



## Писалки и четки

### Улан Батор, Монголия

Нашата следваща дестинация е столицата на Монголия – Улан Батор. Те е разположена на река Туул, приток на Селенга, в долина от полите на планината Богдо Уул.

Вече изучихте как да създавате форми с различни по вид и местоположение контроли, но може би си мислите дали не може да използвате формата като платно, върху което да рисувате свободно. Разбира се, че може!

Нашето пътешествие в света на програмирането ще продължи с изучаването на четките и писалките, но преди това ще ви разкажа за това как и къде във вашия код можете да създадете платно за рисуване и как възниква събитието Paint. Да вървим!





## Събитието Paint

Спомняте ли си какво представляваха събитията. Научихме различни неща за тях, когато бяхме в Казабланка. Но едно от най-загадъчните събития – може би най-малко вероятният кандидат за събитие – е едно от най-важните. Това събитие, известно като Paint, съобщава на вашата програма кога една част или цялата форма е невалидна и трябва да се изрисува.

По какъв начин формата става невалидна? Когато формата бъде създадена за пръв път, тя е невалидна, защото все още програмата не е изрисувала нищо. Първото събитие Paint, което програмата получава, е да ѝ укаже да нарисува нещо във формата.

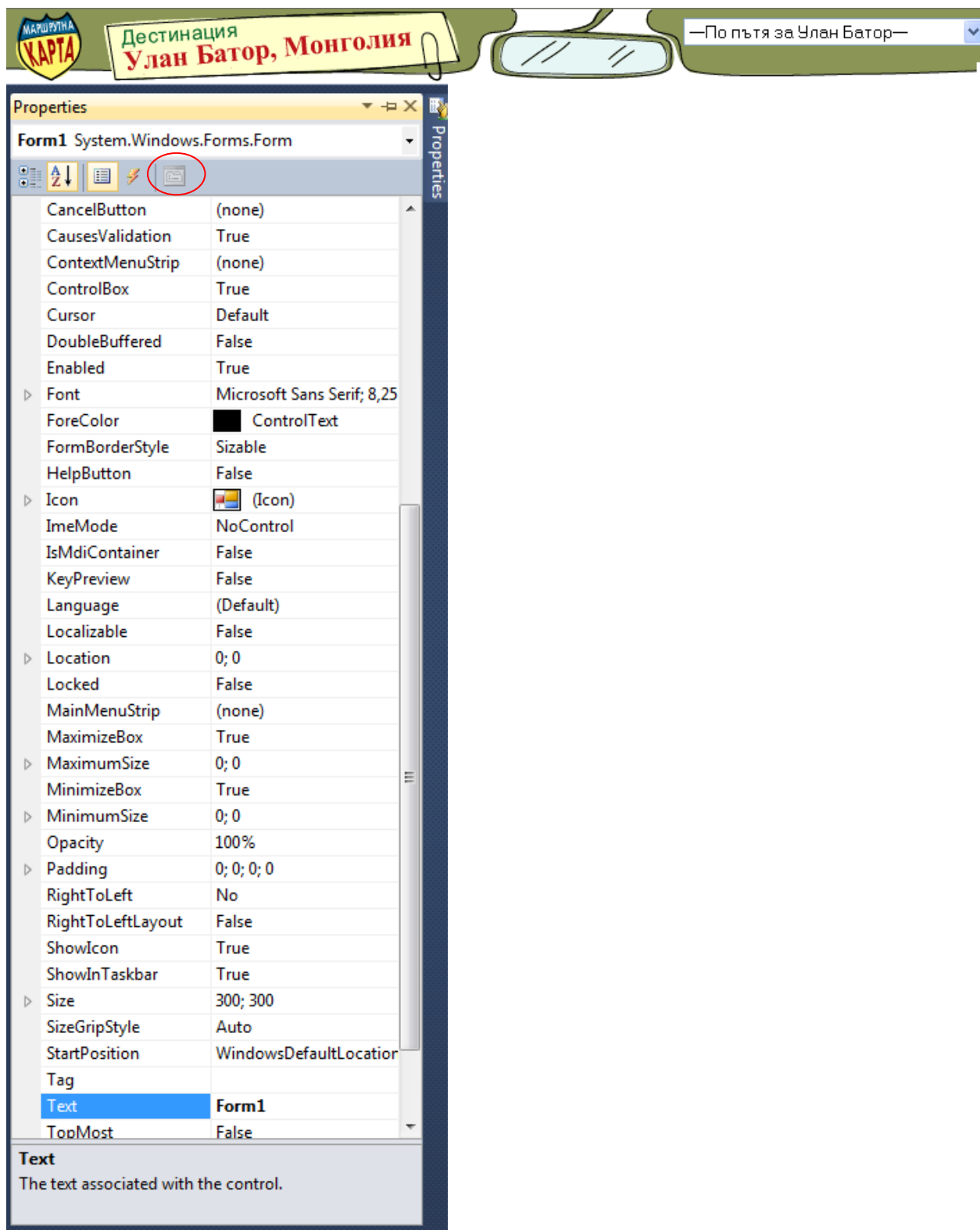
Когато движите формите по екрана така, че да се припокриват една друга, Windows не запазва показаното, което е покрито от друга форма. Когато част или цялата форма по-късно се открие, програмата трябва да възстанови показваното. Поради това тя получава друго събитие Paint. Когато възстановите програма, която преди това е била минимизирана, получавате друго събитие Paint.

Windows програмите трябва да могат изцяло да прерисуват формите си по всяко време. Трябва да запазват или да държат бързо достъпна цялата информация, необходима за това. Да структурирате вашата програма правилно да отговаря на събитието Paint може да звучи много ограничаващо, но ще свикнете.



## Обработване на събитието Paint

Ето как можете да добавите код за обработване на събитието Paint. Нека добавим събитие Paint към форма, която преди това сте създали. Щракнете където и да е във формата, но извън контролите, ако има такива. В прозореца Properties щракнете върху бутончето със светкавицата.



Ще се появи списък с всички събития, които можете да обработвате за формата. Намерете събитието Paint и щракнете двукратно върху него. Ще се генерира код, който изглежда по следния начин:

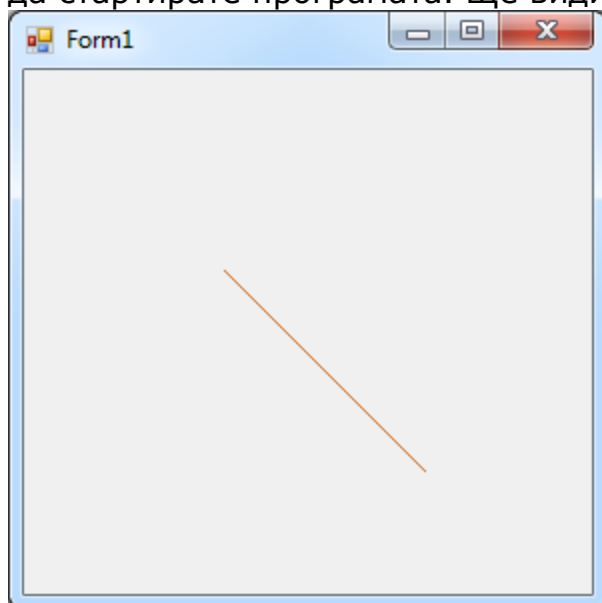


```
Public Class Form1
    Private Sub Form1_Paint(ByVal sender As System.Object, ByVal e _
        As System.Windows.Forms.PaintEventArgs) Handles MyBase.Paint

    End Sub
End Class
```

Въведете следния код:

Не се притеснявайте, ако не разбирате нищо. Изградете приложението. Ако няма грешки изберете опцията Start Debugging от менюто Debug, за да стартирате програмата. Ще видите следния резултат:



## Платното за рисуване

Разгледайте параметрите на процедурата, която обработва събитието Paint. Параметърът с име "e" ви предоставя платно за рисуване с име Graphics. Това е въображаема повърхност, върху която можете да нарисувате всичко, което си пожелаете – от по-прости линии и криви, до сложни фигури с разнообразно запълване. Когато ви е необходимо, сами можете да създадете платно за рисуване с ред, подобен на този:

```
Dim grfx As Graphics = e.Graphics
```

Можете да наречете Graphics платното за рисуване както пожелаете. Някои програми използват graphics с малка буква, но честото му използване често води просто до използване на буквата "g" за обозначаването му. Аз ще използвам Graphics платното за рисуване предоставено със събитието Paint.

