МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ

Федеральное государственное бюджетное образовательное учреждение высшего образования «Московский Авиационный Институт (Национальный Исследовательский Университет)»

Институт №3 Системы управления, информатика и электроэнергетика

Кафедра 304 «Вычислительные машины, системы и сети»

Отчет по лабораторной работе №3

по учебной дисциплине

«Конструирование компиляторов»

Выполнили:

студенты группы М30-411Б-21

Нуриев Н.Н.

Макаренков М.В.

Принял преподаватель:

доцент каф.304, к.т.н., Охотников Д.А.

Оглавление

# Задание 1: Введение и Азбука

Ниже приведены несколько строк текста, напишите шаблон, который соответствует всем предоставленным строкам.

|  |  |  |
| --- | --- | --- |
| Найти | abcdefg | To be completed |
| Найти | abcde | To be completed |
| Найти | abc |  |

Код:  
void main() {

  RegExp regExp = RegExp(r'abc.\*');

  List<String> input = [

    "abcdefg",

    "abcde",

    "abc",

  ];

  for (String str in input) {

    if (regExp.hasMatch(str)) {

      print('Совпадение найдено');

    } else {

      print('Совпадение не найдено');

    }

  }

}

Вывод в консоль:



# Задание 1.2: Числа

Ниже приведены несколько строк текста, содержащих цифры. Напишите шаблон, который соответствует всем цифрам в приведенных ниже строках.

|  |  |  |
| --- | --- | --- |
| **Задача** | **Текст** |  |
| Найти | abc123xyz | To be completed |
| Найти | define "123" | To be completed |
| Найти | var g = 123; |  |

Код:  
void main() {

  RegExp regExp = RegExp(r'\d+');

  List<String> input = [

    "abc123xyz",

    "defint \"123\"",

    "var g = 123",

  ];

  for (String str in input) {

    if (regExp.hasMatch(str)) {

      print('Совпадение найдено');

    } else {

      print('Совпадение не найдено');

    }

  }

}

Вывод в консоль:



# Задание 2: Точка

Ниже приведены несколько строк одинаковой длины с различными символами. Напишите шаблон, который соответствует первым трем строкам, но пропускает последнюю.

|  |  |  |
| --- | --- | --- |
| **Задача** | **Текст** |  |
| Найти | cat. | To be completed |
| Найти | 896. | To be completed |
| Найти | ?=+. | To be completed |
| Пропустить | abc1 |  |

Код:  
void main() {

  RegExp regExp = RegExp(r'.+\.');

  List<String> input = ["cat.", "896.", "?=+.", "abc1",];

  for (String str in input) {

    if (regExp.hasMatch(str)) {

      print('Совпадение найдено');

    } else {

      print('Совпадение не найдено');

    }

  }

}

Вывод в консоль:



# Задание 3: Сопоставление определенных символов

Ниже приведены несколько строк, где необходимо найти только первые три строки, и пропустить последние три. Обратите внимание, что мы не можем избежать сопоставления последних трех строк, если используем метасимвол точку, а должны специально определить, какие буквы должны совпадать, используя приведенное выше правило.

|  |  |  |
| --- | --- | --- |
| **Задача** | **Текст** |  |
| Найти | can | To be completed |
| Найти | man | To be completed |
| Найти | fan | To be completed |
| Пропустить | dan | To be completed |
| Пропустить | ran | To be completed |
| Пропустить | pan |  |

Код:  
void main() {

  RegExp regExp = RegExp(r'[cfm]an');

  List<String> input = [

    "can",

    "man",

    "fan",

    "dan",

    "ran",

    "pan",

  ];

  for (String str in input) {

    if (regExp.hasMatch(str)) {

      print('Совпадение найдено');

    } else {

      print('Совпадение не найдено');

    }

  }

}

Вывод в консоль:



# Урок 4: Исключение определенных символов

|  |  |  |
| --- | --- | --- |
| **Задача** | **Текст** |  |
| Найти | hog | To be completed |
| Найти | dog | To be completed |
| Пропустить | bog |  |

Код:  
void main() {

  RegExp regExp = RegExp(r'[^b]og');

  List<String> input = [

    "hog",

    "dog",

    "bog",

  ];

  for (String str in input) {

    if (regExp.hasMatch(str)) {

      print('Совпадение найдено');

    } else {

      print('Совпадение не найдено');

    }

  }

}

Вывод в консоль:



# Урок 5: Диапазоны символов

В приведенном ниже упражнении обратите внимание на то, что все строки совпадения и пропуска имеют шаблон, и используйте скобочную нотацию, чтобы сопоставить или пропустить каждый символ из каждой строки. Имейте в виду, что шаблоны чувствительны к регистру, и a-z отличается от A-Z с точки зрения символов, которым он соответствует (нижний и верхний регистр).

|  |  |  |
| --- | --- | --- |
| **Задача** | **Текст** |  |
| Найти | Ana | To be completed |
| Найти | Bob | To be completed |
| Найти | Cpc | To be completed |
| Пропустить | aax | To be completed |
| Пропустить | bby | To be completed |
| Пропустить | ccz |  |

Код:  
void main() {

  RegExp regExp = RegExp(r'[A-Z][a-z]{2}');

  List<String> input = [

    "Ana",

    "Bob",

    "Cpc",

    "aax",

    "bby",

    "ccz",

  ];

  for (String str in input) {

    if (regExp.hasMatch(str)) {

      print('Совпадение найдено');

    } else {

      print('Совпадение не найдено');

    }

  }

}

Вывод в консоль:



# Задание 6: Последовательность символов

В приведенных ниже строках последняя строка только с одним z-это не то, что мы считаем правильным написанием сленгового "wazzup?". Попробуйте написать шаблон, который соответствует только первым двум написаниям, используя обозначение фигурной скобки выше.

|  |  |  |
| --- | --- | --- |
| **Задача** | **Текст** |  |
| Найти | wazzzzzup | To be completed |
| Найти | wazzzup | To be completed |
| Пропустить | wazup |  |

Код:  
void main() {

  RegExp regExp = RegExp(r'waz{3,5}up');

  List<String> input = [

    "wazzzzzup",

    "wazzzup",

    "wazup",

  ];

  for (String str in input) {

    if (regExp.hasMatch(str)) {

      print('Совпадение найдено');

    } else {

      print('Совпадение не найдено');

    }

  }

}

Вывод в консоль:



# Задание 7: Звездочка и плюс Клини

Ниже приведены несколько простых строк, которые можно сопоставить с помощью метасимволов «звёздочка» и «плюс».

|  |  |  |
| --- | --- | --- |
| **Задача** | **Текст** |  |
| Найти | aaaabcc | | To be completed |
| Найти | aabbbbc | | To be completed |
| Найти | aacc | | To be completed |
| Пропустить | a | |  |

Код:  
void main() {

  RegExp regExp = RegExp(r'a+b\*c+');

  List<String> input = [

    "aaaabcc",

    "aabbbbc",

    "aacc",

    "a",

  ];

  for (String str in input) {

    if (regExp.hasMatch(str)) {

      print('Совпадение найдено');

    } else {

      print('Совпадение не найдено');

    }

  }

}

Вывод в консоль:



# Задание 8: Символы необязательны

В приведенных ниже строках обратите внимание, как множественность слова "файл" зависит от количества найденных файлов. Попробуйте написать шаблон, который использует опциональный метасимвол, чтобы найти только те строки, где был найден один или несколько файлов.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Задача** | | **Текст** | |  |
| Найти | 1 file found? | | To be completed |
| Найти | 2 files found? | | To be completed |
| Найти | 24 files found? | | To be completed |
| Пропустить | No files found. | |  |

Код:  
void main() {

  RegExp regExp = RegExp(r'\d+ file(s)? found\?');

  List<String> input = [

    "1 file found?",

    "2 files found?",

    "24 files found?",

    "No files found.",

  ];

  for (String str in input) {

    if (regExp.hasMatch(str)) {

      print('Совпадение найдено');

    } else {

      print('Совпадение не найдено');

    }

  }

}

Вывод в консоль:



# Задание 9: Отступы

В приведенных ниже строках вы обнаружите, что содержимое каждой строки отступается некоторым пробелом от индекса строки (число-это часть текста, которая должна соответствовать). Напишите шаблон, который может соответствовать каждой строке, содержащей пробелы между номером и содержимым. Обратите внимание, что символы пробела точно такие же, как и любой другой символ, и специальные метасимволы, такие как звездочка и плюс могут быть использованы.

|  |  |  |
| --- | --- | --- |
| **Задача** | **Текст** |  |
| Найти | 1. abc | | To be completed |
| Найти | 2. abc | | To be completed |
| Найти | 3. abc | | To be completed |
| Пропустить | 4.abc | |  |

Код:  
void main() {

  RegExp regExp = RegExp(r'\d\.\s+abc');

  List<String> input = [

    "1.  abc",

    "2.    abc",

    "3.       abc",

    "4.abc",

  ];

  for (String str in input) {

    if (regExp.hasMatch(str)) {

      print('Совпадение найдено');

