

**УНИВЕРСИТЕТ ПО БИБЛИОТЕКОЗНАНИЕ И ИНФОРМАЦИОННИ ТЕХНОЛОГИИ**

**СПЕЦИАЛНОСТ “ИНФОРМАТИКА И КОМПЮТЪРНИ НАУКИ”**

**КУРСОВА РАБОТА**

**Изготвил:**  **Научен ръководител:**

(проф. Иван Иванов)

Мартин Върбанов

Редовно обучение

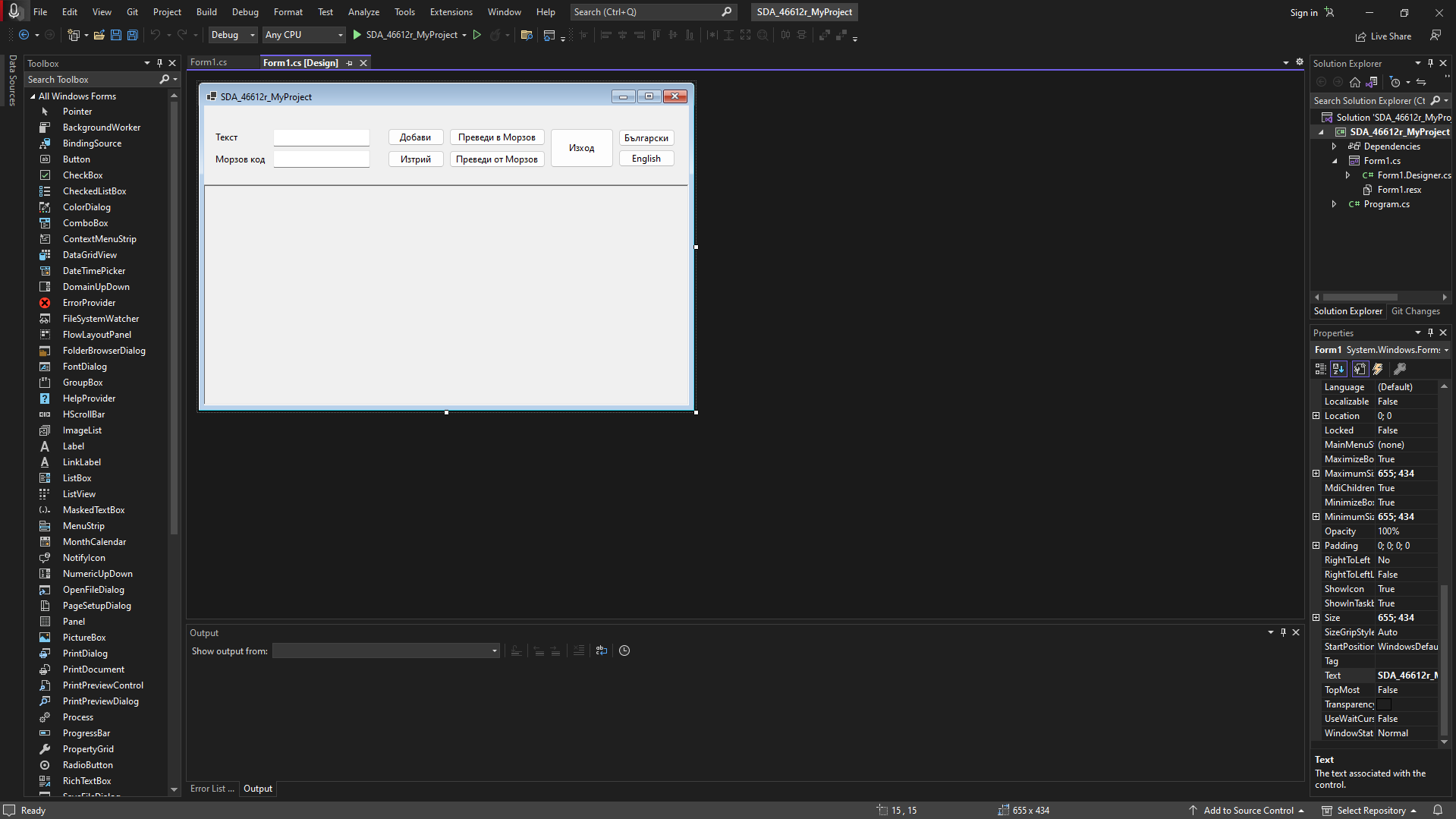
Ф.№ 46612р

София

2023

За дизайна на формата съм използвала:

* 2 Label
* 2 TextBox
* 1 RichTextBox
* 7 Button



Има 2 етикета до текстовите кутии, които показват какво трябва да се въвежда в тях. Двете текстови кутии съответно си имат имена tbText и tbMorse.

Бутонът “Добави” прибавя нов символ към наличните, като за това е необходимо и двете текстови полета да са запълнени, текстът трябва да има символ или няколко такива, докато морзовият код трябва задължително да бъде морзов код, като в този случай разстояния не се допускат (само „-“ и „.“).

Бутонът Изтрий, той работи на по-интересен принцип, за него може и в двете полета да има въведени данни. Като той има следните функции:

* Ако няма въведени данни или данните са грешни изкарва съобщение: „Няма такъв елемент“.
