

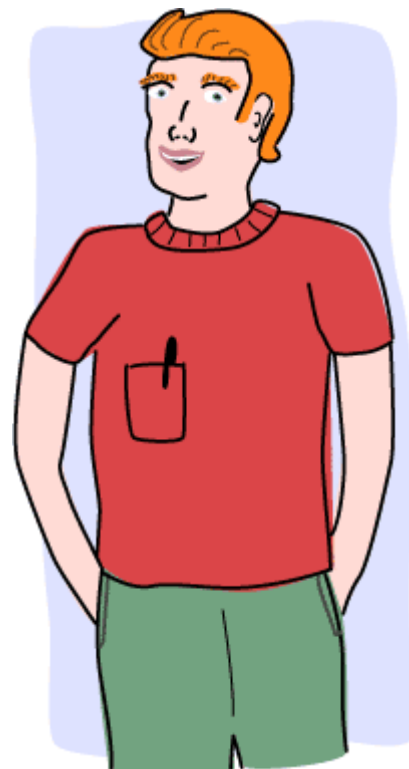




Операторът If

Киев, Украйна

По време на пътуването ни Джин ви показва как да използвате оператори за създаване на математически равенства. Не съм много добър в математиката, но добре се справя с логиката. Логиката е начин за взимане на решения. Ние взимаме решения всеки ден през живота си. Някои от тях са доста добри. Например, реших да участвам в радио игра и спечелих три билета за концерта на групата "Whatever". Взимал съм и недобри решения – сложих джинсите си в пералнята преди да съм извадил билетите от джоба Е, "Whatever" винаги ще бъдат някъде наоколо – ще ги видя следващата година.



Пътуваме за Киев, Украйна. Киев е красив град с много паркове и исторически постройките. Той е разположен на възвишенията около река Днепър. Сигурно си мислите, че няма какво да правите там, но грешите. Съществуват милион възможности. По пътя ще ви покажа как компютърните програми взимат решения с оператора "If...Then".

Мисля си – какво ще стане, ако направим грешен ход по пътя за? Никога няма да стигнем там!



Взимане на решения

Взимам решения непрекъснато. Какъв цвят чорапи да си обуя? Къде да си направя следващата татуировка? Дали да изям цяла много голяма пица с аншоа? Сигурен съм, че вие взимате и доста важни решения!

В действителност хората са късметлии. Взимаме решение при наличие на множество възможности. Като например сладолед с какъв вкус да си купим, каква музика да слушаме или с кого да се закачим. Всъщност хората са късметлии. Компютрите нямат такъв късмет. Те вземат решение въз основа на това дали нещо е истина или лъжа (True или False). True или False е резултатът от изчислението на израза?

Например, представете си, че сте в ресторант и искате да си поръчвате хамбургер, пържени картопки и бира, а сервитьорът ви задава въпроси, на които можете да отговаряте само с да и не!

Желаете ли нещо за пиене? Да.

Желаете ли кафе? Не.

Желаете ли чай? Не.

Желаете ли сода? Не.

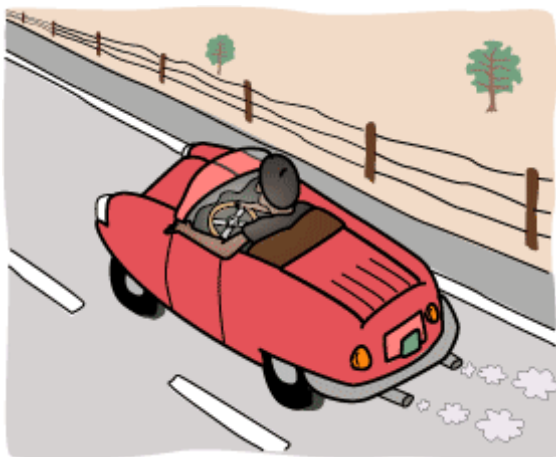
Желаете ли кола? Не.

Желаете ли бира? Да!

Е, до тук отговорих на шест въпроса, а още не съм си поръчал хамбургера и пържените картопки! Може би бавно, но в крайна сметка ще направя поръчката си!

По подобен начин компютърните програми взимат решения – само с да и не. Този начин на “мислене” се нарича Булева логика.

Булева логика



Какво видяхте, компютърните програми се справят чудесно с решаването на математически проблеми. Те също така са добри и в решаването на проблеми от Булевата логика. Отговорът на даден проблем в Булевата логика е винаги истина или лъжа – не съществуват варианти като “подобно на”, “почти като” или “близо до”. Ще ви покажа няколко примера от Булевата логика.

4=4

Is 4=4? Да, това е истина.

3=1+3

Is 3=1+3? Не, това е лъжа.

3=(6+12)/(1+5)

Is 3=(6+12)/(1+5)? Да, това е истина.

Можете да видите, че решението на всеки един от логическите проблеми по-горе е истина или лъжа. Отговорите не са числа, цветове или думи. Всъщност така компютрите взимат решения. Ако отговорът е истина, програмата прави нещо, а ако е лъжа – не прави нищо.



Булеви оператори

Подобно на математиката Булевата логика има свои оператори. Те не работят с числа, а с булеви операнди -- True или False. "Събирането" на два изрази True и False става по специален начин. Булевите изрази могат да се сливат, за да се решат по-сложни булеви проблеми.

