```python
# -*- coding: utf-8 -*-

#import sys
import tkinter as tk
from tkinter import filedialog as fd
from tkinter import ttk
from tkinter import *

from matplotlib import pyplot as plt

"""
On instancie une class pour y créé des méthodes qui vont nous permettre de
stocker des valeur dans l'IHM et de pouvoir
générer une graphique a partir des données stocké dans l'IHM
"""

class generation_graph:
    """
    On crée une méthode avec le constructeur __init__ ou on va venir créer des
variables que l'on va réutiliser dans d'autres méthodes
    et on va ouvrir une IHM qui va nous permettre d'utiliser les méthodes
grâce à des boutons
    """

    def __init__(self):
        self.element_Abscisse = []
        self.element_ordonnee = []
        self.y = []
        self.x = []
        self.nom_des_colones1 = []
        self.nom_des_colones2 = []

        self.root = Tk()
        self.root.title("génération de graphique ")
        self.root.geometry('400x200+550+200')

        #Le bouton 1 va nous permettre d'executer la methode 'open_text_file'

        self.bouton1 = Button(self.root, text='Click to Open File',
activebackground="red",command=self.open_text_file,font=("taille",12))
        self.bouton1.pack(fill=tk.X)

        #Le label 1 nous permet de savoir a quoi correspond le bouton 2

        self.label1 = Label(self.root, text="éléments Abscisses ",
bg="grey",font=("taille",12))
        self.label1.forget
```

```
        #Le bouton 2 sert a selectionne une colone que l'on utilisera dans la
methode 'graphique'

        self.bouton2 = ttk.Combobox(self.root,
values=self.nom_des_colones1,font=("taille",12))
        self.bouton2.forget

        #Le label 2 nous permet de savoir a quoi correspond le bouton 3

        self.label2 = Label(self.root, text="éléments ordonnée ",
bg="grey",font=("taille",12))
        self.label2.forget

        # Le bouton 3 sert a selectionne une colone que l'on utilisera dans la
methode 'graphique'

        self.bouton3 = ttk.Combobox(self.root,
values=self.nom_des_colones2 ,font=("taille",12))
        self.bouton3.forget

        #Le bouton 5 va nous permettre d'executer la methode 'graphique'

        self.bouton4 = Button(self.root, text="générer le graphique",
command=self.graphique,activebackground="red",font=("taille",12))
        self.bouton4.forget

        self.label_error = Label(self.root, text="aucun fichier sélectionner
ou fichier invalide", fg="orange",bg="black",font=("taille", 15))
        self.label_error.forget

        #Le button_destroy permet d'executer la methode 'quit'

        self.button_destroy =
Button(self.root,text="quitter",command=self.quit,fg="red",bg='light
grey',width=36,activebackground="red",font=("taille",15))
        self.button_destroy.pack(side='bottom')


    """
    Une methode qui permet de détruire la fenetre tkinter , se qui permet
d'arreter le programme
    """

    def quit(self):
       #sys.exit()
       self.root.destroy()
```

```
    """
    On crée une méthode qui a pour nom open_text_file et qui va nous permettre
d'aller chercher un fichier dans
    l'explorateur du fichier.
    après avoir sélectionner un fichier on ouvre le fichier puis  on créer une
boucle while
    qui va nous permettre de récupérer la premiere ligne qui contient les noms
des colones pour les stockés dans une
    variable.
    et pour pouvoir les mettre dans les Combobox que l'on à créer dans la
méthode précédente .
    Une fois que la boucle while est fini on ferme le fichier et on fait
apparaitre les bouton que l'on a créer dans la méthode
    précédente et si aucun fichier n'a était sélectionné alors on affiche un
message d'erreur.
    """

    def open_text_file(self):
        self.name = fd.askopenfilename()#on ouvre l'explorateur de fichier et
on selctionne un fichier
        try: #on essaie d'executer le code mais si on y arrive pas on affiche
un message d'erreur et on fait disparaitre les boutons
            self.fichier = open(self.name, 'r')#on ouvre le fichier
selectionner
            print(self.name)#on ecrit se que contient la variable donc le
chemin du fichier selectionner
            self.elements = list()
            compteur = 0
            while 1:  # La condition est donc toujours True

                ligne = self.fichier.readline()  # On tente de lire la ligne

                if not (ligne):  # Si ligne n'existe pas, elle contient False

                    break  # On sort de la boucle

                else:  # Sinon, c'est que ligne contient quelque chose

                    ligne.replace('\n', '')  # On supprime le passage à la
ligne finale de chaque ligne

                    if compteur == 0:
                        self.elements = ligne.split(';')  # On crée une liste
contenant les valeurs de chaque ligne

                    compteur += 1
            """
            On vient mettre les donnees de elements au bouton 2 et 3
            """
```

```
            self.nom_des_colones1 = self.elements
            self.nom_des_colones2 = self.elements
            self.bouton2.config(values=self.nom_des_colones1)
            self.bouton3.config(values=self.nom_des_colones2)

            self.fichier.close()#on ferme le fichier
            """
            On vient afficher tout les boutons et on desactive le premier
bouton pour eviter de clicker dessus
            """
            self.label1.pack(fill=tk.X)
            self.bouton2.pack(fill=tk.X)
            self.label2.pack(fill=tk.X)
            self.bouton3.pack(fill=tk.X)
            self.bouton4.pack(fill=tk.X)
            self.label_error.forget()
        except:
            "On affiche le message d'erreur et on cache tout les boutons sauf
le 1"
            self.label1.forget()
            self.label2.forget()
            self.bouton2.forget()
            self.bouton3.forget()
            self.bouton4.forget()
            self.label_error.pack()
            print("ahahahahahaha")


    """
    Je créé nouvelle méthode et  qui pour nom 'graphique' et le but de celui-
ci est de générer un graphique à partir des
    colonnes que nous avons sélectionnées et pour cela nous commençons en
récupérant les noms des colonnes que nous avons
    sélectionnées.pour être en mesure de récupérer leur numéro de localisation
par la suite.