* Ако има намерен символ ИЛИ морзов код изтрива съответният елемент от дървото и от речника, като излиза съобщение: „Успешно премахнат елемент: “ и след това показва елементът, който е премахнат.
* Ако има немерени символ И морзов код ще се приоритизира въведеният символ и ще игнорира морзовият код (при изтриване се изтриват и символа и морзовият код, приоритизирането е за търсене на елемента).

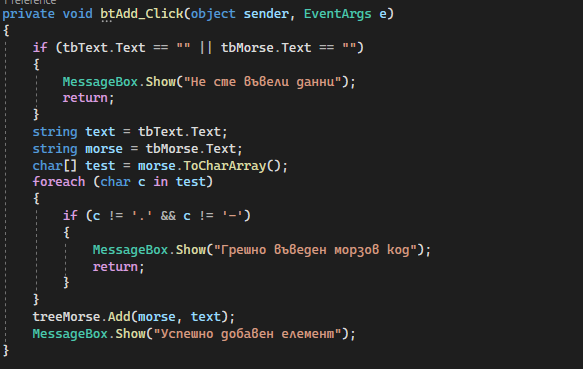
Бутонът “Преведи в Морзов” разделя въведеният текст по разстояния и ги слага в масив, така разделям съобщението на отделните му дими. След това разделя всяка дума на съответните ѝ букви и търси всяка буква в речник, в който срещу всяка буква има въведен съответен морзов код.

Бутонът „Преведи от Морзов“ отново разделя въведеният текст на отделни думи, които слага в масив, след това се вика функцията Find() за всяка една дума, като се търси в дървото („-“ мести на дясно по root, „.“ мести на ляво по root). (Важно е след като се намери елемента да се дървото да се върне в първоначалният си състояние с ReturnToRoot() функцията).

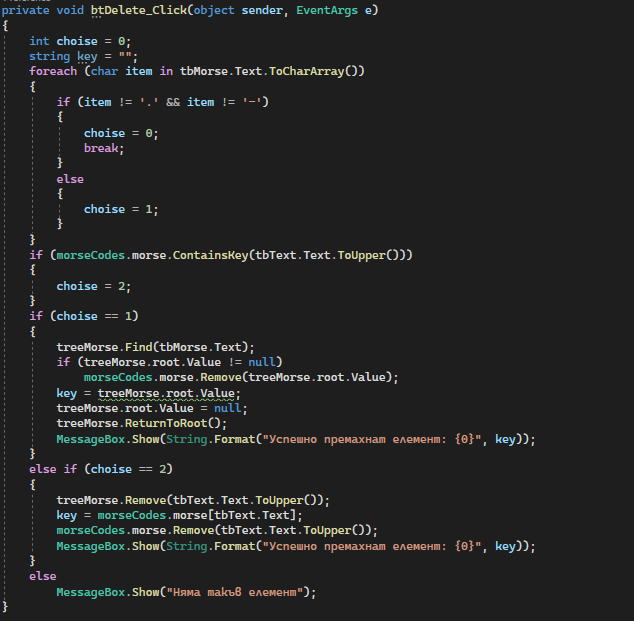
Бутонът „Изход“ излиза от формата.

