



## Рисуване в прозорец

### Москва, Русия

Здравейте отново! Скоро ще разгледаме средновековната крепост Кремъл в Москва. Тя служи за седалище на руското федерално правителство. Катедралата Свети Василий Блажени край Кремъл с елегантните си куполи е един от най-запомнящите се символи на Москва. Москва е сърцето на руските сценични и изящни изкуства. Градът е осеян с театри за драма, опера и балет, сред които най-известен е Большой театър. Искрено се надявам да го посетим, когато стигнем там.

Ди Джей ме заинтригува доста с възможностите във Visual Basic 2010 за рисуване. Аз се постарах да науча още неща, за които ще ви разкажа по време на пътуването ни до следващата дестинация.

Според Евклид "Линията е дължина без ширина". Думите "без ширина" правят това твърдение интересно. Аз, а може би и вие, не съм съгласен с древните гърци. Те не са знаели нищо за компютърната графика. Мислете за пиксела като точка от екрана с определени размери. Компютрите използват тази мерна единица, за да определят дебелината на линиите. Разбира се, възможно е да се използват и други мерни единици. Това обаче, което искам да ви покажа е как бихте могли да рисувате различни линии, криви и запълнени области.

Да вървим!



## Прави линии

Платното за рисуване Graphics предоставя широк набор от инструменти за рисуване. Във Visual Basic те се наричат методи. Всеки един от методите е съпроводен от аргументи, които определят къде и как да се рисува, с каква писалка или четка.

За да начертаете единична права линия, използвайте метода DrawLine. Той получава аргументи, които задават координатите, където започва и завършва линията, и писалката, използвана за чертане на линията.

Например следният ред с код ще изчертае линия, която започва от точка с координати (0,0) и завършва в точка с координати (5,5).

```
grfx.DrawLine(pen, 0, 0, 5, 5)
```

Може би вашият учител ви е разяснил, че есичко което рисувате се разполага в мислена координатна система, началото на която съвпада с горния ляв ъгъл на прозореца. В нея стойностите на x нарастват надясно, а стойностите на y – надолу.

Кодът по-горе оцветява 6 пиксела. Това са пикселите в точки с координати (0,0), (1,1), (2,2), (3,3), (4,4), (5,5). Редът на задаване на двете точки няма значение, затова кодът:

```
grfx.DrawLine(pen, 5, 5, 0, 0)
```

дава същия резултат.



## Правоъгълници

Правоъгълниците не се срещат в природата твърде много, но определено са най-разпространената форма на обекти, проектирани и построени от хората. Правоъгълниците са навсякъде. Страницата, която четете сега, е правоъгълник; параграфите са форматиращи в правоъгълници; прозорецът, през който гледате навън, когато моят текст започне да ви отегчава, вероятно също е правоъгълен.

Определено можете да начертаете правоъгълник с помощта на DrawLine, но по-прост начин предоставя методът DrawRectangle. Той има аргументи, които задават точка, определяща горния ляв ъгъл на правоъгълника, ширината и височината на правоъгълника. Естествено като аргумент трябва да подадете и писалката, с която желаете да изрисувате правоъгълника. Ширината и височината на правоъгълника трябва да бъдат по-големи от 0. При задаване на отрицателни ширини и височини няма да се изчертае нищо. Например:

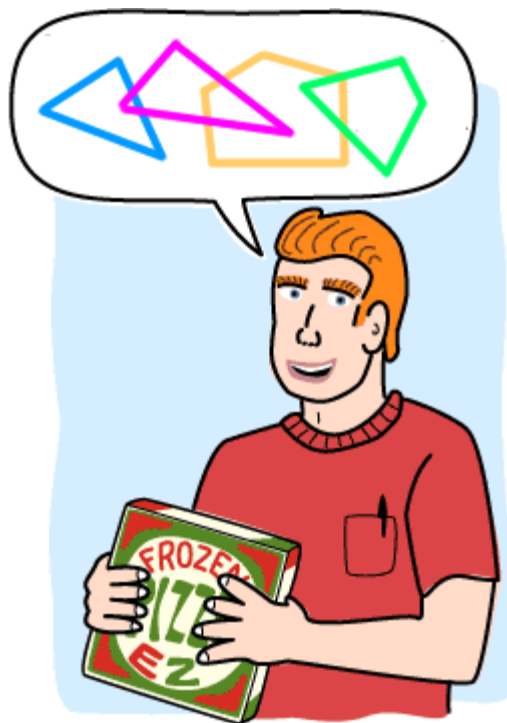


```
grfx.DrawRectangle(pen, 0, 0, 200, 200)
```

изчертава правоъгълник с височина и ширина 200 пиксела, чийто горен ляв ъгъл съвпада с горния ляв ъгъл на формата. Лесно е, нали!

## Многоъгълници

Математическите многоъгълници са затворени фигури с три или повече страни, като триъгълници, четириъгълници, петоъгълници, шестоъгълници, седмоъгълници и т.н. Тях можете да изчертаете с метода `DrawPolygon`. Освен писалката, с която ще чертаете многоъгълника, този метод притежава аргумент, представляващ набор от точки, съответстващи на върховете на многоъгълника.



Ето и пример:

```
Dim points() As Point = {New Point(0, 0), _  
                          New Point(50, 100), _  
                          New Point(100, 0)}  
grfx.DrawPolygon(pen, points)
```

Примерният код изчертава триъгълник, който прилича на буквата "V" в английската азбука. Както виждате триъгълникът се задава с три точки, съответстващи на върховете му.

В някои случаи можете да симулирате извикването на `DrawPolygon` с извикване на `DrawLine`. Ефектът, обаче не винаги е същият.

## Елипси и дъги

Изчертаването на елипса е толкова лесно, колкото и на правоъгълник. Методът `DrawEllipse`, с който можете да направите това, приема същите аргументи както метода `DrawRectangle`. Вие трябва да зададете координатите и размерите на въображаем правоъгълник, в който елипсата ще се изрисува като вписана. Следният код изчертава елипса, която е вписана в правоъгълник с височина и ширина 200 пиксела, чийто горен ляв ъгъл съвпада с горния ляв ъгъл на формата:

### Полезен съвет

Изяснете на учениците си, ако е необходимо понятието: вписана елипса в:

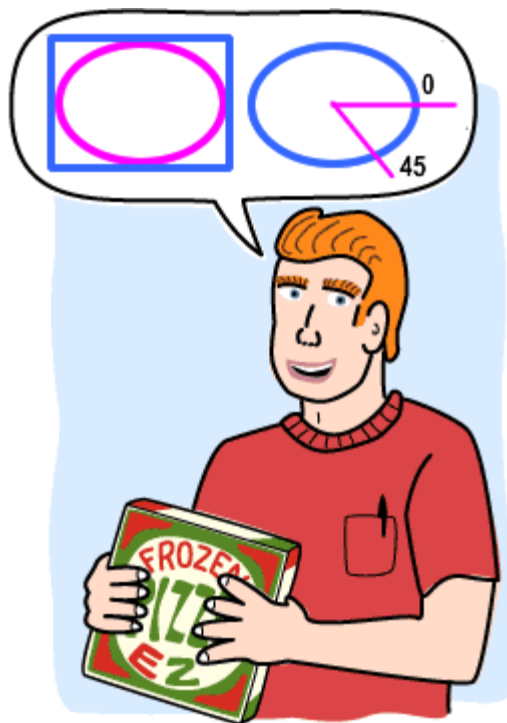
```
gfx.DrawEllipse(pen, 0, 0, 200, 200)
```

Ако правилно сте разбрали това, което ви разказах за елипсите, няма да имате проблем и с дъгите. Дъгата представлява сегмент от елипсата. За изчертаете дъга, трябва да зададете същата информация, необходима и за елипса, плюс това да зададете къде започва и къде завършва дъгата. Поради тази причина методът `DrawArc` изисква същите аргументи, както `DrawEllipse` плюс още два аргумента. Ето и пример:

```
gfx.DrawArc(pen, 0, 0, 200, _  
200, 30, 90)
```

Двата допълнителни аргумента са ъглите, които определят началото на дъгата и нейната дължина. Ъглите могат да бъдат положителни или отрицателни. Те се измерват по посока на часовниковата стрелка в градуси, от началото на хоризонталната ос надясно от центъра на елипсата. Това означава, че ъгъл със стойност 0 градуса съответства на 3:00 часа на часовника.

