

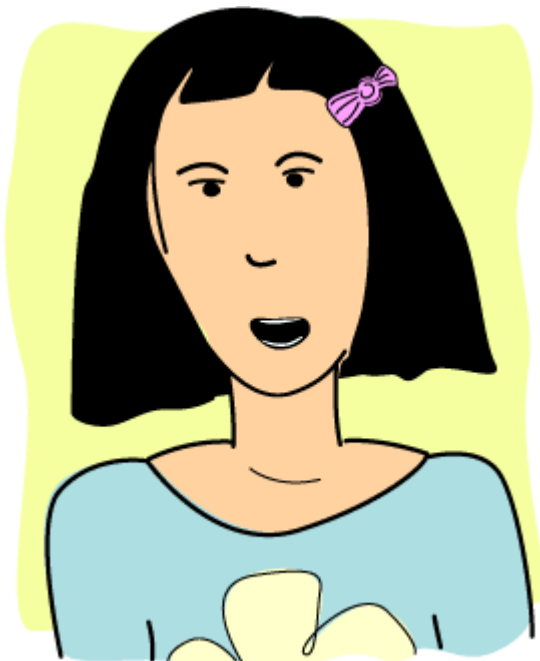
## Дървовиден и списъчен изглед

### *Лондон, Великобритания*

Можете ли да повярвате? Ние сме на последния етап от пътуването ни! Само още една дестинация и нашето пътешествие ще бъде завършено. Сигурен съм, че ви беше забавно! За Клиф и мен беше удоволствие да ви научим на това, което знаем за Visual Basic и Visual Studio 2010 ! Надявам се, че скоро ще се срещнем на друго пътешествие. Вече сте готови да започнете курс за по-напреднали!

По пътя за Лондон ще ви запозная с едни по-специални контроли, наречени дървовиден и списъчен изглед. Със сигурност знам, че сте ги виждали, а сега ще ви науча как да ги използвате.

Да вървим!





## **Windows Explorer**

Приемам, че сте се запознали с програмата Windows Explorer. Клиентската област на Windows Explorer е заета предимно от два големи и сложни контрола. Контролът tree-view (дървовиден изглед) вляво показва йерархичен списък на дисковите устройства и директориите на потребителя. Контролът list-view (списъчен изглед) вдясно показва поддиректориите и файловете в избраната директория по един от четири начина: обикновен списък, таблица с множество колони, имена с малки икони и имена с големи икони.

В тази секция ще разгледаме контролите за дървовиден и списъчен изглед. Големите възможности и гъвкавостта на тези контроли прави невъзможно изчерпателното им представяне. Но със сигурност ще обхвана достатъчно като за начало.

### **Полезен съвет**

Накарайте учениците си да разгледат програмата Windows Explorer и да проучат нейните възможности.



## Дървовидни изгледи

Дървовидният изглед, както споменах, се използва за показване на списък от дискови устройства и директории. Можете, обаче да използвате този контрол за показване на произволна йерархична информация. В йерархията възлите са във взаимоотношения родител-наследник. Visual Studio .NET използва дървовиден изглед за показване на проекти и файлове.

Основният елемент в дървовидния изглед се нарича възел (Node). Възлите на един дървовиден изглед образуват т.нар. колекция. Можете да добавяте нови възли към тази колекция с помощта на метода Add по следния начин:

```
TreeView1.Nodes.Add("Нов възел")
```

В случая ще се добави възел от най-високо ниво. Като аргумент на метода се подава името на новия възел. Ако желаете да добавите възел, който да е наследник на вече съществуващ възел, използвайте код, подобен на този:

```
TreeView1.Nodes(0).Nodes.Add("Нов възел")
```

В този случай ще се добави възел-наследник към възел с индекс 0. Както се досещате, номерирането на възлите в списъка започва от 0.

Съществува и метод за премахване на възел с име RemoveAt. Този метод получава като аргумент индекс на възел, който трябва да бъде изтрит. Кодът, който следва премахва възел с индекс 0, който е от най-високо ниво.

```
TreeView1.Nodes.RemoveAt(0)
```

Премахването на даден възел ще доведе до премахване и на неговите наследници от дървото. Ако желаете да премахнете възел-наследник използвайте следния код:

```
TreeView1.Nodes(0).Nodes.RemoveAt(0)
```

Можете да влагате произволно възлите в дървото. Ако желаете да премахнете всички възли е достатъчно да извикате метода Clear ето така:

```
TreeView1.Nodes.Clear()
```



Ако желаете да премахнете всички възли-наследници на даден възел повикването на метода `Clear` ще изглежда по следния начин:

```
TreeView1.Nodes(0).Nodes.Clear()
```

Този код ще премахне всичко възли наследници на възел с индекс 0, който е от най-горно ниво в дървото.



### ***Дървовиден изглед стъпка по стъпка***

Когато използвате дървовиден изглед, автоматично пред всеки възел се добавя знак "+", ако има наследници. Когато щракнете върху него възлите наследници се показват отместени навътре, а знакът "+" се заменя с "-". Това се дължи на стойността true за свойството ShowPlusMinus.