Платното за рисуване Graphics ви предоставя широк набор от средства за рисуване. С тяхна помощ можете да рисувате линии, правоъгълници, елипси, сложни многоъгълници, текстове с различен наклон и много други. Тях ще разгледаме по-нататък по време на пътуването ни. Разбира се, когато рисуваме винаги имаме нужда от писалки и четки. Нека научим нещо за тях!





## Инструменти за рисуване

Когато чертаете линии или криви, използвате писалка, а когато рисувате запълнени области и текст използвате четка. Писалките и четките във Visual Basic .NET се обозначават съответно с ключовите думи Pen и Brush. Писалките и четките имат конкретен цвят, но заедно с това и много други характеристики. Можете ли да се досетите за някои от тях? В този курс ще изучим най-необходимите.

### Полезен съвет

В този курс учениците не се запознават с понятията клас и обект. Можете да наричате писалките и четките обекти, но бъдете внимателни. Това може да внесе объркване в понятията на учениците.

### Полезен съвет

Провокирайте учениците да се замислят върху характеристиките на писалките и четките. Например, нека дадат отговор на въпроса относно това, което ги кара да сменят моливи с различни цветове или да използват различни техники за рисуване.



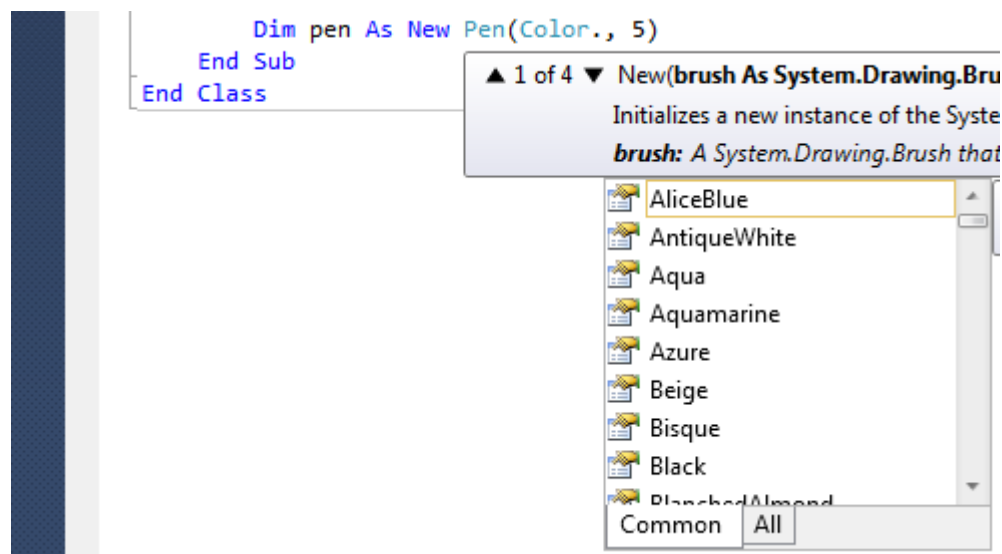
## Писалки

Когато чертаете линия на ръка върху хартия, използвате молив, креда, писалка химикалка, флумастер или нещо друго. Типът на инструмента, който избирате определя най-малко цвета и дебелината на линията. Тези и други характеристики принадлежат и на писалките във Visual Studio 2010. Писалките могат да бъдат създадени от четките, а четките от различни изображения. Това обаче са по-сложни въпроси от програмирането на компютърната графика, които за сега няма да разглеждаме.

Можете да създадете писалка като използвате следния ред с код:

```
Dim pen As New Pen(Color.Aqua, 5)
```

В скобите записвате два параметъра. Първият от тях задава цвета на писалката, а вторият – дебелината. За да зададете цвета на писалката напишете ключовата дума "Color" и след това поставете точка. Visual Studio 2010 ще ви предостави цял списък с цветове, които можете да използвате:



Вторият параметър задава дебелината на писалката. Можете да го пропуснете. Тогава ще се създаде писалка с дебелина 1. В случая съм създал писалка с дебелина 5.



Писалките не е задължително да бъдат задавани плътни линии. Вместо това те могат да се състоят от шаблони с точки и тирета. Този стил се задава със свойство на писалката с име `DashStyle`. Таблицата по-долу описва различните стилове, които можете да задавате на писалките:

| Стил | Описание   |
|------|--|
| 0    | Плътна линия                                     |
| 1    | Пунктирана линия с тире.                         |
| 2    | Пунктирана линия с точка.                        |
| 3    | Пунктирана линия с редуващи се тире и точка.     |
| 4    | Пунктирана линия с редуващи се тире и две точки. |
| 5    | Определен от вас стил.                           |

Стилът на писалката можете да промените със следния ред код:

```
pen.DashStyle = 3
```

За да я използвате обаче, когато рисувате линии и криви, трябва да я зададете като първи параметър на съответния инструмент за рисуване. Различните инструменти за рисуване ще ви покажа в следващите уроци. За сега е необходимо да запомните, че по някакъв начин трябва да кажете на вашата програма да използва създадената от вас писалка. Например редът с код, който видяхте, че изчертава линия би изглеждал по следния начин:

```
e.Graphics.DrawLine(pen, 150, 100, 200, 200)
```



## Четки

Когато чертаете текст или запълнени области, използвате четка. За да създадете четка трябва да напишете код, подобен на следния:

```
Dim brush As New SolidBrush(Color.Aqua)
```

Този ред с код създава четка с плътно запълване. Четките могат да бъдат в пет различни стила, задаващи запълването. Таблицата по-долу описва всеки един от тях.

| Четка               | Описание                                   |
|---------------------|--|
| SolidBrush          | Плътно запълване.                          |
| HatchBrush          | Запълване с шарка.                         |
| TextureBrush        | Запълване с текстурата.                    |
| LinearGradientBrush | Запълване с линеен градиент.               |
| PathGradientBrush   | Запълване с градиент чрез задаване на път. |

Четката с шарка запълва област с малка повтаряща се шарка, най-често състояща се от вертикални, хоризонтални или диагонални линии. Когато използвате такава четка задавате три свойства – стил, цвят на фона и цвят на предния план по следния начин:

### Полезен съвет

Помогнете на учениците си да добавят следните библиотеки в началото на кода за формата:

```
Imports System.Drawing
```

```
Imports System.Drawing.Drawing2D
```

```
Dim brush As New HatchBrush(4, Color.Aqua, Color.Black)
```

Имате възможност да избирате между 53 стила. В таблицата по-долу са описани някои от тях:

| Стил | Описание  |
|------|---|
| 0    | Хоризонтална щриховка                               |
| 1    | Вертикална щриховка.                                |
| 2    | Ляво наклонена щриховка.                            |
| 3    | Дясно наклонена щриховка.                           |
| 4    | Щриховка с хоризонтални и вертикални линии.         |
| 5    | Щриховка с ляво и дясно наклонени вертикални линии. |

Четката, която запълва с текстура използва картинка. Картинката се разполага многократно вертикално и хоризонтално в областта, която



желаете до нейното пълно запълване. Създаването на подобна четка става по следния начин:

```
Dim MyImage As Image = Image.FromFile("image.jpg")  
Dim textureBrush As New TextureBrush(MyImage)
```

Първият ред от кода задава картинката, която ще използвате при създаването на четката. С втория ред създавате четката като и задавате избраната от вас картинка.

Останалите два стила за четки се описват с думата **градиент**, която означава преход между цветовете. Запълването с линеен градиент включва преход между два цвята. На пръв поглед може да изглежда сложно да се създаде начин, по който един цвят се слива с друг, и вероятно затова има два начина за задаване на такава четка. Градиентът между два цвята може да се зададе с двойка успоредни линии. Всяка линия има чист цвят. Цветът прави преход между двете линии. За да създадете подобна четка не е нужно да задавате две успоредни линии. Достатъчно е да зададете две точки. Всяка от двете точки задавате с координати *x* и *y*. Също така трябва да зададете начален и краен цвят, които ще се използват за създаване на преливането:

#### Полезен съвет

Пояснете на учениците, че всичко което рисуват на формата става във въображаема координатна система, чието начало съвпада с горния ляв ъгъл на формата, като *x* стойностите нарастват надясно, а *y* стойностите надолу.

```
Dim LienarGradientBrush As New LinearGradientBrush(_  
    New Point(10, 10), New Point(40, 40),_  
    Color.Aqua, Color.Blue)
```

Последния стил четка изисква да зададете многоъгълник, който ще бъде запълнен с преливащ цвят. Многоъгълника можете да определите като набор от точки по следния начин:

```
Dim points() As Point = {New Point(10, 10),_  
    New Point(10, 300),_  
    New Point(150, 300)}
```

След това наборът от точки можете да използвате, за да създадете четка по следния начин:

```
Dim PathGradientBrush As New PathGradientBrush(points)
```

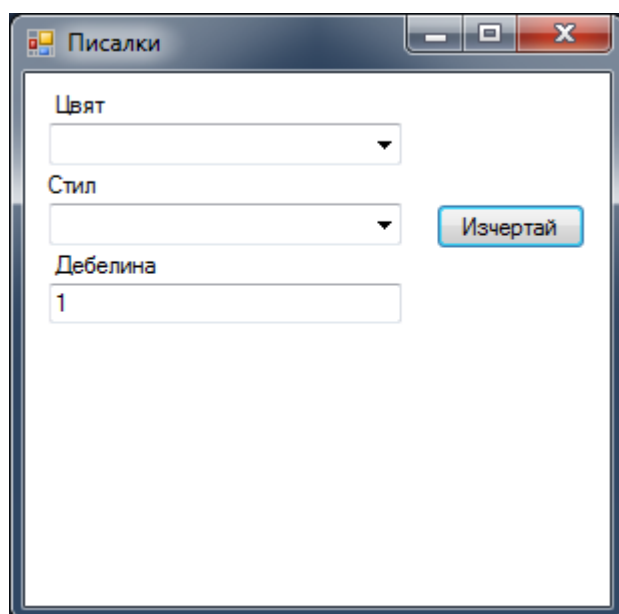


И накрая, използвайте всяка една от четките в предоставените от Visual Basic инструменти за рисуване ето така:

## 13 Пробвайте 01 Писалки

Стига толкова приказки. Време е да напишете вашата първа графична програма.

Създайте форма подобна на показаната по-долу:



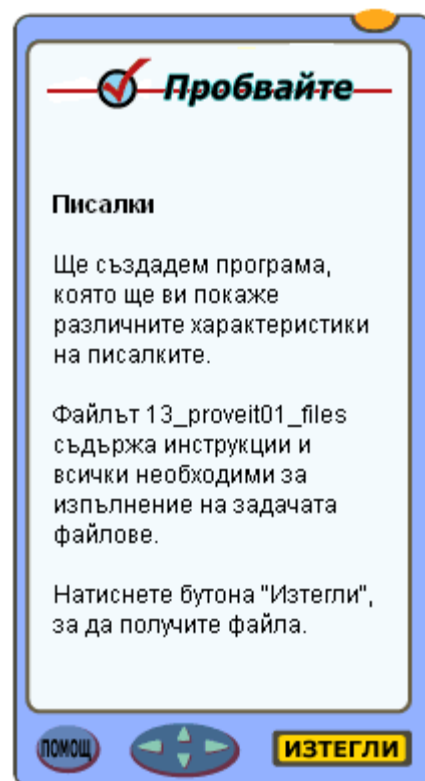
В първия падащ списък въведете следните стойности:

- Тъмно син
- Червен
- Зелен
- Небесно син
- Пурпурен
- Жълт
- Лилав
- Кафяв

Във втория падащ списък въведете следните стойности:

- Плътна
- Пунктирана с тире
- Пунктирана с точка
- Пунктирана с тире и точка
- Пунктирана с тире и две точки

Задайте стойност 1 на свойството Text на текстовото поле.