Методът `DrawPie` има същите аргументи като `DrawArc`, но чертае линии от краищата на дъгата до центъра на елипсата, като създава затворена област.





grfx.DrawPie(pens, 0, 0, 200, 200, 30, 90)



## Запълване на правоъгълници, елипси и сектори

Разгледаните до тук методи, които чертаят затворени области, не запълват тяхната вътрешност. За тях вие можете да укажете писалката, но не и четката, която да се използва. За щастие освен методи започващи с Draw, съществуват и методи, които започват с Fill и се използват за запълване на области. Първият аргумент за тези методи е четката, използвана за запълване на вътрешността.

Методът FillRectangle се използва за запълване на правоъгълна област. С изключение на първия аргумент всички останали съвпадат с тези на метода DrawRectangle. Например кодът

```
grfx.FillRectangle(brush, 10, 10, 400, 400)
```

запълва правоъгълник с височина и ширина 400 пиксела и с координати на горен ляв ъгъл (10,10) с указаната четка.

По аналогичен начин методът FillEllipse запълва затворена област под формата на елипса. Както можете да се досетите този метод получава същите аргументи като DrawEllipse с изключение на първия параметър, който вече е четка, а не писалка.

```
grfx.FillEllipse(brush, 10, 10, 400, 400)
```

Съществува малко припокриване между елипсата, начертана от DrawEllipse и областта, запълнена от FillEllipse. Ако искате да начертаете елипса, която е едновременно запълнена и очертана с контур, извикайте FillEllipse преди извикването на DrawEllipse. Това правило спазвайте и за останалите двойки методи Draw-Fill.

Да не забравяме и метода FillPie, който запълва област под формата на пай.

```
grfx.FillPie(brush, 0, 0, 200, 200, 30, 90)
```

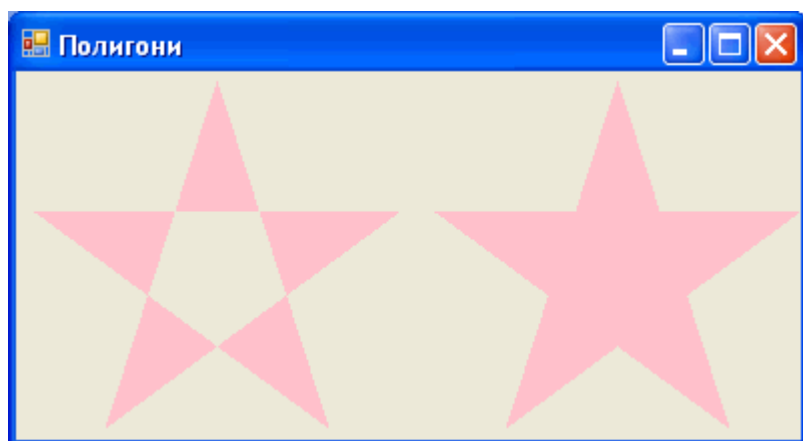
Накрая разполагате и с метод FillPolygon. Това, което прави многоъгълника различен от другите запълнени области е, че линиите, които го определят, могат да се пресичат и застъпват. Това добавя степен на сложност, защото вътрешностите на полигона могат да бъдат запълнени по два различни начина. Начинът на запълване се определя от последния (трети) параметър на метода, който може да бъде 0 или 1. При стойност 0 се реализира редуване на запълнени и незапълнени области, а при стойност 1 се запълват най-много вътрешни области.





Начинът на запълване може да се забележи само, ако линиите, определящи многоъгълника, се застъпват. На практика той определя кои области да се запълнят и кои не. Ако не е зададен начин на запълване в метода FillPolygon, се приема, че стойността му е 0. В този случай дадена затворена област се запълва само ако има нечетен брой граници между затворената област и безкрайността.