Съществуват три булеви оператора: AND, OR и NOT. Операторите AND и OR действат върху два операнда. Операторът NOT действа върху един операнд. И трите оператора дават булев резултат. Всеки един от тях има свои собствени "правила", които определят резултата от действието им върху операндите. Да разгледаме някои булеви изрази, които използват булеви оператори. След това ще ви обясня начина, по който действат.

$$3=1+2 \text{ AND } 2=1+1$$

Резултатът е True.

Имаме два логически изрази, съединени с оператора AND. Първо се изчислява $3=1+2$, което е True. След това се изчислява $2=1+1$, което също е True. Накрая се изчислява True AND True, което според правилата на оператора AND дава отговор TRUE. Ако и двете страни на оператора AND са True, крайният резултат също е True.

$$3=1+2 \text{ AND } 5=2+2$$

Резултатът е False.

Отново имаме два отделни логически проблема, съединени с оператора AND. Първо се изчислява $3=1+2$, което е True. След това се изчислява $5=2+2$, което е False. Накрая се изчислява True AND False, което според правилата на оператора AND дава отговор False. Ако една от двете страни на оператора AND е False, крайният резултат е False.

Един последен пример,

$$3=1+2 \text{ OR } 5=2+2$$

Резултатът е True.

Първо се изчислява $3=1+2$, което е True. След това се изчислява $5=2+2$, което е False. Накрая се изчислява True OR False, което според правилата на оператора OR дава отговор True. Ако една от двете страни на оператора OR е True, крайният резултат е True.

Ще ви е необходима малко практика, но след като поработите с Булевите оператори, ще свикнете да ги използвате. За ваше улеснение, можем да



ви покажем възможните комбинации за оператора AND в таблица. Няма нужда да ги запомняте. Когато ви е необходимо, можете да се върнете и да погледнете таблицата.

Лява страна	Дясна страна	Лява страна AND Дясна страна (Краен отговор)
True	True	True
True	False	False
False	True	False
False	False	False

По същия начин, можем да ви покажем възможните комбинации за оператора OR.

Лява страна	Дясна страна	Лява страна OR Дясна страна (Краен отговор)
True	True	True
True	False	True
False	True	True
False	False	False



Операторът Not не свързва два израза. Той превръща False в True и True във False.

Дясна страна	Not Дясна страна (Краен отговор)
True	False
False	True

Ето и някои примери с оператора Not.

Not(3=1+2) Резултатът е False.

3=1+2 е True. Not True е False.

Not(5=2+2) Резултатът е True.

5=2+2 е False. Not False е True.



Оператори за сравнение

Спомняте ли си, когато Ники ви каза, че програмите много добре решават математически проблеми, но не могат да ги дефинират? Е, същото е в сила и за Булевата логика. Компютърните програми се справят доста добре със сложни логически проблеми, но не могат да ги дефинират. Отново това е работа на програмиста, а именно да дефинира логическия проблем така, че програмата да може да го реши. Когато кодът се изпълнява, програмата решава проблема. Отговорът е True или False. Програмата взима "решение" в зависимост от това дали отговорът е True или False.

Знаете, че знакът равно може да се използва като оператор за присвояване, но той също така може да се използва и като оператор за сравнение. В примерите по-горе знакът равно задава въпроса "Лявата страна равна ли е на дясната? True или False?" Съществуват и други оператори за сравнение, определящи дали лявата страна е по-голяма от дясната ($>$) или лявата страна е по-малка от дясната ($<$). Във всеки един от случаите те поставят въпрос, на който отговорът е True или False. Разгледайте примерите:

$3 < 4$

Поставеният въпрос е: 3 по-малко ли е от 4?
Да, това е истина.

$2 > 5$

Поставеният въпрос е: 2 по-голямо ли е от 5?
Не, това е лъжа.



Ето и списък с най-често използваните оператори за сравнение:

Оператор	Значение	Въпрос
=	Равно на	Лявата страна равна ли е на дясната страна?
>	По-голямо от	Лявата страна по-голяма ли е от дясната страна?
<	По-малко от	Лявата страна по-малка ли е от дясната страна?
>=	По-голямо или равно на	Лявата страна по-голяма ли е или равна на дясната страна?
<=	По-малко или равно на	Лявата страна по-малка ли е или равна на дясната страна?
<>	Различно от	Лявата страна различна ли е от дясната страна?

Може би операторите $>=$, $<=$ и $<>$ са ви чужди. Те обединяват два оператора в един. Те включват два булеви изрази, свързани с оператора OR. Ако един от двата изрази е True, отговорът е True. Например, $>=$ значи: Лявата страна по-голяма ли е от дясната страна? OR лявата страна равна ли е на дясната страна? Имаме два отделни въпроса свързани с оператора OR. Операторът $<>$ значи: Лявата страна по-малка ли е от дясната страна? OR лявата страна по-голяма ли е от дясната страна? Операторът $<>$ всъщност пита: Различни ли са двете страни?



Нека да разгледаме няколко примера, използващи оператори за сравнение. Можете да се върнете назад и да погледнете таблиците, за да проверите как работи оператора OR във всеки един от случаите.

$3 <= 3$

$3 < 3$ е False. $3 = 3$ е True. False OR True = True
True, 3 е по-малко или равно на 3.

