**python实验报告**

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| 课程名称 | **python数据处理编程** | 班级 | **计科1901班** | 学号 | **8208191213** | 姓名 | **王璟文** |
| 实验名称 | **实验1:网络爬虫开发及数据的可视化** | | | 实验时间 | | **2021.5.25** | |
| **实验环境** | | | | | | | |
| **PC/windows/Pycharm/Mysql** | | | | | | | |
| **实验目的 内容要求** | | | | | | | |
| **一.实验目的：**  **开发网络爬虫在东方财富、新浪财经或者纳斯达克等财经网站上爬取一只股票的每天的开盘价，收盘价，最高价，最低价等信息，并存储在数据库中，并开发GUI应用可视化。**  **二.实验内容要求：**  **（1）掌握网络爬虫的开发方法；**  **（2）掌握Python开发数据库的GUI界面；**  **（3）掌握Matplotlib绘制股票的K线图；** | | | | | | | |
| **程序运行结果 源代码 重要函数解释** | | | | | | | |
| 1. **程序运行过程：**   **(1)用户登录界面GUI:**    **(2)输入错误时的重新输入：**    **(3)输入正确登录成功，进入下一个GUI选择需要爬取的股票代码：**    **(4)选择上证，以贵州茅台为例：**    **(5)等待股票爬取，爬取成功：**    **(6)在Mysql数据库中检查爬取到的近80天的数据：**    **(7)检查生成的k线图：**    **四.源代码：**  **(1)Gui\_wx:**  **import re import tushare as ts import mplfinance as mpf from pylab import mpl import wx import pandas as pd from selenium import webdriver from db\_connect import InsertData   class Frame2(wx.Frame):  num = ''  def \_\_init\_\_(self):  super().\_\_init\_\_(parent=None, title='股票爬虫', size=(395, 180))  self.Centre() # 设置窗口居中   #分割窗口  splitter = wx.SplitterWindow(self, -1)  #创建左右面板  leftpanel = wx.Panel(splitter)  rightpanel = wx.Panel(splitter)  splitter.SplitVertically(leftpanel, rightpanel, 100)  splitter.SetMinimumPaneSize(80)   button = wx.Button(rightpanel, label = '确定')  self.tc = wx.TextCtrl(rightpanel, -1, '', pos=(35, 80), size=(150, -1), name='TC', style=wx.TE\_CENTER)   font1 = wx.Font(10, wx.MODERN, wx.NORMAL, wx.BOLD, False, '微软雅黑')  font2 = wx.Font(10, wx.MODERN, wx.NORMAL, wx.NORMAL, True, '微软雅黑')  list2 = ['深证', '上证', '创业板']  lb2 = wx.ListBox(leftpanel, -1, choices=list2, style=wx.LB\_SINGLE)  lb2.SetFont(font1)  self.Bind(wx.EVT\_LISTBOX, self.on\_listbox, lb2)   vbox1 = wx.BoxSizer(wx.VERTICAL)  vbox1.Add(lb2, 1, flag=wx.ALL | wx.EXPAND, border=5)  leftpanel.SetSizer(vbox1)   vbox2 = wx.BoxSizer(wx.VERTICAL)  self.content = wx.StaticText(rightpanel, label='请选择您想要爬取的股票分类')  self.content.SetFont(font2)  vbox2.Add(self.content, 1, flag=wx.ALL | wx.EXPAND, border=10)  vbox2.Add(button, 1, flag = wx.NORMAL | wx.NORMAL, border = 70)  rightpanel.SetSizer(vbox2)   button.Bind(wx.EVT\_LEFT\_UP, self.OnLeftUp) # 按钮左键弹起   def OnLeftUp(self, evt):  '''左键弹起事件函数'''  global num  num = self.tc.GetValue()  self.Destroy()  # ''' # 开始爬数据 # '''   number = num  # chromedriver的安装位置  driver = webdriver.Chrome(executable\_path=r'D:\chromedriver.exe')  # 打开贵州茅台的历史行情表对应的网页  driver.get('https://q.stock.sohu.com/cn/' + number + '/lshq.shtml')  em = driver.find\_element\_by\_id('BIZ\_hq\_historySearch')  trtext = em.text  driver.quit() # ''' # 正则表达式 # '''  def parse\_one\_page(html):  pattern = re.compile('<dd>.