Може би малко ви обърках. Сега ще се опитам да изясня по-добре нещата. Класическият пример е звездата с пет върха. Когато начинът на запълване има стойност 1, вътрешността на петоъгълника се запълва. В противен случай остава празно незапълнено пространство.

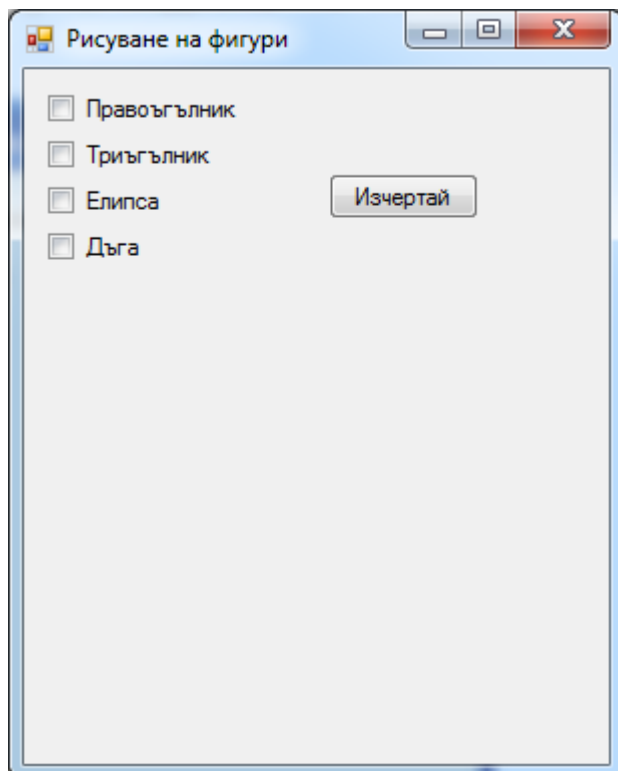


В повечето случаи когато начинът на запълване е със стойност 1, се запълват всички затворени области. Това обаче не е чак толкова просто и съществуват изключения. За да определите дали дадена затворена област ще бъде запълнена, представете си линия, начертана от точка в тази област до безкрайност. Ако въображаемата линия пресича нечетен брой гранични линии, областта се запълва както при зададена стойност 0. Ако въображаемата линия пресича четен брой гранични линии, областта може да бъде запълнена, или незапълнена. Областта се запълва, ако броят на граничните линии, отиващи в едната посока (относно въображаемата линия), не е равен на броя на граничните линии, отиващи в другата посока.

## 14 Пробвайте 01 Рисуване на фигури

Ето, че стана време да приложите наученото до тук.

Създайте форма подобна на показаната по-долу:




Задайте височина и ширина на формата съответно 400 и 300. Използвайте полетата за отметка, за да определите каква фигура ще се изрисува.

Преди всяко изчертаване възстановявайте фона на формата. За целта можете да използвате следния ред с код:

```
g.Clear(Drawing.Color.FromName("Control"))
```

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


Пробвайте


### Рисуване на фигури

Вашата задача е да създадете програма, която изчертава елипса, правоъгълник, триъгълник и дъга. Задължително условие е да сменяте стила, дебелината и цвета на използваната писалка.

**Файлът**  
14\_challenge\_files  
съдържа инструкции и необходимите за изпълнение на задачата файлове.

Натиснете бутона "Изтегли", за да получите файла. След това щракнете двукратно върху него, за да разархивирате инструкциите и помощните файлове.

ПОМОЩ


ИЗТЕГЛИ



Ако програмата работи правилно, я покажете на вашия учител.

### **Продължение / Обобщение**

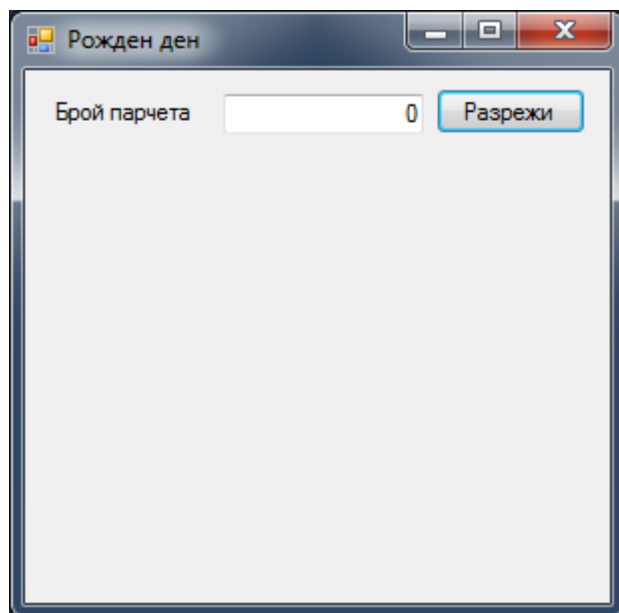
Разширете програмата като осигурите възможност за избор на цвят и стил на писалката от падащо меню.

## 14 Пробвайте 02 Рожден ден

Знаете ли, че Ники има скоро Рожден ден? Замислили сме да я изненадаме с огромна торта. Разбрахме, че ще покани много приятели. А защо не дойдете и вие!

Несъмнено щом има Рожден ден, ще има и торта. Но как ще я разделим на толкова много парчета. Защо да не напишем програма, която да дели торта на толкова равни парчета, колкото са необходими за всички гости. Да започваме работа!

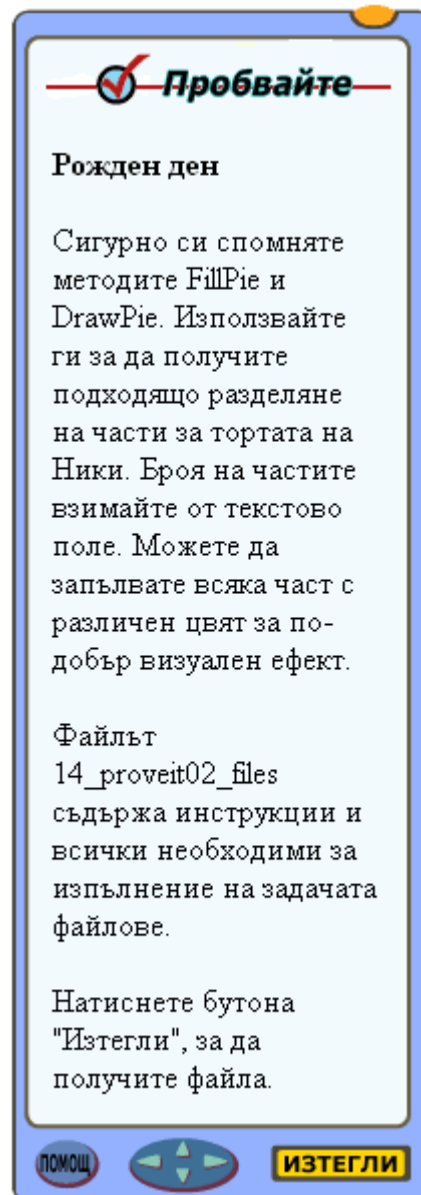
Вашата форма няма да е сложна. Тя може да изглежда така:



Декларирайте три променливи `StartPoint`, `EndPoint` и `Pie` от тип `Integer`. Инициализирайте `StartPoint` със стойност 0, `EndPoint` със стойността на `Pie`, а `Pie` със стъпката, която ще определи размера на частите от тортата:

```
Pie = CInt(360 / Val(TextBox1.Text))
```

Използвайте два цикъла `for` – единия за да изчертаете контурите на парчетата, а втория, за да ги запълните. Не забравяйте, че извикването на метода `DrawPie` трябва да предхожда извикването на метода `FillPie`.