Бутонът „Български“ сменя езика на всички бутони на български, зарежда речникът на български и дървото на български.

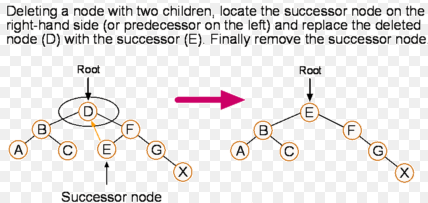
Бутонът „English“ сменя езика на всички бутони на английски, зарежда речникът на английски и дървото на английски.

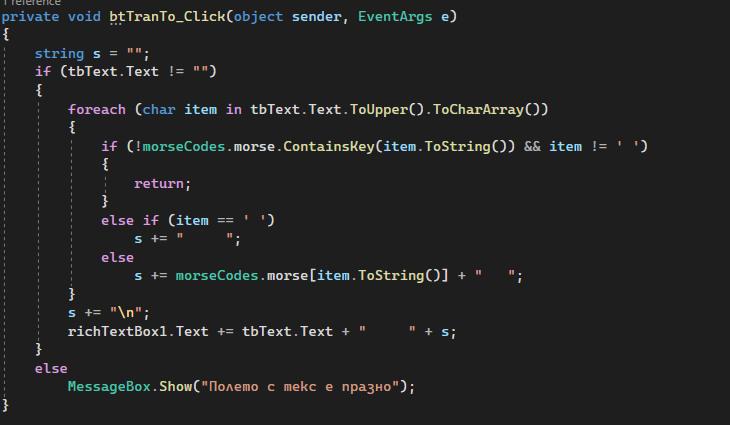


Това е командата за бутона „Добави“. При натискането на бутона ще излезе едно то 3-те съобщения: „Не сте въвели данни“; „Грешно въведен морзов код“; „Успешно добавен елемент“.

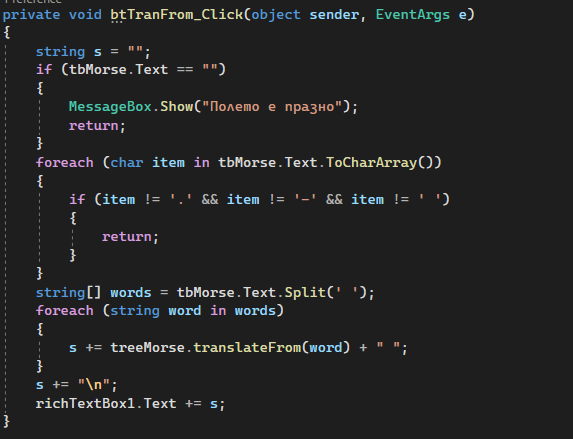


Изтрива като намери елемента в морзов код, след което изтрива съответните стойности в речника. Прави стойността null и се връща в root с функцията ReturnToRoot() важно е да се забележи, че е различно от нормално дърво където изтриването работи като родителя посочва лявото или дясното дете и към дясното или лявото дете съответно на посоченото се слага обратното и след това същото дете се свързва с родителя, трябва да забележим, че използването са същата техника ще развали морзовия код, защото за него е важно всяко листо да си седи на едно и също място.