\*?board-index.\*?>(\d+)</i>.\*?src="(.\*?)".\*?name"><a'  + '.\*?>(.\*?)</a>.\*?star">(.\*?)</p>.\*?releasetime">(.\*?)</p>'  + '.\*?integer">(.\*?)</i>.\*?fraction">(.\*?)</i>.\*?</dd>', re.S)  items = re.findall(pattern, html)  # print(items)  for item in items:  yield {  'index': item[0],  'image': item[1],  'title': item[2],  'actor': item[3].strip()[3:],  'time': item[4].strip()[5:],  'score': item[5] + item[6]  } ###数据处理  tx = trtext.split('\n')  wholedata = {} # 总计的数据//一行  data = [] # 存储数据   v = 0  while v < len(tx):  if v == 0: # 第一行  v += 1  continue  elif v == 1: # 第二行  tr = tx[v].split(' ')   wholedata['date'] = (tr[1].replace('至', '~')).replace('2021', '21')  wholedata['updown\_e'] = tr[2]  wholedata['updown\_f'] = tr[3].replace('%', '')  wholedata['lowest'] = tr[4]  wholedata['highest'] = tr[5]  wholedata['deal\_num'] = tr[6]  wholedata['deal\_money'] = tr[7]  wholedata['change\_rate'] = tr[8].replace('%', '')  else: # 第三行开始  trr = tx[v].split(' ')  data\_eve = {}  data\_eve['date'] = trr[0].replace('-', '/')  data\_eve['open'] = trr[1]  data\_eve['close'] = trr[2]  data\_eve['updown\_e'] = trr[3]  data\_eve['updown\_f'] = trr[4].replace('%', '')  data\_eve['lowest'] = trr[5]  data\_eve['highest'] = trr[6]  data\_eve['deal\_num'] = trr[7]  data\_eve['deal\_money'] = trr[8]  data\_eve['change\_rate'] = trr[9].replace('%', '')   data.append(data\_eve)   v += 1    # print(wholedata)  # print(data)  # ''' # 生成csv文件 # '''**  **name = ['data', 'open', 'close', 'updown\_e', 'updown\_f', 'lowest', 'highest', 'deal\_num', 'deal\_money', 'change\_rate']  test = pd.DataFrame(columns=name, data=data)  test.to\_csv('D:\\桌面\\test.csv', index=False)  # ''' # 生成k线图 # '''   # pd.set\_option()就是pycharm输出控制显示的设置  pd.set\_option('expand\_frame\_repr', False) # True就是可以换行显示。设置成False的时候不允许换行  pd.set\_option('display.max\_columns', None) # 显示所有列  # pd.set\_option('display.max\_rows', None)# 显示所有行  pd.set\_option('colheader\_justify', 'centre') # 显示居中   pro = ts.pro\_api('579ffeb73d0ce7893be7c5565ec91f9164e4f70828be7100dc2299ba')  mpl.rcParams['font.sans-serif'] = ['SimHei'] # 指定默认字体  mpl.rcParams['axes.unicode\_minus'] = False # 解决保存图像是负号'-'显示为方块的问题  if number[0] == '6':  df = pro.daily(ts\_code=number + '.SH', start\_date='20210129', end\_date='20210602')  elif number[0] == '0':  df = pro.daily(ts\_code=number + '.SZ', start\_date='20210129', end\_date='20210602')   # df.sort\_values(by='trade\_date',ascending=False)  datament = df.loc[:, ['trade\_date', 'open', 'close', 'high', 'low', 'vol']] # ：取所有行数据，后面取date列，open列等数据  datament = datament.rename(  columns={'trade\_date': 'Date', 'open': 'Open', 'close': 'Close', 'high': 'High', 'low': 'Low',  'vol': 'Volume'}) # 更换列名，为后面函数变量做准备  datament.set\_index('Date', inplace=True) # 设置date列为索引，覆盖原来索引,这个时候索引还是 object 类型，就是字符串类型。  