СЪЗДАВАНЕ НА *.dll ФАЙЛОВЕ. РАБОТА С БИБЛИОТЕКИ

План

1. **Същност**
2. **Създаване на dll файлове**
3. **Примери**

1. Същност

DLL е съкращение от **Dinamic Link Library** (библиотека за динамично свързване). Те са библиотечни файлове с компилиран код, който може да се употребява в други приложни програми.

Тези модули приличат на библиотечните (LIB) по това че също съдържат определени функционални възможности, пакетирани за употреба в приложните програми. Различията се изразяват при свързване на приложната програма към библиотеката. При обикновените библиотечни модули (LIB) програмата се свързва с функционалните възможности по време на компилиране и изграждане. Всички функционални възможности от библиотечния файл се превръщат в **част от изпълнимия файл на програмата.** При DLL файла приложната програма се свързва се свързва към функционалните възможности на библиотечния файл по време на своето изпълнение. **Библиотеката остава самостоятелен файл,** към който програмата извършва обръщения.

Предимства на DLL файла пред LIB:

Употребата на DLL влияе пряко върху големината на изпълнимия файл, защото в него не се включват функционалните възможности пакетирани в библиотеката, а те от своя страна могат да се използват от различни приложни програми.
Може да актуализирате и модифицирате функционалните възможности на DLL, без да се налагат каквито и да е промени или допълнителни обработки в изпълнимия файл на програмата (разбира се, ако не коригирате експортния интерфейс на DLL).Възможност да използвате DLL с всеки от езиците за програмиране под WINDOWS.

Създаване и употреба на DLL

Библиотеките DLL предлагат определени функции и класове за приложните програми посредством експортиране на функции. Когато дадена функция се експортира, тя се добавя към таблица, включена в DLL. В тази таблица се съдържа списък на всички експортирани функции, съдържащи се в DLL и тя се използва за намиране и извикване на всяка от тях. Всяка функция която не е експортирана, не се добавя в таблицата и поради тази причина тя не може да бъде

видяна или извикана по какъвто и да е начин от външна програма или DLL. Програмите могат да извикват функции от DLL по два способа:

При по неясния от тях се намира мястото в DLL на нужната функция и се получава указател към нея. След това той се използва за обръщение към функцията. При другия се осъществява свързване на приложната програма към файл LIB, създаден с DLL. Файлът LIB се третира от свързващия редактор като стандартен библиотечен файл. Файлът LIB съдържа тапи (stubs) за всяка от експортираните функции в DLL. Тапата представлява псевдо-функция, която притежава наименование и списък с аргументи, еднакви с тези на действителната функция. В тялото на функцията заглушалката е ограничено количество код, който извиква действителната функция от DLL, сякаш тя е част от кода на приложната програма и не е обособена в отделен файл.

2. Създаване на ***.dll файлове**

Класовете са елементи на обектно-ориентираното прграмиране. Всеки клас представлява шаблон за обект, но сам по себе си не е обект. Той дефинира процедури и структури от данни, но никакви конкретни стойности за данните. Например класът "ученик" представя множеството от всички студенти. Класът не съществува реално като физическа същност, а по-скоро можем да го разгледаме като описание на неговите обекти.

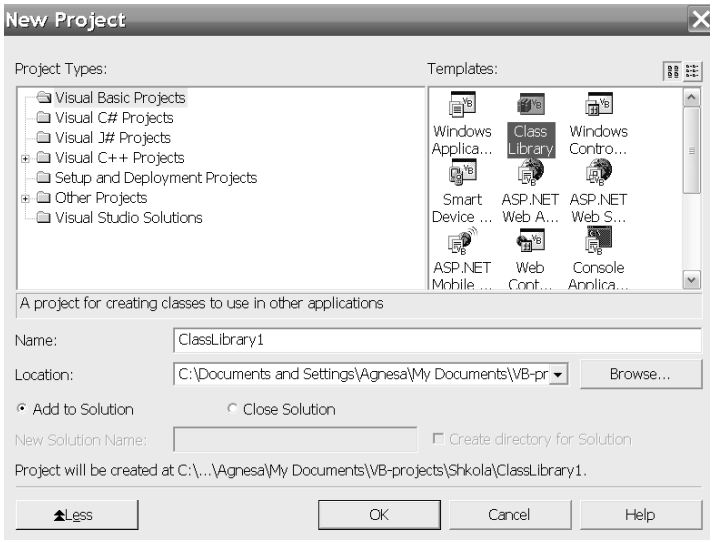
Обектът е една конкретна инстанция на клас, софтуерен модул, който групира всички елементи от данни и процедури, ноебходими за представянето на някакви реални или абстрактни елементи. Те съдържат стойности на данни, които са уникални за конкретна инстанция. Обикновено стойностите за данните, които принадлежат на всеки обект от даден клас, ще се различават. Обаче всички обекти от един и същи клас съдържат един и същи тип данни. Например обект е ученикът Тодор Георгиев, 11 клас, професия Програмист.

Библиотеките се създават като класове, които могат да се използват от други проекти на Visual Basic.Net. За целта те не се компилират като изпълними файлове (с разширение **exe**), а като библиотечни файлове с разширение **dll.**

Има два основни начина на съсдаване на библиотечни файлове.

Първи начин – създаване на нов библиотечен проект като клас.

Стартира се нов проект като се използва шаблон (template) **Class Library,** както е показано на фигура 1:



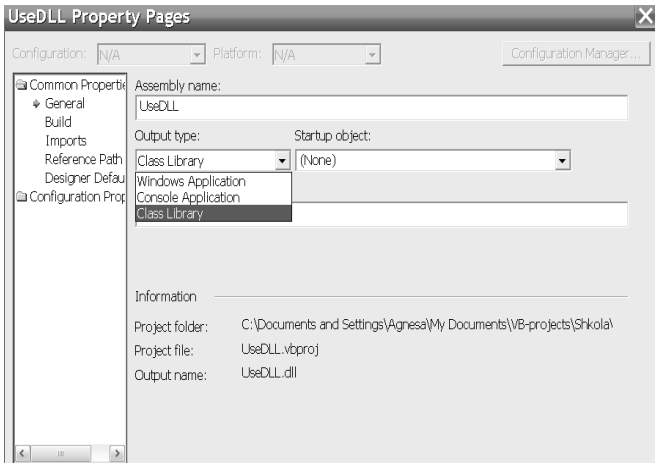
Фиг.1 Създаване на **Class Library** чрез шаблон

Може да се добави нов модул на клас във вече съществуващ проект като се използва меню File-> Add New Item и в появилия се диалогов прозорец, който е подобен на този от фиг.1, да се избере шаблона(template) **Class.**

Втори начин – създаване на съществуващ проект като библиотечен клас

За целта се отваря страницата със свойствата на проекта от меню

Project-><име на проект>Properties, избор **General**, в поле **Output Type** се избира **Class Library** както е показано на фигура 2:



Фиг.2 Създаване на съществуващ проект като библиотечен клас.

И при дадата начина за създаване на библиотека проекта не трябва да съдържа форма. За разлика от обикновените проекти, библиотечните не се стартират , а само се изграждат , защото са създадени да се стартират от други програми, а не самостоятелно. За целта, се избира командата **Build Solution** от меню **Build.** След това, ако проверите в директория **Bin** на проекта, имате файл с разширение **dll** т.е. вече създаден библиотечен файл.

След като сте създали вече библиотечен клас, трябва да отворите редактор на кода му (*Code Editor – фиг. 3*), за да въвеждате елементите му.

Фиг.3 *Code Editor* – прозорец за въвеждане елементи на клас.