$2 >= 3$

$2 > 3$ е False. $2 = 3$ е False. False OR False = False



False, 2 не е по-голямо или равно на 3.

$3 < > 4$

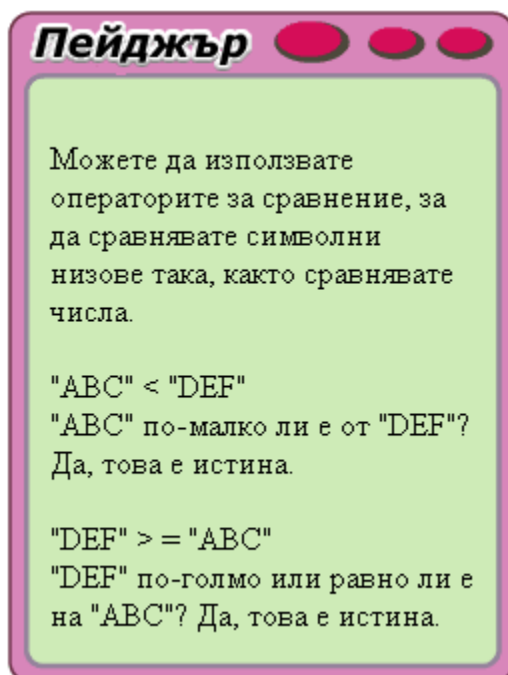
$3 < 4$ е True. $3 > 4$ е False. True OR False = True

True, 3 е различно 4.

$3 < > 3$

$3 < 3$ е False. $3 > 3$ е False. False OR False = False

False, 3 не е различно от 3. 3 е равно на 3



Примерите, които ви дадох са с числа. Операторите за сравнение могат да се използват и при символни низове и други булеви стойности. Разгледайте примерите:

`"ABC" = "DEF"`

"ABC" равно ли е на "DEF"? Не, това е лъжа.

True=True

True равни ли е на True? Да, това е истина.

True=False

True равно ли е на False? Не, това е False.

Полезен съвет

Сравняването на низове става отляво надясно. Първият символ на операнд 1 се сравнява с първия символ на операнд 2, след това втория с втория и т.н. Сравнението е чувствително към малки и големи букви.



Условен код



Сега можете да накарате програмата си да задава въпроси като използва операторите за сравнение и Булевата логика. Какво обаче ще прави с отговорите? Ще взема решение! В зависимост от това дали отговорът е True или False, програмата може да прави избор. В действителност тя няма реален избор. Ако отговорът е True, ще направи едно нещо, а ако отговорът е False – друго нещо. Ето и някои примери от всекидневието.

Ако измия колата на майка ми преди баща ми да се е върнал вкъщи, то ще получа \$5.
Татко е вкъщи. Измих ли колата? True.
Получавам \$5.

Татко е вкъщи. Измих ли колата? False. Не получавам \$5.

Ако Полийн мисли, че съм ѝ симпатичен, то ще излезе с мен.
Мисли ли Полийн, че съм симпатичен? True. Тя ще излезе с мен.
Мисли ли Полийн, че съм симпатичен? False. Тя няма да излезе с мен.

Ето няколко примера, които показват как програмата взема решения:

Ако полето за отметка "Super Size" е маркирано, то показва голяма картинка.

Полето за отметка "Super Size" маркирано ли е? True. Програмата показва голяма картинка.

Полето за отметка "Super Size" маркирано ли е? False. Програмата не показва голяма картинка.

Ако променливата TotalGallons е по-голяма от нула, то изчислява стойността на бензина.

Променливата TotalGallons по-голяма ли е от 0? True. Програмата изчислява стойността на бензина.



Променливата TotalGallons по-голяма ли е от 0? False. Програмата не изчислява стойността на бензина.

Можете да видите, че примерите от всекидневието и програмните примери имат общи неща. Те използват Булева логика, за да задават въпроси, отговорите на които са True или False. Тези въпроси използват оператори за сравнение като равно на и по-малко или равно на. Отговорите могат да бъдат True или False. Ако отговорът е True, нещо се случва. Ако отговорът е False, нищо не се случва.

Операторът If...Then

Забелязахте ли, че примерите ми използват думите "Ако" (If) и "То" (Then)? Ако дадено условие е True, то нещо ще се случи. Ако условието е False, то нищо няма да се случи. Когато искате програмата ви да взема решение, използвате оператора "If...Then" по същия начин. Във Visual Basic синтаксисът е следният:

Полезен съвет

Подсещане. Тази секция показва оператора **If ... Then**. Следващата секция показва оператора

If ... Then ... Else.

```
If condition Then
    conditional statement
End If
```

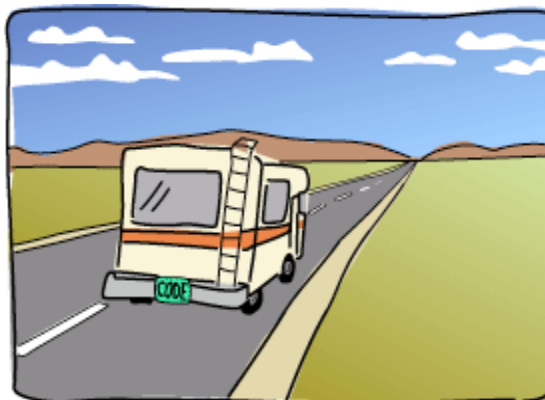
Ето и пример от Visual Basic:

```
If myAge = 3 Then
    MessageBox.Show("I am 3.")
End If
```