datament.index = pd.DatetimeIndex(  datament.index) # 将object类型转化成 DateIndex 类型，pd.DatetimeIndex 是把某一列进行转换，同时把该列的数据设置为索引 index。  datament = datament.sort\_index(ascending=True) # 将时间顺序升序，符合时间序列  mpf.plot(datament, type='candle', mav=(5, 10, 20), volume=True, show\_nontrading=False) # ''' # 将数据导入到mysql数据库中 # '''  InsertData('wholedata', wholedata)   i = 0  while i < len(data):  InsertData('data', data[i])  i += 1  print('成功保存到桌面生成csv文件')  print('数据成功存储到Mysql数据库')   def on\_listbox(self, event):  if event.GetString() == '深证':  s = '选择 深证（股票代码：000001~002946）'  elif event.GetString() == '上证':  s = '选择 上证（股票代码：600001~603927）'  else:  s = '选择 创业板（股票代码：300001~300999）'  self.content.SetLabel(s)   class App2(wx.App):   def OnInit(self):  # 创建窗口对象  frame = Frame2()  frame.Show()  return True  ###登录界面 class Frame1(wx.Frame):  '''程序主窗口类，继承自wx.Frame'''   def \_\_init\_\_(self, parent):  '''构造函数'''   wx.Frame.\_\_init\_\_(self, parent, -1, '股票查询系统——登录界面')  self.SetBackgroundColour(wx.Colour(1000, 1000, 1000))  self.SetSize((520, 220))  self.Center()  wx.StaticText(self, -1, u'用户名：', pos=(40, 50), size=(100, -1), style=wx.ALIGN\_RIGHT)  wx.StaticText(self, -1, u'密码：', pos=(40, 80), size=(100, -1), style=wx.ALIGN\_RIGHT)  self.tip = wx.StaticText(self, -1, u'', pos=(145, 110), size=(150, -1), style=wx.ST\_NO\_AUTORESIZE)  self.tc1 = wx.TextCtrl(self, -1, '', pos=(145, 50), size=(150, -1), name='TC01', style=wx.TE\_CENTER)  self.tc2 = wx.TextCtrl(self, -1, '', pos=(145, 80), size=(150, -1), name='TC02', style=wx.TE\_PASSWORD | wx.TE\_CENTER)  btn\_mea = wx.Button(self, -1, u'登录', pos=(350, 50), size=(100, 25))  # btn\_meb = wx.Button(self, -1, u'鼠标所有事件', pos=(350, 80), size=(100, 25))  btn\_close = wx.Button(self, -1, u'关闭窗口', pos=(350, 80), size=(100, 25))  # 控件事件  self.tc1.Bind(wx.EVT\_TEXT, self.EvtText)  self.tc2.Bind(wx.EVT\_TEXT, self.EvtText)  self.Bind(wx.EVT\_BUTTON, self.OnClose, btn\_close)  # 鼠标事件  btn\_mea.Bind(wx.EVT\_LEFT\_DOWN, self.OnLeftDown) #左键按下  btn\_mea.Bind(wx.EVT\_LEFT\_UP, self.OnLeftUp) #左键弹起  # btn\_mea.Bind(wx.EVT\_MOUSEWHEEL, self.OnMouseWheel)  # btn\_meb.Bind(wx.EVT\_LEFT\_UP, self.OnLeftUp) #左键弹起  # 键盘事件  self.Bind(wx.EVT\_KEY\_DOWN, self.OnKeyDown)  # 系统事件  self.Bind(wx.EVT\_CLOSE, self.OnClose)  self.Bind(wx.EVT\_SIZE, self.On\_size)  # self.Bind(wx.EVT\_PAINT, self.On\_paint)  # self.Bind(wx.EVT\_ERASE\_BACKGROUND, lambda event: None)  def EvtText(self, evt):  '''输入框事件函数'''   obj = evt.GetEventObject()   def On\_size(self, evt):  '''改变窗口大小事件函数'''   self.Refresh()  evt.Skip() # 体会作用   def OnClose(self, evt):  '''关闭窗口事件函数'''   dlg = wx.MessageDialog(None, u'确定要关闭本窗口？', u'操作提示', wx.YES\_NO | wx.ICON\_QUESTION)  if (dlg.ShowModal() == wx.ID\_YES):  self.Destroy()   def OnLeftDown(self, evt):  '''左键按下事件函数'''   # self.tip.SetLabel(u'左键按下')   def OnLeftUp(self, evt):  '''左键弹起事件函数'''   # self.tip.SetLabel(u'左键弹起')  if self.tc1.GetValue() == 'W\_Java' and self.tc2.GetValue() == '123456':  print('登录成功')  self.Destroy()  app = App2()  app.MainLoop()  else:  dlg = wx.MessageDialog(None, u'用户名不存在或密码错误\n是否重新输入？', u'操作提示', wx.YES\_NO | wx.ICON\_QUESTION)  if (dlg.ShowModal() == wx.ID\_NO):  self.Destroy()   def OnMouseWheel(self, evt):  '''鼠标滚轮事件函数'''   vector = evt.GetWheelRotation()  self.tip.SetLabel(str(vector))   def OnMouse(self, evt):  '''鼠标事件函数'''   self.tip.SetLabel(str(evt.EventType))   def OnKeyDown(self, evt):  '''键盘事件函数'''   key = evt.GetKeyCode()  self.tip.SetLabel(str(key))   class App1(wx.App):  def OnInit(self):  self.SetAppName('股票查询系统——登录界面')  self.Frame = Frame1(None)  self.Frame.Show()  return True  if \_\_name\_\_ == '\_\_main\_\_':  app = App1()  app.MainLoop()**  **(2)db\_connect:**  **import pymysql  ###导入数据库的具体函数操作 def InsertData(TableName, dic):  try:  #连接数据库  conn = pymysql.connect(host='localhost', user='root', password='010625', db='datatest', port=3306) # 链接数据库  #创建游标对象  cur = conn.cursor()  COLstr = '' # 列的字段  ROWstr = '' # 行字段   ColumnStyle = ' VARCHAR(20)'  for key in dic.keys():  COLstr = COLstr + ' ' + key + ColumnStyle + ','  ROWstr = (ROWstr + '"%s"' + ',') % (dic[key])   # 判断表是否存在，存在执行try，不存在执行except新建表，再insert  try:  cur.execute("SELECT \* FROM %s" % (TableName))  cur.execute("INSERT INTO %s VALUES (%s)" % (TableName, ROWstr[:-1]))   except pymysql.Error as e:  cur.execute("CREATE TABLE %s (%s)" % (TableName, COLstr[:-1]))  cur.execute("INSERT INTO %s VALUES (%s)" % (TableName, ROWstr[:-1]))  #提交到数据库执行  conn.commit()  #关闭游标  cur.close()  #关闭数据库  conn.close()   except pymysql.Error as e:  print("Mysql Error %d: %s" % (e.args[0], e.args[1]))**   1. **重要函数解释：**   **(1)OnLeftUp:登录过程中判断用户名密码是否正确的代码：**  **def OnLeftUp(self, evt):  '''左键弹起事件函数'''   # self.tip.SetLabel(u'左键弹起')  if self.tc1.GetValue() == 'W\_Java' and self.tc2.GetValue() == '123456':  print('登录成功')  self.Destroy()  app = App2()  app.MainLoop()  else:  dlg = wx.MessageDialog(None, u'用户名不存在或密码错误\n是否重新输入？', u'操作提示', wx.YES\_NO | wx.ICON\_QUESTION)  if (dlg.ShowModal() == wx.ID\_NO):  