i] in set("aeiouáéíóúAEIOU"):
                contador +=1
        return contador

    contador = 0
    for linea in self.f:
        contador +=vocales(linea)
    self.f.seek(0)
    return contador

def cuentaConsonantes(self):
    def consonantes(s):
        contador = 0
        for i in range(len(s)):
            if s[i].lower() in set("bcdfghjklmnpqrstvwxyzBCDFGHJKLMNPQRSTVWXYZ"):
                contador += 1
        return contador

    contador = 0
    for linea in self.f:
        contador +=consonantes(linea)
    self.f.seek(0)
    return contador

```

- I. **def cuentaVocales:** Esta función realizará el conteo de todas las vocales que se encuentran en el archivo de texto. También utilizamos un if que evalué el set que es la estructura de datos ,para este caso las vocales. Necesitamos de un contador que inicie en cero, y vaya sumando 1 cada vez que encuentre una vocal y al final nos regrese el número total de las vocales.

- II. **def cuentaConsonantes:** Esta función realizará el conteo de las consonantes que se encuentren en el archivo de texto. También utilizamos un if que evalué el set que es la estructura de datos, para las consonantes. Necesitamos de un contador que inicie en cero, y vaya sumando 1 cada vez que encuentre una consonantes en el texto y al final nos regrese el número total de las consonantes.

```

def cuentalaspuntuacion(self):
    def signos(s):
        contador = 0
        for i in range(len(s)):
            if s[i] in set(",;.:...¿?;!()*/*/_-"""):
                contador += 1
        return contador

    contador = 0
    for linea in self.f:
        contador +=signos(linea)
    self.f.seek(0)
    return contador

def cuentaespacio(self):
    def espacio(s):
        contador = 0
        for i in range(len(s)):
            if s[i] in set(" "):
                contador += 1
        return contador

    contador = 0
    for linea in self.f:
        contador +=espacio(linea)
    self.f.seek(0)
    return contador

```

- III. **def cuentalaspuntuacion:** Esta función realizará el conteo signos de puntuación que se encuentren en el archivo de texto. También utilizamos un if que evalué el set que es la estructura de datos, para los signos de puntuación. Necesitamos de un contador que inicie en cero, y vaya sumando 1 cada vez que encuentre un signo de puntuacion y al final nos regrese el número total de los signos de puntuación.
- IV. **def cuentaespacio:** Esta función realizará el conteo de los espacios que se encuentren en el archivo de texto. También utilizamos un if que evalué el set que es la estructura de datos, para los espacios. Se implementa un contador que inicie en cero, y vaya sumando 1 cada vez que encuentre un espacio y al final nos regrese el número total de los espacios.

```

def cuentapalabras(self):
    def palabra(s):
        contador = 0
        for i in range(len(s)):
            if s[i] in set(" \n"):
                contador += 1
        return contador

    contador = 0
    for linea in self.f:
        contador += palabra(linea)
    self.f.seek(0)
    return contador + 1

def cuentalinea(self):
    def lineas(s):
        contador = 0
        for i in range(len(s)):
            if s[i] in set("\n"):
                contador += 1
        return contador

    contador = 1
    for linea in self.f:
        contador += lineas(linea)
    self.f.seek(0)
    return contador

```

- V. **def cuentapalabras:** Esta función realizará el conteo las palabras que se encuentren en el archivo de texto. También utilizamos un if que evalué el set que es la estructura de datos, para las palabras. Se implementa un contador que inicie en cero, y vaya sumando 1 cada vez que encuentre una palabra y al final nos regrese el número total de las palabras.
- VI. **def cuentalinea:** Esta función realizará el conteo de las líneas que se encuentren en el archivo de texto. También utilizamos un if que evalué el set que es la estructura de datos, para las líneas. Se implementa un contador que inicie 1, y vaya sumando 1 cada vez que encuentre una palabra y al final nos regrese el número total de las líneas.

```

def cuentamayusculas(self):
    def mayusculas(s):
        contador = 0
        for i in range(len(s)):
            if s[i] in set("AÂBCDEÊFGHIÎJKLMNOÓPQRSTUÚVWXYZ"):
                contador += 1
        return contador

    contador = 0
    for linea in self.f:
        contador += mayusculas(linea)
    self.f.seek(0)
    return contador

def cuentaminusculas(self):
    def minusculas(s):
        contador = 0
        for i in range(len(s)):
            if s[i] in set("aeiouáéíóúbcd fghjklmnpqrstvwxyz"):
                contador += 1
        return contador

    contador = 0
    for linea in self.f:
        contador += minusculas(linea)
    self.f.seek(0)
    return contador

def copiarch(self, copia):
    shutil.copy(self.f.name, copia)
    print("\n9. _ El archivo se a copiado con el nombre de copia")

def convmayusculas(self):
    contador = ""
    for linea in self.f:
        contador += linea.upper()
    self.f.seek(0)
    return contador

```

- VII. **def cuentamayusculas:** Esta función realizará el conteo de las letras mayúsculas que se encuentren en el archivo de texto. También utilizamos un if que evalué el set que es la estructura de datos, para las letras mayúsculas. Se implementa un contador que inicie 1, y vaya sumando 1 cada vez que encuentre una letra mayúscula y al final nos regrese el número total de las letras mayúsculas.
- VIII. **def cuentaminusculas:** Esta función realizará el conteo de todas las letras minúsculas que se encuentren en el archivo de texto. También utilizamos un if que evalué el set que es la estructura de, para este caso las letras minúsculas. Se implementa un contador que inicie 1, y vaya sumando 1 cada vez que encuentre una letra minúscula y al final nos regrese el número total de las letras minúsculas.
- IX. **def copiarch:** *shutil.copy()* Es un método en Python se utiliza para copiar el archivo original con su contenido, el cual se crea en el mismo directorio donde esta el archivo original, el archivo lo llamaremos "copia".
- X. **def convmayusculas:** En este caso solo se tiene que igualar el contador así: contador +=linea.upper() esto hará que nos imprima el texto en mayúsculas.

```

def convminusculas(self):
    contador = ""
    for linea in self.f:
        contador += linea.lower()
    self.f.seek(0)
    return contador

def hexadecimal(self):
    h = []
    def d(s):
        for i in range(len(s)):
            h.append(hex(ord(s[i])))

    for linea in self.f:
        d(linea)
    print(h)
    self.f.seek(0)

```

- XI. **def convminusculas:** En este caso solo se tiene que igualar el contador así: contador += linea.lower() esto hará que nos imprima el texto en mayúsculas.
- XII. **def hexadecimal:** Definimos la función que hará que todo nuestro documento lo muestre a hexadecimal, ponemos una variable que vaya guardando el arreglo en hexadecimal en este caso es h=[]. Después tenemos que definir las siguientes funciones:
- El método append (): Agrega un elemento al final de la lista.
 - La función hex (): Es una de las funciones que se utiliza para convertir un número entero en su hexadecimal.
 - La ord(): Función devuelve un número entero que representa el carácter Unicode.

Ya una vez teniendo estas funciones el ord nos va a ayudar para que le pasemos el código ascii de cada carácter y a esto se le pasa a la función hex para que lo vaya convirtiendo en hexadecimal.

Al final se manda a imprimir la cadena en hexadecimal.

Menú Principal

```
from claseArchivo import *
#main
nomb = input('Nombre del archivo:')
archivo = Archivo(nomb)
archivo.muestra()
print("\n1._ El numero de vocales es: ", archivo.cuentaVocales())
print("\n2._ El numero de consonantes es: ", archivo.cuentaConsonantes())
print("\n3._ El numero de signos de puntuación es: ", archivo.cuentasigpuntuacion())
print("\n4._ El numero de espacios: ", archivo.cuentaespacio())
print("\n5._ El numero de palabras: ", archivo.cuentapalabras())
print("\n6._ El numero de lineas: ", archivo.cuentaslinea())
print("\n7._ El numero de mayusculas es: ", archivo.cuentamayusculas())
print("\n8._ El numero de minusculas es: ", archivo.cuentaminusculas())
archivo.copiararch("copia.txt")
print("\n10._ Convierte a mayusculas:\n", archivo.convmayusculas())
print("\n11._ Convierte a minusculas:\n", archivo.convminusculas())
print("\n12._ Muestra el texto en hexadecimal")
archivo.hexadecimal()
```

Main

Aquí hacemos el menú principal donde mandamos a imprimir todas las funciones definidas que nos ayudan a hacer lo solicitado.

También se crea la ventana donde nos pide el nombre del archivo donde tenemos el texto

RESULTADOS

```
= RESTART: C:\Users\david\OneDrive\Escritorio\Programacion Paralela\Main Clase Archivo.py
Nombre del archivo:texto.txt
1A lo largo de la historia se han escrito obras literarias.
2Algunas han dejado frases para enmarcar,
3es decir,
4frases de libros escritas por grandes autores.
1._ El numero de vocales es: 54

2._ El numero de consonantes es: 73

3._ El numero de signos de puntuación es: 4

4._ El numero de espacios: 22

5._ El numero de palabras: 26

6._ El numero de lineas: 4

7._ El numero de mayusculas es: 2

8._ El numero de minusculas es: 125

9._ El archivo se a copiado con el nombre de copia

10._ Convierte a mayusculas:
A LO LARGO DE LA HISTORIA SE HAN ESCRITO OBRAS LITERARIAS.
ALGUNAS HAN DEJADO FRASES PARA ENMARCAR,
ES DECIR,
FRASES DE LIBROS ESCRITAS POR GRANDES AUTORES.

11._ Convierte a minusculas:
a lo largo de la historia se han escrito obras literarias.
algunas han dejado frases para enmarcar,
es decir,
frases de libros escritas por grandes autores.

12._ Muestra el texto en hexadecimal
['0x41', '0x20', '0x6c', '0x6f', '0x20', '0x6c', '0x61', '0x72', '0x67', '0x6f', '0x20', '0x64', '0x65', '0x20', '0x6c', '0x61', '0x20', '0x68', '0x69', '0x73', '0x74',
'0x6f', '0x72', '0x69', '0x61', '0x20', '0x73', '0x65', '0x20', '0x68', '0x61', '0x6e', '0x20', '0x65', '0x73', '0x63', '0x72', '0x69', '0x74', '0x6f', '0x20', '0x6f',
', '0x62', '0x72', '0x61', '0x73', '0x20', '0x6c', '0x69', '0x74', '0x65', '0x72', '0x61', '0x72', '0x69', '0x61', '0x73', '0x2e', '0xa', '0x41', '0x6c', '0x67', '0x75',
', '0x6e', '0x61', '0x73', '0x20', '0x68', '0x61', '0x6e', '0x20', '0x64', '0x65', '0x6a', '0x61', '0x64', '0x6f', '0x20', '0x66', '0x72', '0x61', '0x73', '0x65', '0x7',
3', '0x20', '0x70', '0x61', '0x72', '0x61', '0x20', '0x65', '0x6e', '0x6d', '0x61', '0x72', '0x63', '0x61', '0x72', '0x2c', '0xa', '0x65', '0x73', '0x20', '0x64', '0x6',
5', '0x63', '0x69', '0x72', '0x2c', '0xa', '0x66', '0x72', '0x61', '0x73', '0x65', '0x73', '0x20', '0x64', '0x65', '0x20', '0x6c', '0x69', '0x62', '0x72', '0x6f', '0x7',
3', '0x20', '0x65', '0x73', '0x63', '0x72', '0x69', '0x74', '0x61', '0x73', '0x20', '0x70', '0x6f', '0x72', '0x20', '0x67', '0x72', '0x61', '0x6e', '0x64', '0x65', '0x',
73', '0x20', '0x61', '0x75', '0x74', '0x6f', '0x72', '0x65', '0x73', '0x2e']
>>>
```