



Сложен If

Катманду, Непал

Клиф свърши чудесна работа като ви запозна с оператора If...Then. Сега вече вашите програми могат да взимат решения! Те почти могат да мислят! Клиф ви показва как да използвате операторите за сравнение и оператора за условен преход, и как да работите с Булевата логика във Visual Basic. Той представи няколко примера, които демонстрират как с оператора "If...Then" се реализира множествен избор и как с Булеви оператори се структурират сложни условия.



Клиф ще кара до нашата следваща дестинация, която е Катманду, Непал. Катманду е разположен в плодородна долина, намираща се в централната част на Непал. Гледката е впечатляваща! Ако имаме достатъчно бензин, пътуването ще бъде прекрасно! Независимо от това, ще ви покажа няколко трика, които правят оператора "If...Then" още по-полезен. В края на пътуването ще можете да накарате програмите си да взимат всякакви решения!



Разширен If...Then оператор

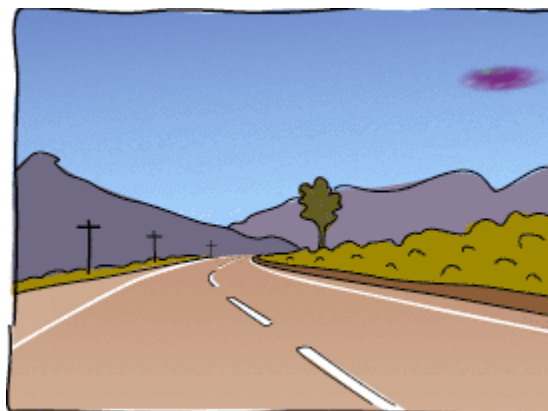
Знаете как да конструирате If...Then изрази във Visual Basic. С един такъв израз не можете да направите особено много. Ако обаче използвате няколко програмата ви ще може да реализира система от решения. Тя би могла да отговаря на въпроси с повече от два възможни отговора! Клиф ви показва как да обединявате няколко изрази в условието на оператора If...Then. Ако условието е True, всички изрази след него ще се изпълнят последователно. Това опростява кода ви и го прави лесен за четене и дебъгване.

Като използвате оператора If...Then можете да напишете код, който кара програмата ви да взима решение без значение колко сложно е то. Подобен код обаче е доста объркващ и притежава повторения. Ще ви науча на няколко трика, които ще улеснят живота ви и ще осигурят на оператора If...Then по-голяма функционалност. Първо ще ви покажа как да влагате If...Then операторите един в друг. След това ще ви науча как да използвате оператора If...Then...Else. Накрая ще ви покажа някои разлики между тези оператор във Visual Basic от една страна и C# от друга. Готови ли сте? Да вървим!



Вложени If...Then оператори

За да запази нещата по-прости, Клиф ни повери малка тайна за оператора If...Then. В един If...Then оператор можете да вмъкнете друг If...Then оператор! Помислете! Вторият If...Then оператор ще се изпълни само, ако условието от първият оператор е True. Първият оператор включва втория оператор. Това се нарича "влагане" на оператори, защото единият оператор обхваща другият оператор. Нека да разгледаме няколко примера.



Ако имам \$200, и ако магазинът е отворен, ще си купя MP3 плеър.

Реализирано с псевдокод това ще изглежда по следния начин:

```
If имам $200 Then
    If магазинът е отворен
    Then
        Ще си купя MP3 плеър
    End If
End If
```

Полезен съвет

Отместванията и подравняванията са от съществено значение при предотвратяване на грешки във

Два оператора If...Then са обединени в един! Първият If...Then оператор обхваща втория. Той има едно условие (If имам \$200) и израз, който ще се изпълни (If магазинът е отворен then ще си купя MP3 плеър) и всъщност представлява втория If...Then оператор. Вторият If...Then оператор има свое собствено условие (If магазинът е отворен) и свой собствен израз, който ще се изпълни (ще си купя MP3 плеър). Както виждате, за да си купя плеър, трябва да имам \$200 и магазинът трябва да е отворен. И двете условия е необходимо да са True. Ако нямам \$200, условието в първия If...Then оператор ще бъде False и независимо дали магазинът е отворен или не, няма да си купя MP3 плеър!

Полезен съвет

Някои ученици ще забележат, че примерът по-горе може да се напише със съставен If оператор: If I have \$200 and the store is open. Влагането на оператори е подходящо, когато се използват серия от If оператори или когато се използва оператор If с клауза Else.



Сега нека да разгледаме малко повече код. Създайте ново Windows приложение с име Nesting. Добавете върху формата две полета за отметка и един бутон. Задайте стойност "Who's Checked?" за свойството Text на бутона. Създайте обработчик на събитието Button1_Click. Добавете следния код в него:



```
If CheckBox1.Checked = True Then
    If CheckBox2.Checked = True Then
        MessageBox.Show("All are checked.")
    End If
End If
```

Изградете и стартирайте приложението. Маркирайте първото поле за отметка и натиснете бутона "Who's Checked?". Какво се случва? Нищо. Маркирайте второто поле за отметка и натиснете отново бутона "Who's Checked?". Този път се появява диалогов прозорец със съобщение "Both are checked.". Както виждате първият ред от кода е условие:

```
If CheckBox1.Checked = True Then
```

След него има израз, който е друг If...Then оператор, вложен в първия. Вложеният If...Then оператор има свое собствено условие:

```
If CheckBox2.Checked = True Then
```

След него също следва израз за изпълнение:

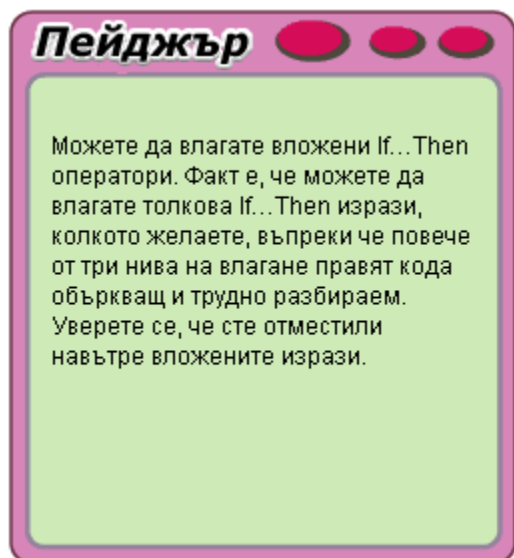
```
MessageBox.Show("All are checked.")
```



И така, какво се случва? Ако първото условие (`CheckBox1.Checked = True`) е `True`, изразът след него се изпълнява. Изразът обаче притежава друг `If...Then` оператор. Ако второто условие (`CheckBox2.Checked = True`) е `True`, изразът след него се изпълнява и диалоговият прозорец се визуализира. Следователно, за да се визуализира диалоговият прозорец, трябва и двете условия да са `True`. Това означава, че и двете полета за отметка трябва да са маркирани.



Забележете, че в кода по-горе вторият If...Then оператор се включва в първия. Ключовите думи End If на първия оператор са след ключовите думи End If на втория оператор, т.е. вторият оператор е вложен в първия. Влагайки множество If...Then оператори можете да изпълните даден израз само, ако всички условия на операторите са True.





Противоположни условия

Условията, които използвахме до тук, бяха малко еднотипни. Ако са True, то изразът след тях се изпълнява. Ако са False, то нищо не се случва. Какво ще стане, ако искаме нещо да се случи, когато условието е False? Какво ще стане, ако искаме да изпълним някакъв друг код?

Една от възможностите е да използваме друг If...Then оператор с противоположно условие и свой собствен израз за изпълнение. Да напишем малко код. Създайте ново Windows приложение с име IfThenOtherwise. Върху формата добавете бутон и поле за отметка. Задайте стойност "IfThen" за свойството Text на бутона. Поставете следния код в обработчика на събитието Button1_Click:

```
If CheckBox1.Checked = True Then
    Form.ActiveForm.BackColor = System.Drawing.Color.Red
End If

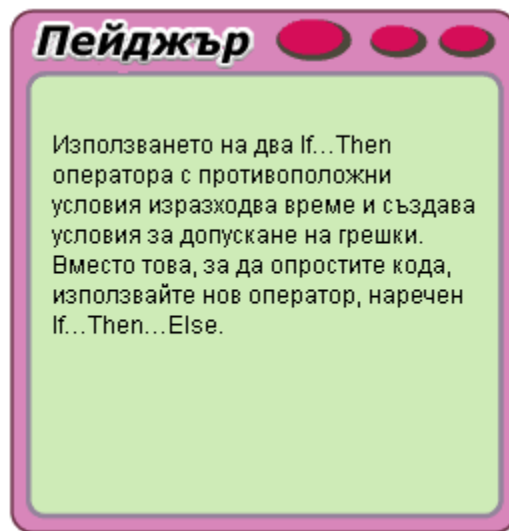
If CheckBox1.Checked = False Then
    Form.ActiveForm.BackColor = System.Drawing.Color.Blue
End If
```

Изградете и стартирайте проекта. Натиснете бутона "If Then". Формата става синя. Маркирайте първото поле за отметка и отново натиснете бутона "IfThen". Формата става червена.

Как работи този код? В първия If...Then оператор се проверява условието CheckBox1.Checked=True. Ако то е изпълнено, цвета на формата става червен. Във втория If...Then оператор се проверява

CheckBox1.Checked=False (противоположно условие). Ако то е изпълнено, цвета на формата става син.

Въпреки повторенията кодът работи! Недостатък в случая е, че вие пишете идентични блокове с код. Това изисква време и създава условия за допускане на грешки.





If...Then...Else

Visual Basic предлага алтернативен начин за отстраняване на повторемостта в кода, което ще направи програмите ви по-ясни и по-лесни за четене. Такъв е операторът **If...Then...Else statement**. Той изисква едно условие, но осигурява два отделни израза за всеки резултат от изпълнението на условието (True and False). Операторът **If...Then...Else** се използва вместо два **If...Then** оператора с противоположни условия. Ето и синтаксиса:

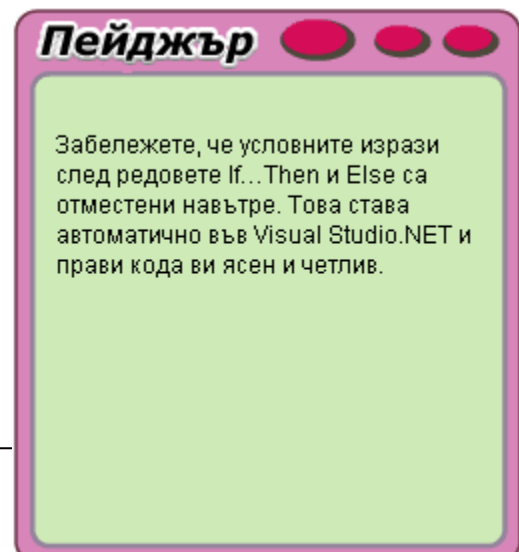
```
If condition Then  
    conditional statement 1  
Else  
    conditional statement 2  
End If
```

Как работи този код? Първо се изпълнява условието. Ако резултатът е True, се изпълнява първият израз (conditional statement 1). В противен случай (else) се изпълнява вторият израз (conditional statement 2).

Нека да модифицираме приложението **IfThenOtherwise** като използваме оператора **If...Then...Else**, за да докажем че работи като два **If...Then** оператора с противоположни условия. Добавете втори бутон. Задайте стойност "IfThenElse" за свойството **Text**. В обработчика на събитието **Button2_Click** добавете следния код:

```
If CheckBox1.Checked = True Then  
    Form.ActiveForm.BackColor = System.Drawing.Color.Red  
Else  
    Form.ActiveForm.BackColor = System.Drawing.Color.Blue  
End If
```

Изградете и стартирайте проекта. Натиснете бутона "IfThenElse". Формата става синя. Маркирайте полето за отметка и отново натиснете бутона "IfThenElse". Формата става червена. Програмата работи по същия начин! Кодът проверява условието **CheckBox1.Checked=True**. Ако то е True, се изпълнява първият израз и формата става червена. В противен случай (else) се изпълнява вторият израз и формата става синя.





Да разгледаме друг пример. Ще променим програмата IfThenOtherwise като добавим няколко реда код в оператора If...Then...Else. След ключовите думи Then и Else на оператора могат да се поставят повече от един израз за изпълнение.

Отворете приложението IfThenOtherwise. Премахнете кода от обработчика на събитието Button2_Click и добавете следния:

```
If CheckBox1.Checked = True Then
    Form.ActiveForm.BackColor = System.Drawing.Color.Red
    MessageBox.Show("I'm red.")
    TextBox1.Text = "I'm red."
Else
    Form.ActiveForm.BackColor = System.Drawing.Color.Blue
    MessageBox.Show("I'm blue.")
    TextBox1.Text = "I'm blue."
End If
```

Изградете и стартирайте проекта. Натиснете бутона "IfThenElse". Случват се три неща. Маркирайте полето за отметка и отново натиснете бутона "IfThenElse". Случват се други три неща.

Полезен съвет

Поощрявайте учениците действително да изпълняват кода стъпка по стъпка, а не само да се плъзгат по повърхността прочитайки го.



Постъпково изпълнение на оператора If

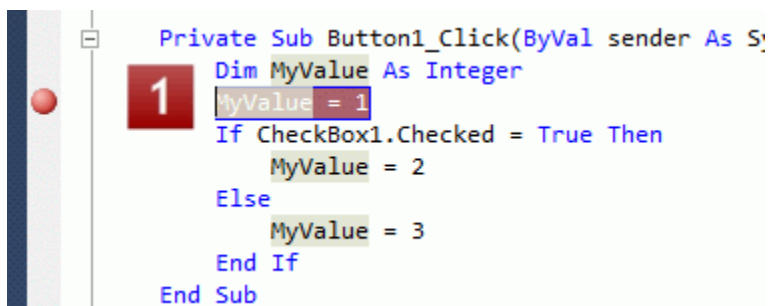
Нека да използваме възможностите за дебъгване във Visual Studio, за да изпълним оператора If...Then...Else стъпка по стъпка. Така ще можете да видите какво действително се случва при изпълнението на кода.

Създайте ново Windows приложение с име StepInIf. Върху формата добавете бутон и поле за отметка. Задайте стойност True за свойството Checked на полето за отметка. Така когато стартирате приложението, полето ще бъде маркирано.

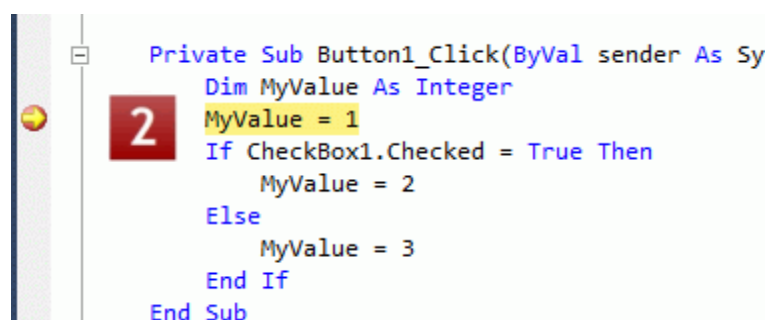
Щракнете двукратно върху бутона, за да създадете обработчик на събитието Button1_Click. Добавете следния код:

```
Dim MyValue As Integer
MyValue = 1
If CheckBox1.Checked = True Then
    MyValue = 2
Else
    MyValue = 3
End If
```

1. Добавете точка за спиране на изпълнението на кода във втория ред (MyValue = 1) като щракнете върху сивата област вляво или щракнете с десния бутон на мишката върху реда и от появилото се контекстно меню изберете командата Breakpoint и след това Insert Breakpoint.



2. Изградете и стартирайте проекта. Когато формата се визуализира, натиснете бутона. Кодът се изпълнява до точката на спиране.





3. Натиснете клавиша F11, за да изпълните текущия ред (MyValue=1). Следващият ред се маркира с жълто.

```
Private Sub Button1_Click(ByVal sender As Sys
Dim MyValue As Integer
3 MyValue = MyValue + 1
If CheckBox1.Checked = True Then
    MyValue = 2
Else
    MyValue = 3
End If
End Sub
```

4. Натиснете клавиша F11, за да изпълните текущия ред (If CheckBox1.Checked = True Then). Тъй като полето за отметка е маркирано, следващият ред се маркира с жълто MyValue=2.

```
Private Sub Button1_Click(ByVal sender As Sys
Dim MyValue As Integer
4 MyValue = 1
If CheckBox1.Checked = True Then
    MyValue = 2
Else
    MyValue = 3
End If
End Sub
```