Обърнете внимание на думата "If". Това е ключова дума във Visual Basic, поради което е синя. Условието е поставено отдясно на ключовата дума "If". Условието извършва сравнение и задава въпрос с два възможни отговора True или False. Променливата myAge равна ли е на 3? Ако отговорът е True, нещо се прави. Забележете, че думата "Then" е вдясно от условието. Във Visual Basic това също е ключова дума, поради което е оцветена в синьо. След ключовата дума "Then" следва изразът, който трябва да се изпълни. Той е отместен навътре, за да бъде по-четлив кодът. Ако условието е True, програмата показва диалогов прозорец със съобщение "I am 3." Забележете, че думите "End If" е поставена след израза за изпълнение. Това са ключови думи, които затварят оператора "If".



А сега опитайте вие! Създайте ново Windows приложение с име IfThen. Добавете върху формата едно текстово поле. Поставете и един бутон. Щракнете двукратно върху бутона, за да създадете обработчик на събитието Button1_Click. Добавете следния код.



```
Dim myAge As Integer
myAge = 3
If myAge = 3 Then
    TextBox1.Text = "I am 3."
End If
```

Изградете и стартирайте проекта си. Натиснете бутона. Ето какво се случва. Кодът ви декларира променлива myAge и ѝ задава стойност 3. Операторът "If" задава въпрос: myAge равно ли е на 3? Отговорът е True, защото променливата myAge е инициализирана с 3. Тъй като отговорът на въпроса е True, изразът след ключовата дума "Then" ще се изпълни. В текстовото поле ще се изпише "I am 3."

Променете кода, както е показано по-долу:

```
myAge = 2
```

Изградете и стартирайте отново проекта. Натиснете бутона. Какво се случва? Нищо? Това е добре! Този път променливата myAge е инициализирана със стойност 2. Операторът "If" задава въпрос: myAge равно ли е на 3? Отговорът е False, така че изразът след ключовата дума "Then" не се изпълнява. Текстът в текстовото поле не е променен, защото резултатът от изчислението на условието е False. Програмата ви прави точно това, което се очаква.

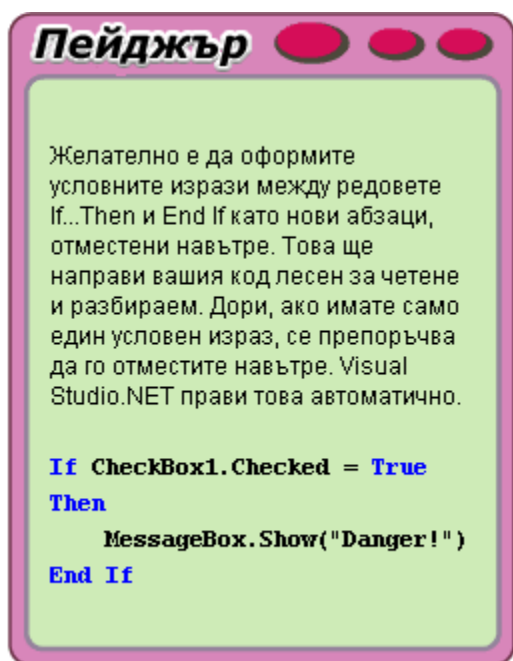


Нека да разгледаме друг пример. Добавете върху формата поле за отметка. Редактирайте обработчика на събитието Button1_Click. Заменете кода в него със следния код:

```
If CheckBox1.Checked = True Then  
    TextBox1.Text="I am checked."  
End If
```

Изградете и стартирайте проекта си. Маркирайте полето за отметка и натиснете бутона. Резултатът от изчислението на условието е True (полето за отметка е маркирано) и в текстовото поле се показва текста "I am checked."

Изградете и стартирайте отново проекта си. Този път не маркирайте полето за отметка. Какво се случва? Нищо. Условието не е изпълнено (полето за отметка не е маркирано) и текстът в текстовото поле не се променя. Програмата ви взима решение (да промени или да не промени текста в текстовото поле) въз основа на условие (дали е маркирано полето за отметка).





Повече от един оператор If

Знам, че ще искате програмата ви да взима няколко решения, когато се изпълнява. Може би ще ви е необходимо да проверявате състоянието на поле за отметка и на радио-бутон, след това да сравните две числа и накрая да установите дали потребителят е попълнил текстово поле. Това са доста проверки! Могат ли всичките да се реализират? Естествено отговорът е "Да". Можете да използвате неограничен брой "If...Then" оператори в кода си. Както всички други оператори, те се изпълняват в реда, в който са поставени в кода. Нека да създадем приложение, което използва повече от един оператор "If...Then".

Нека програмата ни притежава поле за отметка със заглавие "Display All Alarms". Ако то е маркирано, програмата ще промени цвета на формата в червен, текстът в текстово поле ще стане "Red Alert" и ще се визуализира диалогов прозорец със съобщение "Danger! Danger!" Единият от начините да реализирате това е като използвате три отделни оператора "If...Then". В условието на всеки един от тях ще се проверява дали полето за отметка е маркирано. Ако това условие е True, изразът, който следва ще се изпълнява.

