|  |  |  |  |
| --- | --- | --- | --- |
| 姓名：叶超 | 专业：物联网工程 | 班级： | 20181191 |
| 科目：物联网云端处理技术 | | 实验日期：2020-09-22 | |
| 实验题目：物联网云端处理技术实验-2 | | | |
| **【**实验目的**】**  要求：   1. 实验内容及要求：用c语言实现下列功能    1. 读取文件“600519.csv”和“000858.csv”中有关股票的数据。    2. 设计并实现一组命令行下的选择列表，用户通过此列表可以完成如下的功能：       1. 选择一只股票，显示此股票在某个输入的日期的当日数据。       2. 选择一只股票，显示此股票收盘价高于某个输入的价格的所有日期。       3. 选择一只股票，显示此股票股价涨跌幅高于某个输入值的所有日期。       4. 选择一只股票，显示此股票成交量高于某个输入的值的所有日期。       5. 选择一只股票，显示此股票成交金额高于某个输入的值的所有日期。（注意：输入文件中的成交金额为科学计数法。）       6. 选择两只股票，显示并比较这两只股票在某个输入的日期的交易数据。（注意：须考虑三种情况：1、在此日期，两只股票都有数据；2、在此日期，只有一只股票有数据；3、在此日期，两只股票都没有数据。）       7. 选择两只股票，显示这两只股票涨跌幅分别高于某个输入值的日期数。       8. 选择两只股票，显示这两只股票成交量分别高于某个输入值的日期数。       9. 选择两只股票，显示这两只股票成交金额分别高于某个输入值的日期数。 2. 上交内容及要求：    1. 用Word书写实验报告，应包括每个功能的运行截图。    2. C语言程序的源代码。（即“.c”文件）    3. 提交格式：请将实验报告和源代码放入名为“物联网云端数据处理实验-2-学号-姓名”的压缩包中。    4. 请同学们在10月9日24点之前将本次学习报告交给学委。 3. 此实验由各学生独立完成，请勿抄袭。 | | | |
| **【**实验内容**】**  class CsvFile:  def \_\_init\_\_(self, file\_path):  self.file\_path = file\_path  class Solutions:  '''  csv文件处理  '''  def \_\_init\_\_(self, CsvFile):  '''  初始化函数，  属性： CsvFile： CsvFile对象  '''  self.CsvFile = CsvFile      def get\_idx(self, flag, lists):  '''  @brief：返回 flag 在lists第一次出现的下标  @param: flag: 待寻找的元素  lists:集合  '''  idx = 0  while idx<len(temp) and temp[idx]!=char:  idx+=1  return idx    def get\_last\_index(self, flag, lists):  '''  获取 flag 在 lists 最后一次出现的下标  思路： 反转字符串后， 寻找第一次出现的下标  '''  temp = strs[::-1] # 反转字符串  idx = self.get\_idx(self, flag, temp)  return (len(temp)-1-idx)    def get\_id\_name(self):  '''  对属性 file\_path 处理，得到csv文件对应的股票代码  '''  begin\_char = '/'  end\_char = '.'  begin\_index = get\_last\_index(self.CsvFile.file\_path,  begin\_char) + 1  end\_index = get\_last\_index(self.CsvFile.file\_path,  end\_char)  return self.CsvFile.file\_path[begin\_index, end\_index]      def get\_df(self):  '''  通过属性 file\_path 处理csv数据，得到 DataFrame  '''  file = pd.read\_csv(self.CsvFile.file\_path)  df = pd.DataFrame(file)  return df    def get\_df\_head\_list(self):  '''  无参构造，得到列标题  '''  df = self.get\_df()  head\_list = df.columns.values.tolist()  return head\_list    def search\_equal(self, head, val, show\_head\_list):    '''  寻找在 head 这一列中 值为 val的行，显示内容由 show\_head\_list 决定  '''  #-----------------------------------------  '''  df\_head\_list = self.get\_df\_head\_list()  head = df\_head\_list[head\_idx]  '''  #-----------------------------------------  df = self.get\_df()  return df.loc[ (df[head] == val),  show\_head\_list]  def search\_cmp(self,head, val, show\_head\_list):  '''  寻找在 head 这一列中 值> val的行，显示内容由 show\_head\_list 决定  '''  df = self.get\_df()  return df.loc[ (df[head] > val),  show\_head\_list ]  def marge(csv\_list, outputfile):  for inputfile in csv\_list:  f = open(inputfile, 'r', encoding='gb18030',errors='ignore')  data = pd.read\_csv(f,encoding='gb18030')  data.to\_csv(outputfile, mode='a', index=False)  print('完成合并')    #去重保留一个表头  def distinct(file):  df = pd.read\_csv(file, header=None)  datalist = df.drop\_duplicates()  datalist.to\_csv( index=False, header=False)  print('完成去重')  csv\_list = glob.glob(os.getcwd()+'/data/\*.csv')  output\_csv\_path = 'data/margin.csv'  print(csv\_list)  marge(csv\_list, output\_csv\_path)  distinct(output\_csv\_path)  def main():  csv\_list = glob.glob(os.getcwd()+'/data/\*.csv')  margin\_file\_path = os.getcwd()+'/data/margin.csv'  if margin\_file\_path not in csv\_list:  output\_csv\_path = 'data/margin.csv'  marge(csv\_list, output\_csv\_path)  distinct(output\_csv\_path)  pwd = os.getcwd()+'/data' # csv文件所在目录  csv\_list = glob.glob(os.getcwd()+'/data/\*.csv')  print(csv\_list)  file = CsvFile(csv\_list[0]) # 读取第一个csv文件，新建CsvFile对象，  test = Solutions(file)  show\_list = test.get\_df\_head\_list()  #pprint.pprint(test.get\_df())  print(test.search\_cmp('成交金额', '912323410', show\_list).head())  if \_\_name\_\_ =='\_\_main\_\_':  main() | | | |
| **【**实验结果**】**          5    **选取2个股票：**  **合并2个csv文件：**    6.    7.    8：  9： | | | |
| **【**教师评语和成绩**】**  **成绩：** **指导教师：** **日期：** | | | |