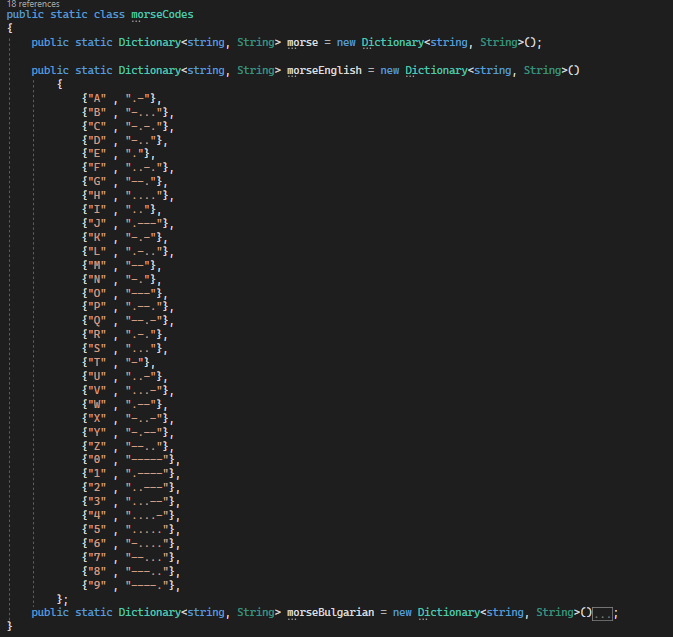




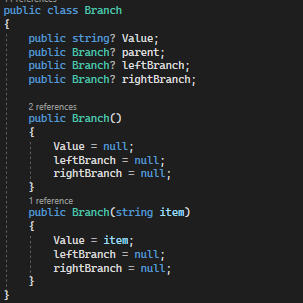
Кода за превеждане към морзов код, търси буквите в речника.



Проверява дали полето е празно. Минава през всички символи и проверява дали са „.“; „-“; „ “. При което ги разделя на букви и превежда всяка буква като изпълнява инструкциите за намиране на буква и връща стойността.

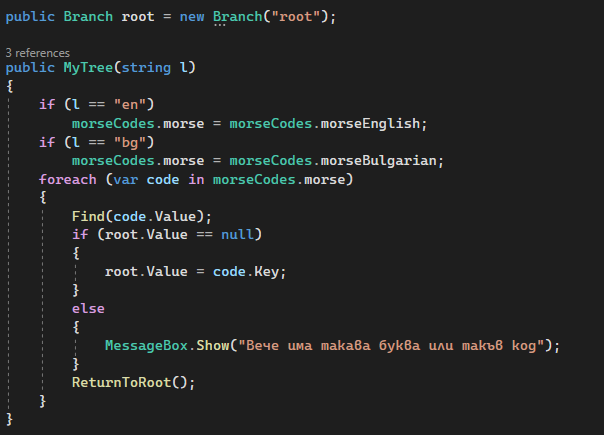


Използвам public static class morseCodes, за да мога да го достигам отвътре на класа MyTree, както и извън класа, какъвто е примера за някой от бутоните. Имам един речник morse, който може да приема стойностите на morseEnglish или morseBulgarian, което зависи от бутоните за език. В този скриншот речникът morseBulgarian е минимизиран за по-голяма четливост.