Ще ви покажа как става това. Създайте ново Windows приложение с име FireAlarm. Добавете върху формата по едно поле за отметка, текстово поле и бутон. Задайте стойност "Three If's" за свойството Text на бутона. Щракнете двукратно върху него, за да създадете обработчик на събитието Button1_Click. Добавете следния код.

```
If CheckBox1.Checked = True Then
    Form.ActiveForm.BackColor =
System.Drawing.Color.Red
End If
If CheckBox1.Checked = True Then
    TextBox1.Text = "Red Alert"
End If
If CheckBox1.Checked = True Then
    MessageBox.Show("Danger! Danger!")
End If
```

Изградете и стартирайте проекта си. Маркирайте полето за отметка и натиснете бутона "Three If's". Операторите "If...Then" се изпълняват последователно, при което се изпълняват и изразите след всеки един от тях. Първият оператор "If...Then" променя цвета на формата в червен. Вторият оператор променя текста в текстовото поле на "Red Alert".



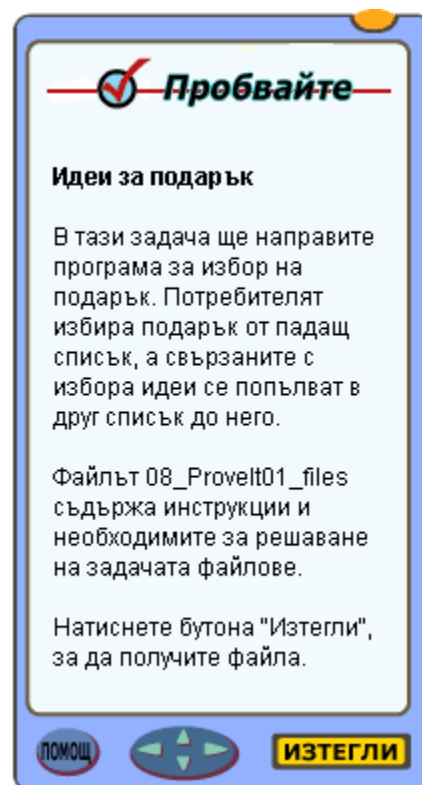
Третият оператор показва диалогов прозорец със съобщение "Danger! Danger!".



08 Пробвайте 01 Идеи за подарък

Докато бяхме в околностите на Киев, Украйна, спряхме в магазин за подаръци. Ако напишете програма, която да помага на купувачите в избора им на подарък, собственикът на магазина ще ви даде автентична картичка с красивия търговски център на Киев.

1. Създайте форма със заглавие "Идеи за подарък". Поставете един етикет с текст "Хората, които купуват" и друг етикет с текст "Също купуват".
2. Поставете на формата компонент ComboBox (Виж секция 04). Променете текста му на "Избери подарък". Попълнете списъка с неговите елементи като използвате таблицата с подаръци по-долу или дайте свои идеи.
3. Поставете текстово поле. Направете текстовото поле многоредово.
4. Винаги когато е избран подарък от падащия списък, попълвайте текстовото поле с най-малко два елемента, които ще бъде подарени на купувачите. Можете да използвате таблицата по-долу или да създадете своя собствена.



Шапка	Шал Ръкавици
Ботуши	Боя за обувки Обувалка
Книга	Хартия Поздравителна картичка
Пощенска картичка	Пощенска марка Пощенски плик
Ваза	Цветя Панделка



Номерът на селектирания елемент се получава от свойството `SelectedIndex` на компонента `ComboBox`. Номерацията започва от 0.

Името на селектирания елемент се получава от свойството `SelectedItem` на компонента `ComboBox`. Ако желаете в текстовото поле да се появяват два елемента, в което съм сигурен, поставете специална Visual Basic константа между тях, като например:

```
TextBox1.Text = "A" & vbCrLf & "B"
```

Разбира се, А и В ще замените според случая.

Ако програмата работи правилно, я покажете на вашия учител.

Полезен съвет

За някои ученици инструкциите няма да са ясни е очакват серия от `If` оператори. Ученикът може да сравнява и по индекс, и по име. Вижте обяснителните бележки за примерите относно двата метода.



Повече от един оператор за сравнение

Забелязахте ли, че операторите "If...Then" в примера FireAlarm имат едно и също условие, което е:

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

Единствената разлика между тях е в израза, който се изпълнява след проверката и условието. Получава се дублиране на код, нали? Няма ли по-лесен начин? Да, разбира се! Visual Basic и повечето съвременни езици за програмиране допускат обединяването на няколко условия. Ето и синтаксиса:

```
If condition Then  
    conditional statement 1  
    conditional statement 2  
    conditional statement etc.  
End If
```

Имайки това предвид, ние можем да пренапишем кода от примера FireAlarm като отстраним повторението на условията и направим програмата по-разбираема. Отворете приложението FireAlarm. Добавете върху формата втори бутон. Задайте стойност "One If" за свойството му Text. Щракнете двукратно върху бутона, за да създадете обработчик на събитието Button2_Click. Добавете следния код:

```
If CheckBox1.Checked = True Then  
    Form.ActiveForm.BackColor =  
System.Drawing.Color.Red  
    TextBox1.Text = "Red Alert"  
    MessageBox.Show("Danger! Danger!")  
End If
```

Изградете и стартирайте проекта. Маркирайте полето за отметка и натиснете бутона "One If". Проверява се едно условие. То е True, поради което всички изрази след условието се изпълняват последователно. Програмата работи точно както преди. Този път, обаче трите условия са обединени! Обединяването на няколко условия в едно прави кода по-кратък и по-разбираем.