5. Натиснете клавиша F11, за да изпълните текущия ред (MyValue=2). Следващият ред се маркира с жълто.

```
Private Sub Button1_Click(ByVal sender As Sy
Dim MyValue As Integer
5 MyValue = 1
If CheckBox1.Checked = True Then
    MyValue = 2
Else
    MyValue = 3
End If
End Sub
```

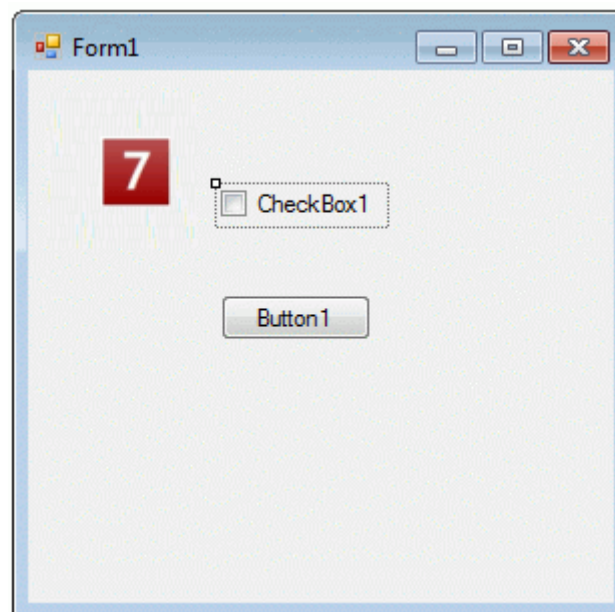
6. Натиснете клавиша F11, за да изпълните текущия

```
Private Sub Button1_Click(ByVal sender As Sys
Dim MyValue As Integer
6 MyValue = 1
If CheckBox1.Checked = True Then
    MyValue = 2
Else
    MyValue = 3
End If
End Sub
```



ред (End If). Следващият ред се маркира с жълто.

7. Натиснете клавиша F11, за да изпълните текущия ред (End Sub).
Формата се появява отново.



8. Махнете отметката от полето за отметка и натиснете бутона. Повторете описаните по-горе стъпки като натискате клавиша F11 и наблюдавайте кои редове ще се изпълнят. Този път се изпълнява клаузата Else, следвана от реда MyValue=3.

```

Private Sub Button1_Click(ByVal sender
8  Dim MyValue As Integer
    MyValue = 1
    If CheckBox1.Checked = True Then
        MyValue = 2
    Else
        MyValue = 3
    End If
End Sub

```

Сега вече наистина видяхте как работи оператора If...Then...Else!



А сега да приложим това, което сме научили.

09 Пробвайте 01 Резултати от игра на голф

Забавлявахме ли се, когато играехме голф тази сутрин? Добре, вие се забавлявахте, а какъв беше моят резултат? Нека да пресметнем резултата.

Създайте форма подобна на показаната по-долу:

Изчислете средния резултат от играта при натискането на бутона "Изчисли". Тъй като съдържанието на текстовите полета е от тип string, а не integer, вие трябва да го конвертирате. Visual Basic осигурява няколко начина за това. Ето един от тях.

X = Val (TextBox1.Text)

Функцията Val преобразува символни стойности, след което вие можете да ги присвоите на променливи като X.

За всеки играч:

- Ако резултатът му е по-голям от средния, визуализирайте съобщение като например "Отлично" в съответния етикет, в противен случай визуализирайте съобщение – "Тренирай още".

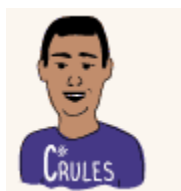
Текстовите полета трябва да бъдат празни, когато стартирате програмата.



Ако програмата работи правилно, я покажете на вашия учител.



Операторът If в C#



Покажах ви синтаксиса и няколко примера за операторите If...Then и If...Then...Else във Visual Basic. Сега ще накарам Дърк и Джин да ви научат как да използвате тези оператори в C#.

Полезен съвет

Напредналите ученици могат да приключат програмата на Visual Basic по-бързо и да я напишат на C#.