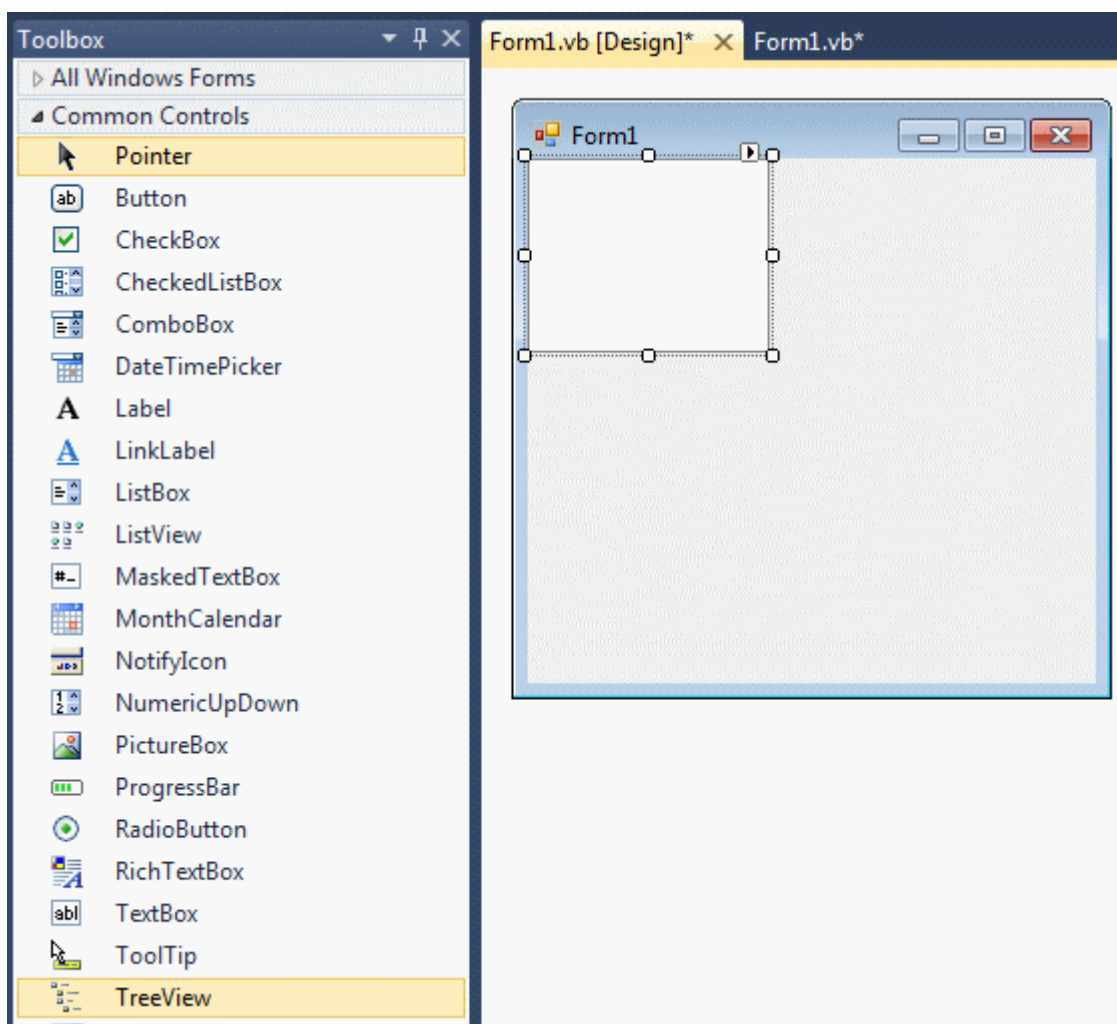
Дървовидния изглед притежава още две свойства – ShowLines и ShowRootLines, които при зададена стойност true предизвикват свързване на възлите с линии. Свойството ShowRootLines се отнася за възлите от най-високо ниво, които често се наричат коренови, а свойството ShowLines – за останалите възли.

Дървовидните изгледи често притежават малки картинки вляво от възлите. Те могат да се променят при разширяването на възела, когато се щракне върху знака "+". Например, Windows Explorer показва затворена папка за възел на затворена директория и отворена папка за възел на разширена директория.

За да добавите картинки, трябва да използвате контрол с име ImageList. Той представлява списък с изображения. За да стане всичко по-ясно, ще проследим необходимите за това действия стъпка по стъпка.

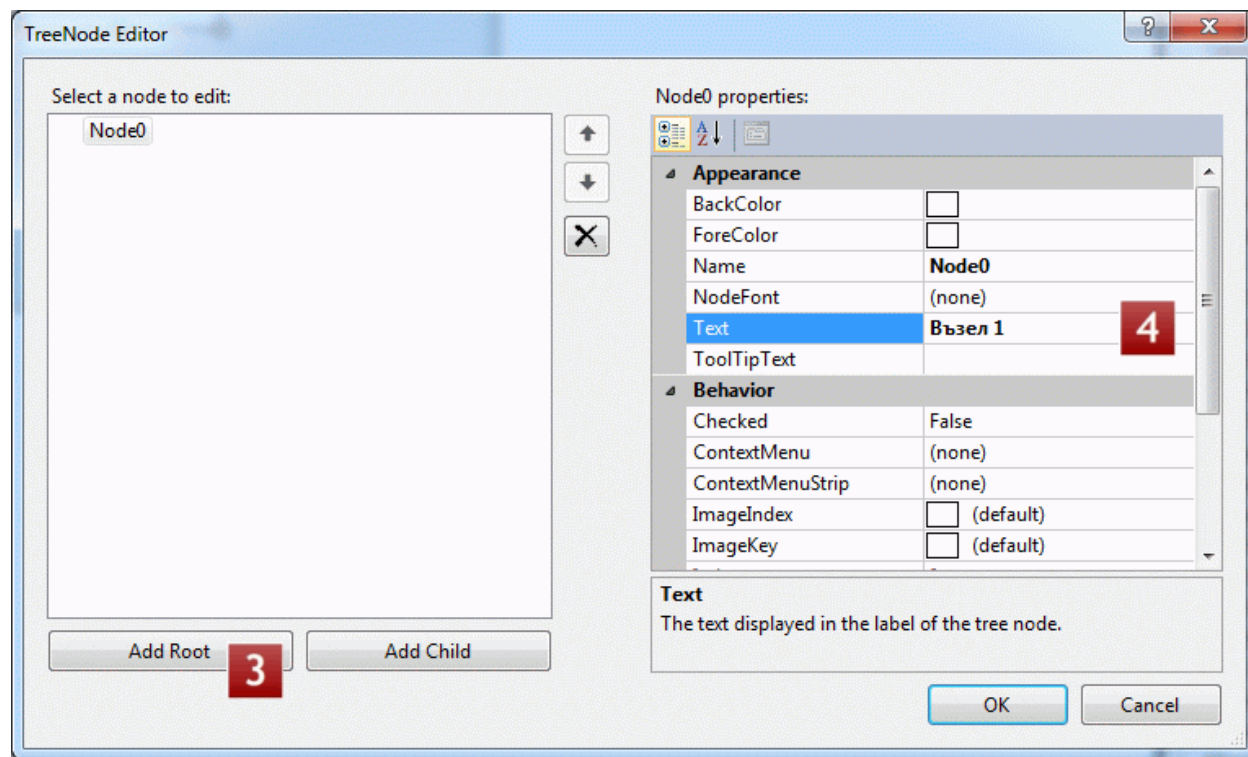


1. Създайте нов проект с Visual Studio 2010 и от прозореца Toolbox добавете контрол TreeView.