Множествен избор

Спомняте ли си, че когато си поръчвах бира отговарях само с да и не на въпросите? Използвани бяха серия от въпроси, за да се изброят възможните варианти. Въпреки, че стана по-бавно аз все пак получих това, което исках. Можете да реализирате въпрос с няколко възможни отговора в кода си като използвате серия от "If...Then" оператори. Имам чудесен пример за това.

Създайте ново Windows приложение с име ChooseAColor. Добавете върху формата три радио-бутона и един обикновен бутон. Използвайте прозореца Properties, за да зададете следните стойности на свойствата. Задайте стойност "Red" за свойството Text на първия радио-бутон. Задайте стойност "Yellow" за свойството Text на втория радио-бутон. Задайте стойност "Green" за свойството Text на третия радио-бутон. Задайте стойност "Color Me" за свойството Text на обикновения бутон. Задайте стойност True за свойството Checked на първия радио-бутон. Цракнете двукратно върху бутона, за да създадете обработчик на събитието Button1_Click. Добавете следния код:

```
Dim myColor As System.Drawing.Color
If RadioButton1.Checked = True Then
    myColor = System.Drawing.Color.Red
End If
If RadioButton2.Checked = True Then
    myColor = System.Drawing.Color.Yellow
End If
If RadioButton3.Checked = True Then
    myColor = System.Drawing.Color.Green
End If

Form.ActiveForm.BackColor = myColor
```

Изградете и стартирайте проекта. Забележете, че радио-бутонът "Red" е маркиран. Натиснете бутона "Color Me". Цветът на формата става червен. Селектирайте друг радио-бутона натиснете бутона "Color Me". Цветът на формата се променя!

Разгледайте кода. За да се отговори на въпрос с множество възможни отговора, са използвани няколко оператора "If...Then". При последователното си изпълнение програмата определя кой от радио-



бутоните е маркиран. След това в зависимост от маркирания радио-бутон
тя изпълнява определен израз.



Булеви оператори в кода

Говорихме за това как да използваме Булевите оператори за решаване на сложни логически задачи. Нека да разгледаме пример, който използва оператора AND за получаване на сложно условие.

Ще променим програмата ChooseAColor като добавим поле за отметка, което ще включва или изключва възможността за промяна на цвета. Ще използваме булевия оператор AND, за да създадем условие, в което ще проверяваме дали полето за отметка е маркирано и дали цветът е променен.

Отворете приложението ChooseAColor. Добавете върху формата поле за отметка. Задайте стойност "Enable Colors" на свойството Text. Редактирайте обработчика на събитието Button1_Click. Заменете кода по следния начин:

```
Dim myColor As System.Drawing.Color
If RadioButton1.Checked = True And
CheckBox1.Checked Then
    myColor = System.Drawing.Color.Red
End If
If RadioButton2.Checked = True And
CheckBox1.Checked Then
    myColor = System.Drawing.Color.Yellow
End If
If RadioButton3.Checked = True And
CheckBox1.Checked Then
    myColor = System.Drawing.Color.Green
End If
Form.ActiveForm.BackColor = myColor
```

Изградете и стартирайте приложението. Натиснете бутона "Color Me". Нищо не се случва! Цветът не се променя. Маркирайте полето за отметка "Enable Colors" и отново натиснете бутона "Color Me". Цветът на формата става червен. Ето и какво се случва. Добавихме следния код в условието на оператора If...Then:

And CheckBox1.Checked

Полезен съвет

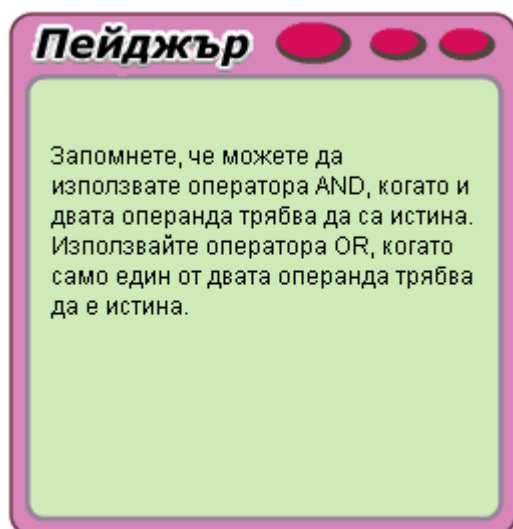
Някои ученици се затрудняват в използването на сложен If оператор. Те го избягват чрез създаване на по-сложни програми. Използвайте програмата за игра на морски шах от секция Unit 4, за да покажете



Така условието се усложни. За да се промени цвета, и двете страни на оператора AND трябва да са True. Например,

```
RadioButton2.Checked = True And CheckBox1.Checked
```

Резултатът от изпълнението на оператора AND е True. Запомнете, че True AND True=True. Ако една от страните на оператора AND е False, то целият израз от условието ще се изчисли като False и цвета няма да се промени.