    } else {

      print('Совпадение не найдено');

    }

  }

}

Вывод в консоль:



# Задание 10: Начало и конец

Обработайте каждую из приведенных ниже строк с помощью этих новых специальных символов.

|  |  |  |
| --- | --- | --- |
| **Задача** | **Текст** |  |
| Найти | Mission: successful | To be completed |
| Пропустить | Last Mission: unsuccessful | To be completed |
| Пропустить | Next Mission: successful upon capture of target |  | |

Код:  
void main() {

  RegExp regExp = RegExp(r'^Mission: successful$');

  List<String> input = [

    "Mission: successful",

    "Last Mission: unsuccessful",

    "Next Mission: successful upon capture of target"

  ];

  for (String str in input) {

    if (regExp.hasMatch(str)) {

      print('Совпадение найдено');

    } else {

      print('Совпадение не найдено');

    }

  }

}

Вывод в консоль:



# Задание 11: Сопоставление групп

Продолжайте и попробуйте использовать это для написания регулярного выражения, которое соответствует только именам файлов (не включая расширение) приведенных ниже PDF-файлов.

|  |  |  |  |
| --- | --- | --- | --- |
| **Задача** | **Текст** | **Результат** |  |
| Захватить | file\_record\_transcript.pdf | file\_record\_transcript | To be completed |
| Захватить | file\_07241999.pdf | file\_07241999 | To be completed |
| Пропустить | testfile\_fake.pdf.tmp |  |  |

Код:  
void main() {

  RegExp regExp = RegExp(r'^(.\*)\.pdf$');

  List<String> input = [

    "file\_record\_transcript.pdf",

    "file\_07241999.pdf",

    "testfile\_fake.pdf.tmp"

  ];

  for (String str in input) {

    final match = regExp.firstMatch(str);

    if (match != null) {

      print('${match.group(1)}');

    } else {

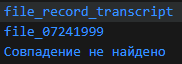
      print('Совпадение не найдено');

    }

  }

}

Вывод в консоль:



# Задание 12: Вложенные группы

Для следующих строк напишите выражение, которое соответствует и фиксирует как полную дату, так и год даты.

|  |  |  |  |
| --- | --- | --- | --- |
| **Задача** | **Текст** | **Результат** |  |
| Захватить | Jan 1987 | Jan 1987 1987 | To be completed |
| Захватить | May 1969 | May 1969 1969 | To be completed |
| Захватить | Aug 2011 | Aug 2011 2011 |  |

Код:  
void main() {

  RegExp regExp = RegExp(r'^([A-Za-z]{3} (\d{4}))$');

  List<String> input = [

    "Jan 1987",

    "May 1969",

    "Aug 2011",

  ];

  for (String str in input) {

    var match = regExp.firstMatch(str)!;

    print('${match.group(1)} ${match.group(2)}');

  }

}

Вывод в консоль:



**Урок 13: Групповая работа**

Ниже приведены несколько различных общих разрешений дисплея, необходимо захватить ширину и высоту каждого дисплея.

|  |  |  |  |
| --- | --- | --- | --- |
| **Задача** | **Текст** | **Результат** |  |
| Захватить | 1280x720 | 1280 720 | To be completed |
| Захватить | 1920x1600 | 1920 1600 | To be completed |
| Захватить | 1024x768 | 1024 768 |  |

Код:  
void main() {

  RegExp regExp = RegExp(r'^(\d+)x(\d+)$');

  List<String> input = [

    "1280x720",

    "1920x1080",

    "1024x768",

  ];

  for (String str in input) {

    var match = regExp.firstMatch(str)!;

    print('${match.group(1)} ${match.group(2)}');

  }

}

Вывод в консоль:



**Задание 14: Условие**

Напишите шаблон, который соответствует только первым двум строкам.

|  |  |  |
| --- | --- | --- |
| **Task** | **Text** |  |
| Найти | I love cats | To be completed |
| Найти | I love dogs | To be completed |
| Пропустить | I love logs | To be completed |
| Пропустить | I love cogs |  |

Код:  
void main() {

  RegExp regExp = RegExp(r'I love (cats|dogs)$');

  List<String> input = [

    "I love cats",

    "I love dogs",

    "I love logs",

    "I love cogs",

  ];

  for (String str in input) {

    if (regExp.hasMatch(str)) {

      print('Совпадение найдено');

    } else {

      print('Совпадение не найдено');

    }

  }

}

Вывод в консоль:

