

## Диалогови прозорци

### Стокхолм, Швеция

С моята група отдавна сме замислили да посетим родината на някои от легендите в световното музикално пространство. Несъмнено познавате АББА, Юръп и Роксет, а какво ще кажете за Ace of Base, Army of Lovers и The Cardigans?

Ще използваме нашето пътуване до Швеция, за да реализираме идеята си. В Стокхолм задължително ще посетим първия в света музей на открито Скансен, в който има къщи от различните краища на Швеция и зоологическа градина.

По пътя ще ви разкажа какво представляват и как се използват диалоговите прозорци. При положение, че можете да декорирате главната форма на програмата си с бутони и други контроли по същия начин, по който можете да създадете дизайна и на диалогов прозорец, какво прави една форма различна от диалогов прозорец?





## Диалоговите прозорци

От гледна точка на управление на събитията на диалоговите прозорци и техните контроли разликата между тях и формите е малка.

Диалоговите прозорци биват модални и немодални. Най-често използваните са модалните диалогови прозорци и затова ще обърна повече внимание на тях. Както подсказва името, модалния диалогов прозорец превключва режима на въвеждане (вход) от главната форма на програмата към диалоговия прозорец. Когато програмата показва модален диалогов прозорец, потребителят не може да превключва между диалоговия прозорец и друга форма на програмата. Потребителят трябва явно да напусне диалоговия прозорец, обикновено като натисне команден бутон с надпис ОК (или Open, или Save) или бутон с надпис Cancel. Но потребителят може да превключи на друга програма докато диалоговият прозорец продължава да бъде показан. Някои диалогови прозорци дори не позволяват превключване към други програми: системните модални диалогови прозорци съобщават за сериозни проблеми и първо трябва да бъдат завършени, за да може потребителят да прави нещо друго в Windows.

Немодалните диалогови прозорци са много подобни на допълнителните форми, които създава програмата. Можете да превключвате между немодалните диалогови прозорци, които програмата създава, и главната форма. Програмистите често използват модални диалогови прозорци, когато програмата трябва да получи от потребителя допълнителна информация.

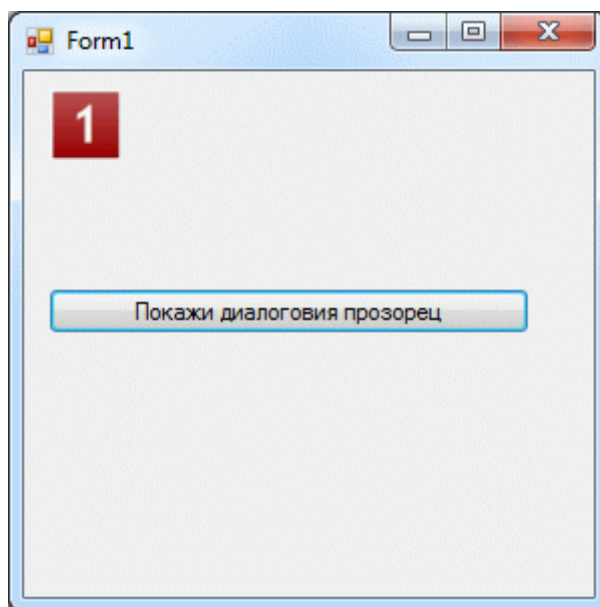


## Вашият първи модален диалогов прозорец

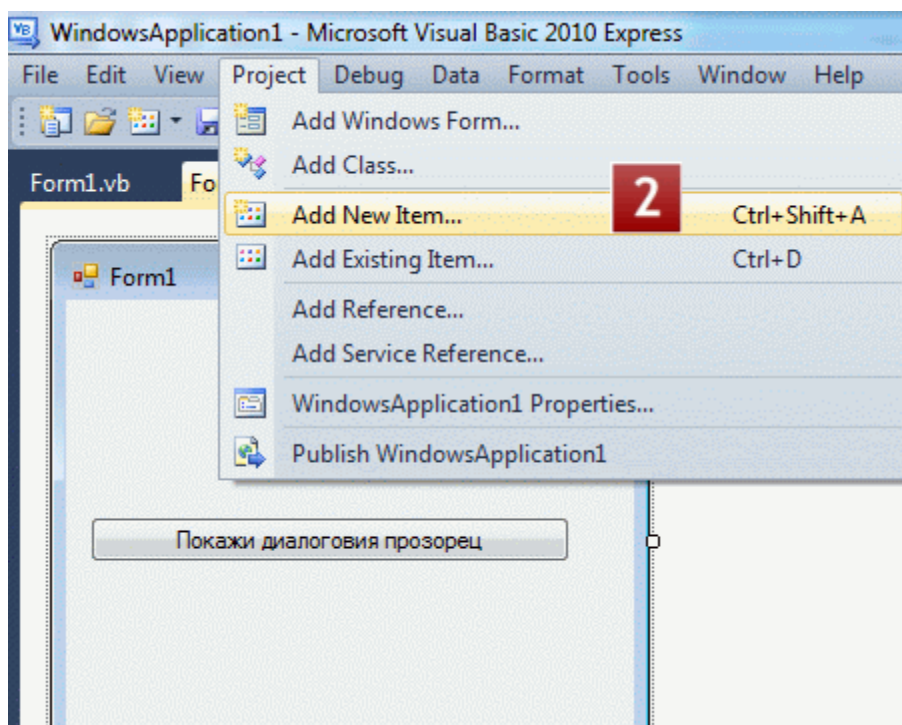
Диалоговите прозорци се различават от формите най-вече по начина, по който се извикват и прекратяват.

