

## Windows приложения

Пътуваме към едно от любимите ми места – Казабланка, Мароко! То се намира в западната част на Мароко на Атлантическия океан. В миналото Казабланка е бил проспериращ град, известен като Алфа. Той е бил разрушен от португалците, а по-късно възстановен от тях. След това е окупиран от французите под чието административно управление започва бързо да се разраства. Така старото мавританско селище се превръща в модерен град.

Сега Казабланка е най-големият град в Мароко! Той притежава едно от най-мащабните изкуствени пристанища в света.

За мен Казабланка е една от основните причини да се присъединя към пътуването.



Спомняте ли си, че по пътя за Милано използвахте Visual Studio 2010 , за да напишете първата си програма? Едно от първите неща, които трябваше да определите беше типа на проекта и вие избрахте Windows приложение. Windows приложението е тип програма, която притежава потребителски интерфейс, осигуряващ начин за взаимодействие между потребителя и програмата. Не всички програми притежават потребителски интерфейс. Някои от тях се стартират еднократно или работят във фонов режим и не изискват потребителски вход. О! Колко скучно!



## Дизайн на форми

Всички програми, които пиша са Windows приложения. Windows приложенията използват форми, което ви позволява да проектирате интерфейса, който ще вижда потребителят, и начина, по който той ще взаимодейства с програмата.

Когато създавате ново Visual Basic приложение за Windows, Visual Studio 2010 автоматично ще добави форма с име Form1.vb към проекта ви. За да осигурите взаимодействието на потребителя с програмата, вие добавяте контроли като бутони, етикети, текстови полета, падащи списъци и др. Да предположим, че вашата форма трябва да осигури въвеждането на информация за спуканата гума от примера в предходната секция. Ето описанието на контролите, които са ви необходими, за да направите това:

- В текстово поле въведете регистрационния номер на колата.
- От група радио-бутони изберете този, който показва коя гума е спукана (лява предна, дясна предна, лява задна, дясна задна)
- Използвайте поле за отметка, за да определите дали "поръчката е експресна"
- От падащ списък изберете размера на гумата
- Натиснете бутон, който претърсва описа с наличните гуми
- В текстово поле визуализирайте списък с наличните гуми, който включва производител, номер на модел, размер, инвентарен номер и цена

И така, формата за въвеждане на информацията за спуканата гума има две текстови полета, четири радио-бутона, един падащ списък и един бутон. Можете да ги подредите по различни начини върху формата, забавлявайки се и показвайки своите творчески способности.



За мен един от най-предизвикателните аспекти в програмирането е дизайна на формите. Това изисква да определя как ще изглежда формата и как ще бъдат подредени контролите върху нея. Формите са като платна за рисуване, а контролите като бои. И все пак трябва да внимавам. Трябва да съм сигурен, че формата ще прави точно това, което трябва. Тя трябва да отговаря на определени изисквания – да бъде лесна за използване и да не съдържа излишни компоненти. Когато създавам дизайна на формата, си задавам въпроса "Какво е предназначението на формата и какво трябва да прави?"



Също така трябва да съм сигурен, че формата изглежда добре – като мен! Тя не трябва да съдържа много цветове и неприятни визуални ефекти, като например мигащи текстове, които разсейват потребителя. Контролите върху формата трябва да са подходящо и логически подредени. Замислям се върху реда, в който потребителят въвежда информация и взаимодейства с формата. Понякога предупреждавам потребителя, че не е въвел цялата необходима информация или че информацията е грешна. Когато създавате дизайна на вашите форми трябва да мислите като потребител и да се опитате да си представите възможните начини, по които той ще взаимодейства с нея.

Накрая трябва да запомните, че всички Windows приложения изглеждат и действат по стандартен начин. Потребителят очаква вашето приложение да притежава определен дизайн и да се държи по определен начин. Например, първата команда от главното меню да бъде "File", а по-долу от нея да бъде разположена командата "Exit", с която програмата се затваря.

Потребителят ще бъде учуден и раздразнен, ако при натискането на бутон със заглавие "Calculate Results", цвета на формата стане червен, а заглавията на всички останали бутони станат латински.



## Използване на форми

Както вече знаете, когато създавате ново Windows приложение, Visual Studio 2010 автоматично добавя форма с име Form1.vb към проекта ви. Това е главната форма за вашето приложение. Потребителят ще я види, когато за стартира програмата ви.

Главната форма е като централно управляващо табло. От нея потребителят може да достигне до всяка друга част на програмата. Често в горната си част главната форма притежава меню или лента с инструменти, които препращат потребителя към различните части на програмата.

### Полезен съвет

Form1.vb може да се преименува с по-описателно име. Това е желателно за всички приложения.

В главната форма потребителят избира елементи от менюто или натиска бутони, за да отвори други форми. Тези форми могат да се използват за въвеждане и визуализиране на информация или за взаимодействие с програмата. Обикновено те имат специфично предназначение като показване на резултати, събиране на информация или определяне на програмни настройки. След като потребителят приключи работата си с тях, той ги затваря и се връща към главната форма. Независимо, че Visual Studio 2010 добавя към проекта ви само една форма, вие можете да създадете много други в зависимост от нуждите на приложението, което създавате.

Друг важен тип форма е диалоговият прозорец MessageBox. Вие вече го използвахте в програмата Hello World, за да визуализирате съобщение с текст "Hello World". Диалоговият прозорец MessageBox се използва за извеждане на съобщения със специална форма. След като потребителят прочете съобщението той може да затвори формата посредством натискане на бутона OK.

Да започваме! Ще създадете ваша форма, а именно формата за описание на спуканата гума. Върху нея ще добавите контроли и ще настроите свойствата им. Ще ви покажа как контролите отговарят на събития като например натискане на бутон, и ще преговорим наученото за диалоговия прозорец MessageBox.

### Полезен съвет

Учениците ще четат материалите с интерес, но на практика няма да правят това, което е описано в тях. За тях ще бъде по-добре, ако наистина правят това, което е описано в текста, затвърждавайки и преговаряйки материала от курса.

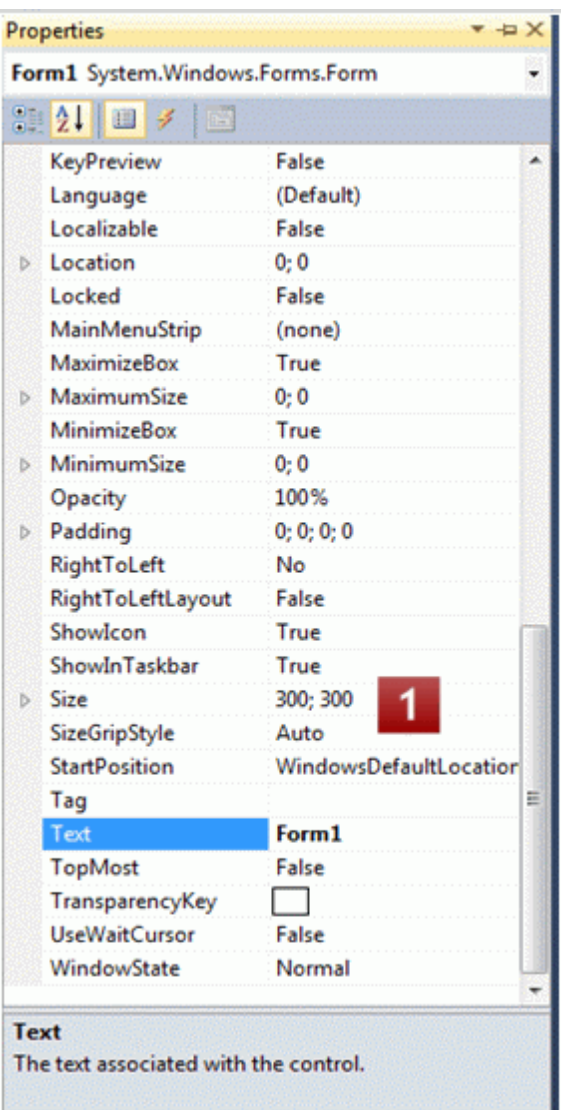


## Свойства на формите

Когато създавате ново Windows приложение, Visual Studio 2010 автоматично добавя форма с име Form1.vb към проекта ви. Нейните размери по подразбиране са 300 на 300 пиксела. Ще попитате как разбрах какъв е размера на формата. Е, това е лесно да се провери като се използва прозореца Properties. Нещо повече, вие можете да промените размера на формата като използвате този прозорец.

Ще ви покажа как може да стане това. Първо, създайте ново Windows приложение с име Flat Tire. След това щракнете еднократно върху формата, за да я селектирате. Натиснете клавиша F4, за да отворите прозореца Properties.

1. Докато формата е селектирана, прозорецът Properties показва нейните свойства. Забележете, че свойството Text е маркирано по подразбиране.





Дестинация  
**Казабланка, Мароко**

|                   |                        |
|-------------------|------------------------|
| RightToLeftLayout | False                  |
| ShowIcon          | True                   |
| ShowInTaskbar     | True                   |
| Size              | 300; 300               |
| SizeGripStyle     | Auto                   |
| StartPosition     | WindowsDefaultLocation |
| Tag               |                        |

2. Скролирайте нагоре списъка със свойствата и изберете свойството Size. Забележете, че пред него има знак плюс. Това показва, че свойството Size притежава няколко полета за въвеждане на стойности.

|               |                        |
|---------------|------------------------|
| ShowIcon      | True                   |
| ShowInTaskbar | True                   |
| Size          | 300; 300               |
| Width         | 300                    |
| Height        | 300                    |
| SizeGripStyle | Auto                   |
| StartPosition | WindowsDefaultLocation |

3. Натиснете знака плюс до свойството Size. Ще видите свойствата Height и Width. Височината на формата е 300 и ширината ѝ е 300. Променете ги.

|               |          |
|---------------|----------|
| ShowInTaskbar | True     |
| Size          | 450; 300 |
| Width         | 300      |
| Height        | 300      |
| SizeGripStyle | Auto     |

4. Щракнете двукратно върху свойството Width, за да маркирате стойността му. Натиснете клавиша backspace, за да изтриете текущата стойност.

|               |          |
|---------------|----------|
| ShowInTaskbar | True     |
| Size          | 450; 300 |
| Width         | 450      |
| Height        | 300      |
| SizeGripStyle | Auto     |

5. Въведете стойност 450 и натиснете клавиша Enter. Това е! Току що променихте ширината на формата. Забелязахте ли как се промени

съответстващата ѝ стойност в свойството Size след като натиснахте клавиша Enter.



Сега нека променим цвета на формата посредством нейното свойство BackColor. Скролирайте прозореца Properties, докато откриете свойството BackColor. Щракнете върху думата "Control". Отдясно ще се появи бутон с насочена надолу стрелка. Ще се отвори прозорец с палитра от цветове. Изберете страницата Custom. Щракнете двукратно върху цветния квадрат, за да изберете цвят. Аз бих избрал светло жълт. Погледнете формата сега!

Изградете проекта и стартирайте програмата. Забелязахте ли новия цвят на формата. Освен това тя вече не е квадратна, а правоъгълна!



Сега вече знаете как да променяте свойствата на формата като използвате прозореца Properties. Скролирайте списъка на прозореца и се запознайте с него. Той ви позволява да променяте свойствата на формата по време на дизайн, т.е. когато я създавате. Както ще видите, прозорецът Properties не се използва само за промяна на свойствата на формите, а също така и

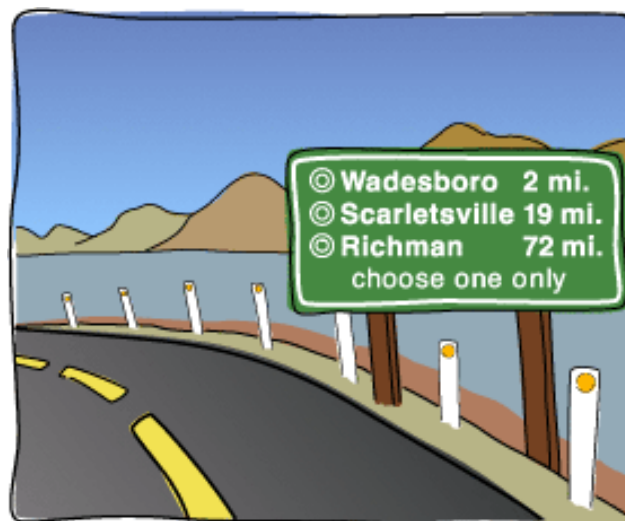
на бутоните, текстовите полета и други контроли.





## Контроли и свойства

Знам, че можете да добавяте бутони върху формата, но нека направим една репетиция! Както виждате прозорецът Toolbox притежава и други контроли. Сега ще ви покажа как да използвате най-популярните от тях и как да настройвате най-често използваемите им свойства. Да започнем с етикетите.



## Етикети

Контролът Label се използва за озаглавяване на самостоятелни или групи от контроли. Обикновено той се използва за издаване на инструкции или определяне на реда, в който трябва да се въвежда информацията във формата.

### Полезен съвет

Всички контроли са обекти, но не всички обекти са контроли. Обект може да бъде всичко с изключение на свойство. Свойството е характеристика на обекта, като например цвят.

Отворете прозореца Toolbox и щракнете двукратно върху контрола Label. Контролът ще се появи на формата ви. Преместете го като държите натиснат левия бутон на мишката. Селектирайте го и отворете прозореца Properties. Прозорецът Properties визуализира списък със свойствата на контрола Label. Скролирайте го докато достигнете до свойството Text. Щракнете двукратно върху свойството и изтрийте стойността му. Въведете следния текст: "Enter License"

## Текстово поле

Текстовото поле се използва, за да визуализирате или получите информация от потребителя. Отворете прозореца Toolbox и щракнете двукратно върху контрола Textbox. На формата ви ще се появи контрол с име Textbox1. Преместете го като държите натиснат левия бутон на мишката и го поставете отдясно на етикета Label1. Селектирайте го и отворете прозореца Properties. Скролирайте го докато достигнете до свойството Text. Щракнете двукратно върху свойството и изтрийте стойността му. Установете стойност False за свойството ReadOnly за да позволите на потребителя да въвежда текст в текстовото поле. False е стойността по подразбиране.



Текстовите полета често се използват за визуализиране на голямо количество информация. Ако действително текстът е дълъг, вие сигурно ще добавите ленти за скролиране. Поставете върху формата друго текстово поле. Задайте стойност True на неговото свойство Multiline. Установете ширината и височината му на 100 пиксела. Ако свойството Multiline е със стойност False, вие няма да можете да промените височината на текстовото поле. Задайте стойност Vertical на свойството Scrollbars. Установете стойност True за свойството ReadOnly, така че потребителят да не може да въвежда текст в полето.

## Радио-бутони

Радио-бутоните се използват, за да се предостави избор на потребителя. Потребителят може да селектира само един радио-бутон от дадена група подобно на тест с множество отговори. Щракнете двукратно върху контрола Radio Button в прозореца Toolbox. На формата ви ще се появи контрол с име RadioButton1. Селектирайте контрола, щракнете върху него с десния бутон на мишката и изберете командата Copy от контекстното меню. Щракнете на произволно място върху формата. След това щракнете с отново с десния бутон на мишката и изберете командата Paste от контекстното меню. На формата ви ще се появи контрол с име RadioButton2. Повторете операцията Paste за да добавите още два радио-бутона с имена RadioButton3 и RadioButton4. Селектирайте контрола с име RadioButton1 и отворете прозореца Properties. Скролирайте списъка в прозореца до достигане на свойството и го променете от "RadioButton1" на "Left, Front". Променете свойството Text на останалите радио-бутони по следния начин: от RadioButton2 на "Left, Rear", от RadioButton3 на "Right, Front", от RadioButton4 на "Right, Rear".

## Полета за отметка

Полетата за отметка също се използват, за осигуряване на потребителски избор. Потребителят може да селектира едно или повече полета за отметка от дадена група. Щракнете двукратно върху контрола CheckBox в прозореца Toolbox. На формата ви ще се добави контрол с име CheckBox1. Отворете прозореца Properties и задайте стойност "Rush Job" на свойството му Text.

## Падащи списъци

Падащият списък представя списък от елементи, които могат да се избират. Избраният от потребителя елемент се визуализира в текстовото поле на падащия списък. Щракнете двукратно върху контрола Combobox



в прозореца Toolbox. На формата ви ще се добави контрол с ме ComboBox1. Отворете прозореца Properties и задайте стойност "Choose Tire Size" за свойството Text на контрола. Изберете свойството Items. Щракнете върху думата "Collection". Вдясно от нея ще се появи бутон с три точки. Натиснете този бутон, за да отворите прозорец, в който можете да въведете елементите на падащия списък. Всеки елемент трябва да е на отделен ред. На нов ред се преминава посредством натискане на клавиша Enter. Въведете стойностите 12, 13, 14 и 15. Натиснете бутона ОК, за да затворите прозореца.

## Бутони за команди

Добавете бутон на формата. Задайте стойност "Get Tires" на свойството му Text. Задайте червен цвят за свойството му BackColor. Изберете свойството Font. Щракнете върху него, за да визуализирате бутончето с трите точки. В появилата се палитра на шрифта задайте стойност Bold за негов стил. Натиснете бутона ОК, за да затворите палитрата.

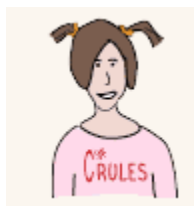
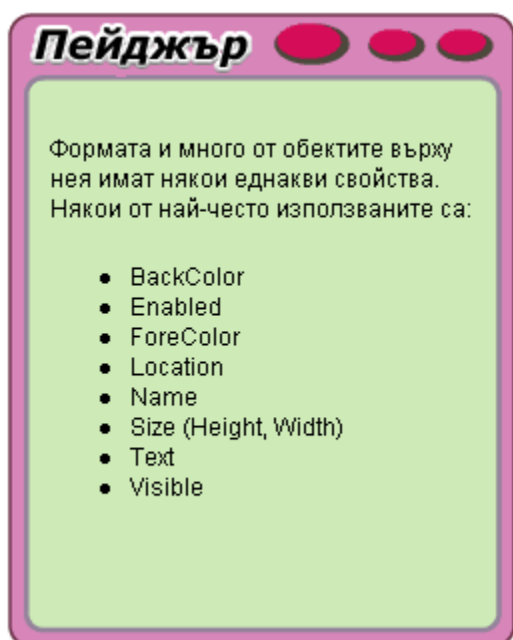
Изградете приложението и стартирайте програмата. Щракнете върху различните контроли на формата. Какво се случва, когато щракнете върху падащия списък? Какво се случва, когато изберете един от радиобутоните или маркирате полето за отметка? Въведете текст в текстовото поле до етикета Enter License. Забелязахте ли, че другото текстово поле е сиво и че не можете да въвеждате текст в него. Щракнете върху знака "X" в горния десен ъгъл на формата, за да затворите приложението.

Трябва да отбележим, че вие само създадохте формата. През това време, обаче се запознахте с няколко контроли и техните свойства. Следващата стъпка, за да направите формата работеща, е да добавите Visual Basic.NET код за формата и контролите.

Уверете се, че сте се запознали с различни свойства на контролите. Можете да разместите контролите върху формата така, че тя да стане по-подредена и лесна за използване. Можете да добавите и етикети, които показват предназначението на контролите и информацията, която представят. Как си представяте подредбата на контролите? Кой от тях ще бъде първи и кой последен?

### Полезен съвет

Учениците трябва да използват прозореца Property, за да разгледат свойствата на контролите.



В C# дизайна на форми е аналогичен на този във Visual Basic 2010. Това се дължи на Visual Studio 2010 средата. Отново трябва да отворите прозореца Toolbox и да поставяте контролите от него на формата. След това отворете прозореца Properties и настройте свойствата на контролите.



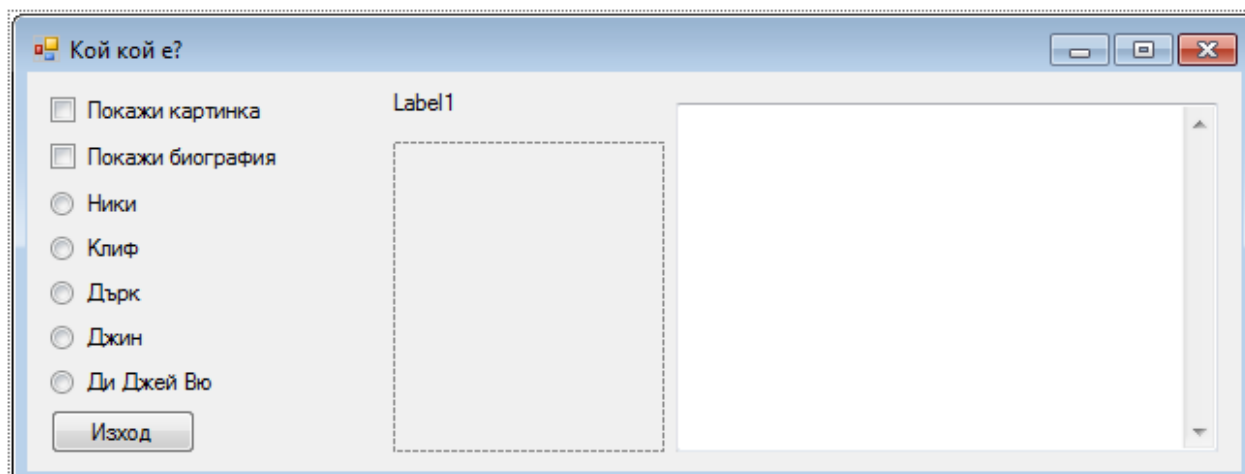
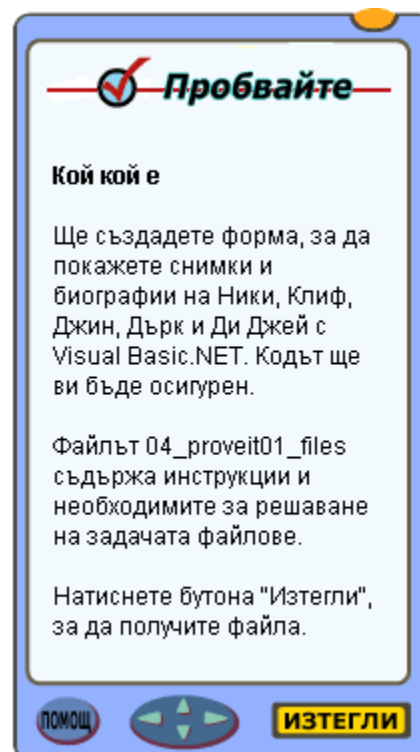
## 04 Пробвайте 01 Кой кой е

Желаете ли да научите повече за хората, които ви съпътстват в курса по програмиране? На този етап трябва да докажете, че можете да създадете форма и да инициализирате свойствата на обектите, поставени на нея като използвате Прозореца за свойства. Ако се справите успешно, ще научите повече за приятелите ви от курса.

Вземете шаблон 01 от секция 04 с изходен код от вашия учител или от уеб сайта на курса. Поставете проекта в самостоятелна директория. Ще получите две поддиректории (bin, obj) и 7 файла.

Отворете Visual Studio, изберете Existing Project и намерете файл с име WhosWho.sln. Като алтернативен вариант можете да използвате My Computer, за да откриете файла WhosWho.sln, след което да щракнете два двукратно върху него. проектът WhosWho ще се отвори в средата на Visual Studio.

Вашата задача е да добавите обекти върху формата посредством прозореца Toolbox така, че формата да изглежда като показаната по-долу. Също така трябва да актуализирате свойствата на обектите, както е описано по-долу.





1. Щракнете еднократно върху формата, за да я селектирате и променете името ѝ на "Кой кой е?" от прозореца Properties.
2. Поставете две полета за отметка. Оставете имената им каквито са по подразбиране – CheckBox1 и CheckBox2. Променете текстовите им свойства съответно на "Покажи картинка" и "Покажи биография".
3. Поставете пет радио-бутона. Променете текстовите им свойства на Ники, Клиф, Дърк, Джин и Ди Джей Вю. Оставете имената и каквито са по подразбиране: RadioButton1 – RadioButton5.
4. Поставете текстово поле с име TextBox1.
5. Поставете контрол за картинка с име PictureBox1. Използвайте Прозореца за свойствата, за да промените височината му на 160 и ширината му на 140. За целта щракнете върху знака "+", намиращ се в полето за модифициране на размера.
6. Променете някои от свойствата на TextBox1. Редът, в който ще направите това е важен! Текстовото поле трябва да бъде достатъчно голямо, за да покаже няколко реда от биографията на съответния човек. Visual Basic няма да ви позволи да промените неговата височина преди да сте установили стойност True за свойството му Multiline. Ако текстовото поле е едноредово, то няма причина да се променя височината му. След като направите това, променете височината на текстовото поле на 180, а ширината му на 280. Накрая задайте стойност Vertical на свойството Scrollbars, за да позволите на текстовото поле да се скролира, ако текстът в него е по-голям.
7. Поставете бутон с име Button1. Променете текста му на "Изход".

Изградете проекта и ако няма грешки изберете опцията Start Debugging от менюто Debug, за да стартирате програмата.

Ако програмата работи правилно, я покажете на вашия учител.

#### Полезен съвет

Учениците добавят обекти в частично завършена програма. Те не пишат код.





## Обработка на събития

Събитията се генерират от действията на потребителя. Например, когато потребителят натисне бутон, избира елемент от падащо меню или въвежда текст в текстово поле, се генерира събитие. Събития се генерират и когато потребителят щракне еднократно или двукратно върху контрол или премества курсора на мишката към и от контрол. Вие ще използвате тези събития, за да изпълнявате асоциирания към тях VB.NET код. Кодът на VB.NET определя това, което вашата програма трябва да прави.

Когато писахте програмата Hello World, щракнахте двукратно върху бутона, при което се отвори прозорец, в който въведохте кода за събитието Button1\_Click event. Събитието Button1\_Click е подразбиращо се за бутона. Всеки контрол има събитие по подразбиране, към което можете да асоциирате код като щракнете двукратно върху контрола в дизайн режим.

Например:

- За контрола Button, подразбиращото се събитие е Click
- За контрола Textbox, подразбиращото се събитие е TextChanged
- За контрола RadioButton, подразбиращото се събитие е CheckedChanged
- За контрола CheckBox, подразбиращото се събитие е CheckedChanged
- За контрола ComboBox, подразбиращото се събитие е SelectedIndexChanged

За да ви покажа как става това, нека да добавим код за събитията по подразбиране на контролите от формата за "спуканата гума". Кодът ще визуализира диалогов прозорец със съобщение за контрола, който е генерирал събитието. В дизайн режим щракнете двукратно върху всеки един от контролите, които сте добавили на формата. Прозорецът за въвеждане на код ще се отвори за обработка на събитията за всички контроли. Копирайте следния ред код в обработчика на всяко едно от събитията.

**MessageBox.Show(sender.GetType.Name)**

Изградете приложението и го стартирайте. Щракнете върху отделните контроли на формата, променете текста в текстовото поле и изберете елемент от падащия списък. Подразбиращото се събитие за всеки един

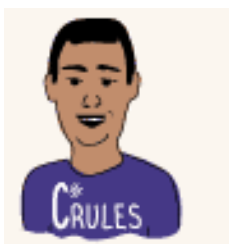


от контролите ще бъде обработено, а типа на контрола, който го генерира, ще се визуализира в диалогов прозорец MessageBox.

Ако формата е част от истинско приложение, вие ще трябва да добавите конкретен код, за да му осигурите необходимата функционалност. Например, когато натиснете бутона Get Tires, в текстовото поле трябва да се визуализира списък с наличните гуми.

### Полезен съвет

Понякога учениците щракват върху грешен контрол като етикет, създавайки обработчик на събитие. Обикновено, когато изтрият ненужния код, всичко е наред. Понякога VB създава допълнителен код към обработчика на събитие. Този код също трябва да бъде изтрил. При изграждането на приложението възникват грешки, така че такъв код лесно се открива.



В C# контролите имат същите събития както във VB. Това се дължи на общата среда за изпълнение и общата среда за разработка – Visual Studio 2010 .

### Диалоговият прозорец MessageBox

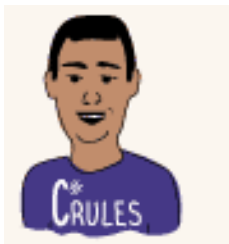
Често пъти ще ви се наложи да осигурите диалог между програмата и потребителя. Програмата може да изпрати съобщение на потребителя, да съхрани информацията от него или да получи потребителски вход. На практика тази комуникация във Visual Studio се реализира посредством диалоговия прозорец MessageBox.

В първата си програма вие написяхте ред от код, който визуализираше диалогов прозорец MessageBox.

`MessageBox.Show("Hello, World.")`

Диалоговият прозорец MessageBox е тип форма, която е вградена във VB.NET. Вие не трябва да я добавяте, за да я използвате. Всичко, което трябва да направите, е да извикате метода Show() и да специфицирате текста, който желаете да се покаже. Синтаксисът, който трябва да използвате, е:

Обърнете внимание, че текстът, който желаете да се покаже, е обграден с кавички.



Хей, вижте кода за диалоговия прозорец MessageBox в C#.

```
{  
MessageBox.Show("Hello World");  
}
```



### Полезен съвет

Диалоговият прозорец MessageBox е по-сложна форма, която показва бутони, които потребителят може да натисне (Yes, No, Cancel). Програмистът може да установи кой бутон е натиснат от потребителя.

Например,

```
Dim UserChoice as Integer
```

```
UserChoice = MessageBox.Show ("Въпрос?", vbYesNo, "заглавие")
```

```
If UserChoice = vbYes Then
```

```
    ' "да" код
```

```
Else
```

```
    ' "не" код
```

```
End If
```



### Пейджър

Контролите върху формата имат и други събития, освен подразбиращото се. Повечето от тях притежават множество събития. Например текстовото поле има следните събития:

- Промяна на текста (подразбиращо се събитие)
- Щракване
- Двукратно щракване
- Влизане на курсора на мишката в контрола
- Напускане на контрола от курсора на мишката
- и още много!

За да видите всички събития за даден контрол, първо го селектирайте. След това в прозореца Properties натиснете бутона Events (изглежда като светкавица).



## Влизане във Казабланка

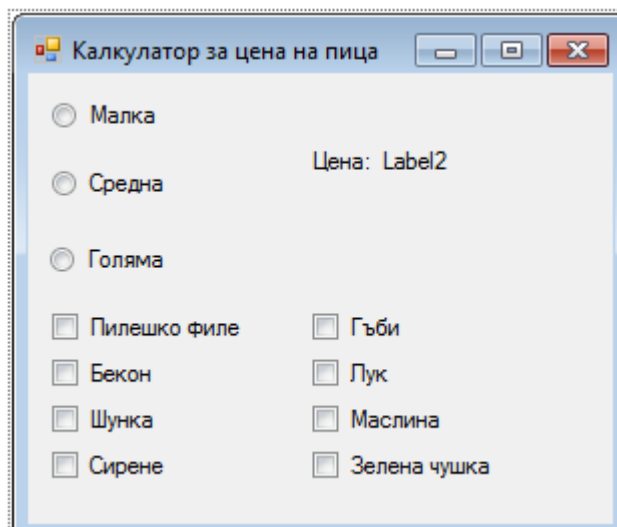
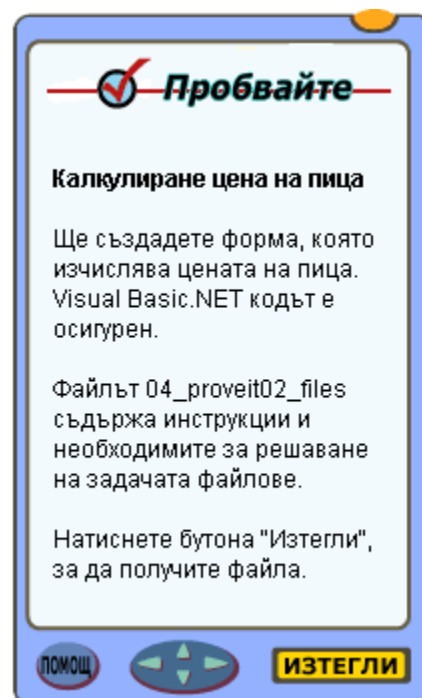
Свършихме чудесна работа! Вече знаете защо обичам дизайна на формите толкова много! Знам, че сте готови за Казабланка! Пробвайте задачите, направете теста и вече ще бъдете готови да поемете по пътя на професионалния дизайнер на форми!

### 04 Пробвайте 02 Калкулиране цена на пица

Обичате ли пица? Съществуват програми, които позволяват на купувачите да пресметнат цената на поръчаната от тях пица.

Обзалагам се, че вие също можете да напишете подобна програма. Можете да подредите компонентите върху формата по ваше желание. Ние ще ви осигурим изходния код на програмата. Използвайте шаблон 02 от секция 04. намерете файла Pizza Price.sln. Щракнете двукратно върху него, за да го отворите с Visual Studio.

Вашата задача е да поставите обекти върху формата като използвате прозореца така, че тя да изглежда по следния начин.







Необходими са ви три радио-бутона, два етикета и 8 полета за отметка. Не забравяйте да промените заглавието на формата. Текстът на втория етикет ще се променя в процеса на работа на програмата.

Програмата калкулира 5 лв. за базовия вариант на пицата, по 1 лв. за всяка месна допълнителна съставка и по 0.50 лв. за всяка зеленчукова допълнителна съставка. Изборът на средна пица увеличава цената и с 25%, а на голяма – 50%.

Изградете проекта и ако няма грешки изберете опцията Start Debugging от менюто Debug, за да стартирате програмата. Ако програмата работи правилно, я покажете на вашия учител.

## Продължение / Обобщение

Покажете формираната цена така, че да изглежда като парична сума. Visual Basic.NET предоставя функции за това. За целта подайте променлива на функцията `FormatCurrency` по следния начин:  
`FormatCurrency(MyVariable)`

Добавете компонент за картинка, който ще визуализира подходяща картинка на пица.

### Полезен съвет

Функцията `FormatCurrency` работи със стойностите на променливите, а не с обекти. Прилагането и върху текстови полета не означава, че те ще бъдат форматираны. Това означава само, че текста в текстовото поле е форматиран в момента на прилагане на функцията. Ако съдържанието на текстовото поле се промени по-късно, форматирането ще изчезне, освен ако не се приложи отново.



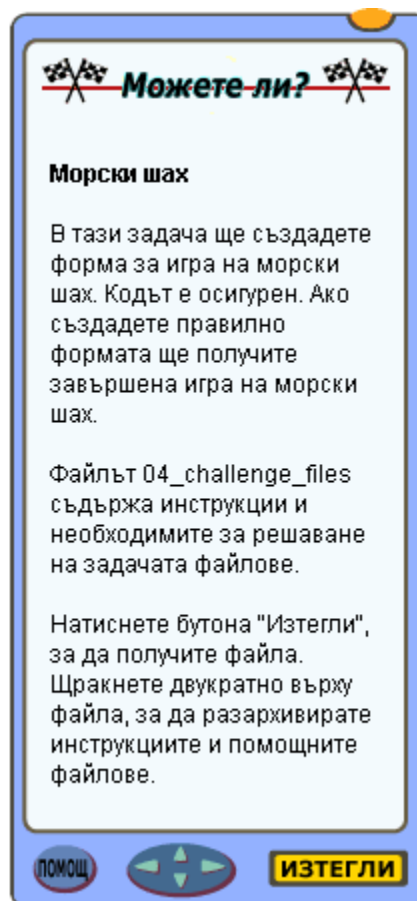
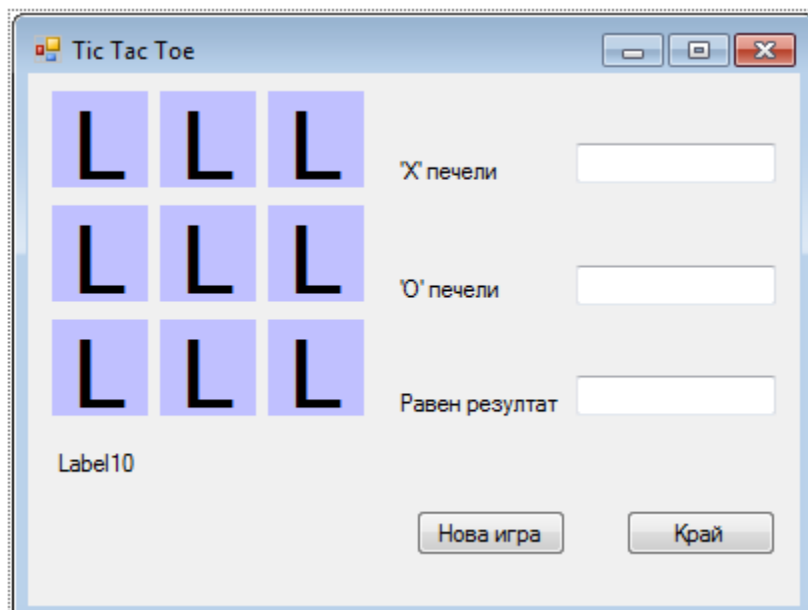
## 04 Можете ли? Морски шах

Преносимият компютър е удобен, когато човек пътува. Компютърната игра е начин за прекарване на времето, когато ви е скучно.

Ваша задача, ако се съгласите е да напишете класическата игра на Морски шах с Visual Basic. Дърк откри кода на програмата на неизвестен уеб сайт, но изходния код на формата липсваше. Вие трябва да създадете формата така, че да съответства на кода. Ако направите всичко правилно ще имате работеща игра Морски шах.

Вземете от вашия учител или от сайта на курса, ако се обучавате в Интернет, шаблона с изходен код от секция 04. Поставете го в собствена директория.

Отворете Visual Studio, от менюто File изберете командата Open Project и отворете файла TicTacToe.sln. Като алтернативен вариант можете да използвате My Computer, за да изберете файла TicTacToe.sln, след което да щракнете два пъти върху него. Средата на Visual Studio ще се отвори автоматично.



Вашата задача е да добавите обекти върху формата като използвате Прозореца Toolbox, така че формата да изглежда като тази по-долу. Също така трябва да обновите елементите в прозореца на свойствата както е описано по-долу.



- Щракнете върху формата, за да установите фокуса върху нея. Променете нейното свойство текст на Tic Tac Toe.
- Създайте 9 етикета и ги номерирайте от 1 до 9. Подредете ги така, че етикет 1 да бъде в горния ляв ъгъл на формата, а етикет 9 – в долния десен. Можете да спестите време като създадете един етикет, настроите свойствата му и след това го размножите 8 пъти. Задайте следните стойности на свойствата:

|           |  |
|-----------|--|
| AutoSize  | False  |
| Height    | 48   |
| Width     | 48   |
| Font      | Microsoft Sans Serif Bold 36                 |
| BackColor | Може да оставите сивия цвят по подразбиране. |
| Text      | Можете да оставите текста по подразбиране.   |

#### Полезен съвет

Главното L' на предишната страница представя първата буква на етикета, който е с главни букви.

- Създайте етикет с номер 10 под предходните 9.

|           |  |
|-----------|--|
| AutoSize  | False  |
| Height    | 25   |
| Width     | 175  |
| Font      | Microsoft Sans Serif Bold 16                 |
| BackColor | Може да оставите стойността по подразбиране. |
| Text      | Може да оставите стойността по подразбиране. |

- Създайте три етикета с номера от 11 до 13. Променете текста им на "Х печели", "О печели" и "Равен резултат", както е показано в примера.
- Създайте три текстови полета с номера от 1 до 3. Поставете ги непосредствено до етикетите , създадени в стъпка 4.
- Създайте бутон с номер 1, както е показано. Променете текста му на "Нова игра".
- Създайте бутон с номер 2, както е показано. Променете текста му на "Край".



Веднъж след като сте създали формата, изградете приложението. Ако няма грешки изберете опцията Start Debugging от менюто Debug, за да стартирате програмата.

Ако програмата ви работи правилно, я покажете на учителя си.

### Продължение / Обобщение

- Променете цвета на фона на формата.
- Променете шрифта на Microsoft Sans Serif Bold 10 на всички бутони, текстови полета и др. Можете ли да промените шрифта на всички елементи едновременно като преди това щракнете върху тях при натиснат клавиш CTRL?
- Добавете текстови полета за имената на играчите.



## Пристигане в Казабланка

Вече пристигнахме в Казабланка, Мароко и вие трябва да направите друг тест. Той ще ви помогне да отговорите на въпросите, които Комисията по състезанията ще ви зададе. Късмет!



## Проверка на знанията

НАПРАВЕТЕ ТЕСТА ОТНОВО

- |  |  |
|--|--|
| <p><b>1</b> Какво се използва, за да се осигури взаимодействие между формата и потребителя?</p> <ul style="list-style-type: none"><li><input type="radio"/> А. Контроли</li><li><input type="radio"/> В. Обекти</li><li><input type="radio"/> С. И двете</li></ul> | <p><b>3</b> Кой контрол се използва за показване и въвеждане на информация?</p> <ul style="list-style-type: none"><li><input type="radio"/> А. Текстово поле</li><li><input type="radio"/> В. Етикет</li><li><input type="radio"/> С. Бутон</li></ul>                                      |
| <p><b>2</b> Формата и контролите ѝ притежават:</p> <ul style="list-style-type: none"><li><input type="radio"/> А. Дълбочина</li><li><input type="radio"/> В. Свойства</li><li><input type="radio"/> С. Дъщерни прозорци</li></ul>                                  | <p><b>4</b> Какво правят събитията във Visual Basic?</p> <ul style="list-style-type: none"><li><input type="radio"/> А. Променят свойствата на контролите</li><li><input type="radio"/> В. Актуализират папката с кода</li><li><input type="radio"/> С. Карат кода да се изпълни</li></ul> |