Променете цвета на формата на Window като използвате свойството `BackColor`.

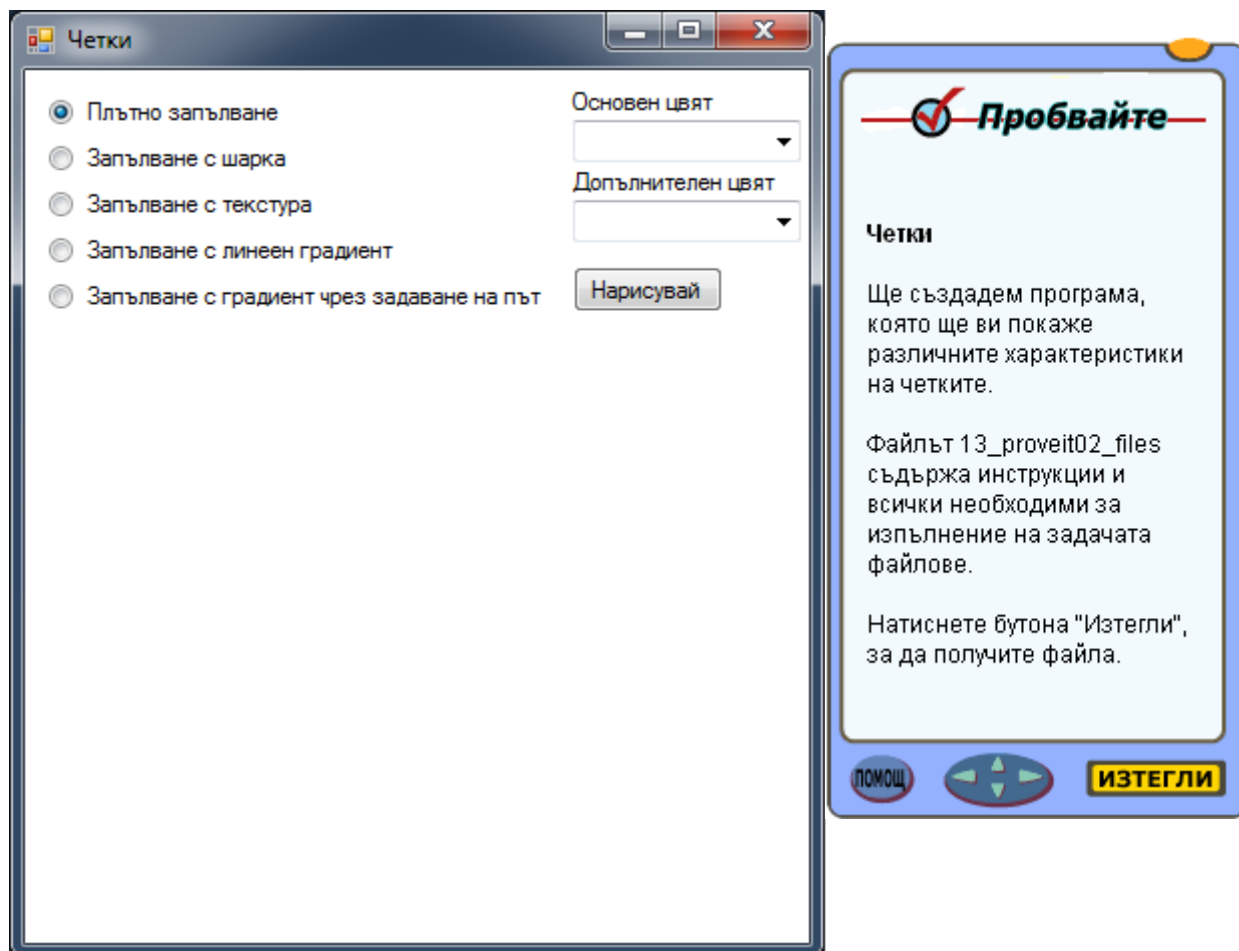
Задайте подходящо заглавие на формата и на бутона върху нея.