C# има оператори, които работят както операторите If...Then и If...Then...Else във Visual Basic. Те се използват по същия начин. Синтаксисът им обаче е малко по-различен. Нека първо да ви покажа кратък пример на C#, след което ще ви опиша особеностите в него.

```
int MyAge;  
MyAge=3;  
if (MyAge==3)  
{  
    MessageBox.Show("I am 3.");  
    MessageBox.Show("I am still 3.");  
}
```

На първия ред е декларирана променлива с име My Age. На вторият ред тя е инициализирана със стойност 3. След това започва оператора "if". Забележете, че той се пише с малки букви. След него следва условието (MyAge==3). Условието е поставено в скоби. Сигурно се чудите каква е ролята на двойния знак равно. В C# този символ се използва като оператор за сравнение вместо единичния знак равно във Visual Basic. Изразите които се изпълняват след условието са обградени с фигурни скоби – { и }. Някои програмисти използват фигурните скоби, за да направят кода по-четлив. Забележете, че всеки един от изразите завършва с точка и запетая. Също така ключовата дума "Then" липсва! А сега да разгледаме оператора "If...Then...Else" в C#. Потърсете разликите и приликите между съответстващите си оператори в C# и Visual Basic.

```
int MyAge;  
MyAge=3;  
if (MyAge==2)  
{  
    MessageBox.Show("I am 2.");  
    MessageBox.Show("I am still 2.");  
}  
else  
{
```




```
    MessageBox.Show("I am 3.");  
    MessageBox.Show("I am still 3.");  
}
```

Отново се забелязва, че изразите за изпълнение се обграждат с фигурни скоби. Виждат се две групи от изрази – една след условието на оператора if, и друга след клаузата else. Всяка група е поставена между фигурни скоби. Всеки един от изразите завършва с точка и запетая. Също така ключовата дума "Then" не се използва.

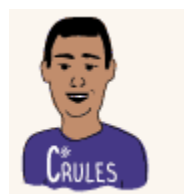


Оператори в C#

Има още някои различия при конструиране на изрази с операторите "If...Then" и "If...Then...Else" във Visual Basic от една страна и C# от друга. Първо, Булевите оператори използват различни символи. Вместо And, Or, и Not, C# използва &&, ||, и ! Разгледайте таблицата по-долу, за да установите разликите.

Булев оператор	Visual Basic	C#
AND	And	&&
OR	Or	
NOT	Not	!

Ето и пример, който използва Булевите оператори AND (&&) и NOT (!) в C#.



```
bool isBilled=true;
bool isLate=false;
bool isDone;
isDone = isBilled && !isLate;
MessageBox.Show(isDone.ToString());
```

Забележете, че резервираните думи bool (Булев тип), true, и са с малки букви. Променливите isBilled, isLate и isDone са декларирани от Булев тип. В C# вместо оператора AND се използва оператора &&, а вместо оператора NOT – оператора !. Тъй като C# не поддържа автоматично преобразуване в символен низ, се използва метода ToString.



Съществуват различия и между операторите за сравнение. Вече видяхте, че представяне на равенство се използва знака == вместо знака =. За представяне на неравенство в C# се използва знака != вместо знака <>. Таблицата по-долу описва изброените различия.

Оператор сравнение	за	Visual Basic	C#
Равно		=	==
По-голямо		>	>
По-малко		<	<
По-голямо или равно		>=	>=
По-малко или равно		<=	<=
Различно		<>	!=

Разгледайте кода на C#, който използва оператора за неравенство.

```
int MyAge;
MyAge=2;
if (MyAge!=3)
{
    MessageBox.Show("I am not 3.");
}
```



Както виждате кодът визуализира съобщение "I am not 3."



Получихме доста разяснения. Виждам доста разлики в начина на използване на операторите "If...Then" и "If...Then...Else" във Visual Basic 2010 и C#. Не може да се отрече и наличието на прилики! Тези оператори правят едно и също нещо по един и същ начин, но имат различен синтаксис.



Пристигане в Катманду, Непал

Ето ни в Катманду, Непал! Научихте ли доста по време на пътуването? Сега вече знаете как да използвате доста програмни техники. Можете да пишете вложени If...Then оператори и да конструирате изрази с оператора If...Then...Else. Вашият код вече наистина би могъл да взема решения!