self.Destroy()**  **(2)从对应网站爬取数据的代码：**  **# ''' # 开始爬数据 # '''  number = num  # chromedriver的安装位置  driver = webdriver.Chrome(executable\_path=r'D:\chromedriver.exe')  # 打开贵州茅台的历史行情表对应的网页  driver.get('https://q.stock.sohu.com/cn/' + number + '/lshq.shtml')  em = driver.find\_element\_by\_id('BIZ\_hq\_historySearch')  trtext = em.text  driver.quit()**  **(3)正则表达式处理：**  **# ''' # 正则表达式 # '''  def parse\_one\_page(html):  pattern = re.compile('<dd>.\*?board-index.\*?>(\d+)</i>.\*?src="(.\*?)".\*?name"><a'  + '.\*?>(.\*?)</a>.\*?star">(.\*?)</p>.\*?releasetime">(.\*?)</p>'  + '.\*?integer">(.\*?)</i>.\*?fraction">(.\*?)</i>.\*?</dd>', re.S)  items = re.findall(pattern, html)  # print(items)  for item in items:  yield {  'index': item[0],  'image': item[1],  'title': item[2],  'actor': item[3].strip()[3:],  'time': item[4].strip()[5:],  'score': item[5] + item[6]  }**  **(4)对爬到的数据进一步处理：**  **###数据处理  tx = trtext.split('\n')  wholedata = {} # 总计的数据//一行  data = [] # 存储数据   v = 0  while v < len(tx):  if v == 0: # 第一行  v += 1  continue  elif v == 1: # 第二行  tr = tx[v].split(' ')   wholedata['date'] = (tr[1].replace('至', '~')).replace('2021', '21')  wholedata['updown\_e'] = tr[2]  wholedata['updown\_f'] = tr[3].replace('%', '')  wholedata['lowest'] = tr[4]  wholedata['highest'] = tr[5]  wholedata['deal\_num'] = tr[6]  wholedata['deal\_money'] = tr[7]  wholedata['change\_rate'] = tr[8].replace('%', '')  else: # 第三行开始  trr = tx[v].split(' ')  data\_eve = {}  data\_eve['date'] = trr[0].replace('-', '/')  data\_eve['open'] = trr[1]  data\_eve['close'] = trr[2]  data\_eve['updown\_e'] = trr[3]  data\_eve['updown\_f'] = trr[4].replace('%', '')  data\_eve['lowest'] = trr[5]  data\_eve['highest'] = trr[6]  data\_eve['deal\_num'] = trr[7]  data\_eve['deal\_money'] = trr[8]  data\_eve['change\_rate'] = trr[9].replace('%', '')   data.append(data\_eve)   v += 1**  **(5)生成csv文件：**  **# ''' # 生成csv文件 # '''  name = ['data', 'open', 'close', 'updown\_e', 'updown\_f', 'lowest', 'highest', 'deal\_num', 'deal\_money', 'change\_rate']  test = pd.DataFrame(columns=name, data=data)  test.to\_csv('D:\\桌面\\test.csv', index=False)**  **(6)生成k线图：**  **# ''' # 生成k线图 # '''  # pd.set\_option()就是pycharm输出控制显示的设置  pd.set\_option('expand\_frame\_repr', False) # True就是可以换行显示。设置成False的时候不允许换行  pd.set\_option('display.max\_columns', None) # 显示所有列  # pd.set\_option('display.max\_rows', None)# 显示所有行  pd.set\_option('colheader\_justify', 'centre') # 显示居中   pro = ts.pro\_api('579ffeb73d0ce7893be7c5565ec91f9164e4f70828be7100dc2299ba')  mpl.rcParams['font.sans-serif'] = ['SimHei'] # 指定默认字体  mpl.rcParams['axes.unicode\_minus'] = False # 解决保存图像是负号'-'显示为方块的问题  if number[0] == '6':  df = pro.daily(ts\_code=number + '.SH', start\_date='20210129', end\_date='20210602')  elif number[0] == '0':  df = pro.daily(ts\_code=number + '.SZ', start\_date='20210129', end\_date='20210602')   # df.sort\_values(by='trade\_date',ascending=False)  datament = df.loc[:, ['trade\_date', 'open', 'close', 'high', 'low', 'vol']] # ：取所有行数据，后面取date列，open列等数据  datament = datament.rename(  columns={'trade\_date': 'Date', 'open': 'Open', 'close': 'Close', 'high': 'High', 'low': 'Low',  'vol': 'Volume'}) # 更换列名，为后面函数变量做准备  datament.set\_index('Date', inplace=True) # 设置date列为索引，覆盖原来索引,这个时候索引还是 object 类型，就是字符串类型。  datament.index = pd.DatetimeIndex(  datament.index) # 将object类型转化成 DateIndex 类型，pd.DatetimeIndex 是把某一列进行转换，同时把该列的数据设置为索引 index。  datament = datament.sort\_index(ascending=True) # 将时间顺序升序，符合时间序列  mpf.plot(datament, type='candle', mav=(5, 10, 20), volume=True, show\_nontrading=False)**  **(7)数据导入mysql数据库中：**  **# ''' # 将数据导入到mysql数据库中 # '''  InsertData('wholedata', wholedata)   i = 0  while i < len(data):  InsertData('data', data[i])  i += 1  print('成功保存到桌面生成csv文件')  print('数据成功存储到Mysql数据库')   def on\_listbox(self, event):  if event.GetString() == '深证':  s = '选择 深证（股票代码：000001~002946）'  elif event.GetString() == '上证':  s = '选择 上证（股票代码：600001~603927）'  else:  s = '选择 创业板（股票代码：300001~300999）'  self.content.SetLabel(s)**  **(8)连接到数据库 并创建游标对象：**  **#连接数据库 conn = pymysql.connect(host='localhost', user='root', password='010625', db='datatest', port=3306) # 链接数据库 #创建游标对象 cur = conn.cursor() COLstr = '' # 列的字段 ROWstr = '' # 行字段**  **(9)数据库中的建表操作：**  **# 判断表是否存在，存在执行try，不存在执行except新建表，再insert try:  cur.execute("SELECT \* FROM %s" % (TableName))  cur.execute("INSERT INTO %s VALUES (%s)" % (TableName, ROWstr[:-1]))  except pymysql.Error as e:  cur.execute("CREATE TABLE %s (%s)" % (TableName, COLstr[:-1]))  cur.execute("INSERT INTO %s VALUES (%s)" % (TableName, ROWstr[:-1])) #提交到数据库执行 conn.commit()** | | | | | | | |
| **实验思路 在实验过程中遇到的问题 参考资料** | | | | | | | |
| 1. **实验思路：**   **本实验通过selenium实现对单只股票的近80天静态数据的爬取，通过正则表达式对爬到的代码进行处理，通过mysql数据库存储爬取的数据，进而生成csv文件，通过mplfinance和Matplotlib以及pandas对csv文件中的股票数据进行绘图操作，绘出股票k线图。通过wxpython实现界面GUI，实现人机交互的友好界面。本项目可以实现三大板块（深证，上证，创业板）任何一只股票的爬取，并实现数据库的存储化以及k线图的可视化分析。**   1. **实验过程中遇到的问题：**   **(1)刚开始对正则表达式的应用并不熟练，后来自学了《python从小白到大牛》**  **中正则表达式的部分，做到了对杂乱数据的处理及定位。**   1. **在GUI的设计过程中，起初打算使用pyqt5进行制作，但是由于本人并不适应直接拖拽创造界面的方式，后来换用了wxpython进行制作。**   **七．参考文献：**  **《python从小白到大牛》**  **《python数据处理编程》**  **csdn中相关博文** | | | | | | | |
| **指导老师： 王斌 2021 年 6 月 2 日** | | | | | | | |

**2020-2021学年第 1学期 任课老师：**

**填写内容时，可把表格扩大。**