

# Декларативно програмиране в процедурните езици

## Курсова задача №1: Регулярни изрази

### Задача 1:

От даден текст да се извлекат всички синоними  $Y_i$  на подадена дума  $X$  и всички думи  $Z_i$ , за които условието “ $X$  е различно от  $Z_i$ ” е налично в текста. Това, дали две думи са синоними, се разпознава чрез вече реализирана двуместна статична функция `AreSynonyms`. По извлечените данни да се формират два списъка - този от всички  $Y_i$  и този от всички  $Z_i$ .

Вход: Низ, който съдържа текста, и низ  $X$ .

Вече реализирани методи: `static bool AreSynonyms(string, string)`.

Изход (на екран): Два списъка от  $Y_i$  и  $Z_i$ .

### Задача 2:

Да се изведат имената на всички възли на подаден xml, които имат поне два атрибута. За всяко име да се изведат стойностите на всички атрибути от възела.

Вход: Низ в xml формат.

Изход (на екран): Поредица от имена на възли, последвани от списък от стойности.

### Задача 3:

В подаден текст всички дати във формат mm/dd/yyyy да се приведат във формат dd/mm/yyyy. Това да се направи само за датите, които със сигурност са във формат mm/dd/yyyy. Спорните дати (тези за които не е ясно в какъв формат са) да се изведат.

Вход: Низ.

Изход: Низ с извършените замени.

Изход (на екран): Поредица от спорните дати.

### Задача 4:

От даден текст да се извлекат всички условия от вида “ $X \geq n$ ” или “ $X \leq m$ ”, където  $n$  и  $m$  са числа, а  $X$  е подадена дума. По извлечените данни да се определят точните долна и горна граници за  $X$ .

Вход: Низ, който съдържа текста, и низ  $X$ .

Изход (на екран): Точните долна и горна граници на  $X$ .

### Задача 5:

В подаден текст, всички суми в евро или долари (€, EUR, \$, USD) да се приведат в лева по подадени курсове. След преобразуването да се изведат всички суми над 1000лв.

Вход: Низ, коефициент за преобразуване от евро в лева и коефициент за преобразуване от долари в лева.

Изход: Низ с преобразувани суми.

Изход (на екран): Поредица от сумите над 1000лв.

### Задача 6:

В параграфите на подаден html, всички получерни текстове да се преобразуват до нормален текст, а курсивните текстове да станат в получер фонт. Счита се, че няма вложени параграфи в html-a.

Вход: Низ в html формат.

Изход: Преобразуван низ в html формат.

### Задача 7:

Да се провери, че корена на подаден xml е правилно затворен и че всички листа на xml-a са правилно затворени.

Вход: Низ в xml формат.

Изход: True/False

**Задача 8:**

В даден текст, да се проверят за грешни изчисления всички изречения от вида “Корените на уравнението  $a.x + b.x + c = 0$  са  $x_1 = d$  и  $x_2 = e$ .”. Грешните изречения да се изведат.

Вход: Низ.

Изход (на екран): Поредица от грешните изречения.

**Задача 9:**

От даден код на C# да се извлекат имената на всички атрибути на публични методи (без значение от броя на аргументите) и имената на всички атрибути на protected методи, които имат поне два аргумента.

Вход: Низ, който съдържа C# код.

Изход (на екран): Поредица от имената на намерените атрибути.

**Задача 10:**

Да се “инвертират” всички редове в таблици в даден html (т.е. последните колонии стават първи, предпоследните - втори и т.н.). Счита се, че няма вложени таблици в html-a.

Вход: Низ в html формат.

Изход: Преобразуван низ в html формат.

**Задача 11:**

В текст на английски език да се заменят всички изрази от вида “Xly Y” с “Y, in a X way,” или “Y, in an X way,” (където X и Y са думи).

Вход: Низ.

Изход: Низ с извършените замени.

**Задача 12:**

От даден текст да се извлекат всички условия от вида “X е или  $Y_1$ , или  $Y_2$ , ... или  $Y_k$ ” или “X не е нито  $Z_1$ , нито  $Z_2$ , ... нито  $Z_k$ ”, където X е подадена дума, а  $Y_i$  и  $Z_i$  са думи от текста. По извлечените данни да се формират два списъка - този от всички  $Y_i$  и този от всички  $Z_i$ .

Вход: Низ, който съдържа текста, и низ X.

Изход (на екран): Двата списъка от  $Y_i$  и  $Z_i$ .

**Задача 13:**

Да се коригират (белите пространства да се заменят със съответното escape-ване, а “/” да се измени на “/”, освен след “http:” и “https:”) и нормализират (да се добави “http://” там, където липсва, а има “www.”) всички хипервръзки в подаден html.

Вход: Низ в html формат.

Изход: Преобразуван низ в html формат.

**Задача 14:**

От даден текст да се извлекат всички съществителни, за които в същото изречение, в което се намират, има спомената сума в лева, евро или долари (€, EUR, \$, USD, лв). Това, дали една дума е съществително, се разпознава чрез вече реализирана едноместна статична функция IsNoun.

Вход: Низ, който съдържа текста.

Вече реализирани методи: static bool IsNoun(string).

Изход (на екран): Поредица от всички съществителни.

**Задача 15:**

Да се изведат текстовете от всички листа на подаден xml, в които е дефиниран подаден атрибут с подадена стойност, и също така в текста се спомената тази стойност като отделна дума. При извеждането да се изведе и името на възела с атрибута.

Вход: Низ в xml формат, низ с име на атрибут и низ със стойност на атрибут.

Изход (на екран): Поредица от имената на листата и текстовете.

**Задача 16:**

Да се сортират всички списъци от дати в подаден текст (дайте са разделени със запетайки).

Вход: Низ.

Изход: Низ със сортирани списъци.

**Задача 17:**

От даден текст да се извлекат всички изречения, които съдържат поне един синоним и поне един антоним на дадена дума *X*. Това, дали две думи са синоними или антоними, се разпознава чрез вече реализирани двуместни статични функции `AreSynonyms` или `AreAntonyms`.

Вход: Низ, който съдържа текста, и низ *X*.

Вече реализирани методи: `static bool AreSynonyms(string, string)`, `static bool AreAntonyms(string, string)`.

Изход (на екран): Поредица от намерените изречения.

**Задача 18:**

В параграфите на подаден `html`, текста да се “инвертира” (т.е. получените текстове да се преобразуват до нормален текст, а нормалните текстове да станат в получер фонт). Считат се, че няма вложени параграфи в `html`-а.

Вход: Низ в `html` формат.

Изход: Преобразуван низ в `html` формат.

**Задача 19:**

Да се изведат всички изречения от подаден текст, в които се срещат поне 4 дати, като между датите има поне една дума.

Вход: Низ.

Изход (на екран): Поредица от изреченията.

**Задача 20:**

В подаден текст да се коригират всички сгрешени думи и сгрешена пунктоация. Сгрешени думи се разпознават с вече реализирана едноместна статична функция `HasWrongSpelling` на която се подава една дума, а сгрешената пунктоация се разпознава с функция `HasWrongPunctuation`, на която се подава цяло изречение. Функции `CorrectSpelling` и `CorrectPunctuation` връщат съответно коригираните варианти на дума и изречение.

Вход: Низ, който съдържа текста.

Вече реализирани методи: `static bool HasWrongSpelling(string)`, `static bool HasWrongPunctuation(string)`, `static string CorrectSpelling(string)`, `static string CorrectPunctuation(string)`.

Изход: Коригирания текст.

**Задача 21:**

Да се изведат всички стойности на атрибути от подаден `xml`, които са размер (KB, MB, GB и т.н.). Да се изведат и имената на атрибутите и възлите, в които се намира стойността.

Вход: Низ в `xml` формат.

Изход (на екран): Поредица от име на възел, име на атрибут и стойност-размер.

**Задача 22:**

От подаден код на `C++/C#/Java` да се извлекат имената на всички статични методи, чиято реализация не започва на същия ред, на който са декларирани.

Вход: Низ, който съдържа `C++/C#/Java` код.

Изход (на екран): Поредица от имената на намерените методи.

**Задача 23:**

В подаден текст на български език да се коригират всички грешки с неправилна употреба на “с” или “със”, както и на “в” или “във”. Също така да се коригират всички неправилни членувания на думи,

завършващи на “т” или “щ”. За тази цел да се изпълва вече реализирана едноместна статична функция `GetUndefined`, която по подадена дума дава нейния нечленуван вариант.

Вход: Низ, който съдържа текста.

Вече реализирани методи: `static string GetUndefined(string)`.

Изход: Кorigирания текст.

#### **Задача 24:**

От подаден `html` да се извлекат всички съществителни, които се намират в параграф и са оцветени в червено. Това, дали една дума е съществително, се разпознава чрез вече реализирана едноместна статична функция `IsNoun`. Счита се, че няма вложени параграфи в `html-a`.

Вход: Низ в `html` формат.

Вече реализирани методи: `static bool IsNoun(string)`.

Изход (на екран): Поредица от всички съществителни.

#### **Задача 25:**

От даден код на `C#`, който съдържа само публични методи, да се извлекат имената на всички методи, които са `yield` генератори.

Вход: Низ, който съдържа `C#` код.

Изход (на екран): Поредица от имената на намерените методи.