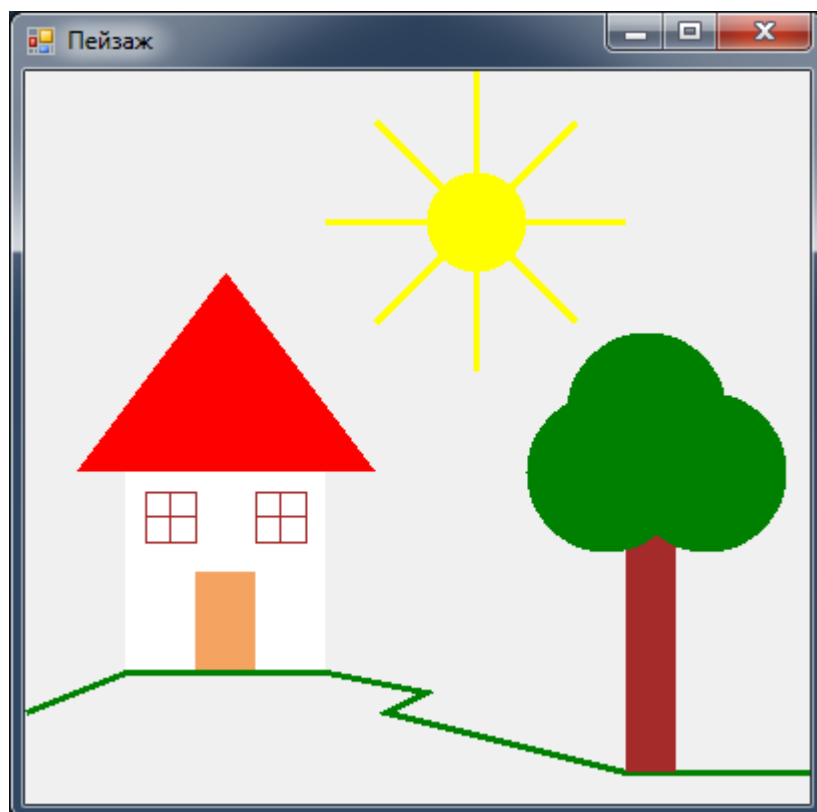
Изградете проекта и ако няма грешки изберете опцията Start Debugging от менюто Debug, за да стартирате програмата.  
Ако програмата работи правилно, я покажете на вашия учител.

### **Продължение / Обобщение**

Спомнете си как се създава запълване с градиент. Опитайте се да запълните частите на тортата с различни преливащи се цветове.

## 14 Можете ли?    Пейзаж



Ди Джей е истински художник, но дали се справя толкова добре, когато рисува върху формата, както когато рисува върху платно с боите си? А вие? Сега ще проверим!



Ако се затруднявате да си представите координатите на фигурите, начертайте предварително върху лист с координатна система отделните компоненти на картината. Изчислете местоположението и големината на всяка една от тях.

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.


**Можете ли?**


**Пейзаж**

Вашата задача е да изрисувате картина подобна на тази във файла с инструкциите

Файлът  
14\_challenge\_files  
съдържа инструкции и необходимите за изпълнение на задачата файлове.

Натиснете бутона "Изтегли", за да получите файла. След това щракнете двукратно върху него, за да разархивирате инструкциите и помощните файлове.

ПОМОЩ
▶ ◀ ◂ ◃
ИЗТЕГЛИ



## Проверка на знанията

НАПРАВЕТЕ ТЕСТА ОТНОВО

- 1** С кой метод можете да изчертаете правоъгълник?

  - ☐ A. DrawRectangle
  - ☐ B. DrawLine
  - ☐ C. И двата
- 2** Какви аргументи приема методът DrawPolygon?

  - ☐ A. x и y
  - ☐ B. Pen и Points
  - ☐ C. x1, y1, x2 и y2
- 3** Какво изчертава методът DrawArc?

  - ☐ A. Елипса
  - ☐ B. Дъга
  - ☐ C. Пай
- 4** Какво е предназначението на метода FillEllipse?

  - ☐ A. Запълва полигон с определена писалка.
  - ☐ B. Изчертава контура на елипса.
  - ☐ C. Запълва елипса с определена четка