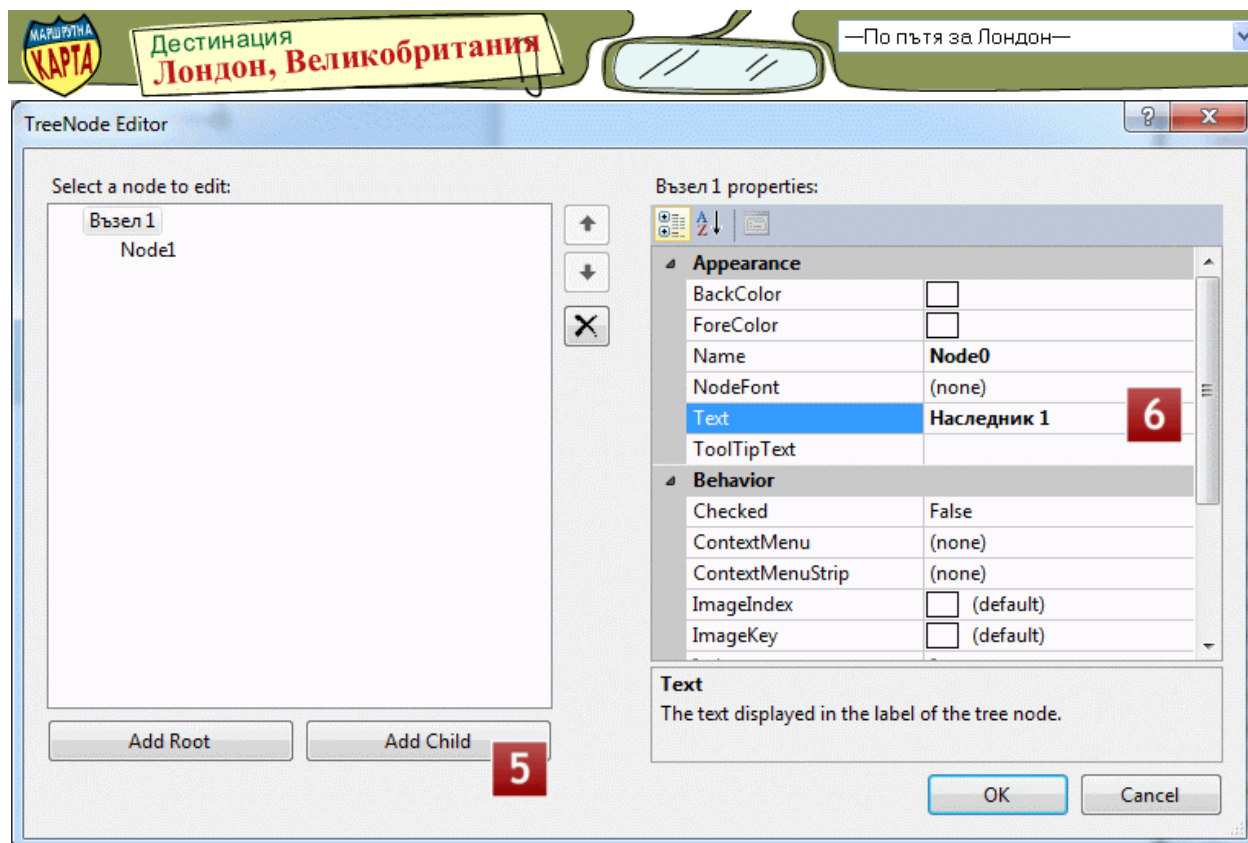


- Щракнете върху свойството Nodes на TreeView контрола в прозореца Properties. Ще се появи диалогов прозорец с име TreeNode Editor.
- Натиснете бутона Add Root, за да добавите възел от най-високо ниво.
- Можете да смените заглавието му от стойството Text, което е достъпно в дясната част на прозореца TreeNode Editor.

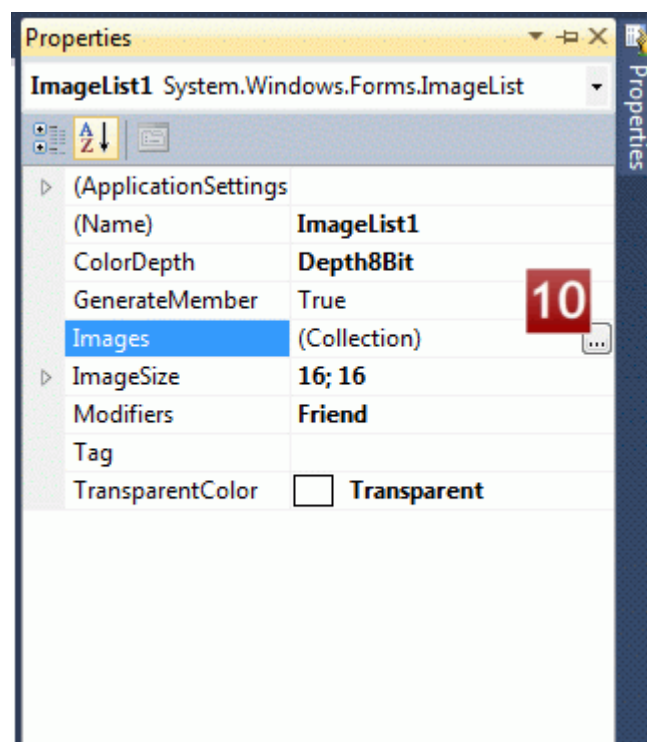


- Щракнете възху добавения възел и натиснете бутона Add Child.
- Променете заглавието на новия възел по аналогичен на т.4 начин.

