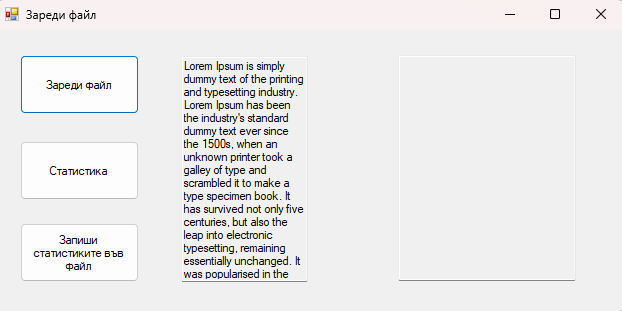
# Упражнения: Графични приложения (GUI Apps)

## Четене на файл

Създайте GUI приложение, което да **зарежда текст**, да **показва статистика** и да я **записва във файл**.

### Пример



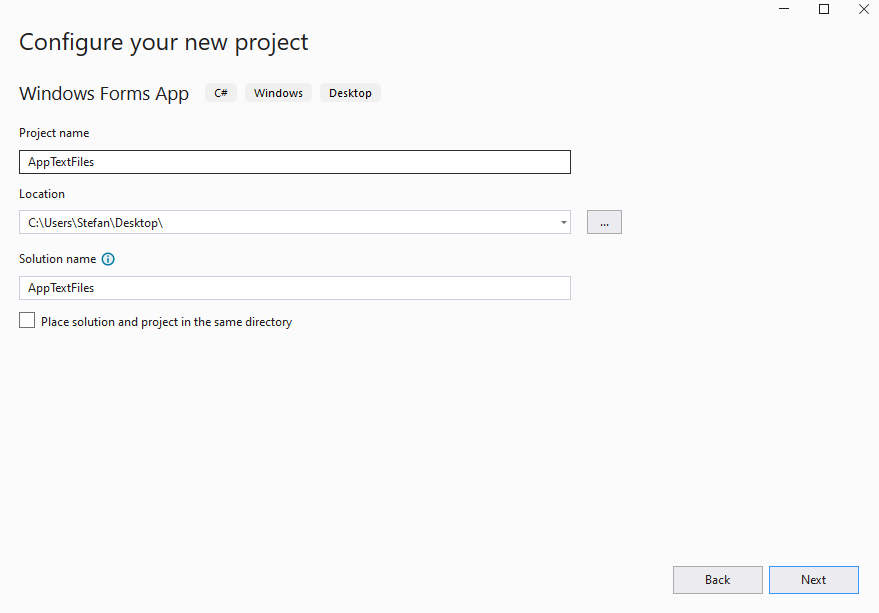
### Насоки

1. Изберете **[Create a new project]**. В търсачката напишете "Windows Forms", изберете **първата опция** и натиснете **[Next]**

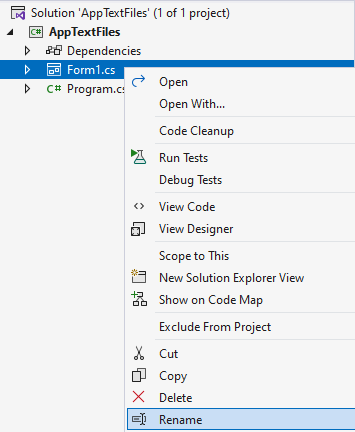
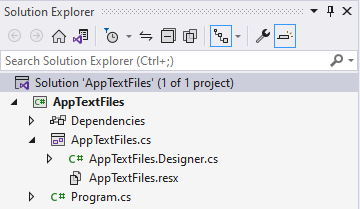
Graphical user interface, application

Description automatically generated

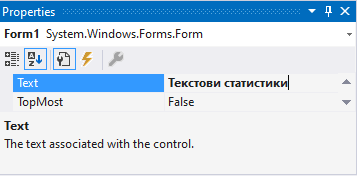
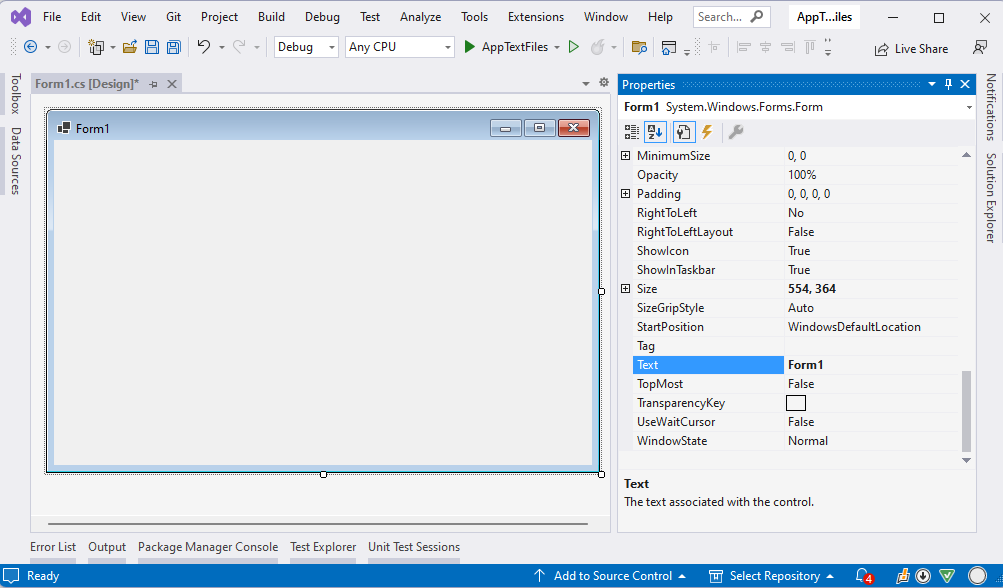
1. Задайте **смислено име** на приложението, което описва неговото предназначение – например **AppTextFile**. Натиснете **[Next]** 🡪 **[Create]**



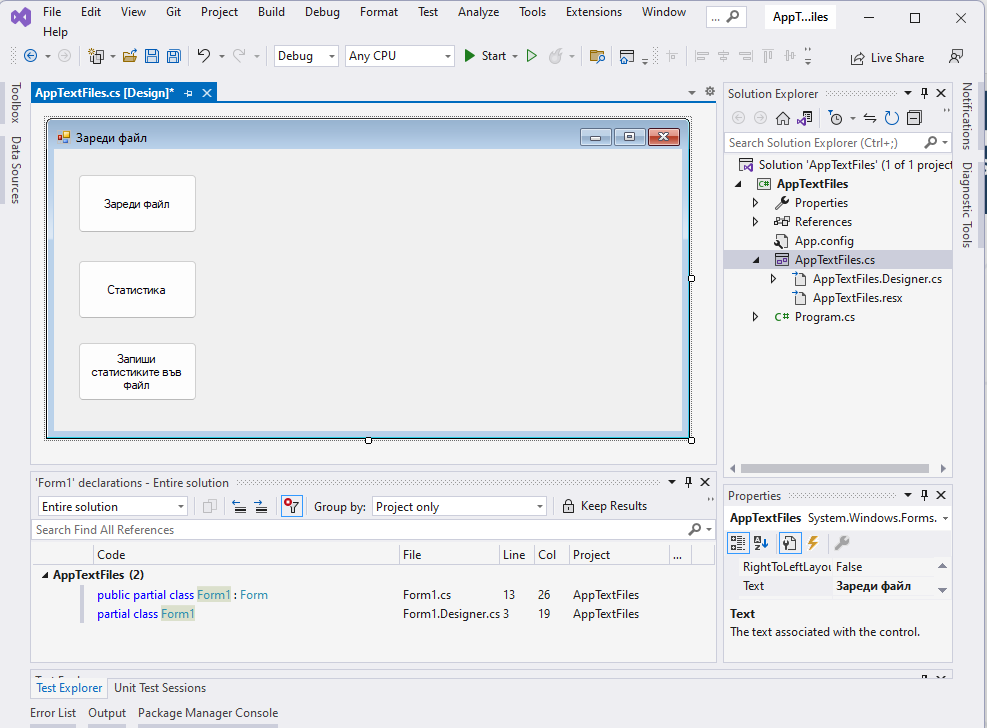
1. Променете **името** на формата: "**Form1**" 🡪 "**AppTexrFiles**"



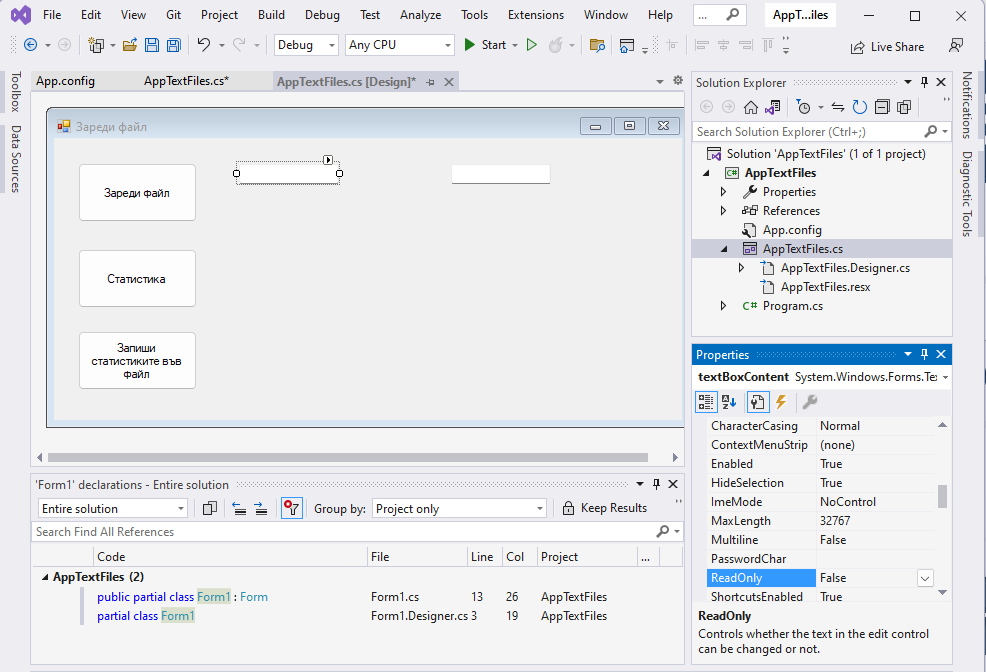
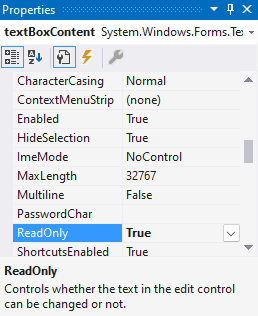
1. Променете **заглавието** на формата: "**Form1**" 🡪 "**Текстови статистики**"



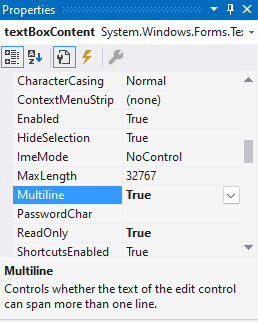
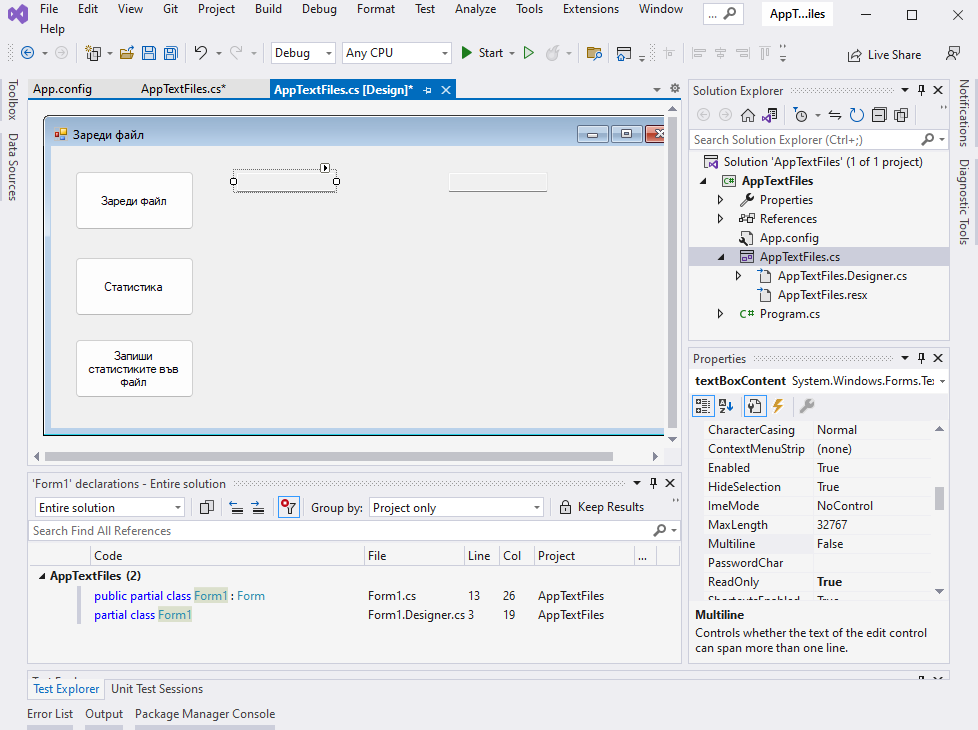
1. Добавете необходимите контроли – два текстови полета (**TextBox**) и три бутона (**Button**)



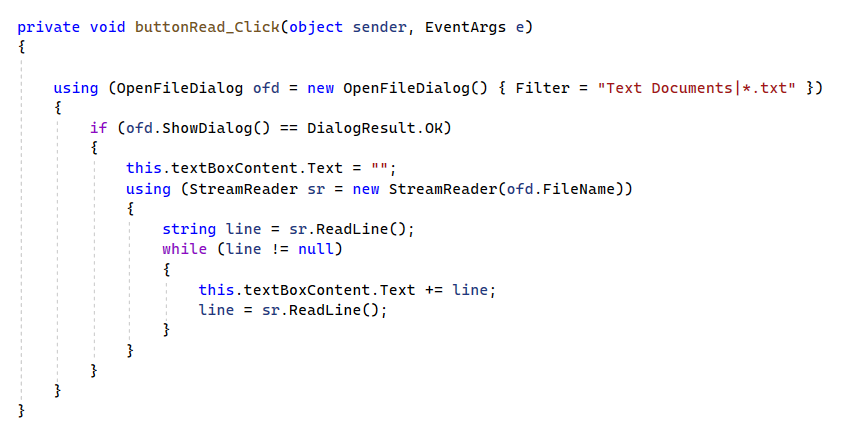
1. Променете **имената** на контролите
   * textBox1 🡪 textBoxContent
   * textBox2 🡪 textBoxData
   * button1 🡪 buttonRead
   * button2 🡪 buttonStatistics
   * button3 🡪 buttonWrite
2. Променете **текста** на контролите:
   * "button1" 🡪 "Зареди файл"
   * "button2" 🡪 "Статистика на думите"
   * "button3" 🡪 "Запиши статистиките във файл"
3. След като включим свойството **ReadOnly** на textbox, съдържанието му няма да може да се **променя**



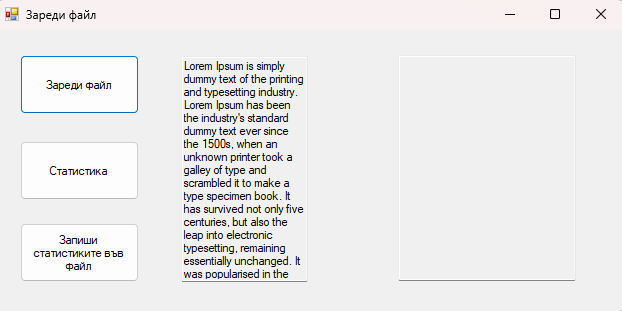
1. След като включим свойството **Multiline** на textbox, съдържанието на файла ще бъде на редове



1. Добавете метод-обработчик с име **buttonRead\_Click()** на събитието **Click** на бутона за добавяне на събитие.



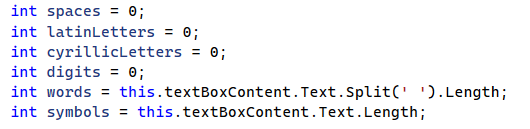
1. За да стартирате приложението, изберете **Debug** 🡪 **Start Without Debugging** или [**Ctrl+F5**]. Натиснете **заредете файл**

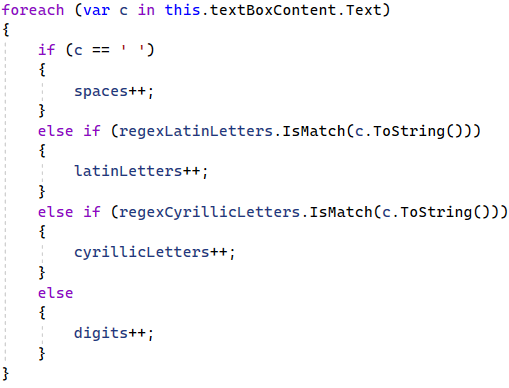


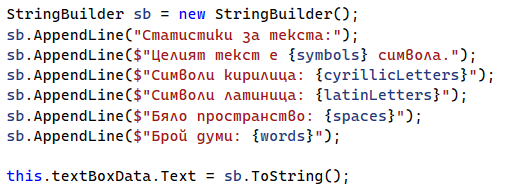
## Четене на файл

1. Добавете **метод-обработчик** с име **buttonStatistic\_Click()** на събитието **Click** на бутона за добавяне на събитие.

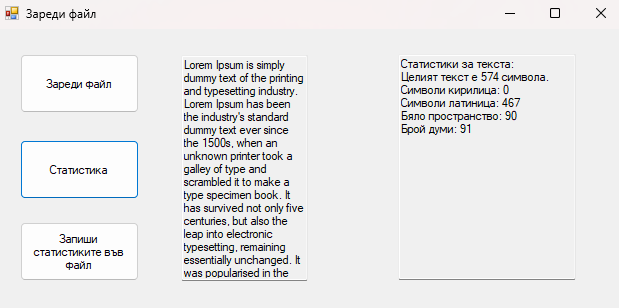
* Създайте променливи за броене на символи



* Използвайте regex за проверка на символите
* Започнете да обхождате символите
* Използваме **StringBuilder**, за да визуализираме текста

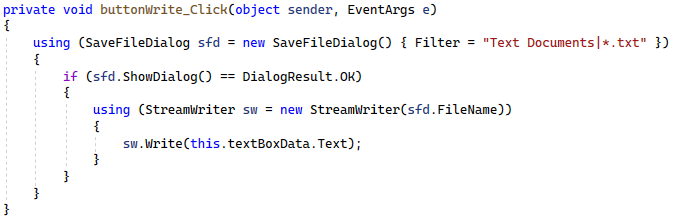


1. За да стартирате приложението, изберете **Debug** 🡪 **Start Without Debugging** или [**Ctrl+F5**]. Натиснете **заредете файл.** След това **статистика**.

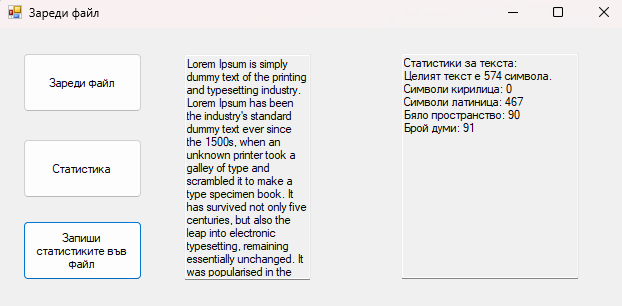


## Четене на файл

1. Добавете **метод-обработчик** с име **buttonWrite\_Click()** на събитието **Click** на бутона за добавяне на събитие.



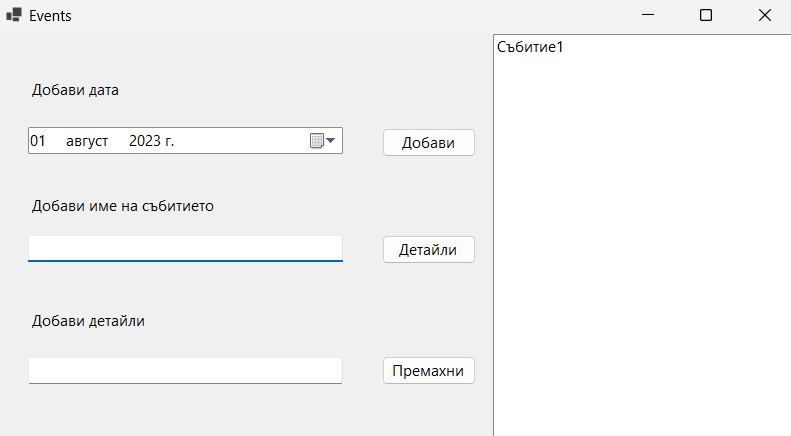
1. За да стартирате приложението, изберете **Debug** 🡪 **Start Without Debugging** или [**Ctrl+F5**]. Натиснете **заредете файл.** След това **статистика.** Накрая натиснете **запиши статистиките във файл**



## Събития

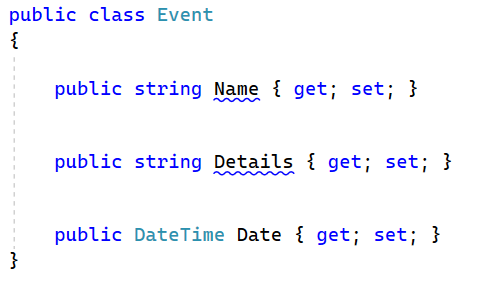
Напишете GUI приложение, която запазва и премахва събития във файл, въведени от потребителя.

### Пример

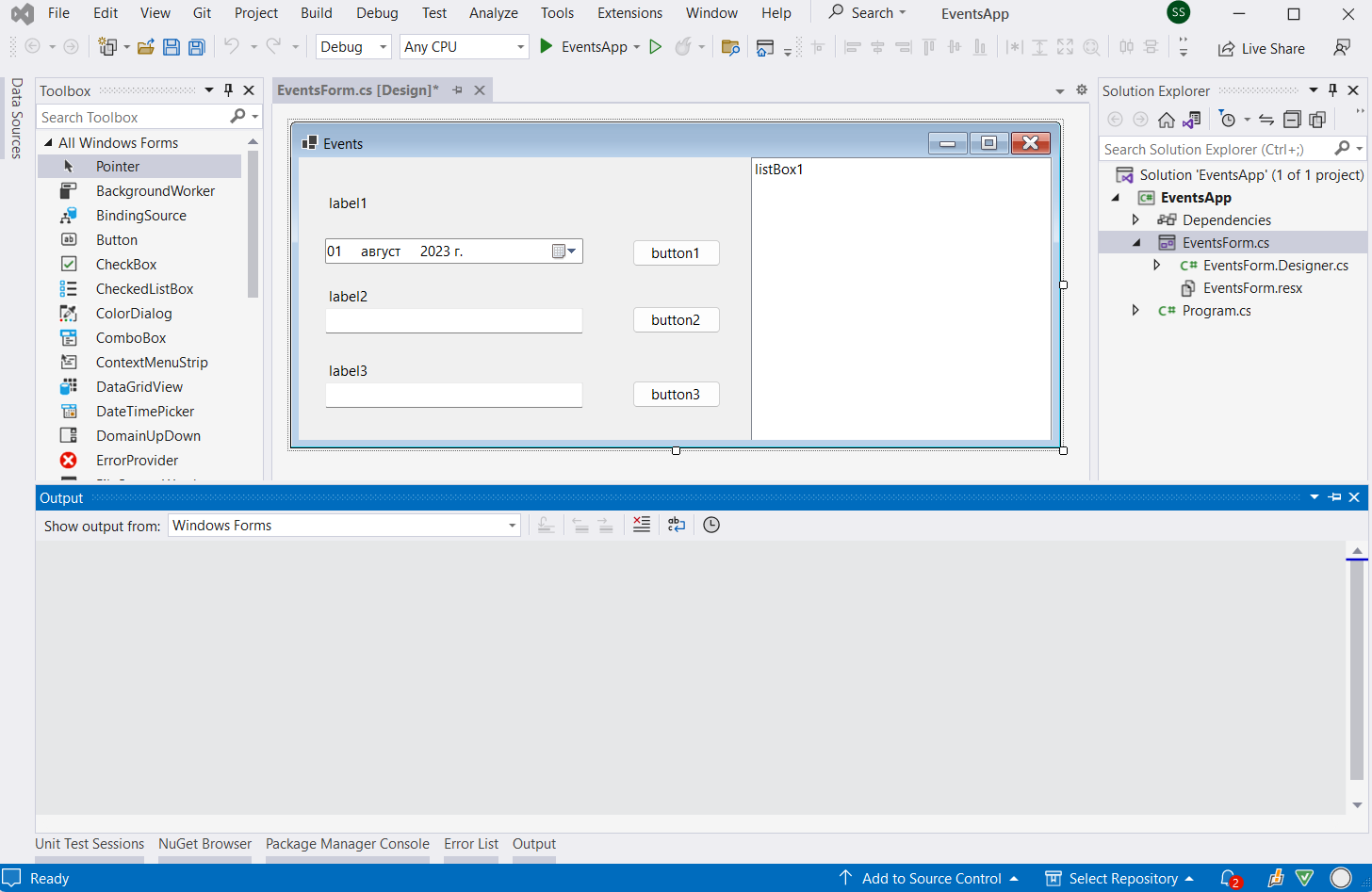


### Насоки

1. Създайте нов **Windows Forms проект** и му задайте **подходящо име**, например "**EventsApp**":
2. Създайте папка “Models”. В нея създайте обект Event. Трябва да има **име**, **детайли** и **дата**:



1. Добавете необходимите **контроли** – **два текстови** **полета** (TextBox), **три** **надписа** (Label), **три бутона** (Button), **един** **dateTimePicker** и **един listbox**:



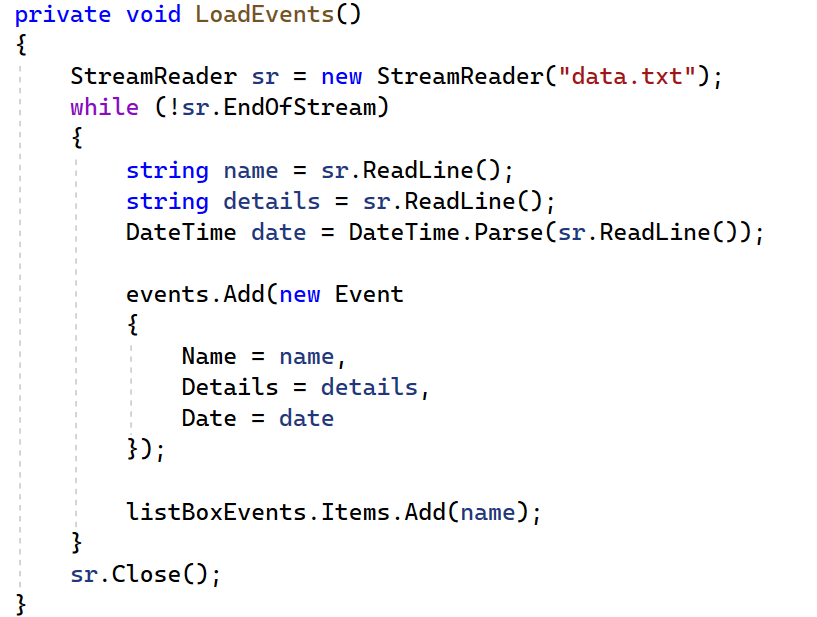
1. Променете **имената** на контролите:

* textBox1 🡪 textBoxName
* textBox2 🡪 textBoxDetails
* dateTimePicker1 🡪 dateTimePicker
* listBox1 🡪 listBoxEvents
* label1 🡪 labelDate
* label2 🡪 labelName
* label3 🡪 labelDetails
* button1 🡪 buttonAdd
* button2 🡪 buttonDetails
* button3 🡪 buttonRemove

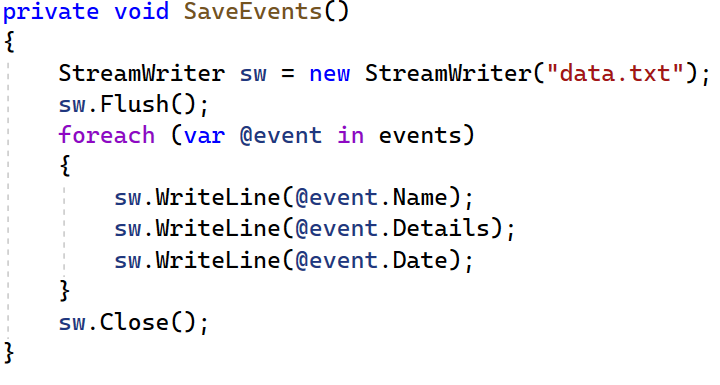
1. Променете **текста** на контролите:

* button1 🡪 Добави
* button2 🡪 Детайли
* button3 🡪 Премахни
* label1 🡪 Добави дата
* label2 🡪 Добави име на събитието
* label3 🡪 Добави детайли

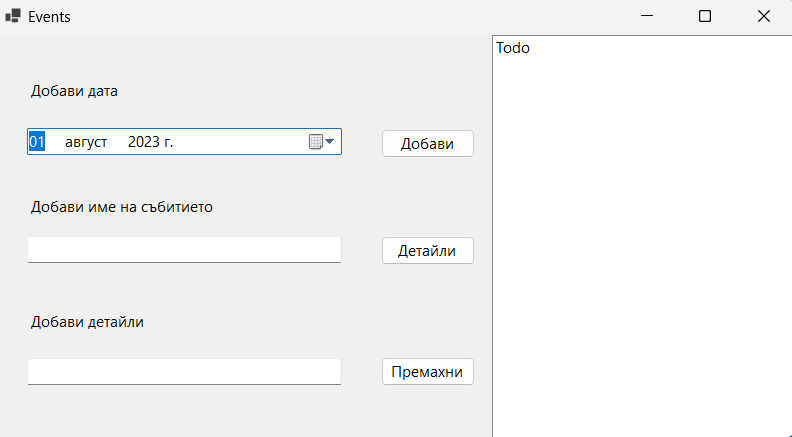
1. Добавете **метод** с име **LoadEvents()** за зареждане на събитията



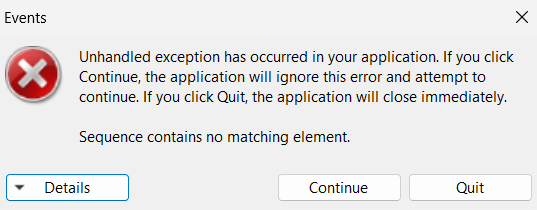
1. Добавете **метод** с име **SaveEvents()** за запаметяване на събитията



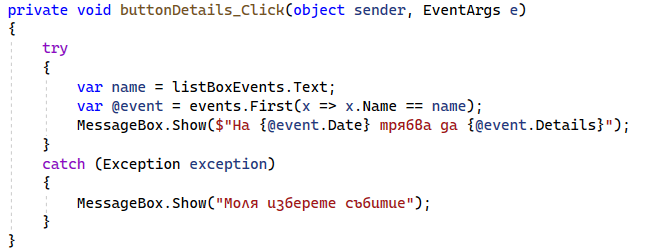
1. Добавете **метод-обработчик** с име **buttonAdd\_Click** на събитието **Click** на бутона за добавяне на събитие. След всяко събитие добавяне трябва да извикате метода **SaveEvents()**
2. Добавете **метод-обработчик** с име **buttonDelete\_Click** на събитието **Click** на бутона за премахва на събитие. След всяко събитие изтриване трябва да извикате метода **SaveEvents()**
3. Добавете **метод-обработчик** с име **buttonDetails\_Click** на събитието **Click** на бутона за показване на детайли на събитие. Използвайте message box и отпечатайте: "На **{дата на събитието}** трябва да **{детайли на събитието}**"
4. За да **стартирате** и **тествате** приложението, използвайте клавишната комбинация **[Ctrl+F5]**



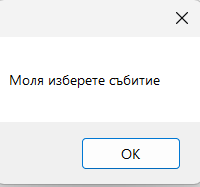
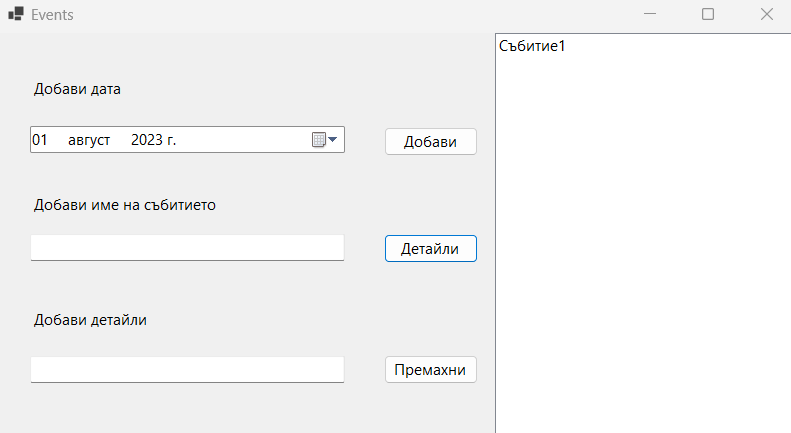
1. **Тествайте** приложението. Какво става, ако опитаме да натиснем детайли без да сме избрали събитие? Ще **възникне грешка** (exception) по време на изпълнение и приложението:



1. Как можем **да поправим този проблем**? Искаме вместо приложението „да се чупи“, да излиза подходящо съобщение, не са избрали събитие:



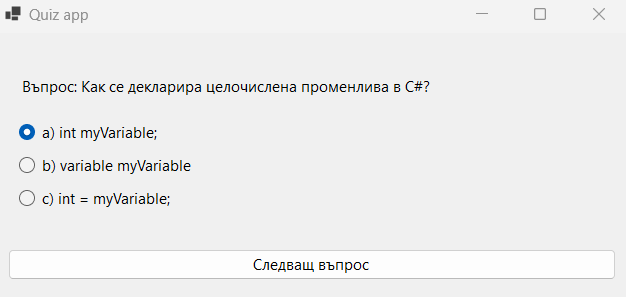
Кодът по-горе хваща възможния **проблем при парсване на число** и **пресмятане на сумата**, след което визуализира резултат “**error**“. Ще учим по-нататък в нашия курс как се работи с **try-catch** конструкцията в C# за хващане на грешки.

1. Стартирайте отново и тествайте какво се получава при натискане на бутона детайли:

## Quiz игра

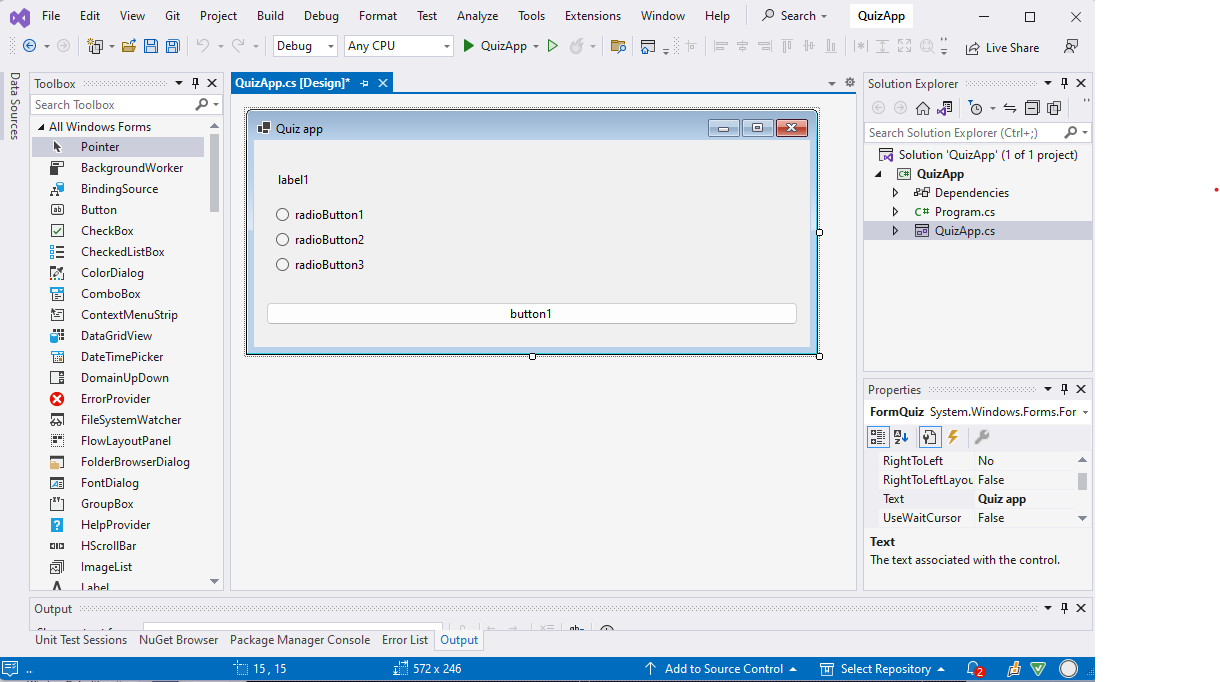
Създайте **GUI приложение**, което да прочита въпроси и отговори от текстови файл ("quiz\_questions.txt") и да предоставя на потребителя възможност да избере верния отговор.

### Пример



### Насоки

1. Създайте нов **Windows Forms проект** и му задайте **подходящо име** (например "**QuizApp**")
2. Създайте променлива за броя на въпросите
3. Създайте променлива за броя на правилните отговорени въпроси
4. Създайте променлива за четене на файла с име **sr**
5. Променете **името** на формата: "**Form1**" 🡪 "**FormQuiz**"
6. Променете **заглавието** на формата: "**Form1**" 🡪 "**Quiz app**"
7. Добавете необходимите контроли – **4 надписа** (Label), **3 радио бутона** (Radio button) **1 бутон** (Button)



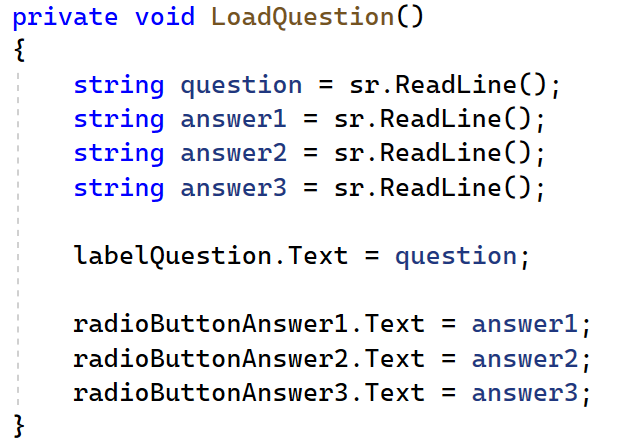
1. Променете **имената** на контролите:

* "label1" 🡪 "labelQuestion"
* "button1" 🡪 "buttonChangeQuestion"
* "radioButton1" 🡪 "radioButtonAnswer1"
* "radioButton2" 🡪 "radioButtonAnswer2"
* "radioButton3" 🡪 "radioButtonAnswer3"

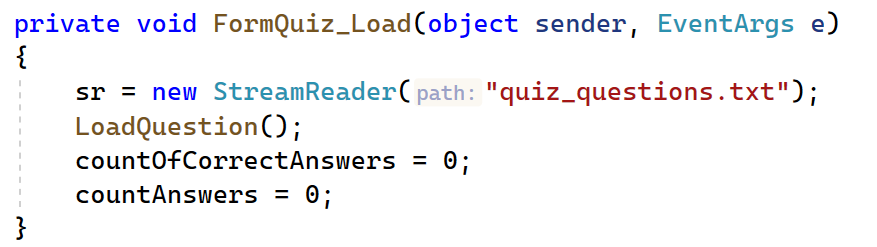
1. Променете **текста** на контролите:

* " button1" 🡪 "Следващ въпрос"
* "radioButtonAnswer1" 🡪 ""
* " radioButtonAnswer2" 🡪 ""
* " radioButtonAnswer3" 🡪 ""
* "label1" 🡪 ""

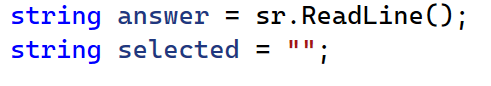
1. Добавете **метод** с име **LoadQuestion(),** име който да зарежда следващия въпрос

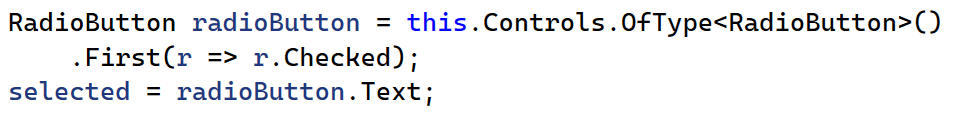


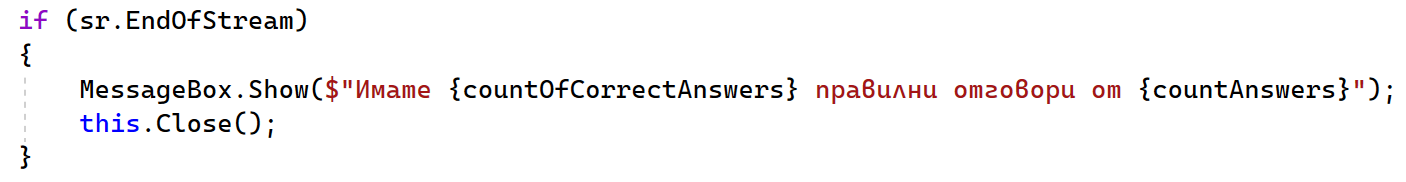
1. Добавете **метод обработчик** с име **FormQuiz\_Load**, който инициализира променливите



1. Добавете **метод-обработчик** с име **buttonChangeQuestion\_Click** на събитието **Click** на бутона за следващ въпрос
   * Създаваме променлива, която чете отговора от файла
   * Създаваме променлива, която ще използваме за отговора на потребителя



* + Увеличаваме броя на отговорите с едно
  + Присвояваме на променливата selected текста на избрания радио бутон
  + 
  + Ако избрания отговор е правилен, увеличаваме броя на правилните отговорени въпроси и изписваме с **message box** "Вие отговорихте правилно"
  + В противен случай, изписваме "Правилният отговор е **{answer}**"
  + След това правим проверка да сме прочели целия файл. Ако е така, затваряме програмата и отпечатваме "Имате **{countOfCorrectAnswers}** правилни отговори от **{countAnswers}**"



* + Накрая извикваме метода **LoadQuestion()**, за да заредим следващия въпрос

1. За да **стартирате** и **тествате** приложението, използвайте клавишната комбинация **[Ctrl+F5]**