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.

## 13 Пробвайте 01 Четки

Дойде време да се занимаем и с четките.  
 Създайте форма подобна на показаната по-долу:



При поставянето на радио-бутоните проверете дали свойството им Name е в диапазона от RadioButton1 до RadioButton5.

В двата падащи списъка въведете следните стойности:

- Тъмно син
- Червен
- Зелен
- Небесно син
- Пурпурен
- Жълт
- Лилав
- Кафяв





Променете цвета на формата на Window като използвате свойството `BackColor`. Задайте стойности 400 и 470 съответно за ширината и височината ѝ.

Задайте подходящо заглавие на формата и на бутона върху нея.

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.

Забелязахте ли разликата между отделните стилове четки?

### 13 Можете ли? Четката с шарка

Спомняте ли си четката с шарка. Можете да я използвате с 53 стила. За да видите всеки един от тях, отговорете на следващото предизвикателство.

Кодът, който ще напишете поставете в обработчика на събитието Click на формата. В секция 9 научихте как се генерира случайно число в даден интервал. В случая трябва да генерирате случайно число в интервала от 0 до 53, което след това да използвате като стойност за стила на четката с шарка.

Не забравяйте да създадете платно за рисуване, преди да създадете и използвате четката.

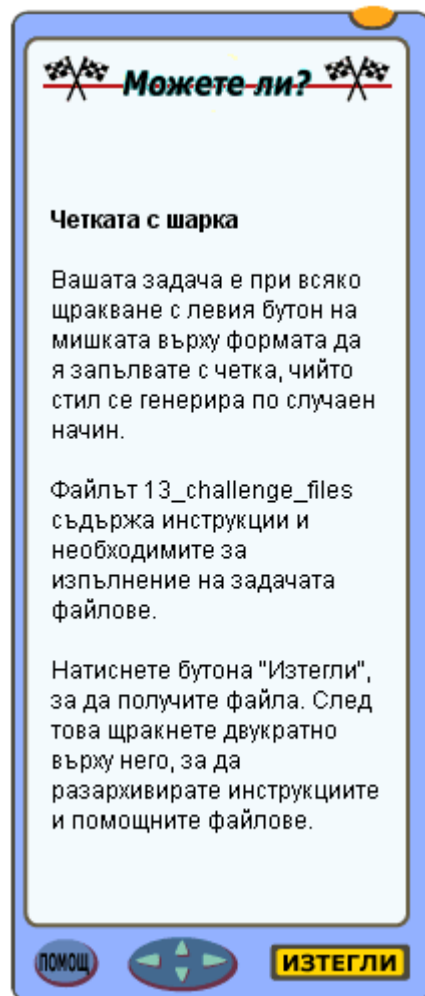
За попълване на формата със създадената от вас четка можете да използвате следния ред с код:

```
g.FillRectangle(brush, 0, 0, 300, 300)
```

В кода по-горе "g" съответства на платното за рисуване Graphics. В следващата секция ще научите как да използвате метода FillRectangle.

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.



МАРШРУТНА  
КАРТА

Дестинация  
Улан Батор, Монголия

—По пътя за Улан Батор—

?

Проверка на знанията

НАПРАВЕТЕ ТЕСТА ОТНОВО

1

Кога възниква събитието Paint?

☐ А. Когато щракнете с левия бутон на мишката.

☐ В. Когато формата е невалидна.

☐ С. Когато щракнете върху клавиш от клавиатурата.

2

Как се нарича платното за рисуване във Visual Basic .NET?

☐ А. Graphics

☐ В. Rectangle

☐ С. Brush

3

Контурът на фигурите, които рисувате се определя от:

☐ А. Писалка

☐ В. Платно за рисуване

☐ С. Четка

4

Колко стила четки има във Visual Basic .NET?

☐ А. 3

☐ В. 8

☐ С. 5