Пристигане в Киев

Е, свършихме всичко! Ето ни в Киев, Украйна. Не се съмнявах, че ще успеем. По време на пътуването ни научихте как да използвате оператора за условен преход "If...Then". Сега вече можете да накарате програмите си да разсъждават. Те могат да взимат решения, така че внимавайте какво ги карате да правят!

Получих забавна задача на джобния си компютър. Мисля, че ще я направите с удоволствие. Не забравяйте и теста, който трябва да решите за Комисията по състезанията.



08 Пробвайте 02 Мотивационна програма

Знаем, че вие се обучавате по курса "Светът на програмирането", защото сте проявили интерес за това в клас. Напишете обзорна програма, за да подпомогнете други ученици във взимането на решение за изучаване на курса по програмиране.

Пробвайте

Мотивационна програма

В тази задача ще направите интерактивна обзорна форма. Използвайки скала от 1 до 10, потребителят ще оцени предпочитанията си по отношение на различни програмни техники. Програмата ще натрупва резултата и ще изчислява нивото на възможностите му, свързани с програмирането.

Файлът 08_proveit02_files съдържа инструкции и необходимите за решаване на задачата файлове.

Натиснете бутона "Изтегли", за да получите файла.

ПОМОЩ ИЗТЕГЛИ

Създайте форма подобна на показаната по-долу:



Тест за програмист

Би ли записал курс по програмиране?

Въведете число от 0 до 10 във всяко текстово поле.

0 означава Никога, 5 означава Понякога, а 10 - Винаги.

Обичате ли да решавате алгебрични задачи?

Можете ли да отделяте внимание на детайлите и подробностите?

Разбирате ли упътванията за използване на DVD, мобилни телефони и др.?

Обичате ли да разрешавате проблеми?

Прецизно ли изпълнявате последователност от задачи?

Можете ли да се концентрирате върху едно нещо дълго време?

Общо Тълкуване на резултата

Попълнете шестте текстови полета и натиснете бутона "Резултат". 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.

- Ако общата получена стойност е по-голяма от 40, в текстовото поле Evaluation покажете съобщение, което показва, че ученикът е отличен кандидат за курса по програмиране.
- Ако общата получена стойност е между 20 и 39, то уведомете ученика, че възможността за успех в курса по програмиране е несигурна, но би могъл да опита.



- Ако общата получена стойност е под 20, то препоръчайте на ученика да промени пътя на своето професионално развитие.

Какъв е вашия собствен резултат?

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

Полезен съвет

Когато оценявате тази задача проверете дали, точно 20 е стойността за "неуверен" и точно 40 – за "атакуем".

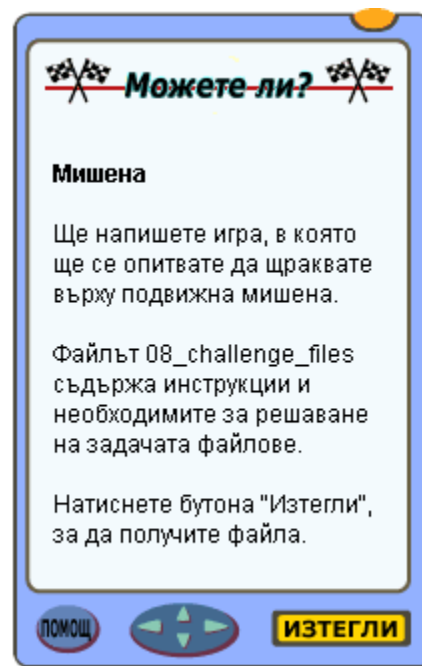
08 Можете ли? Мишена

Помните ли играта, която наблюдавахме по време на вечеря миналата вечер, когато трябваше да щраквате с мишката по подвижна мишена? Нека да създадем такава с Visual Basic. Това ще изисква някои по специални действия така, че прочетете внимателно инструкциите по-долу.

Полезен съвет

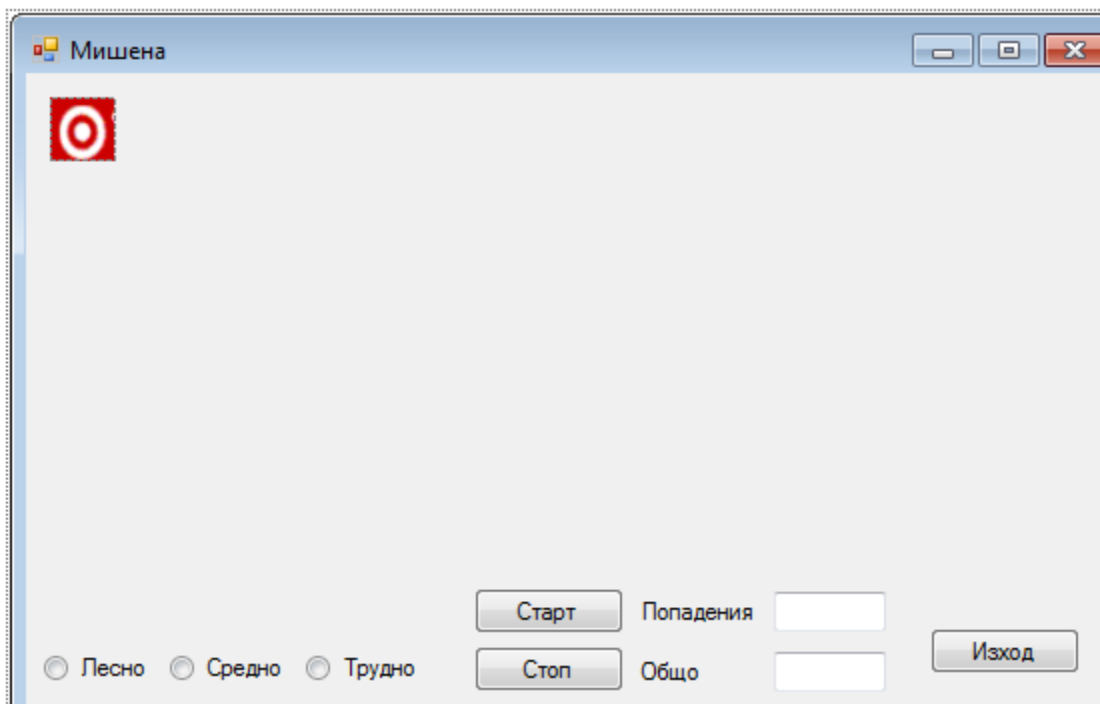
Тази програма е по-голяма от предходната. Тя може да бъде полезна за преговор на обяснителните бележки, за да се подготвите за въпросите на

