HỌC VIỆN CÔNG NGHỆ BƯU CHÍNH VIỄN THÔNG

KHOA CÔNG NGHỆ THÔNG TIN

BỘ MÔN LẬP TRÌNH PYTHON

BÁO CÁO BÀI TẬP LỚN

Giảng viên hướng dẫn : **Kim Ngọc Bách**
Nhóm : **11**
Người thực hiện : **Nguyễn Xuân Khương**
Mã SV : **B22DCCN474**

Hà Nội – 2024

Bài I:

1.Nhập thư viện:

Mã bắt đầu bằng việc nhập các thư viện cần thiết:

- Selenium: Dùng để tự động hóa trình duyệt web
- BeautifulSoup: Dùng để phân tích cú pháp HTML và XML.
- Webdriver_manager: Quản lý driver cho trình duyệt tự động.

```
from selenium import webdriver
from selenium.webdriver.chrome.service import Service
from webdriver_manager.chrome import ChromeDriverManager
import time
from bs4 import BeautifulSoup
```

2.Thiết lập điều khiển web:

Trình điều khiển Chrome được khởi tạo và cài đặt tự động thông qua ChromeDriverManager.

```
driver = webdriver.Chrome(service=Service(ChromeDriverManager().install()))
```

3.Định nghĩa các lớp quản lý người chơi và đội:

Các lớp Player, Player_Manager, Squad, và Squad_Manager được nhập từ một mô-đun khác (giả định là ob). Những lớp này được sử dụng để quản lý dữ liệu người chơi và đội bóng.

```
from ob import Player
from ob import Player_Manager
from ob import Squad
from ob import Squad_Manager

player_manager = Player_Manager()
squad_manager = Squad_Manager()
```

4.Hàm kiểm tra dữ liệu:

Hàm validdata kiểm tra xem giá trị đầu vào có rỗng hay không. Nếu rỗng, nó trả về "N/a"; nếu không, nó chuyển đổi giá trị thành kiểu float.

```
def validdata(n): if n == '': return "N/a" return float(n)
```

5.Hàm thu thập dữ liệu từ trang web:

Hàm getDataFromWeb thực hiện việc truy cập URL, thu thập và phân tích dữ liệu thống kê từ các bảng HTML. Hàm này trả về danh sách dữ liệu của người chơi và đội.

```
def getDataFromWeb(url, idPlayerTable, idSquadTable, lengthPlayerData,
DataName):
    ...
```

5.1.Thu thập dữ liệu người chơi:

Dữ liệu người chơi được thu thập từ bảng có ID idPlayerTable và được thêm vào danh sách resultPlayerData.

5.2.Thu thập dữ liệu đội:

Dữ liệu đội bóng được thu thập từ bảng có ID idSquadTable và được thêm vào danh sách resultSquadData.

6.Thu thập các thống kê khác nhau:

Mã gọi hàm getDataFromWeb nhiều lần với các URL khác nhau để thu thập các thống kê khác nhau như Thời gian Chơi, Thống kê Cơ bản, Thống kê Thủ môn, và nhiều hơn nữa.

7.Xử lý dữ liệu đội:

Dữ liệu đội được xử lý để kiểm tra xem đội đã tồn tại trong bộ quản lý đội hay chưa. Nếu chưa, một đối tượng đội mới được tạo và thêm vào bộ quản lý.

```
for i in list_squad_result:
    s = squad_manager.findSquadByName(i[0])
    if s == None:
        new_s = Squad(*i[0:3])
        ...
```

8.Xử lý dữ liệu người chơi:

Tương tự như xử lý dữ liệu đội, mã kiểm tra xem người chơi đã tồn tại hay chưa trước khi thêm người chơi mới vào bộ quản lý và liên kết với đội tương ứng.

```
for i in list_player_result:
    p = player_manager.findPlayerByNameandTeam(i[0], i[3])
    ...
```

9.Lưu dữ liệu:

Dữ liệu đã thu thập được lưu vào tệp bằng cách sử dụng mô-đun pickle, cho phép lưu trữ và phục hồi dữ liệu dễ dàng.

```
import pickle

with open("squads.pkl", "wb") as file:
    pickle.dump(squad_manager.list_squad, file)
```

10.Xuất dữ liệu người chơi ra tệp CSV:

Dữ liệu người chơi được xuất ra tệp CSV để dễ dàng xem xét và phân tích. Tệp này bao gồm tiêu đề và các hàng dữ liệu cho mỗi người chơi.

```
import csv
with open('result.csv', mode='w', newline='', encoding='utf-8') as file:
    writer = csv.writer(file)
    ...
```

11.Mở tệp CSV:

Cuối cùng, mã mở tệp CSV đã xuất bằng cách sử dụng lệnh hệ thống.

```
import subprocess
subprocess.Popen(["start", r"result.csv"], shell=True)
```

Bài II:

1.Nhập dữ liệu:

Mã bắt đầu bằng việc nhập các thư viện cần thiết cho việc xử lý dữ liệu và vẽ biểu đồ:

```
import pickle
import csv
import statistics
from common import header, row, row2, header2, rowsquad
import pandas as pd
import os
```

-pickle: Để tải dữ liệu đã lưu trữ.

-csv: Để xuất dữ liệu ra định dạng CSV.

-statistics: Để tính toán các thống kê như trung bình, trung vị, độ lệch chuẩn.

-pandas: Để xử lý dữ liệu dạng bảng.

-os: Để thao tác với hệ thống tệp và thư mục.

2.Tải dữ liệu đội:

Dữ liệu đội bóng được tải từ tệp squads.pkl:

```
list_squad = []
with open("squads.pkl", "rb") as file:
    list_squad = pickle.load(file)
```

3.Đọc và tiền xử lý dữ liệu:

Dữ liệu người chơi được đọc từ tệp result.csv. Các giá trị "N/a" được thay thế bằng 0 và các cột dữ liệu sau cột thứ 5 được chuyển đổi thành kiểu số:

```
df = pd.read_csv('result.csv')
df.replace("N/a", 0, inplace=True)
ATTR_NUMBER = 172
for i in range(5, ATTR_NUMBER):
    df[df.columns[i]] = pd.to_numeric(df[df.columns[i]], errors='coerce').fillna(0)
```

4.Tính toán và xuất dữ liệu thống kê:

4.1.Tìm top 3 và bottom 3:

Mặc dù mã đã được chuẩn bị để tìm ra các cầu thủ đứng đầu và đứng cuối cho từng thuộc tính, đoạn mã này được tắt đi bằng cách sử dụng điều kiện if 1==0. Nếu được bật, nó sẽ in ra tên các cầu thủ top 3 và bottom 3 cho từng thuộc tính:

```
if 1 == 0:
    for i in range(5, ATTR_NUMBER):
        top_3_rows = df.nlargest(3, df.columns[i])
        print("Top3 cao nhất thuộc tính", header[i])
        print(top_3_rows.iloc[:, 0].values)
        print("Top 3 thấp nhất thuộc tính", header[i])
        bot_3_rows = df.nsmallest(3, df.columns[i])
        print(bot_3_rows.iloc[:, 0].values)
```

4.2.Tính trung bình, trung vị, độ lệch chuẩn:

Đoạn mã tiếp theo tính toán giá trị trung bình, trung vị và độ lệch chuẩn cho từng thuộc tính, sau đó xuất kết quả ra tệp CSV result2.csv:

```
with open('result2.csv', mode='w', newline='', encoding='utf-8') as file:
    all_attr = []
    for i in range(5, ATTR_NUMBER):
        arr = df.iloc[:, i]
        all_attr.append(arr)
  ...
 writer = csv.writer(file)
 writer.writerow(header2)
  ...
```

Dữ liệu thống kê được tính toán cho toàn bộ dữ liệu cũng như cho từng đội bóng riêng lẻ.

5.Vẽ biểu đồ:

5.1. Vẽ Biểu Đồ Histogram cho Từng Thuộc Tính:

Đoạn mã này tạo ra biểu đồ histogram cho từng thuộc tính và lưu chúng vào thư mục histograms:

```
if 1 == 0:
    import matplotlib.pyplot as plt
    output_directory = 'histograms'
    os.makedirs(output_directory, exist_ok=True)
    ...
```

5.2. Vẽ Biểu Đồ cho Từng Đội:

Mã cũng tạo ra biểu đồ cho từng đội bóng và lưu chúng vào các thư mục riêng biệt:

```
        for squad in squads:
            output_squad_directory = f'histogramsof{squad}'
            os.makedirs(output_squad_directory, exist_ok=True)
            ...
```

6.Tìm đội bóng có chỉ số cao nhât:

Cuối cùng, đoạn mã tìm đội bóng có chỉ số cao nhất cho từng thuộc tính:

```
        if 1 == 0:
            biggest_attr_value = [-1e9] * 167
            best_team_in_one_attr = [""] * 167
            ...
```

Kết quả được in ra và đội bóng có phong độ ấn tượng nhất được xác định bằng cách đếm số lượng chỉ số cao nhất mà mỗi đội đạt được.