Практическое занятие №14

Тема: Составление программ с использованием регулярных выражений в IDE PyCharm Community.

Цель: Закрепить усвоенные знания, понятия, алгоритмы, основные принципы составления программ, приобрести навыки составления программ с использованием регулярных выражений в IDE PyCharm Community.

Постановка задачи. Из исходного текстового файла (expansion.txt) выбрать имена файлов, соответствующие типам: .xls, .xml, .html, .css, .py. Посчитать количество полученных элементов.

```
Текст программы:
```

```
import re
with open('expansion.txt', 'r') as file:
  content = file.read()
# Определяем регулярное выражение для поиска имен файлов с расширениями .xls, .xml, .html, .css, .py
pattern = re.compile(r'\b[a-zA-ZO-9 \.-]+\.(xls|xml|html|css|py)\b')
# Ищем все соответствия регулярному выражению в тексте
matches = pattern.findall(content)
# Подсчет количества совпадений
count = len(matches)
print(f'Haшeл {count} файлов со следующими расширениями: .xls, .xml, .html, .css, .py')
print('Имена файлов:', ', '.join(matches))
Упражнение по регуляркам:
from typing import Pattern
import re
with open("regex.txt") as f:
  data = f.read()
def get_data(pattern: Pattern[str]) -> str:
  return ' | '.join(pattern.findall(data))
ex 1 = re.compile(r'' d+'')
print(f"Bce натуральные числа: {get_data(ex_1)}\n")
ex 2 = re.compile(r''[A-ZA-9]+'')
print(f"Bce «слова», написанные капсом: {get_data(ex_2)}\n")
ex_3 = re.compile(r''[A-9][0-9].'', re.I)
print(f"Все слова, в которых есть русская буква, а за ней цифра: {get_data(ex_3)}\n")
ex 4 = \text{re.compile}(r'' \backslash b[A-AA-ZE] \backslash w+'')
print(f"Все слова, начинающиеся с русской или латинской большой букв: {get_data(ex_4)}\n")
ex 5 = re.compile(r"\b[AEËИОУЫЭЮЯаеіои]\w+", re.l)
print(f"Все слова, которые начинаются на гласную: {get_data(ex_5)}\n")
ex_6 = re.compile(r"\b[a-sA-Z]+\d{2,}[a-sA-Z]+", re.l)
print(f"Все натуральные числа, не находящиеся на границе слова: {get_data(ex_6)}\n")
ex_7 = re.compile(r"\S^*\*\S", re.DOTALL)
print(f"Haйдите строчки, в которых есть символ* (. — это точно не конец строки!): {get_data(ex_7)}\n")
```

```
ex_8 = re.compile(r"\(.*?\)+", re.DOTALL)

print(f"Hайдите строчки, в которых есть открывающая и когда-нибудь потом закрывающая скобки:

{get_data(ex_8)}\n")

ex_9 = re.compile(r'<a href="#\d+">.*</a>', re.DOTALL)

print(f"Выделите одним махом весь кусок оглавления (в конце примера, вместе с тегами):

\n{get_data(ex_9)}\n")

ex_10 = re.compile(r'<a href="#\d+">(.*?)</a>', re.DOTALL)

print(f"Выделите одним махом только текстовую часть оглавления, без тегов: {get_data(ex_10)}\n")

ex_11 = re.compile(r"^\s*$, re.MULTILINE)

print(f"Hайдите пустые строчки: {get_data(ex_11)}\n")

ex_12 = re.compile(r'(<a href="#\d+">).*?(</a>)')

print("Bce теги, не включая их содержимое: ")

[print(f"{tags}") for tags in map(lambda x: " ".join(x), ex_12.findall(data))]
```

Протокол работы программы:

Нашел 7 файлов со следующими расширениями: .xls, .xml, .html, .css, .py Имена файлов: xml, html, css, py, html, css, py

Process finished with exit code 0

Упражнение:

Все натуральные числа: 123 | 123 | 12345 | 11223344 | 1 | 2 | 3 | 11 | 22 | 33 | 44 | 10 | 10 | 11 | 11 | 12 | 12 | 13 | 13 | 14 | 14 | 15 | 15 | 16 | 16 | 2

Все слова, в которых есть русская буква, а за ней цифра:

Все слова, начинающиеся с русской или латинской большой букв: евСЃР | СЏСЂРЅС | Рµ | РІС | СЂР | енРёСЏ | РїСЪРµРГСЃС | РІР | СЏСЋС | СЃРЅР | РЅР | РїРЅС | РЅР | РЁР | РЅРЅ | РІРЅСЪР | РГРЅ | РЅР | РІРЅСЪР | РІРЅСЪРЅЪР | РІРЅРІР | РЅР | РІРЅРІР | РІРІР | РІРОТЬР | РІРОТ

Все слова, которые начинаются на гласную: Expressions | under_score

Все натуральные числа, не находящиеся на границе слова:

```
Найдите строчки, в которых есть символ * (. — это точно не конец строки!): *** \mid (***) \mid a*(b+[c+d])*e
| !!"""####$$$$%%%%%&&&'''(((())***+
Найдите строчки, в которых есть открывающая и когда-нибудь потом закрывающая скобки: () | (***) |
(b+[c+d]) | (((())
Выделите одним махом весь кусок оглавления (в конце примера, вместе с тегами):
<a href="#10">10: CamelCase -> under score</a>;
<a href="#11">11: PJPrP°P»PµPSPëPµ PïPsPIC,PsCъPsPI</a>;
<a href="#12">12: P'P»PëP·PePëPμ CΓP»PsPIP°</a>;
<a href="#13">13: P¤PsCЂPjP°C,PëCЂPsPIP°PSPëPμ P±PsP»CЊC€PëC... C‡PëCЃPμP»</a>;
<a href="#14">14: P P°P·PrPμP»PëC,CHb C,PμPεCΓC, PSP° PïCЂPμPrP»PsP¶PμPSPëCЏ</a>;
<a href="#15">15: P¤PsCЪPjP°C,PёСЪPsPIP°PSPёРµ PSPsPjPµСЪP° C,PµP»PµС"PsPSP°</a>;
<a href="#16">16: РџРѕРёСЃРє e-mail'РѕРІ вЪ" 2</a>
Выделите одним махом только текстовую часть оглавления, без тегов: 10: CamelCase -> under_score | 11:
РЈРrP°P»РµPSPëPµ PïPsPiC,PsCъPsPi | 12: P′P»PëP·PePëPµ CЃP»PsPiP° | 13: P¤PsCъPjP°C,PëCъPsPiP°PSPëPµ
P±PsP»CHC€PëC... C‡PëCЃРμP» | 14: P P°P·PrPμP»PëC,CH C,PμP€CЃC, PSP° PïCЂΡμPrP»PsP¶PμPSPëCЏ | 15:
P \texttt{xPsC} \texttt{bPjP} \texttt{°C,PEC} \texttt{bPsPIP} \texttt{°PSPEP} \mu \ P \texttt{SPsPjP} \mu \texttt{C} \texttt{bP} \texttt{°C,P} \mu \texttt{P} \texttt{»P} \mu \texttt{C} \textit{"PsPSP} \texttt{°} \ | \ 16: \ P \mu P \texttt{sPEC} \acute{\mathsf{P}} \texttt{e-mail} \texttt{'PsPI} \ \texttt{b} \texttt{b} \texttt{"} \ 2
Найдите пустые строчки: | | | | |
Все теги, не включая их содержимое:
<a href="#10"> </a>
<a href="#11"> </a>
<a href="#12"> </a>
<a href="#13"> </a>
<a href="#14"> </a>
<a href="#15"> </a>
<a href="#16"> </a>
```

Process finished with exit code 0

Вывод: В процессе выполнения практического занятия выработала навыки составления программ с использованием регулярных выражений в IDE PyCharm Community.Выполнены разработка кода, отладка, тестирование, оптимизация программного кода. Готовые программные коды выложены на GitHub