

Министерство образования Республики Беларусь
Учреждение образования
«Белорусский государственный университет
информатики и радиоэлектроники»
Филиал
«Минский радиотехнический колледж»

Учебный предмет
«Конструирование программ и языка программирования»

Инструкция
по выполнению лабораторной работы №7
«Разработка, отладка и испытание программ с применением регулярных
выражений»

Минск 2024 г.

Лабораторная работа № 7

Тема работы: «Разработка, отладка и испытание программ с применением регулярных выражений»

1 Цель работы

Сформировать умения создавать регулярные выражения и применять их при разработке программ.

2 Задание

Номер варианта соответствует номеру по списку в журнале.

1. Напишите регулярное выражения для проверки строки на: время в формате ММ::СС.
2. Напишите регулярное выражения для проверки строки на: МАС-адрес
3. Напишите регулярное выражения для проверки строки на: IP-адрес (v4)
4. Напишите регулярное выражения для проверки строки на: IP-адрес (v6)
5. Напишите регулярное выражения для проверки строки на: номер телефона в формате +37517-xxx-xx-xx
6. Напишите регулярное выражения для проверки строки на: номер телефона оператора «МТС» в формате +37529-xxx-xx-xx
7. Напишите регулярное выражения для проверки строки на: номер телефона оператора «VELCOM» в формате +37544-xxx-xx-xx
8. Напишите регулярное выражения для проверки строки на: номер телефона оператора «LIFE» в формате +37525-xxx-xx-xx
9. Напишите регулярное выражения для проверки строки на: дата рождения в формате ДД.ММ.ГГГГ или ДД/ММ/ГГГГ
10. Напишите регулярное выражения для проверки строки на: номер телефона оператора «МТС» в формате +37533-xxx-xx-xx.
11. Напишите регулярное выражения для проверки строки на: номер автомобиля в формате YYYUxx-Y, где x – буква, y – цифра.
12. Напишите регулярное выражения для проверки строки на: адрес электронной почты
13. Напишите регулярное выражения для проверки строки на: код цвета в шестнадцатеричном формате
14. Напишите регулярное выражения для проверки строки на: начальные и конечные пробелы
15. Напишите регулярное выражения для проверки строки на: проверку номера кредитной карты

3 Оснащение работы

ПК, среда Visual Studio 2019, MSword.

4 Основные теоретические сведения

Регулярные выражения представляют эффективный и гибкий метод по обработке больших текстов, позволяя в то же время существенно уменьшить объемы кода по сравнению с использованием стандартных операций со строками.

Основная функциональность регулярных выражений в .NET сосредоточена в пространстве имен `System.Text.RegularExpressions`. А центральным классом при работе с регулярными выражениями является класс `Regex`.

Параметры `RegexOptions`

Класс `Regex` имеет ряд конструкторов, позволяющих выполнить начальную инициализацию объекта. Две версии конструкторов в качестве одного из параметров принимают перечисление `RegexOptions`. Некоторые из значений, принимаемых данным перечислением:

- `Compiled`: при установке этого значения регулярное выражение компилируется в сборку, что обеспечивает более быстрое выполнение
- `CultureInvariant`: при установке этого значения будут игнорироваться региональные различия
- `IgnoreCase`: при установке этого значения будет игнорироваться регистр
- `IgnorePatternWhitespace`: удаляет из строки пробелы и разрешает комментарии, начинающиеся со знака `#`
- `Multiline`: указывает, что текст надо рассматривать в многострочном режиме. При таком режиме символы `"^"` и `"$"` совпадают, соответственно, с началом и концом любой строки, а не с началом и концом всего текста
- `RightToLeft`: приписывает читать строку справа налево
- `Singleline`: устанавливает однострочный режим, а весь текст рассматривается как одна строка

Элементы синтаксиса регулярных выражений:

`^`: соответствие должно начинаться в начале строки (например, выражение `@"^пр\w*"` соответствует слову "привет" в строке "привет мир")

`$`: конец строки (например, выражение `@"\w*ир$"` соответствует слову "мир" в строке "привет мир", так как часть "ир" находится в самом конце)

`.`: знак точки определяет любой одиночный символ (например, выражение `"м.р"` соответствует слову "мир" или "мор")

`*`: предыдущий символ повторяется 0 и более раз

`+`: предыдущий символ повторяется 1 и более раз

`?`: предыдущий символ повторяется 0 или 1 раз

`\s`: соответствует любому пробельному символу

`\S`: соответствует любому символу, не являющемуся пробелом

`\w`: соответствует любому алфавитно-цифровому символу

`\W`: соответствует любому не алфавитно-цифровому символу

`\d`: соответствует любой десятичной цифре

\D: соответствует любому символу, не являющемуся десятичной цифрой.

5. Порядок выполнения работы

1. Выделить ключевые моменты задачи.
2. Построить алгоритм и теоретическую объектную модель решения задачи.
3. Запрограммировать полученные алгоритмы и объектную модель.

6. Форма отчета о работе

Лабораторная работа № _____

Номер учебной группы _____

Фамилия, инициалы учащегося _____

Дата выполнения работы _____

Тема работы: _____

Цель работы: _____

Оснащение работы: _____

Результат выполнения работы: _____

7. Контрольные вопросы и задания

1. Регулярное выражение – это...?
2. Какой класс необходимо подключить к проекту для работы с регулярными выражениями?

8. Рекомендуемая литература

1. **Рихтер, Дж.** CLR via C#. Программирование на платформе Microsoft .NET Framework 4.5 на языке C# / Дж. Рихтер. СПб. : Изд-во Питер, 2021. 896 с.
2. **Прайс, М. Дж.** C# 10 и .NET 6. Современная кросс-платформенная разработка / М. Дж. Прайс. СПб : Изд-во Питер, 2023. 848 с.
3. **Васильев, А.Н.** Программирование на C# для начинающих. Особенности языка / А.Н. Васильев. М. : Эксмо, 2022. 528 с.
4. **Фримен, А.** ASP.NET Core 3 с примерами на C# для профессионалов / А. Фримен. СПб. : Изд-во Вильямс, 2021. 1184 с.