Променете размера на формата. За височина задайте стойност 350, а за ширина – 500. За заглавие на формата използвайте "Мишена".





Добавете компоненти върху формата така, че да изглежда по следния начин:



Целта на играта е да се щракне върху подвижен обект. Кодът на програмата ще придвижва картинката, правейки я трудна за уцелване. Бутонът "Старт" инициализира и стартира играта. Бутонът "Стоп" приключва играта. Бутонът "Изход" затваря програмата. Трите радиобутони определят сложността на нивото, до което е достигнала играта. Добавете компонент за картинка, който ще визуализира подвижния обект.

Изображението на подвижния обект се намира във файла 08_Challenge_Target.gif. Използвайте го, за да зададете картинката, която ще се визуализира от контрола с име PictureBox. Задайте стойност StretchImage за неговото свойство SizeMode.

Програмата използва таймер, за да се осигури изпълнението на кода на специфични интервали. След като създадете формата, отворете контрол с име Timer в прозореца Toolbox и щракнете двукратно върху него. Той няма да бъде видим за потребителя.

1. Създайте променлива от тип Integer, за да преброявате успешните удари.



2. Създайте променлива от тип Integer, за да определяте общия брой на направените опити.
3. В процедурата, която обработва събитието Button_Click на "Старт" бутона добавете код, който прави следното:
 - Нулира броячите Hits и Total.
 - Установява стойност True за свойството Enable на компонента Timer.
4. В процедурата, която обработва събитието Button_Click на "Старт" бутона добавете код, който прави следното:
 - Ако радио-бутон с име RadioButton1 е натиснат, установява стойност 900 за свойството Interval на компонента Timer, и стойност 40 за височината и ширината на картинката.
 - Ако радио-бутон с име RadioButton2 е натиснат, установява стойност 850 за свойството Interval на компонента Timer, и стойност 35 за височината и ширината на картинката.
 - Ако радио-бутон с име RadioButton3 е натиснат, установява стойност 800 за свойството Interval на компонента Timer, и стойност 30 за височината и ширината на картинката.
5. В процедурата, която обработва събитието Button_Click на "Стоп" бутона добавете код, който установява стойност False за свойството Enable на компонента Timer.
6. В процедурата, която обработва събитието Button_Click на "Изход" бутона добавете код, който затваря програмата посредством командата End.
7. В процедурата, която обработва събитието PictureBox_Click на компонента PictureBox добавете код, който увеличава с единица стойността на брояча Hits и я визуализира в текстовото поле с етикет "Попадения".
8. В процедурата, която обработва събитието, генерирано от компонента Timer, добавете следния код, за да осигурите движение на обекта. Предполага се, че формата има размер 350 x 550.

```
Dim MyRandomGenerator As System.Random
MyRandomGenerator = New System.Random
Dim RandomX As Integer
Dim RandomY As Integer
```

```
RandomX = MyRandomGenerator.Next(1, 550)
```



```
RandomY = MyRandomGenerator.Next(1, 250)
PictureBox1.SetBounds(RandomX, RandomY,
    Me.Width, Me.Height,
    System.Windows.Forms.BoundsSpecified.Location)
```

9. След горния код, увеличете общия брой на направените опити с 1 и го визуализирайте в текстовото поле с етикет "Общо".

Продължение / Обобщение

В началото на играта оставете цвета на формата сив. Осигурете промяна на цвета в червен, когато потребителя уцели подвижния обект.

- 1 Компютрите взимат решения, основавайки се на?
- ☐ A. True / False
 - ☐ B. Do / Until
 - ☐ C. състоянието на байтовете
- 2 Какво означава този оператор: <>?
- ☐ A. Степенуване
 - ☐ B. Различно
 - ☐ C. Равно
- 3 Кога операторът AND връща стойност True?
- ☐ A. Когато лявата или дясната страна е True
 - ☐ B. Когато и двете страни са True
 - ☐ C. Не може да се определи
- 4 Кога операторът OR връща стойност True?
- ☐ A. Когато лявата или дясната страна е True
 - ☐ B. Когато и двете страни са True
 - ☐ C. Не може да се определи