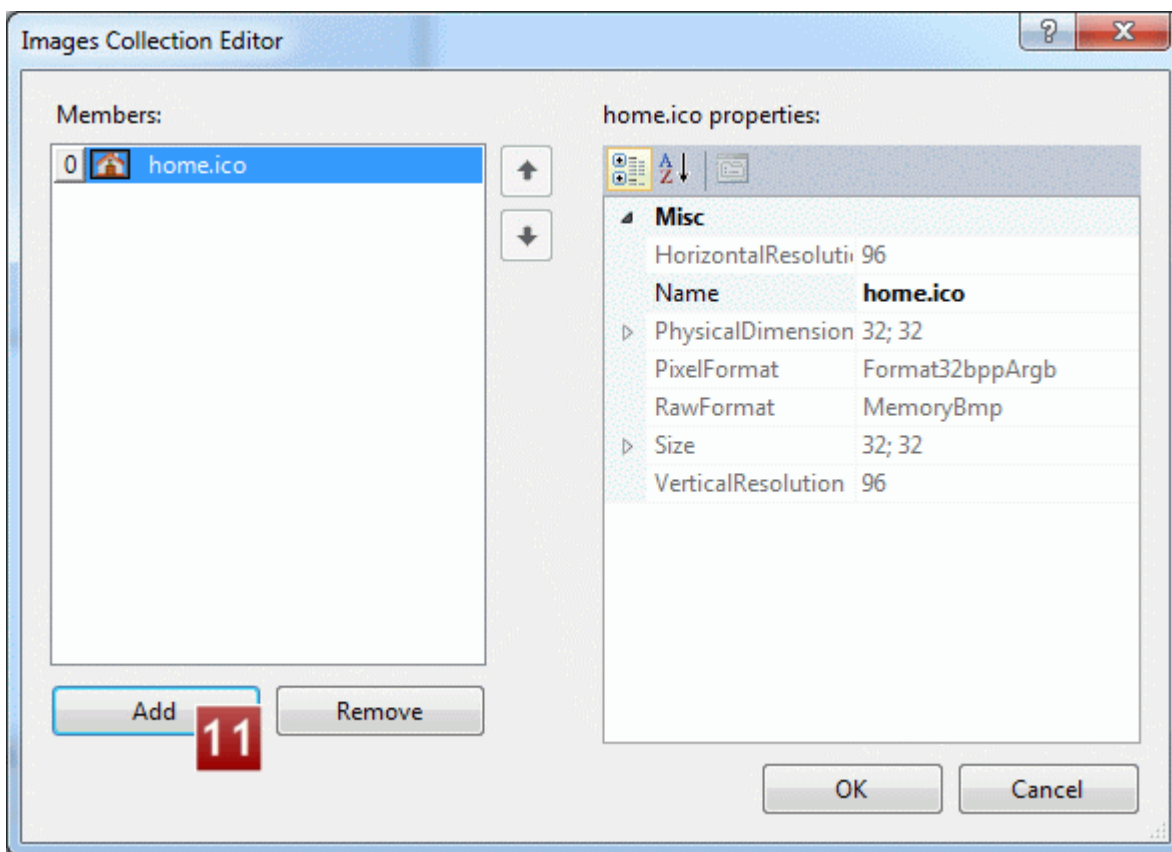




7. Добавете по същия начин още възли-наследници.
8. Повторете стъпките от 3 до 7 няколко пъти.
9. От прозореца Toolbox -> Components добавете ImageList контрол.
10. Щракнете върху свойството Images на ImageList контрола е прозореца Properties. Ще се появи диалогов прозорец с име Image Collection Editor.



11. Натиснете бутона Add, за да добавите картинка. Използвайте диалоговия прозорец Open, който ще се появи за да изберете подходяща картинка.



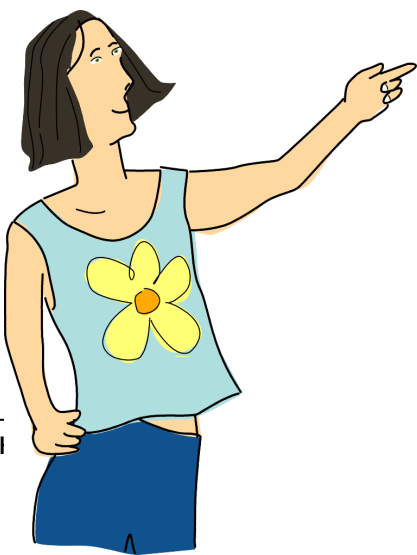
12. Добавете толкова картинки, колкото са ви необходими и натиснете бутона OK.

#### Полезен съвет

Ако учениците ви не могат да открият картинки на компютрите си, ги насочете към папките с инструкциите за решаване на

13. Щракнете върху TreeView контрола и след това отворете

свойството му ImageList в прозореца Properties. Задайте му стойност ImageList1.



14. Изградете и стартирайте приложението.

Забелязахте ли какво се случи, когато зададохте стойност за свойството ImageList на TreeView контрола?



Разширете възлите от дървото със знак "+" пред тях. Както виждате пред всеки един от възлите вече има картинка. Картинката обаче е една и съща за всички възли. За да я промените отново покажете прозореца TreeNode Editor. Селектирайте възел чиято картинка искате да промените. След това в дясната част на прозореца TreeNode Editor открийте свойствата ImageIndex и Selected ImageIndex. Първото свойство задава картинката, която ще се показва пред възела без да е избран, а второто – когато е избран.

Беше забавно и лесно. Нали!



## Списъчни изгледи

В най-разширената си форма контролът `ListView` показва текстова информация в редове и колони със заглавия на колоните. Първата колона с информация съдържа елементите, а другите колони – поделементите, прикрепени към всеки елемент. Например в `Windows Explorer` името на файла е елемент, а размера на файла, датата на модификация и атрибутите са поделементи. Контролът `ListView` може да показва и обикновен списък само от елементите (без поделементите), елементите в множество колони с малки икони и елементите в множество колони с големи икони.

Елементите на списъчния изглед формират колекция, представена от свойството му `Items`. Подобно на дървовидния изглед, списъчният изглед притежава свойства `SmallImageList` и `LargeImageList`, с помощта на които можете да задавате картинките за елементите.

За да настроите вида на списъчния изглед използвайте свойството му `View`. То може да приема една от следните стойности:

| Стойност               | Описание                                                                                                                                                        |
|------------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <code>LargeIcon</code> | Всеки елемент се показва с голяма картинка, която се взима от свойството <code>LargeImageList</code> и заглавие.                                                |
| <code>Details</code>   | Всеки елемент се показва на отделен ред с детайлна информация, подредена в колони. Колоните имат свои заглавия.                                                 |
| <code>SmallIcon</code> | Всеки елемент се показва с малка картинка, която се взима от свойството <code>SmallImageList</code> и заглавие.                                                 |
| <code>List</code>      | Всеки елемент се показва с малка картинка, която се взима от свойството <code>SmallImageList</code> и заглавие. Елементите се разполагат в колони без заглавия. |
| <code>Tile</code>      | Всеки елемент се показва с голяма картинка, заглавие и детайлна информация вдясно от картинката.                                                                |

Картинките в `SmallImageList` и `LargeImageList` трябва да съвпадат. Например, третата картинка от `LargeImageList` трябва да е по-голяма версия на третата картинка от `SmallImageList`.

Заглавията на колоните в списъчния изглед също формират колекция, представена от свойството му `Columns`. Някои от свойствата на колоните са описани в таблицата по-долу:

| Свойство          | Описание                                         |
|-------------------|--------------------------------------------------|
| <code>Text</code> | Задава текста, който ще се ползва за заглавие на |



|           |                                                                            |
|-----------|----------------------------------------------------------------------------|
|           | колоната.                                                                  |
| Width     | Задава ширина на колоната.                                                 |
| TextAlign | Задава подравняване на текста, който ще се ползва за заглавие на колоната. |

Добавянето на нов елемент към списъчния изглед става с помощта на метода Add ето така:

```
ListView1.Items.Add("Нов елемент")
```

Ако желаете да добавите поделемент използвайте следния код:

```
ListView1.Items(1).SubItems.Add("Нов поделемент")
```

В този случай ще се добави поделемент към елемент с индекс 1. И тук номерацията на елементите започва от 0.

Премахването на елемент става с метода RemoveAt по следния начин:

```
ListView1.Items.RemoveAt(0)
```

Този метод получава като аргумент индекс на елемент, който трябва да бъде изтрит. Ако желаете да премахнете всички елементи от списъка е достатъчно да извикате метода Clear ето така:

```
ListView1.Items.Clear()
```

Някои от свойствата на елементите на списъчния изглед са описани в таблицата по-долу:

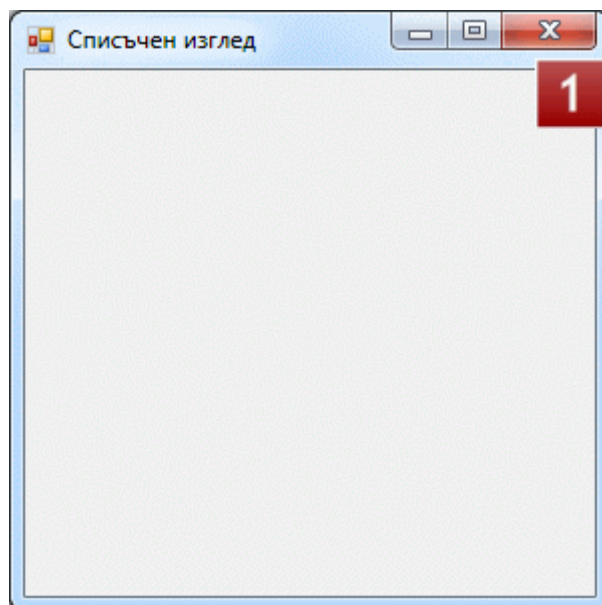
| Свойство   | Описание                                                            |
|------------|---------------------------------------------------------------------|
| Text       | Задава текста, който ще се ползва за заглавие на елемента.          |
| ImageIndex | Задава номера на картинката, която ще се показва пред елемента.     |
| SubItems   | Задава поделементите на даден елемент.                              |
| ForeColor  | Задава цвета на текста, който ще се ползва за заглавие на елемента. |
| BackColor  | Задава цвета на фона на елемента.                                   |



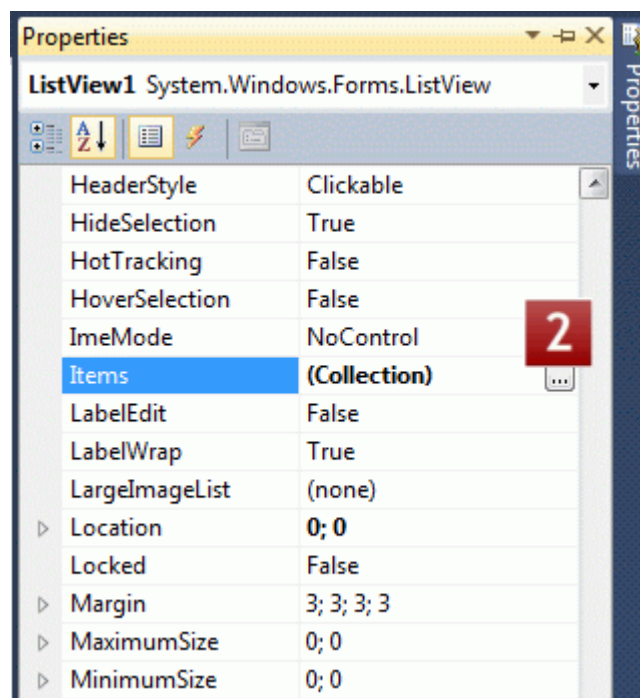
## Списъчен изглед стъпка по стъпка

Ето и необходимите действия за създаване на списъчен изглед в средата на Visual Studio 2010.

1. Създайте нов проект с Visual Studio 2010 и от прозореца Toolbox добавете контрол ListView.



2. Щракнете върху свойството Items на ListView контрола в прозореца Properties. Ще се появи диалогов прозорец с име ListViewItem Collection Editor.

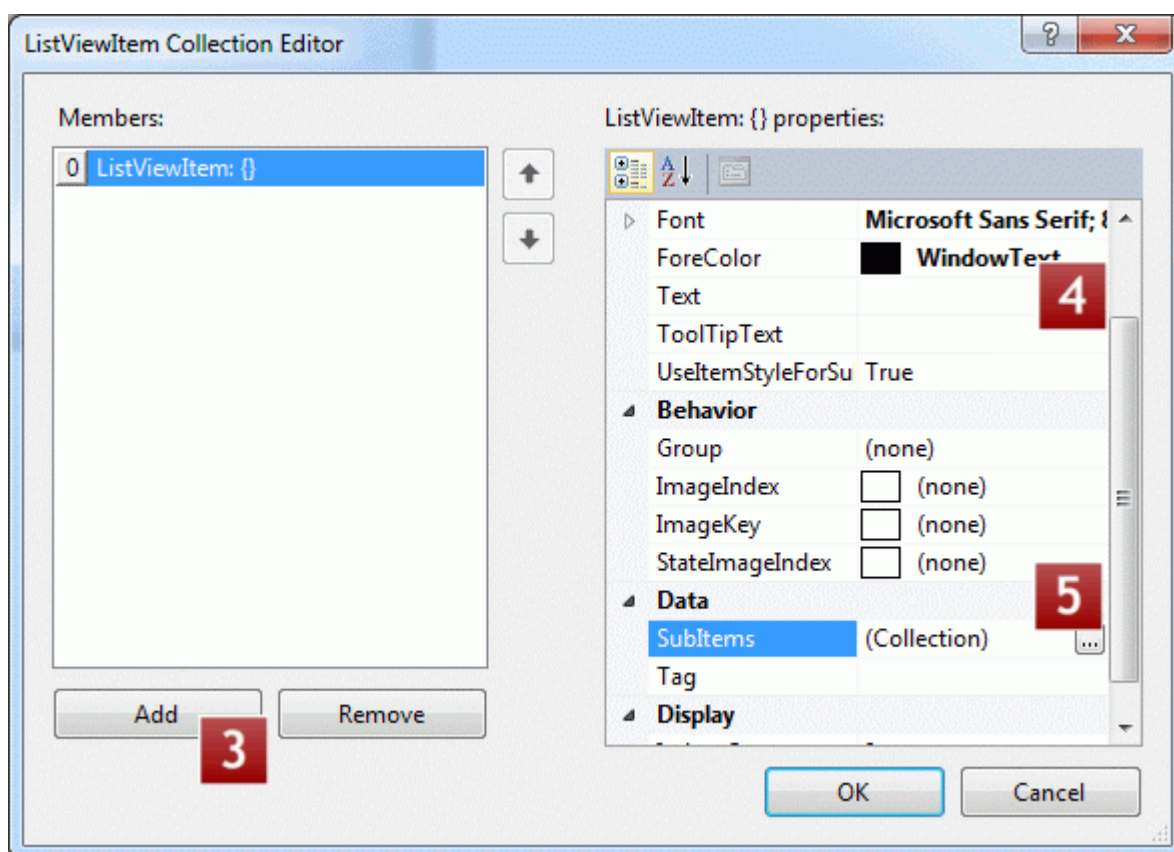


3. Натиснете бутона Add, за да добавите нов елемент.