Сега ще ви покажа колко лесно можете да създадете диалогов прозорец!

1. Създайте нов проект и поставете бутон на формата, която Visual Basic създава за вас.



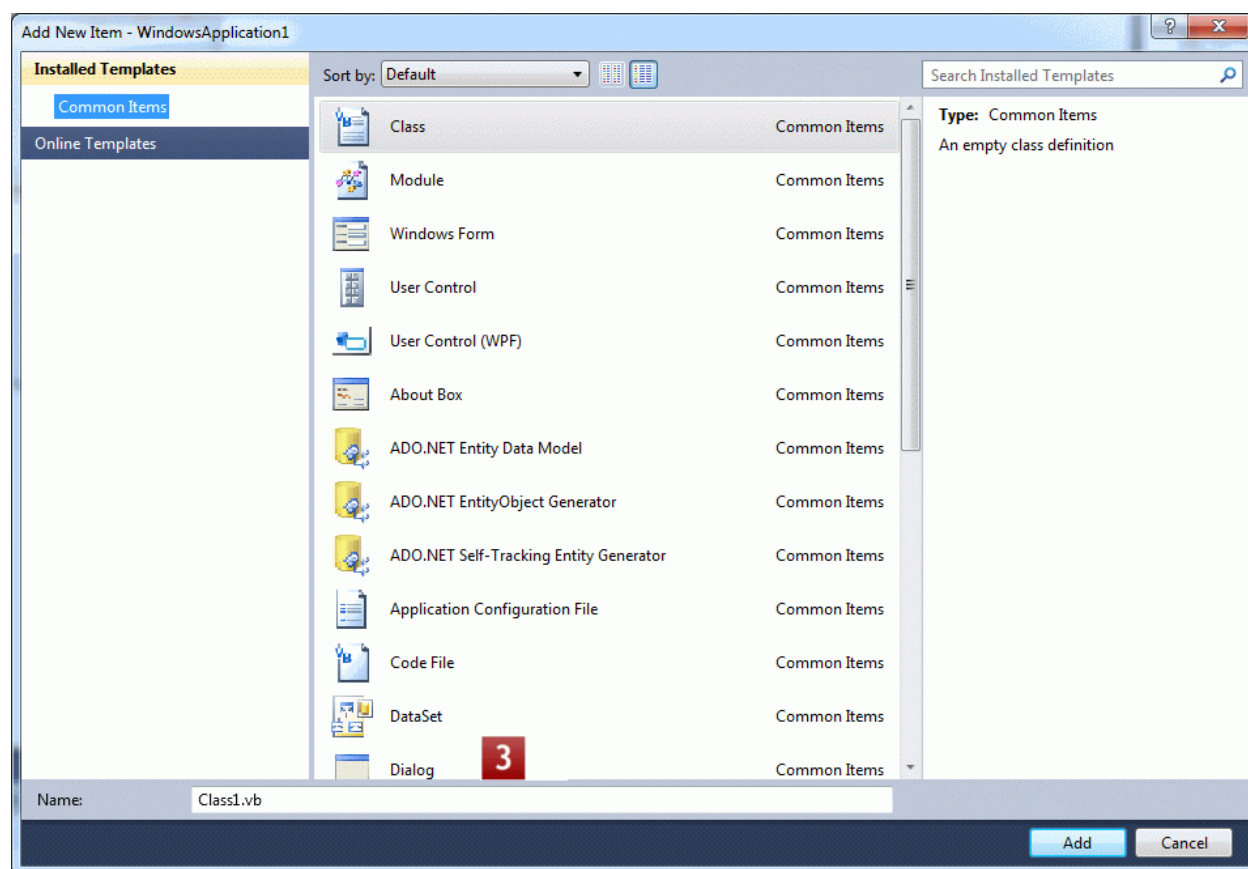
2. От менюто Project на Visual Studio 2010





изберете командата Add New Item...

3. В появилия се прозорец със заглавие Add New Item изберете шаблона Dialog и натиснете бутона Add.





4. Ще се създаде нов диалогов прозорец с два бутона Ok и Cancel. Върнете се отново при главната форма и щракнете два пъти върху бутона. В обработчика на събитието Click въведете следния код:

```
Dialog1.ShowDialog()
```

5. Изградете и стартирайте програмата.

Какво се случва, когато натиснете бутона на формата?



## Методът ShowDialog

Методът ShowDialog казва на диалоговия прозорец да стане видим. Той не връща стойност, докато не бъде прекратен диалоговият прозорец.

ShowDialog е методът, който трябва да използвате, за да извикате модален диалогов прозорец. По времето, докато е показан модален диалогов прозорец, не можете да превключвате обратно в главната форма на програмата. Ето какво означава да е модален. Докато е визуализиран модалният диалогов прозорец, формата на приложението не може да приема никакво въвеждане от клавиатурата или мишката.

Двата бутона на диалоговия прозорец притежават обработчици на събитието Click. Щракнете двукратно върху всеки един от тях, за да видите кода, генериран от Visual Basic 2010.

Двата обработчика задават стойност на свойството DialogResult на диалоговия прозорец. Това свойство може да приема една от следните стойности:

Стойност
None
OK
Cancel
Abort
Retry
Ignore
Yes
No

Методът ShowDialog връща същата стойност както свойството DialogResult на диалоговия прозорец, когато е прекратен. Така програмата, която извиква диалоговият прозорец може да провери при какво условие е затворен. Най-общо казано така тя получава информация от диалоговия прозорец, ако DialogResult е OK, или продължава своя път, ако DialogResult е Cancel.

## Приемане и отказване

Стартирайте отново създадената от вас програма. След като покажете диалоговия прозорец, натиснете клавиша Enter. Какво се случва? Диалоговият прозорец се затваря, нали!

Покажете още веднъж диалоговия прозорец и след това натиснете клавиша Esc. Диалоговият прозорец се затваря отново.

А сега разгледайте свойствата AcceptButton и CancelButton на диалоговия прозорец. Виждате, че свойството AcceptButton приема като стойност името на бутона OK, а свойството CancelButton – името на бутона Cancel. Това означава, че свойството AcceptButton определя бутона, който трябва да бъде активиран при натискането на клавиша Enter, а свойството CancelButton – бутона, който трябва да бъде активиран при натискането на клавиша Esc.







## Местоположение на екрана

Вероятно сте забелязали, че новозаредените Windows приложения често се появяват в различни местоположения на екрана. Това се случва и с диалоговите прозорци. Диалоговият прозорец по подразбиране се появява подходящо отместен от формата на приложението.

Можете да замените подразбиращото се поведение, управляващо местоположението на форми и диалогови прозорци като използвате свойството `StartPosition`. То може да приема всяка една от следните стойности:

Стойност	Описание
Manual	Позицията на диалоговия прозорец се определя от стойността на свойството <code>Location</code> .
CenterScreen	Диалоговият прозорец се позиционира в центъра на екрана, а размерът му се определя от стойността на свойството <code>Size</code> .
WindowsDefaultLocation	Диалоговият прозорец се позиционира на подразбиращо се за Windows местоположение, а размерът му се определя от свойството <code>Size</code> .
WindowsDefaultBounds	Диалоговият прозорец се позиционира се позиционира на подразбиращо се за Windows местоположение и има подразбиращ се за Windows размер.
CenterParent	Диалоговият прозорец се позиционира в центъра на формата, от която се извиква.

Подразбиращата се стойност е `WindowsDefaultLocation`, което означава, че Windows позиционира диалоговия прозорец, а вашата програма задава размера му. Ако искате Windows да зададе и размера на диалоговия прозорец вместо вас, задайте стойност `WindowsDefaultBounds` на свойството `StartPosition`.

Стойността `CenterParent` позволява на програмата да позиционира диалоговия прозорец в центъра на формата на програмата без да извършва никакви изчисления. Но това разположение може да не е оптималното. `CenterScreen` намира приложение при немодалните диалогови прозорци, които понякога се появяват на екрана при зареждане на приложение.



Стойността Manual дава на диалоговият прозорец пълна свобода при задаване на неговата позиция и размер. Най-общо даден диалогов прозорец ще предпочита да използва тази стойност, за да позиционира себе си спрямо формата, която го извиква.

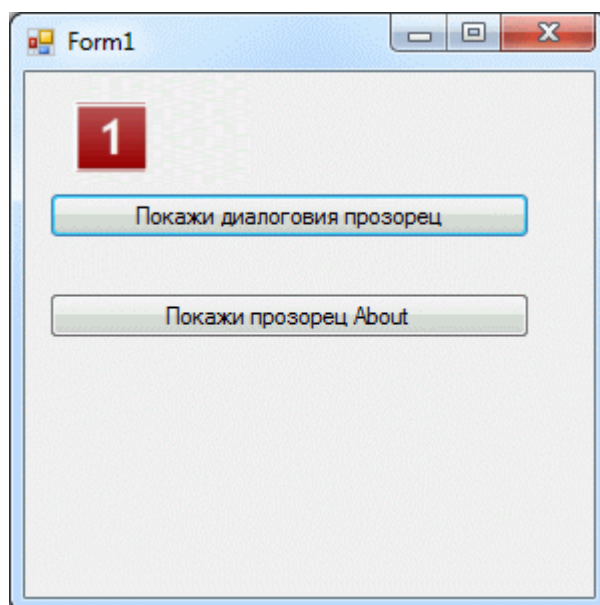


## Прозорецът About

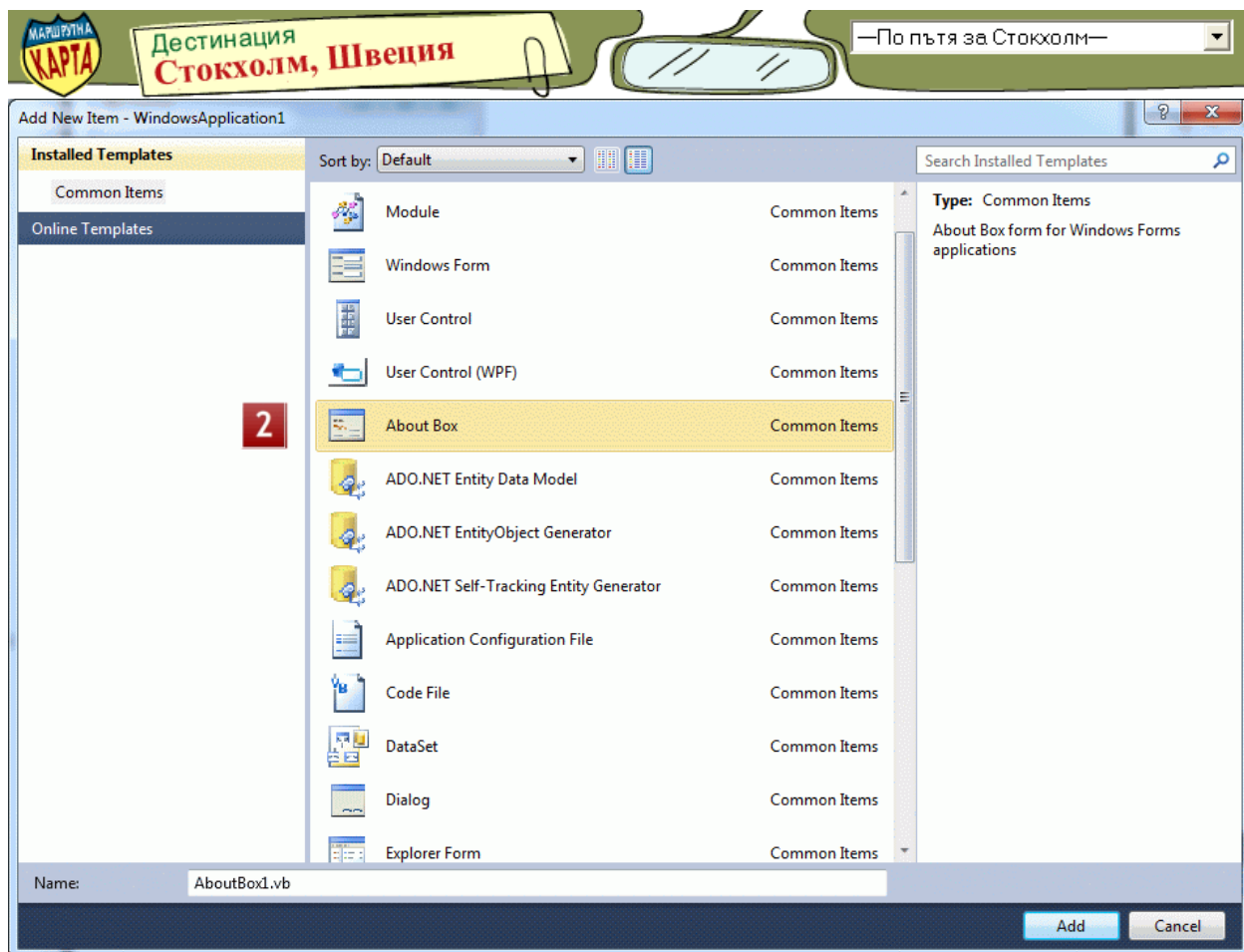
Един често срещан диалогов прозорец, който може да бъде открит в почти всички приложения, се нарича "About". Той може да бъде просто бележка с името на автора, създал програмата или сложен екран на системните ресурси и вероятно телефонен номер или уеб сайт за техническа поддръжка.

За да създадете такъв прозорец, следвайте стъпките по-долу.

1. Отворете създаденото вече от вас приложение и поставете още един бутон на формата.



2. От менюто Project на Visual Studio 2010 изберете командата Add Component.



3. В появилия се прозорец със заглавие Add New Item изберете шаблона About Box и натиснете бутона Add.
4. Ще се създаде нов диалогов прозорец с бутон Ok и полета, които можете да попълните с информация за вас. Върнете се отново при главната форма и щракнете два пъти върху новия бутон. В обработчика на събитието Click въведете следния код:

**AboutBox1.ShowDialog()**

5. Изградете и стартирайте програмата.

Какво се случва, когато натиснете новия бутон на формата?

## Стандартни диалогови прозорци

Една от основните цели на Windows винаги е била да предлага стандартизиран потребителски интерфейс. Ето защо съществуват стандартни диалогови прозорци, които можете да използвате наготово и които позволяват отваряне и затваряне на файлове, избор на цветове и шрифтове и настройка на печата.

В тази секция ще ви запозная с диалоговите прозорци за избор на шрифт и цвят. Да започваме!





## Диалог за избор на шрифт

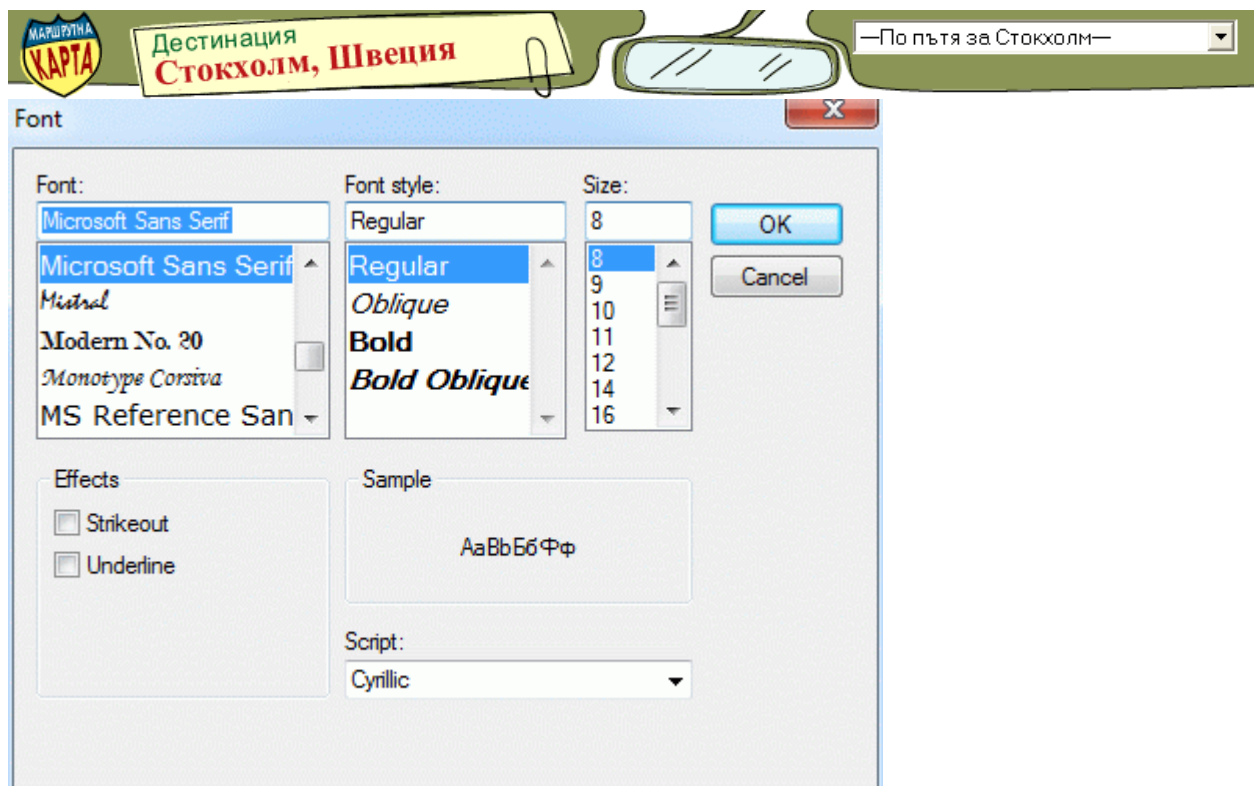
Стандартният диалогов прозорец FontDialog позволява избор както на шрифт, така и на цвят. Избраните шрифт и цвят са достъпни посредством свойствата му Font и Color. Можете да задавате настройките на диалоговия прозорец като използвате свойствата от таблицата по-долу:

Свойство	Описание	По подразбиране
ShowEffects	Показва или скрива опцията за избор на подчертан или зачертан шрифт.	True
ShowColor	Показва или скрива опцията за избор на цвят.	False
ShowApply	Показва или скрива бутон Apply.	False
ShowHelp	Показва или скрива бутон Help.	False

Създаването и извикването на диалогов прозорец за избор на шрифт е лесно. Достатъчно е да изтеглите компонента FontDialog от прозореца Toolbox и да добавите само един ред код в обработчика на събитието, при възникването на което ще се появява прозорецът:

**FontDialog1.ShowDialog()**

Преди извикването на метода ShowDialog можете да настроите описаните в таблицата свойства. Прозорецът по-долу е със зададена стойност True на свойствата ShowColor и ShowApply.





## Диалог за избор на цвят

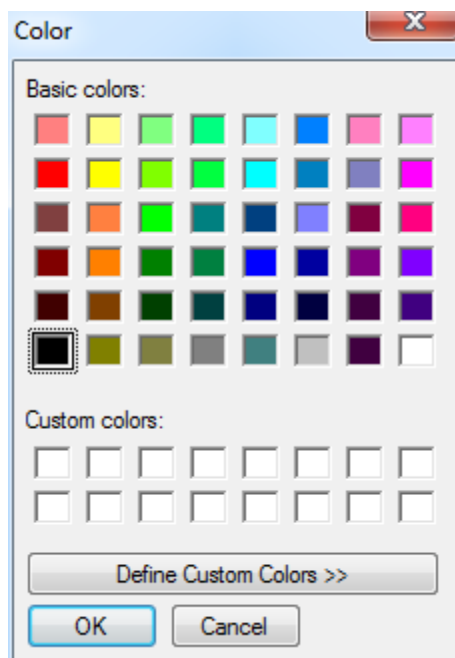
Стандартният диалогов прозорец ColorDialog позволява избор на цвят, който е достъпен посредством свойството Color. В таблицата по-долу са описани някои от по-съществените свойства на този прозорец.

Свойство	Описание	По подразбиране
FullOpen	Показва или скрива опцията за създаването на потребителски цветове при извикването на прозореца.	False
AllowFullOpen	Позволява или забранява възможността за създаването на потребителски цветове.	True
SolidColorOnly	Позволява или забранява възможността за избор само на чисти цветове.	False
AnyColor	Позволява или забранява показването на всички цветове от множеството на основните цветове.	False
ShowHelp	Показва или скрива бутон Help.	False





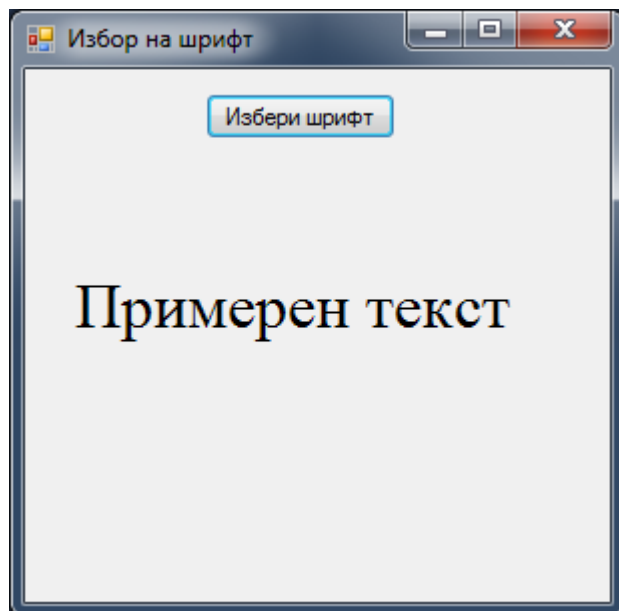
Създаването и извикването на диалогов прозорец за избор на цвят е аналогичен. Изтеглете компонента ColorDialog от прозореца Toolbox и добавете следния ред с код в обработчика на събитието, при възникването на което ще се появява прозорецът:



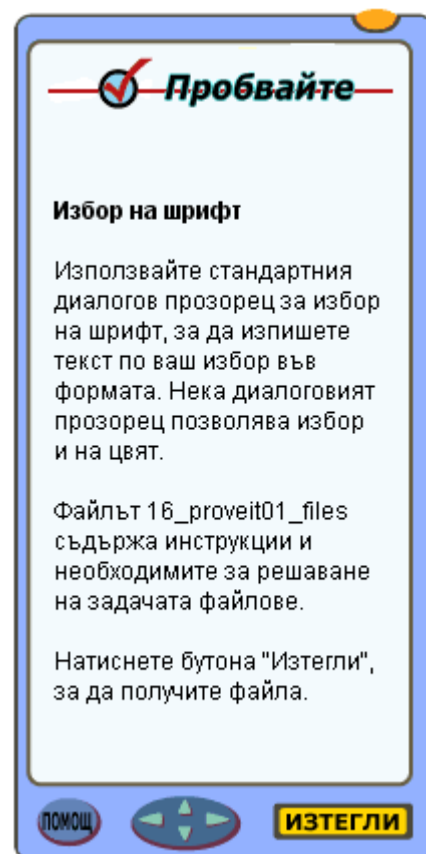
Диалоговият прозорец, който виждате е с настройките си по подрабиране.

## 16 Пробвайте 01 Избор на шрифт

Ето и вашата първа задача.



Декларирайте две глобални променливи MyFont и MyBrush съответно за шрифт и четка като им зададете начални стойности.



В обработчика на събитието Paint изпишете текст с метода DrawString като използвате за аргументи променливите MyFont и MyBrush.

Поставете бутон на формата. Създайте диалогов прозорец за избор на шрифт. В обработчика на неговото събитие Click използвайте върнатата от метода ShowDialog стойност, за да определите дали потребителят е потвърдил или отказал избора на шрифт. В случай, че е потвърдил актуализирайте променливите MyFont и MyBrush със стойностите на свойствата Font и Color на диалоговия прозорец. Извикайте метода Invalidate, за да предизвикате извикване на обработчика на събитието Paint, при което текста ще се изпише с новите шрифт и цвят.

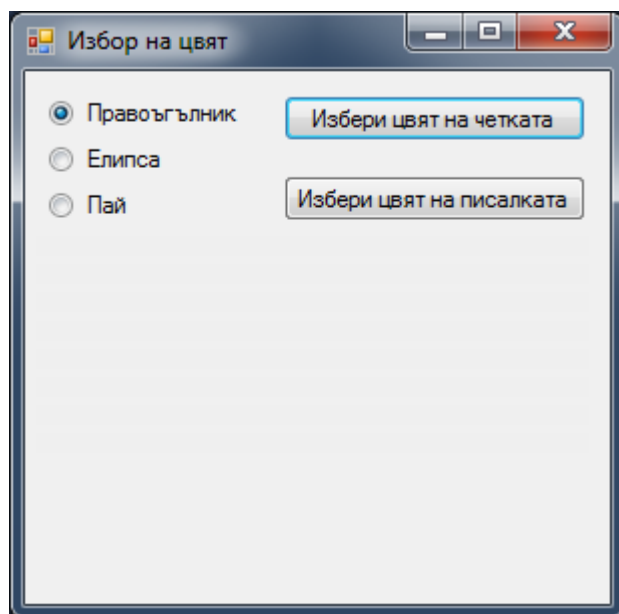
Изградете проекта и ако няма грешки изберете опцията Start Debugging от менюто Debug, за да стартирате програмата.

Ако програмата работи правилно, я покажете на вашия учител.

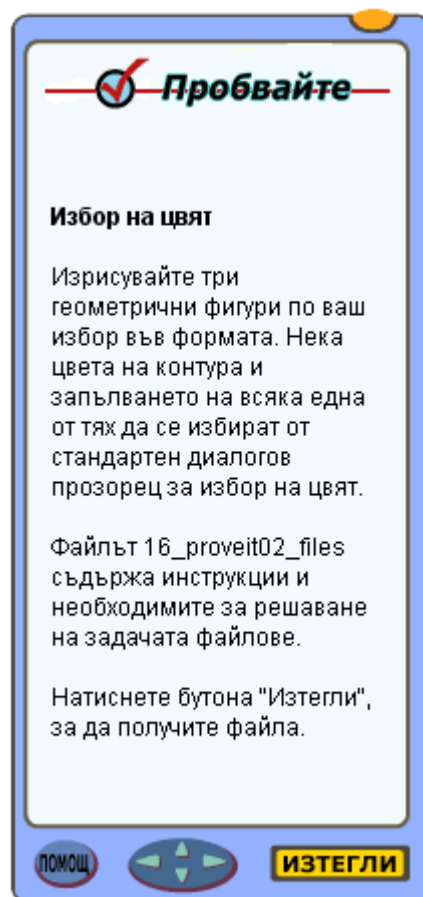
## 16 Пробвайте 02 Избор на цвят

Вече сме в Стокхолм! Сигурен съм, че се забавлявахте. Нека да направим останалите задачи, които са на джобния компютър, след което да решим теста, за да ни перфорират контролната карта.

Създайте форма подобна на следната:



Декларирайте две глобални променливи MyPen и MyBrush съответно за писалка и четка като им зададете начални стойности.



В обработчика на събитието Paint изрисувайте правоъгълник, елипса и пай със създадените писалка и четка.

Създайте диалогов прозорец за избор на цвят. В обработчика на събитието Click на двата бутона отново използвайте върнатата от метода ShowDialog стойност, за да определите дали потребителят е направил или отказал избора на цвят.

В обработчика на събитието Click на единия бутон актуализирайте стойността на свойството Color на четката със стойността на свойството Color на диалоговия прозорец за избор на цвят.

В обработчика на събитието Click на другия бутон актуализирайте стойността на свойството Color на писалката със стойността на свойството Color на диалоговия прозорец за избор на цвят.



И в двата обработчика направете проверка за това кой от радио-бутоните е избран и в зависимост от това създайте променлива от тип Rectangle като я инициализирате с обграждащия правоъгълник на съответната фигура.

Например ако е избран радио-бутон със заглавие "Правоъгълник", а правоъгълника се изрисува на позиция (10,10) с размер (100,100), то вашата променлива ще бъде декларирана така:

```
Dim Rect As New Rectangle(10, 10, 100, 100)
```

След това подайте променливата Rect като аргумент на метода Invalidate.

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

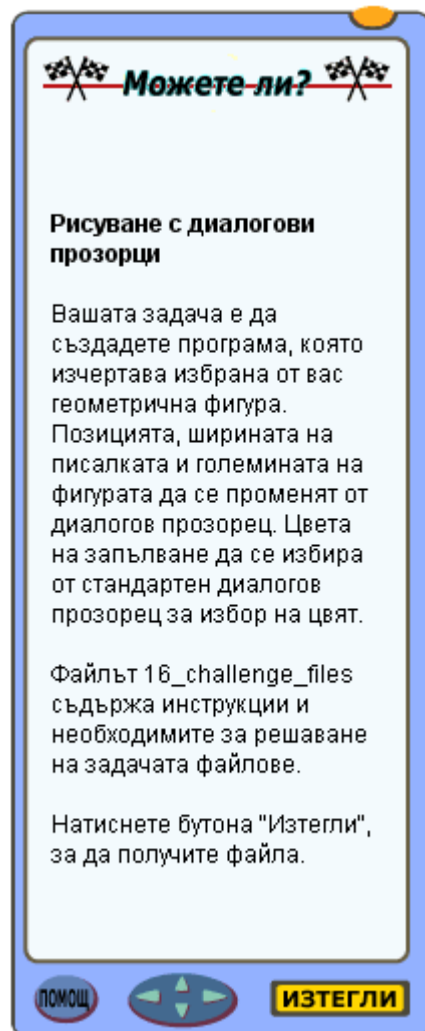
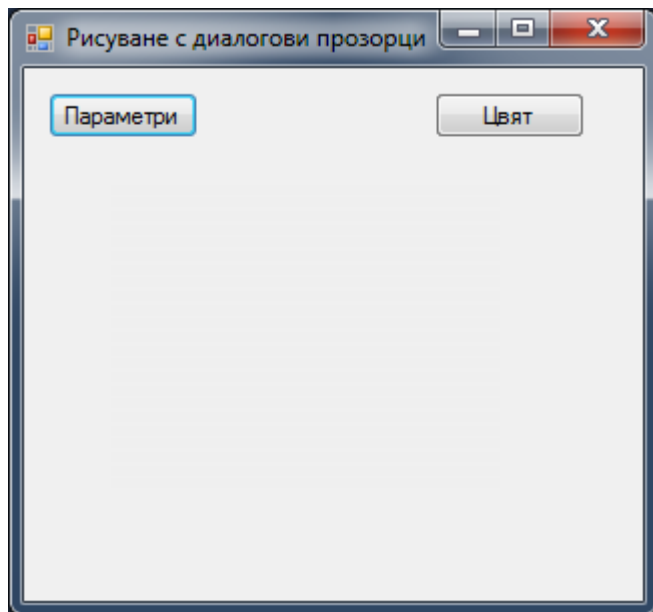
Ако програмата работи правилно, я покажете на вашия учител.



## 16 Можете ли? Рисуване с диалогови прозорци

Време е за поредното предизвикателство.

Създайте форма подобна на показаната по-долу:

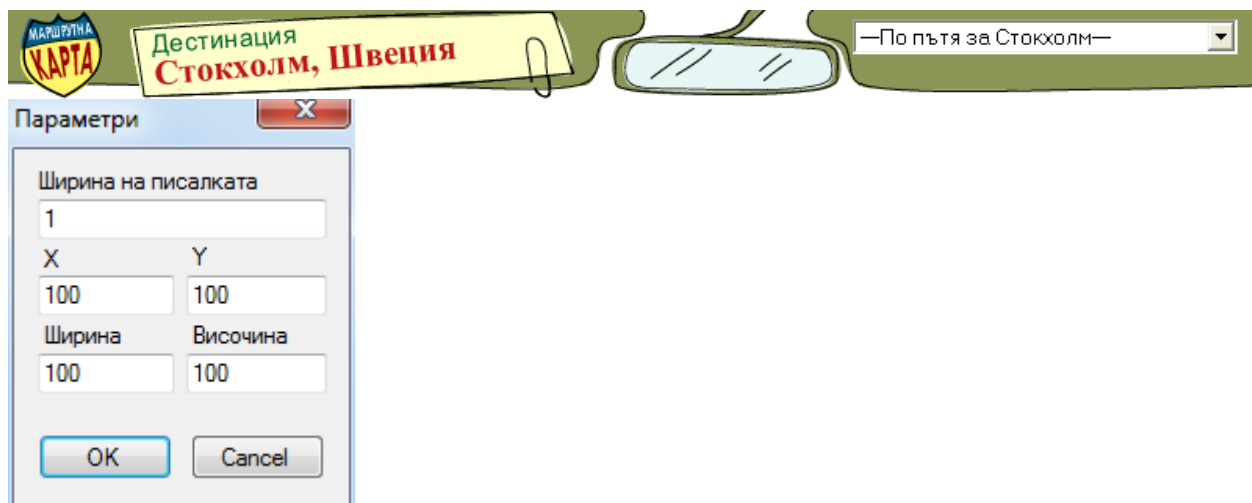


Декларирайте шест глобални променливи като използвате следната таблица:

Променлива	Тип	Предназначение
x	Integer	Задава позиция по x на фигурата.
y	Integer	Задава позиция по y на фигурата.
FigureWidth	Integer	Задава ширина на фигурата.
FigureHeight	Integer	Задава височина на фигурата.
Brush	SolidBrush	Задава четка за запълване на фигурата.
Pen	Pen	Задава писалка за контура на фигурата.

В обработчика на събитието Paint изрисуйте фигурата.

Създайте диалогов прозорец, който изглежда така:



В обработчика на събитието Click на единия бутон прехвърлете стойностите от глобалните променливи (с изключение на Brush) към текстовите полета на диалоговия прозорец. Не забравяйте, че стойността, която се въвежда в текстовото поле на диалоговия прозорец за ширината на писалката съответства на свойството Width на писалката Pen. След това покажете диалоговия прозорец с метода ShowDialog и ако потребителят е затворил прозореца с бутона OK, прехвърлете стойностите на текстовите полета към глобалните променливи. Накрая извикайте метода Invalidate.

Създайте диалогов прозорец за избор на цвят. В обработчика на събитието Click на другия бутон използвайте върнатата от метода ShowDialog стойност, за да определите дали потребителят е направил или отказал избора на цвят. Използвайте стойността на свойството Color на диалоговия прозорец, за да зададете стойност на свойството Color на четката. Извикайте метода Invalidate.

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.

- 1 Кой диалогови прозорци не позволяват преминаване към дадена форма преди да бъдат затворени?
  - ☐ A. Модалните
  - ☐ B. Немодалните
  - ☐ C. И двата
- 2 Кой метод реализира показване на диалогов прозорец?
  - ☐ A. OpenFileDialog
  - ☐ B. ShowDialog
  - ☐ C. PrintDialog
- 3 С кое свойство се управлява местоположението на диалоговия прозорец?
  - ☐ A. Position
  - ☐ B. StartPosition
  - ☐ C. Location
- 4 За какво се използва диалоговият прозорец ColorDialog?
  - ☐ A. Избор на цвят
  - ☐ B. Избор на шрифт
  - ☐ C. Извеждане на помощна информация