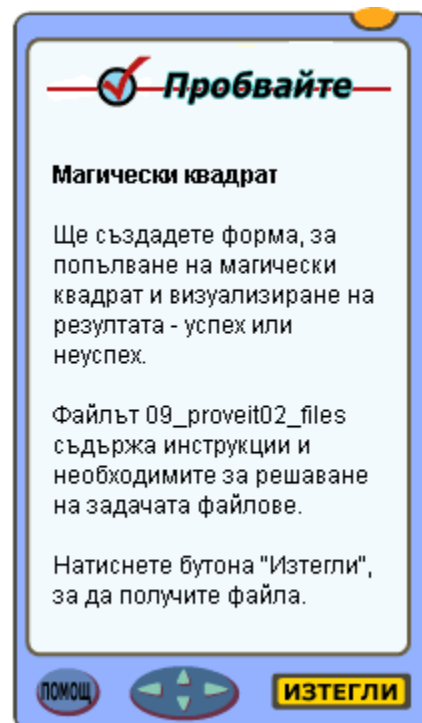
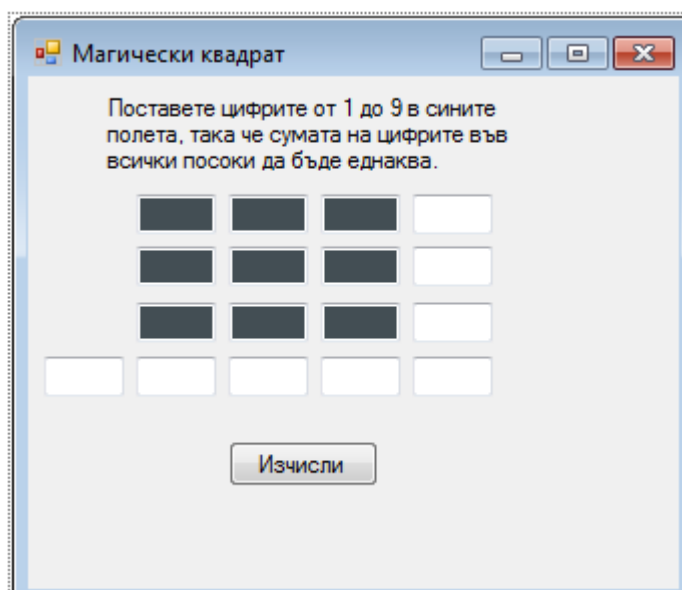
Добре, време е за малко практика. Ето ви една математическа задача.



09 Пробвайте 02 Магически квадрат

Аз не съм професионален фокусник като този, който видяхме в Увеселителния парк на Катманду, но знам един магически трик. Мога да подреда цифрите от 1 до 9 в решетка от 3 реда и 3 колони така че сумата им във всички посоки да бъде една и съща. Можете ли да напишете подобна програма?

Създайте форма подобна на показаната по-долу:



При натискане на бутона "Изчисли" се изчислява сумата на числата по трите реда, трите колони и двата диагонала, като резултатът се визуализира в текстовите полета номерирани от 10 до 17 (от най-долния ляв до най-горния десен).

Тъй като съдържанието на текстовите полета е от тип string, а не integer, вие трябва да го конвертирате. Visual Basic осигурява няколко начина за това. Ето един от тях.

```
X = Val (TextBox1.Text)
```

Функцията Val преобразува символни стойности, след което вие можете да ги присвоите на променливи или да ги използвате в конструкции If.



Всяка сума трябва да е равна на 15. Сумата на всички девет текстови полета трябва да е 45.

Етикетът с име Label2 в долната част на формата може да бъде използван за визуализиране на успеваемостта, за да поощри играча да опита отново или за да отбележи дублирането на цифри.

Ако програмата работи правилно, я покажете на своя учител.

Полезен съвет

Ако сумата на всички 9 полета не е 45, то вероятно е дублирано число. Учениците също трябва да проверят за числа, които са по-големи от 9 или по-малки

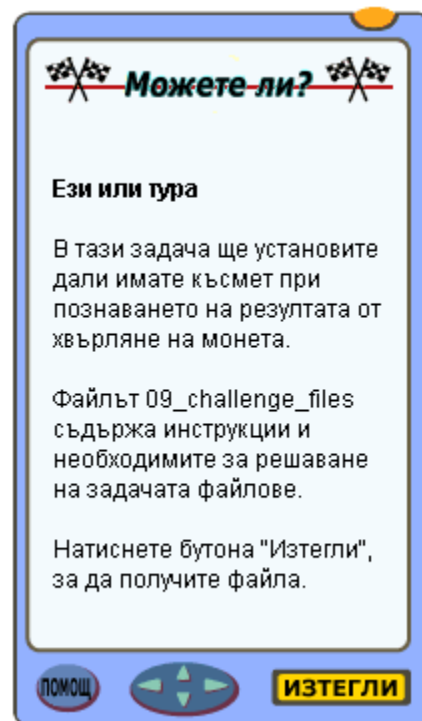


Готови ли сте за истинско предизвикателство?

09 Можете ли? Ези или тура

Всеки път, когато между нас има несъгласие, като например кой да изхвърли боклука, теглим чоп като използваме монета. Нека да накараме компютъра да направи хвърляне на монетата. Ние ще трябва да предположим дали резултатът от хвърлянето ще бъде ези или тура.

Създайте форма подобна на показаната по-долу:



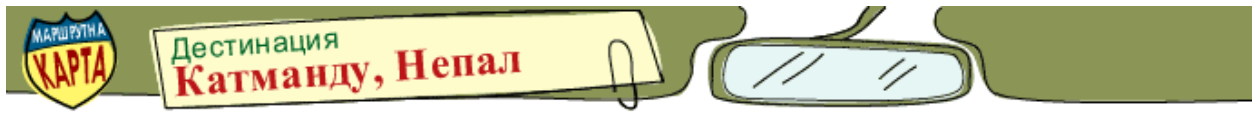
Когато съответния радио-бутон е натиснат променете стойността на контрола TextBox1 на "Ези" или "Тура".

При натискането на бутона "Играй", генерирайте случайно число като използвате следния код:

```
Dim MyRandomGenerator As System.Random
MyRandomGenerator = New System.Random
Dim RanNum As Integer
```

```
RanNum = MyRandomGenerator.Next(0, 2)
```

- Ако случайното число е 0, установете текста в компонента TextBox2 на "Ези", в противен случай – на "Тура".
- Ако текстът в компонента TextBox1 е еквивалентен на текста в компонента TextBox2, добавете 1 към стойността, отчитаща победите, в противен случай извадете 1.



Визуализирайте броя на победите и загубите в подходящи текстови полета.

Забележка: В секция 7 научихте, че не можете да смесвате числови и символни стойности. Текстовите свойства винаги са от тип String. Visual Basic ви позволява директно да въвеждате числови стойности в текстовите полета, но възниква проблем при тяхното прочитане. Дефинирайте две променливи от тип Integer за победите и загубите. Използвайте ги, за да преизчислявате резултата, след което визуализирайте стойностите им в текстовите полета.

Ако програмата работи правилно, я покажете на учителя си.

Продължение / Обобщение

Добавете полета, за да покажете процента на победите и процента на загубите.



Проверка на знанията

НАПРАВЕТЕ ТЕСТА ОТНОВО

- | | |
|--|--|
| <p>1 Вложеният If оператор?</p> <ul style="list-style-type: none"><input type="radio"/> A. Не се използва<input type="radio"/> B. Поставя се между фигурни скоби<input type="radio"/> C. Отмества се, за да бъде четлив | <p>3 Когато If изразът е false, кой код се изпълнява?</p> <ul style="list-style-type: none"><input type="radio"/> A. Кодът след Then<input type="radio"/> B. Кодът след Else<input type="radio"/> C. Кодът след целия if блок |
| <p>2 Когато един If израз е вложен в друг, колко са съществуващите End If изрази?</p> <ul style="list-style-type: none"><input type="radio"/> A. 0<input type="radio"/> B. 1<input type="radio"/> C. 2 | <p>4 Колко редове могат да се поставят след изрази Else ?</p> <ul style="list-style-type: none"><input type="radio"/> A. 0<input type="radio"/> B. 1<input type="radio"/> C. Колкото е необходимо |