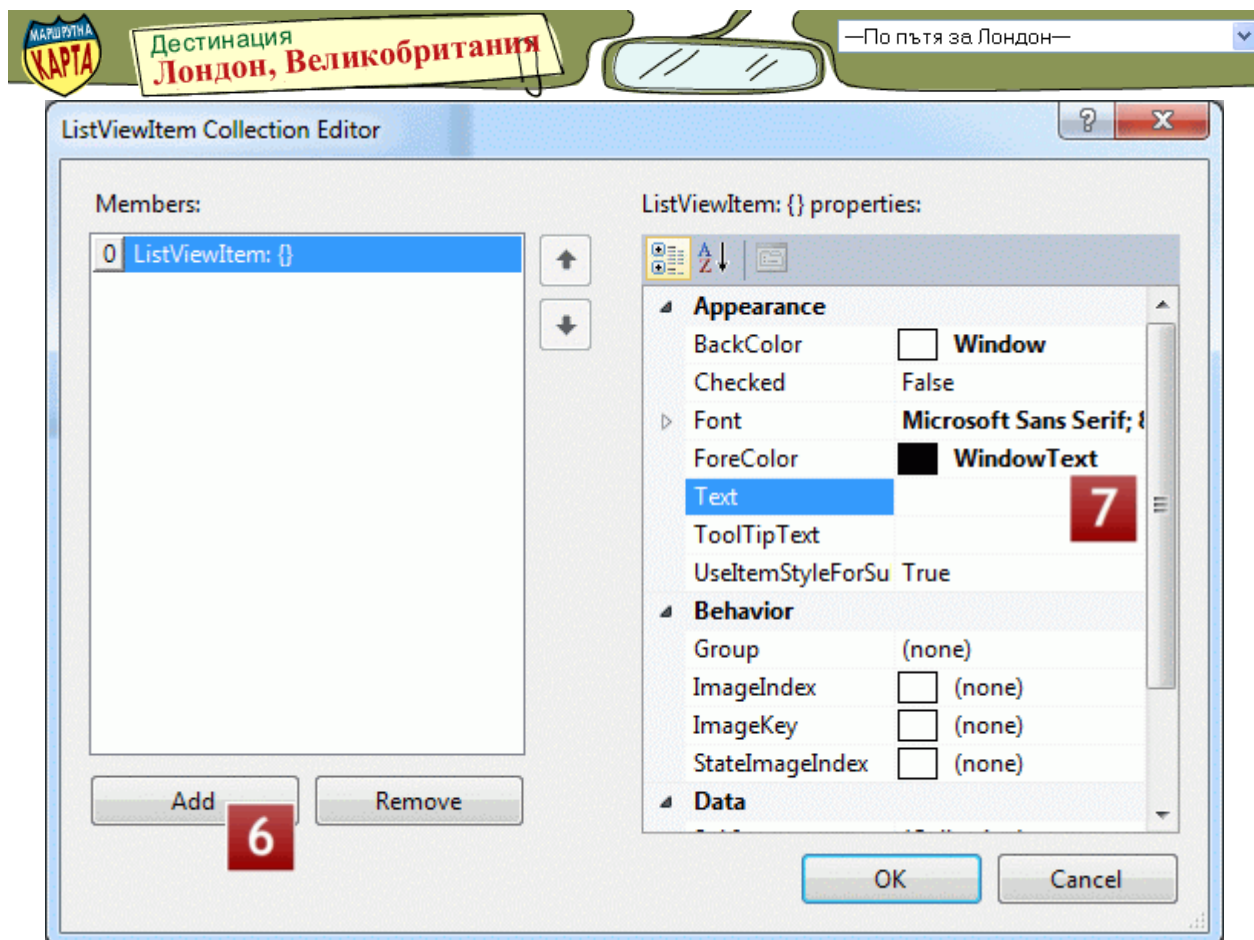




4. Задайте заглавие на елемента от свойството Text, което е достъпно в дясната част на прозореца ListViewItem Collection Editor.
5. Щракнете върху свойството SubItems. Ще се появи диалогов прозорец с име ListViewSubItem Collection Editor.



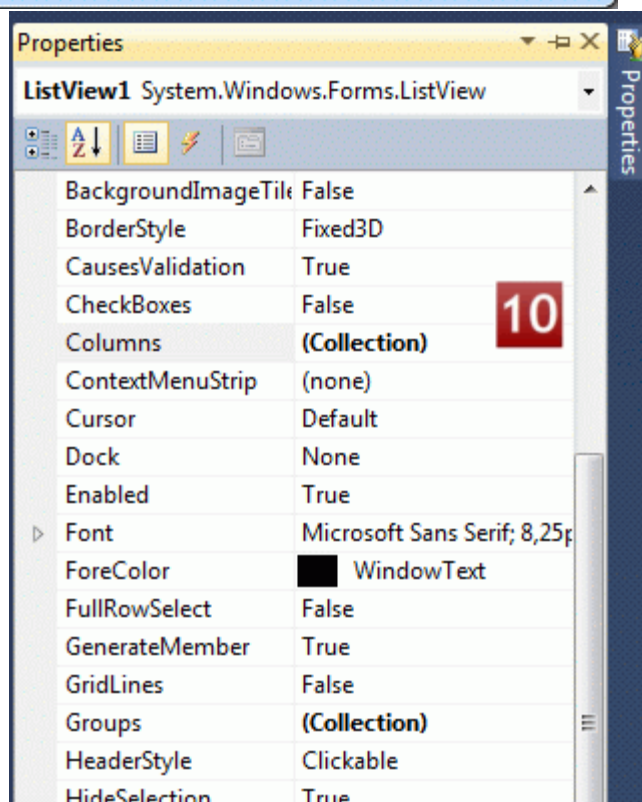
6. Натиснете бутона Add, за да добавите нов поделемент.
7. Задайте заглавие на поделемента от свойството Text от дясната част на прозореца ListViewSubItem Collection Editor.



8. Повторете стъпка 6 и стъпка 7 няколко пъти и натиснете бутона OK.

9. Повторете стъпките от 3 до 8 също няколко пъти и натиснете бутона OK.

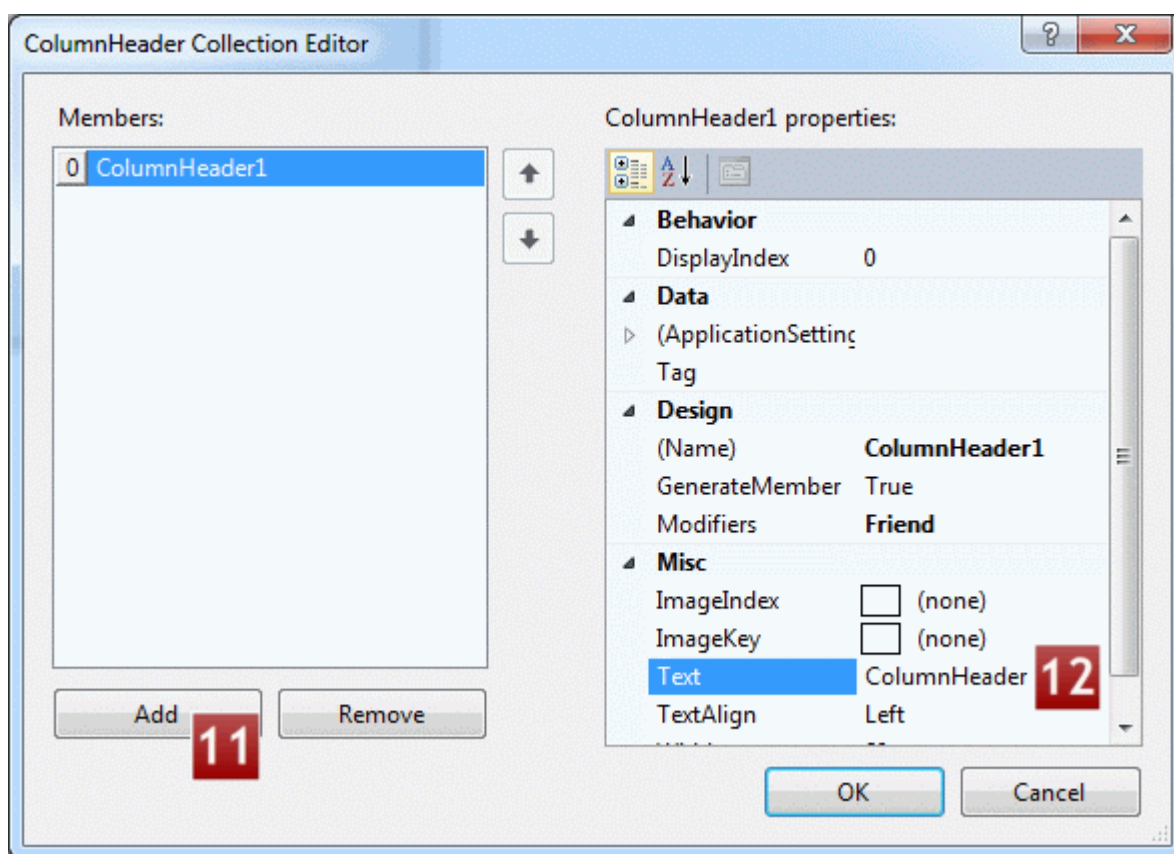
10. Щракнете върху свойството Columns на ListView контрола в прозореца Properties. Ще се появи диалогов прозорец с име ColumnHeader Collection Editor.







11. Натиснете бутона Add, за да добавите ново заглавие на колона.
12. Задайте стойност за свойството Text на колоната.



13. Повторете стъпка 11 и стъпка 12 в зависимост от броя на поделементите, които сте задали на главните елементи и натиснете бутона OK.
14. Задайте стойност Details на свойството View на ListView контрола.
15. Изградете и стартирайте приложението.

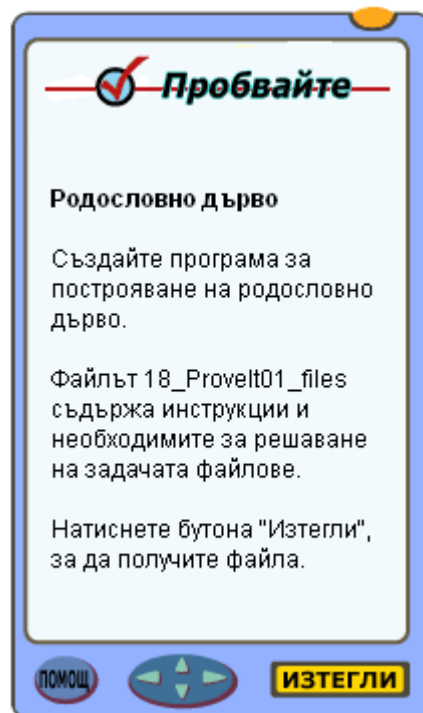
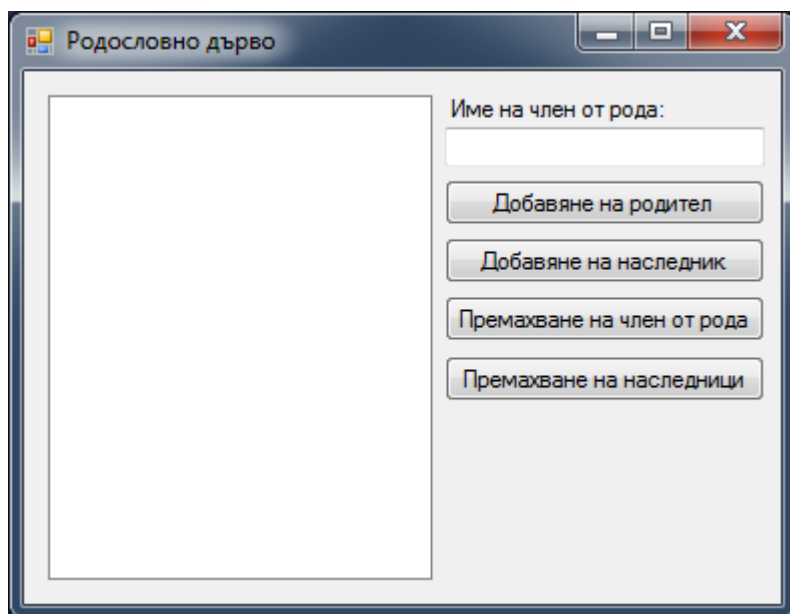


Добавянето на картинки пред елементите на ListView контрола става по начина, описан за TreeView контрола. За да не ви отегчавам ще ви оставя сами да разучите как става това.

## 18 Пробвайте 01 Родословно дърво

Време е да проверим знанията ви.

Създайте форма с дървовиден изглед, която изглежда по следния начин:



Използвайте бутона “Добавяне на родител”, за да позволите добавяне на прародителите от родословното дърво.

Използвайте бутона “Добавяне на наследник”, за да позволите добавяне на наследник към избран член от родословното дърво.

Името на добавения член взимайте от текстовото поле.

Използвайте бутона “Премахване на член от рода”, за да позволите премахване на член от родословното дърво заедно с наследниците му.

Използвайте бутона “Премахване на наследник”, за да премахнете всички наследници на избран член от родословното дърво.

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.

## 18 Пробвайте 02 Списък със задачи

Получих още една задача джобния си компютър.

Създайте форма със списъчен изглед, която изглежда по следния начин:

**Пробвайте**

**Списък със задачи**

Напишете програма, която позволява да създавате списък със задачи, които имате да изпълните. Нека всяка една задача се описва с наименование, описание и срок за изпълнение.

Файлът 18\_ProveIt02\_files съдържа инструкции и необходимите за решаване на задачата файлове.

Натиснете бутона "Изтегли", за да получите файла.

ПОМОЩ
**ИЗТЕГЛИ**

Задайте стойност Details за свойството View на списъчния изглед и добавете три колони като използвате колекцията Columns от прозореца Properties.

В обработчика на събитието Click на бутона "Редактирай" добавете код, който променя текста на поделементите за селектирания елемент. Използвайте свойството SelectedItems на контрола ListView.

В обработчика на събитието Click на бутона "Добави" добавете код, който добавя нов елемент към списъчния изглед.

В обработчика на събитието Click на бутона "Премахни" добавете код, който премахва селектирания елемент от списъчния изглед. Използвайте метода Remove на колекцията Item.

Не забравяйте да изчистите съдържанието на трите текстови полета при натискането на бутоните "Редактирай" и "Добави", след като съответно е редактиран или добавен елемент.



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

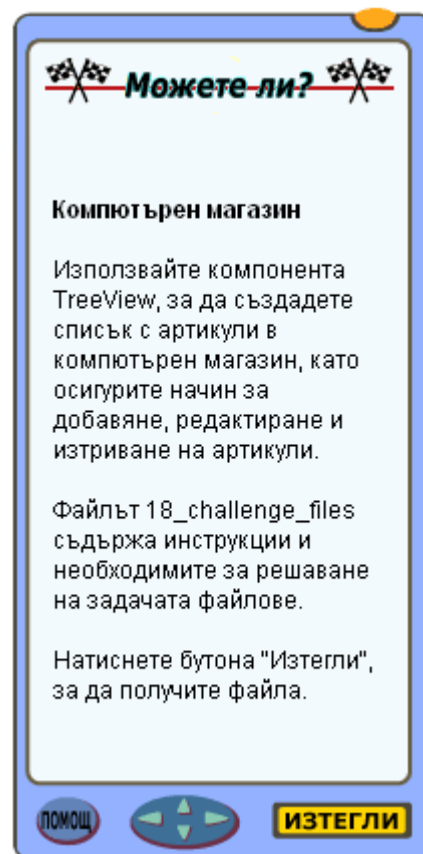
Ако програмата работи правилно, я покажете на вашия учител.



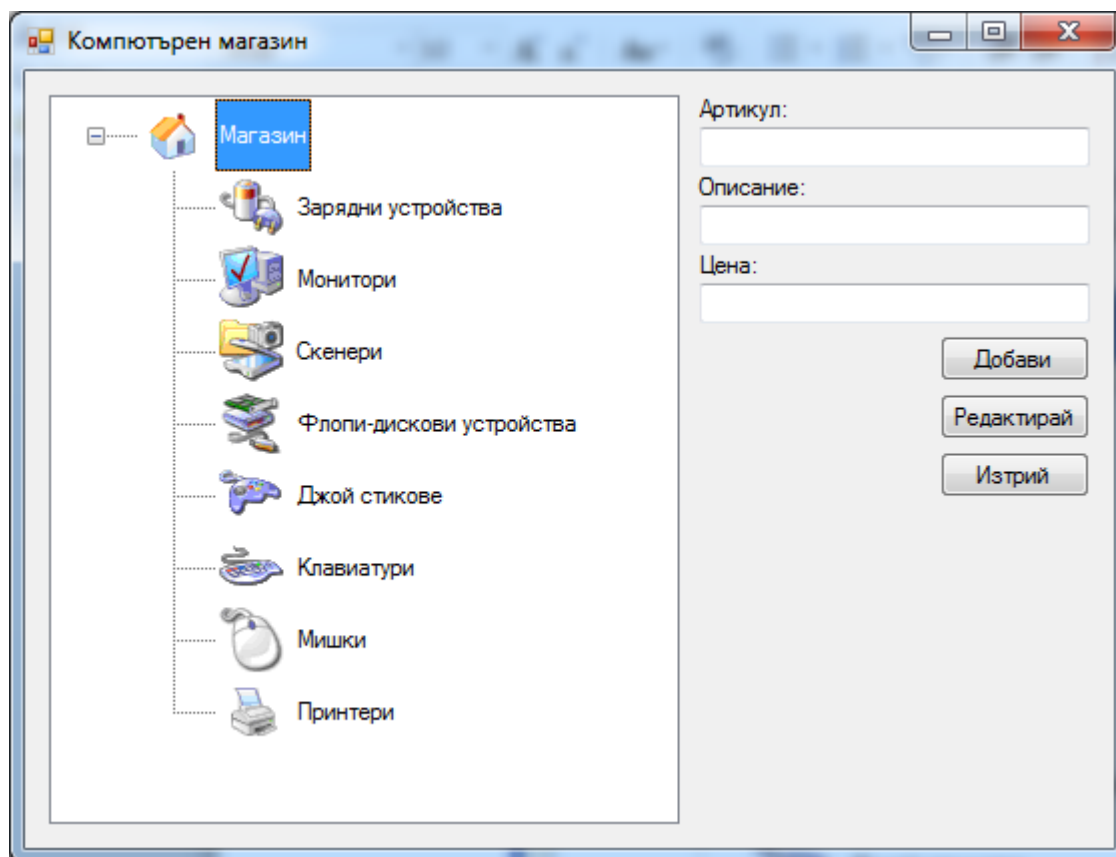
## 18 Можете ли? Компютърен магазин

Ето, че стигнахме до последната задача, която трябва да решите по време на нашето пътешествие.

В папката с инструкциите за изпълнение на задачата имате 10 картинки, които можете да използвате, за да обозначите групите компютърни артикули.



Създайте форма, подобна на следната:



Създайте списък с изображения за дървовидния изглед. Изображенията са с размер 36x36. Той определя стойността, която трябва да зададете за свойството `ImageSize` на дървовидния изглед. Нека последното изображение в списъка да бъде `folder.bmp`, а първото – `home.bmp`.

Добавете елементи към дървовидния изглед както е показано на формата по-горе. За всеки елемент, обозначаващ група артикули, задайте еднаква стойност за свойствата `ImageIndex` и `SelectedImageIndex`.

В обработчика на събитието `Click` на бутона "Добави" добавете код, който добавя нов елемент с два поделемента към избрания възел от дървовидния изглед. Заглавията на елемента и поделементите му вземете от текстовите полета като знаете, че заглавието на елемента е име на артикул, а заглавието на поделементите са съответно описание и цена на артикула. Задайте стойност 9 за свойствата `Imageindex` и `SelectedImageIndex` на новия елемент. Задайте стойност

**`TreeView1.SelectedNode.Index`**



свойствата `ImageIndex` и `SelectedImageIndex` на двата поделемента.

В обработчика на събитието `Click` на бутона "Редактирай" добавете код, който променя вече добавен и селектиран елемент.

В обработчика на събитието `Click` на бутона "Изтрий" добавете код, който изтрива селектиран елемент.

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.





## Проверка на знанията

НАПРАВЕТЕ ТЕСТА ОТНОВО

- 1** Каква е структурата на информацията, показвана от контрола `TreeView`?

  - ☐ А. Информацията не е структурирана
  - ☐ В. Списъчна
  - ☐ С. Йерархична
- 2** Какво прави следният ред с код `TreeView1.Nodes.RemoveAt(0)`?

  - ☐ А. Премахва произволен възел.
  - ☐ В. Премахва възел от най-високо ниво с позиция 0.
  - ☐ С. Премахва всички възли.
- 3** Какво се задава със свойството `ImageIndex` на контрола `ListView`?

  - ☐ А. Номер на картинка, която ще се показва пред елемента
  - ☐ В. Цвета на фона на елемента
  - ☐ С. Заглавието на елемента
- 4** Колко стойности може да приема свойството `View` на контрола `ListView`?

  - ☐ А. 2
  - ☐ В. 7
  - ☐ С. 5