实验一：python 爬虫与 GUI 界面

PB18000169　陈晨曦

任务说明：

利用 python 实现网络爬虫，获取相应网站的所有图片，并利用图形化界面 GUI 来实现图片数据的获取与信息反馈。

实验细节：

get_jmp_strurl_list 函数：

函数的输入参数为字符串 str_url，即为网页的 URL；函数的返回为字符串数组，其中存放了相应网页的所有图片的 URL。

通过 python 自带的函数库 requests 和 BeautifulSoup 来实现 get_jmp_strurl_list 函数，在利用 select 函数获得网页的所有图片信息后，还需要将相应的图片信息利用 re 库来实现正则化处理，使得最后返回的 URL 符合规范。

源代码：

```
# -*- coding: utf-8 -*-
"""
Created on Sun Apr 11 14:08:03 2021

@author: Lenovo
"""
import requests
from bs4 import BeautifulSoup
import re

#从待爬网页获得图片内容
def get_jmp_strurl_list(str_url):
    #str_url="https://www.ustc.edu.cn"   #待爬网页的URL,这是一个字符串
    URL=requests.get(str_url)   #这是一个URL对象

    #URL.text为URL对象中的html超文本文件内容
    soup=BeautifulSoup(URL.text,'lxml') #转化成Unicode编码格式

    #看看此时的strhtml，可见其文件结构满足html超文本文件的结构
    #print(soup)

    data=soup.select('img')
    jmp_strurl_list=[]
    for item in data:
        result=str_url+"/"+item.get('src')
        jmp_strurl_list.append(result)

    #for jmp_strurl in jmp_strurl_list:
        #print(jmp_strurl)
    return jmp_strurl_list
```

save_img 函数：

函数的输入参数为 filename 和 jmp_strurl_list，其中 filename 代表要创建的文件夹的文件夹名（该文件夹用来存储下载的图片），jmp_strurl_list 是个字符串数组，存放要下载的图片的 URL。

为了实现该函数，首先要调用 os 库实现文件夹的创建和路径的设置，其次利用 requests 库来访问图片的 URL 并完成图片的下载。在实现该函数的过程中，还调用了 time

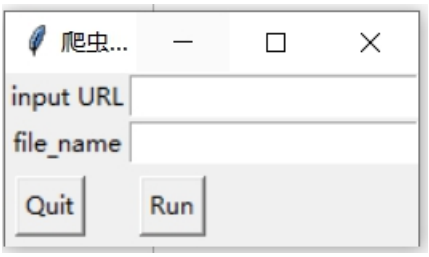
库来对无法访问 URL 进行处理，并在函数最后显示无法下载的图片的 URL。

源代码：

```python
# -*- coding: utf-8 -*-
"""
Created on Sun Apr 11 14:13:51 2021

@author: Lenovo
"""
import requests
import time
import os

def save_img(file_name,jmp_strurl_list):
    #创建保存下载图片的文件夹
    count=0
    mark=1
    unable_jmp_strurl_list=[]
    able_jmp_strurl_list=[]
    if (os.path.exists("./"+file_name)):
        print("the file_name already exists,please change the file_name")

    else:
        os.makedirs("./"+file_name)

        #开始下载图片
        for jmp_strurl in jmp_strurl_list:
            print(" try downloading----- %s" %(jmp_strurl))
            #通过时延语句判断图片是否可下载
            try:
                jmp_url=requests.get(jmp_strurl,timeout=10)
                time.sleep(1)
            except:
                mark=0;

            if(mark==1):
                jmp_full_path="./"+file_name+"/"+str(count)+".jpg"
                with open(jmp_full_path,'wb') as f:
                    f.write(jmp_url.content)
                if not (jmp_strurl in able_jmp_strurl_list):
                    able_jmp_strurl_list.append(jmp_strurl)
                count=count+1

            elif(mark==0):
                if not (jmp_strurl in unable_jmp_strurl_list):
                    unable_jmp_strurl_list.append(jmp_strurl)
                    mark=1

        print()
        print("successfully download %d pictures" %(count))
        if(unable_jmp_strurl_list!=[]):
            print("unable download jmp url list:")
            for unable_jmp_strurl in unable_jmp_strurl_list:
                print("    "+unable_jmp_strurl)

    return able_jmp_strurl_list
```

图形化界面部分：

实现图形化交互界面，实现方法为调用 tkinter 库，创建窗口，利用库中的 Label 类，Button 类，Entry 类完成整体的交互功能，交互界面如下：



在 tkinter 的具体使用，通过窗口中再生成窗口的方法来实现图片数据的反馈处理，为了反馈下载成功的信息，我们生成新窗口 info_window：

在 info_window 中的最后一行，我们还通过 os.getcwd()命令返回了保存下载图片的文件夹的路径信息。

源代码：

```python
# -*- coding: utf-8 -*-
"""
Spyder Editor

This is a temporary script file.
"""
import get_jmp
import download
import os

from tkinter import*

def GUI_func():
    strurl=entry1.get()
    file_name=entry2.get()
    jmp_strurl_list=get_jmp_strurl_list(strurl)
    able_jmp_strurl_list=save_img(file_name,jmp_strurl_list)

    #显示成功下载的图片信息
    info_window=Tk()
    info_window.title("成功下载----图片的url")
    i=0
    for jmp_strurl in able_jmp_strurl_list:
        Label(info_window,text=jmp_strurl).grid(row=i,column=0)
        i=i+1

    #窗口中空一行
    Label(info_window,text="").grid(row=i,column=0)
    i=i+1

    #显示图片的路径
    Label(info_window,text="path: "+os.getcwd()).grid(row=i,column=0)
    i=i+1
    pass

myWindow=Tk()
myWindow.title("爬虫----获取网页所有图片")
Label(myWindow,text="input URL").grid(row=0)
Label(myWindow,text="file_name").grid(row=1)

entry1=Entry(myWindow)
entry1.grid(row=0, column=1)
entry2=Entry(myWindow)
entry2.grid(row=1, column=1)

Button(myWindow, text='Quit', command=myWindow.quit).grid(row=2, column=0,sticky=W, padx=5, pady=5)
Button(myWindow, text='Run', command=GUI_func).grid(row=2, column=1, sticky=W, padx=5, pady=5)

myWindow.mainloop()

```

实验总结：

1.学习了 python 的基本语句，了解了 python 的使用方法和常规函数。

2.掌握了爬虫的基础原理和简单实用。

3.熟练了 GUI 图形交互界面的简单使用。