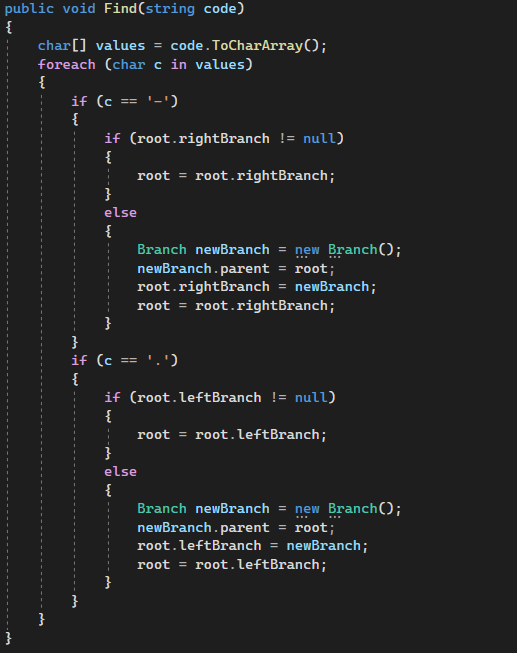
Клас Branch, този клас има променливи Value; parent; leftBranch; rightBranch има два конструктора за създаване със и без задаване на стойност на Value.

КОДЪТ ОТ ТУК Е САМО ЗА КЛАСА public class MyTree



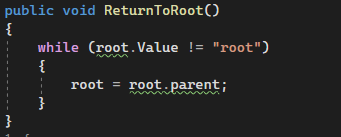
Създавам root със стойност „root“ която после ще използвам, за да се връщам в началото на дървото.

Конструкторът на класа „MyTree“ приема стрингова стойност, с която после се зарежда Българският или Английският речник. След това намира стойността на всеки ключ в речника и го зарежда в дървото, като използва стойността в речника за да намери правилното място.

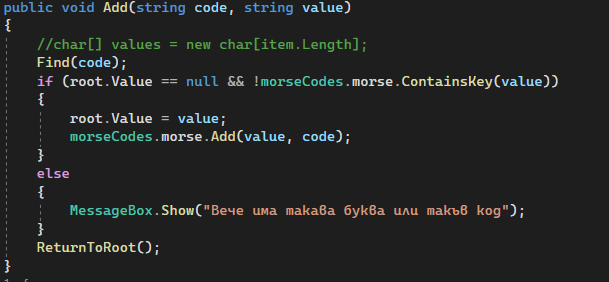


Във Find() се слага стринг от комбинация в морзов код, като ако стойността е „-“ ще тръгне на дясно, ако е „.“ Ще тръгне наляво. Ако има създаден клон ще тръгне по клона, в противен случай ще създаде нов клон, на който за стойността parent се слага root, така след като root стане съответно leftBranch или rightBranch ще може да се върне обратно ползвайки

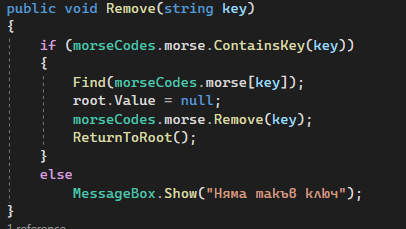
root = root.parent



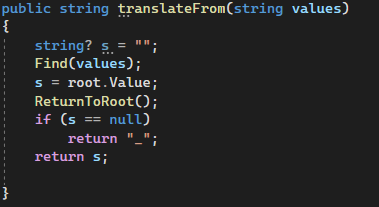
Функцията ReturnToRoot(),тя използва while цикъл, докато стойноста на root.Value НЕ е равна на „root“ и след това root = root.parent, така, винаги, когато се извика функцията състоянието на дървото ще се върне в началното си пложение.



За добавяне първо намира мястото, където трябва да добави, като Find() ще създава клони докато стигне до правилното място. След това ще провери дали няма записана стойност на това място, и дали същата стойност вече не е записана в речника, което значи, че същата стойност е налична някъде другаде в дървото.



За изтриване на стойност при наличен ключ, първо намира стойността в речника и след това намира мястото в дървото, при което го прави равно на null и след това премахва същата стойност от речника, и тъй като е използва Find() използваме ReturnToRoot(), за да се върнем в началото на root.



При превод първо намира дадената стойност прави променлива string s да е равна на тази стойност и след това пак се връща в началното положение, ако стойността не е намерена (равна на null), вместо стойност връща долна черта, което изглежда по този начин:

