федеральное государственное автономное образовательное учреждение высшего образования

Санкт-Петербургский национальный исследовательский УНИВЕРСИТЕТ информационных технологий, механики и оптики

**Факультет программной инженерии и компьютерной техники**

**ЛАБОРАТОРНАЯ РАБОТА № 3**

**ПО ДИСЦИПЛИНЕ «Технологии программирования»**

Выполнили:

Юсюмбели В. И.

Плохотнюк В. С.

Толстухин М.С.

Группа: P3411

Преподаватель:

Оголюк А. А.

Санкт-Петербург

2019

import sys  
from html.parser import HTMLParser  
from enum import Enum  
  
  
class TAG(Enum):  
 html = 0,  
 tr = 1,  
 td = 2,  
 table = 3,  
  
  
class STATUS(Enum):  
 HTML\_START = 0,  
 TABLE\_START = 1,  
 TABLE\_FOUND = 2,  
 TABLE\_ROW\_FOUND = 3,  
 RANK = 4,  
 RANK\_END = 5,  
 MALE = 6,  
 MALE\_END = 7,  
 FEMALE = 8,  
 FEMALE\_END = 9,  
 ROW\_END = 10,  
  
  
class COLUMN(Enum):  
 NONE = -1,  
 RANK = 0,  
 MALE = 1,  
 FEMALE = 2,  
  
  
def is\_int(s):  
 try:  
 int(s)  
 return True  
 except ValueError:  
 return False  
  
  
class TableHTMLParser(HTMLParser):  
 status = STATUS.HTML\_START  
 resultList = []  
 currentRank = 0  
  
 def handle\_starttag(self, tag, attrs):  
  
 if tag == TAG.html.name:  
 self.status = STATUS.HTML\_START  
  
 if tag == TAG.table.name and self.status == STATUS.HTML\_START:  
 for attr in attrs:  
 if attr[0] == 'summary' and attr[1] == 'Popularity for top 1000':  
 self.resultList = []  
 self.status = STATUS.TABLE\_FOUND  
  
 if tag == TAG.tr.name and (self.status == STATUS.TABLE\_FOUND or self.status == STATUS.ROW\_END):  
 self.status = STATUS.TABLE\_ROW\_FOUND  
  
 if tag == TAG.td.name:  
 if self.status == STATUS.TABLE\_ROW\_FOUND:  
 self.status = STATUS.RANK  
 if self.status == STATUS.RANK\_END:  
 self.status = STATUS.MALE  
 if self.status == STATUS.MALE\_END:  
 self.status = STATUS.FEMALE  
  
 def handle\_data(self, data):  
 if self.status == STATUS.RANK:  
 if is\_int(data):  
 self.currentRank = int(data)  
 else:  
 self.currentRank = -1  
 if self.status == STATUS.MALE:  
 if self.currentRank != -1:  
 self.resultList.append([data])  
 if self.status == STATUS.FEMALE:  
 if self.currentRank != -1:  
 self.resultList[self.currentRank - 1].append(data)  
  
 def handle\_endtag(self, tag):  
 if tag == TAG.td.name:  
 if self.status == STATUS.RANK:  
 self.status = STATUS.RANK\_END  
 if self.status == STATUS.MALE:  
 self.status = STATUS.MALE\_END  
 if self.status == STATUS.FEMALE:  
 self.status = STATUS.FEMALE\_END  
 if tag == TAG.tr.name:  
 if self.status == STATUS.FEMALE\_END:  
 self.status = STATUS.ROW\_END  
 if tag == TAG.table.name and self.status == STATUS.ROW\_END:  
 del self.resultList[-1]  
  
  
def extr\_name(filename):  
 file = open(filename, 'rt')  
 html = file.read()  
 file.close()  
 parser = TableHTMLParser()  
 parser.feed(html)  
 return parser.resultList  
  
  
def list\_names\_with\_rank(result, is\_male):  
 return [f'{item[int(is\_male)]} {i + 1}' for i, item in enumerate(result)]  
  
  
def main():  
 # print(1)  
 args = sys.argv[1:]  
 if not args:  
 print('use: [--file] file [file ...]')  
 sys.exit(1)  
 else:  
 if args[0] == '--file':  
 args = args[1:]  
  
 output = ''  
 output2 = ''  
 for filename in args:  
 output += '\'' + filename[4:8] + '\''  
 output2 += filename + '\n'  
 result = extr\_name(filename)  
 count = 0  
 for name in result:  
 if count < 10:  
 output2 += f"{count + 1} {name[0]} {name[1]}"  
 output2 += '\n'  
 else:  
 output2 += '\n'  
 break  
 count += 1  
  
 names = sorted(list\_names\_with\_rank(result, False) + list\_names\_with\_rank(result, True))  
 for name in names:  
 output += f", '{name}'"  
 output += '\n'  
  
 print(output)  
 print('TOP-10 RATED')  
 print(output2)  
  
  
if \_\_name\_\_ == '\_\_main\_\_':  
 main()