|  |  |
| --- | --- |
|  |  |
|  |

ИСТОРИЯ ИЗМЕНЕНИЙ

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| Версия | Описание изменений | Автор | Дата | Утверждено   |  |  | | --- | --- | | Имя | Дата | | |
| <1.0> | Initial version | Alexander\_Kuznetsov1 | <02-Mar-2012> |  |  |
|  |  |  |  |  |  |

СВЯЗАННЫЕ ДОКУМЕНТЫ

|  |  |
| --- | --- |
| Номер | Название документа |
|  |  |
|  |  |

# Задания

Выполните приведенные ниже задания. Задания, помеченные «\*», являются заданиями повышенной сложности и не обязательны для выполнения.

# Задание 1

Написать программу, которая определяет среднюю длину слова во введенной текстовой строке. Учесть, что символы пунктуации на длину слов влиять не должны. Используйте стандартные методы класса String.

# Задание 2

Написать программу, которая удваивает в первой введенной строке все символы, принадлежащие второй введенной строке.

Пример:

Введите первую строку: написать программу, которая

Введите вторую строку: описание

Результирующая строка: ннааппииссаать ппроограамму, коотоораая

# Задание 3

Разработать консольное приложение, которое выводит на экран (в виде таблицы) отличия в параметрах культур:

"ru" vs "en"

"en" vs "invariant"

"ru" vs "invariant"

Необходимо вывести на экране отличия в:

1) формат отображения даты и времени

2) формат отображения числовых данных (разделитель дробной и целой части, разделитель групп разрядов и т.п.)

Целесообразно реализовать отдельный метод, который принимает на входе название культур и выводит отличия на экран. Повторно использовать этот метод (Code Reuse) для вывода различных пар культур.

# Задание 4\*

Проведите сравнительный анализ скорости работы классов String и StringBuilder для операции сложения:

string str = "";

StringBuilder sb = new StringBuilder();

int N = 100;

for (int i = 0; i < N; i++)

{

str += "\*";

}

for (int i = 0; i < N; i++)

{

sb.Append("\*");

}

# Задание 5\*

Напишите программу, которая заменяет все найденные в тексте HTML теги на знак “\_”. Обязательно использовать регулярные выражения.

Пример

Введите HTML текст: **<b>Это</b> текст <i>с</i> <font color=”red”>HTML</font> кодами**

Результат замены: \_Это\_ текст \_с\_ \_HTML\_ кодами

# Задание 6\*

Напишите программу, которая проверяет текстовую строку на соответствие имеющегося в ней текста формату вещественного числа и выводит, в каком формате оно записано. Обязательно использовать регулярные выражения.

1. Число может быть записано в обычной нотации.
2. Число может быть записано в научной нотации (например, 127 = 1.27\*102 = 1.27e2, -0.0055 = -5.5\*10-3 = -5.5e-3).

Пример 1

Введите число: **5**

Это число в обычной нотации

Пример 2

Введите число: **-2.5**

Это число в обычной нотации

Пример 3

Введите число: **5.75e-5**

Это число в научной нотации

Пример 4

Введите число: **\***

Это не число

# Задание 7\*

Напишите программу, которая определяет, сколько раз в тексте встречается время. Постарайтесь учесть, что в сутках только 24 часа, а в часе – 60 минут. Обязательно использовать регулярные выражения.

Пример

Введите текст: **В 7:55 я встал, позавтракал и к 10:77 пошел на работу.**

Время в тексте присутствует 1 раз.

# Требования к оформлению

1. Для работы автоматических фильтров необходимо, чтобы тема письма выглядела следующим образом: [STYYYY\_N]Фамилия\_Имя\_TaskNN. Здесь ST – сокращенное наименование программы (Students Training), YYYY – номер года, N – номер группы, NN – порядковый номер задания в виде двух цифр, напр. 01, 05, 12 и т.д.
2. Каждое задание должно представлять собой отдельный project в рамках общего solution. Название проектов: Task1, Task2 и т.д. Название solution должно формироваться по принципу: Фамилия\_Имя\_Task1.
3. Проект должен быть прислан в zip архиве (rar, 7z и др. не допускаются). Имя архива должно совпадать с именем solution.
4. Не ленитесь сделать ввод/вывод дружественным по отношению к пользователю. Программы, выводящие мигающий курсор на черном фоне, сразу получают «ноль».