    Ensuite, nous allons créer une variable 'Nom_de_fichier' qui sera utilisé
lors de l'enregistrement du graphique.
    puis nous créerons une variable.element_graph_1 et element_graph_2 qui
contiendront le numéro d'emplacement
    de colonne du fichier.pouvoir extraire toutes les données contenues dans
les colones.
    Puis nous ouvrons le fichier que nous avons sélectionné dans la méthode
précédente et puis nous créons une boucle
    while qui va nous permettre de lire tout le fichier ligne par ligne et des
les stockés dans une variable 'élements'.
    Après avoir stocké toutes les données du fichier dans une variable,nous
allons venir stocker les données d'une colonnes
```

```
    que nous avons sélectionné dans une liste  x et.on fait pareil pour
l'autres colones mais cette fois on le
    stocke dans une liste  y .
    Ensuite on va venir convertir la liste x en floats si les données sont
différentes d'une date sinon
    on ne convertie pas la liste x et on fait pareil pour la liste y
    ce qui nous permettra d'avoir les nombres dans l'ordre dans le graphique.
    Après avoir fait sa nous allons fermer le fichier et nous allons générer
le graphique a partir des liste y et x.
    nous plaçons une légende sur le graphe grâce aux variables
"elements_Abscisse" et "elements_ordonnee".
    Enregistre le graphique que nous venons de générer qui aura comme nom la
variable "nom_fichier".
    Et pour finir on clear les liste x et y pour pouvoir refaire un graphique.

    """

    def graphique(self):  # Fonction qui permet de générer le graphique
        """
        on met les donnees que l'on a selectionner dans les boutons 2 et 3
dans deux nouvelles variable pour eviter de
        modifier les donnees des boutons.
        """
        self.element_Abscisse = (self.bouton2.get())
        self.element_ordonnee = (self.bouton3.get())
        """
        On recupere le numero de l'emplacement des colones selctionne pour
pouvoir les retrouver dans le fichier
        """
        element_graph_1 = self.nom_des_colones1.index(self.element_Abscisse)
        element_graph_2 = self.nom_des_colones2.index(self.element_ordonnee)
        """
        On réouvre le fichier pour cette fois lire tout le fichier
        """
        self.fichier = open(self.name, 'r')
        compteur = 0
        while 1:  # La condition est donc toujours True

            ligne = self.fichier.readline()  # On tente de lire la ligne


            if not (ligne):  # Si ligne n'existe pas, elle contient False

                break  # On sort de la boucle

            else:  # Sinon, c'est que ligne contient quelque chose

                ligne.replace('\n', '')  # On supprime le passage à la ligne
finale de chaque ligne
```

```
                self.elements = ligne.split(';')  # On crée une liste
contenant les valeurs de chaque ligne

                """
                On ajoute les colonnes que l'on a sélectionné au liste x et y
si le compteur et différents de 0
                """
                if compteur != 0:
                    self.y.append(self.elements[element_graph_1])
                    self.x.append(self.elements[element_graph_2])

                compteur += 1
        """
        Sa va nous permettre  de convertir les chaînes de caractère en str
sauf les dates
        """
        if self.element_Abscisse != 'DATE':
            self.y = list(map(float, self.y))

        if self.element_ordonnee != 'DATE':
            self.x = list(map(float, self.x))


        self.fichier.close()  # on  ferme le fichier

        """
        On génénere le graphiaue a partir des lis x et y
        """
        plt.plot(self.y, self.x)  # on génére un graphique à partir des listes
        """
        On vient créé une légende avec les variable element_Abscisse et
element_ordonne
        """
        plt.xlabel(self.element_Abscisse)
        plt.ylabel(self.element_ordonnee)

        self.Nom_fichier = self.name.replace(".csv", "") + " " +
self.element_ordonnee + " " + self.element_Abscisse + \
                           ".jpg"
        plt.savefig(self.Nom_fichier)  # on sauvegarde le graph en lui mettant
comme nom la variable

        plt.show()  # on ferme la fenêtre
        self.y.clear()  # on efface la liste
        self.x.clear()  # on efface la liste
```

```
a = generation_graph()  # On appel la class generation_graph
tk.mainloop()  # On ferme l'IHM
```