Там е зададена рамката за създаване на елементите:

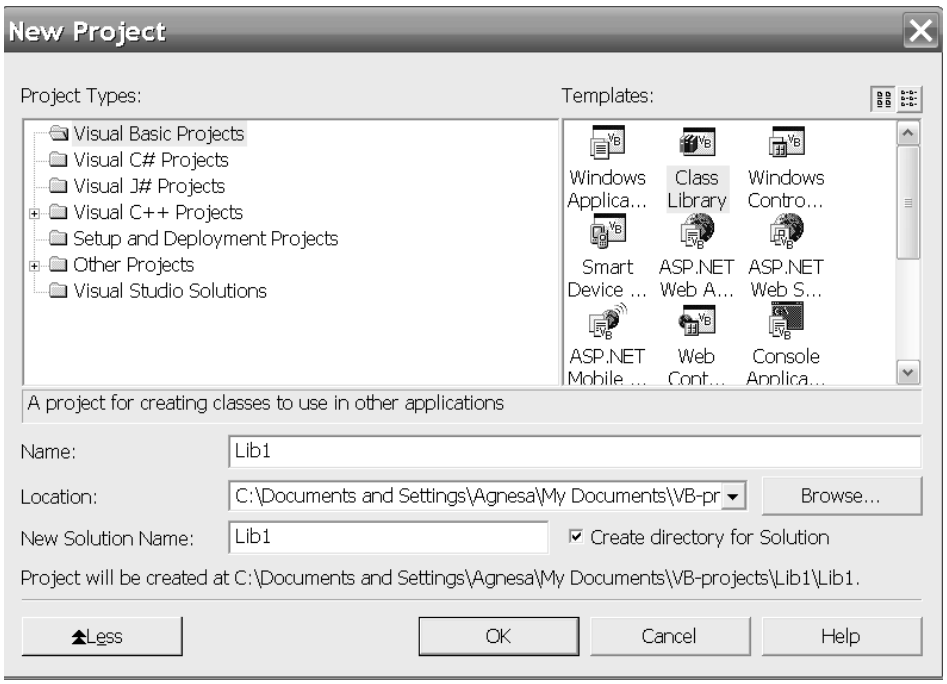
Public Class Class1

End Class

Може да променяте името на класа. В прозореца за кода ако смените в реда на дефиницията на класа *Public Class Class1* името му и ако погледнете в прозореца *Solution Explorer*, ще видите, че името на файла все още е *Class1.vb*. Извикайте с десен бутон контекстното меню в *Solution Explorer* и преименувайте с *Rename* с желаното от вас име. Така се сменя името на файла, който съдържа един или повече класове. По-късно ще обясним елементите на класа, затова сега ще се задоволим с използването само на функции и подпрограми в него.

<u>Пример</u>: Да се създаде библиотека, като проекта се именува *Lib1*. След това класа и файлът му да се преименуват на *ShowMe.* Да съдържа функция *M(),* която извежда в диалогов прозорец, съобщението подадено ѝ като аргумент, с икона за информация.

1-ва стъпка: Създаване на библиотечен клас.



2-ра стъпка: Преименуване на *ShowMe.*

3-та стъпка: Написване на функцията

```
Public Class ShowMe
Public Sub M(ByVal txt As String)
      MsgBox(txt, MsgBoxStyle.Information, "Аз съм библиотека")
   End Sub
End Class
```

4-та стъпка: Изграждане на библиотеката с командата **Build Solution** от меню **Build**.

3. Използване на библиотеки .

За да използвате вашата библиотека, трябва да добавите обръщение (**reference**) към нея като в *Solution Explorer*, дясно кликване върху *References* върху името на проекта, избера се *Add Reference.* Появява се диалоговия прозорец *Add Reference*. Клик върху страница *Projects* и можете да намерите чрез бутона *Browse* желаната библиотека. Кликнете бутон *Select* за да преместите желаната библиотека в кутия *Selected Components*, и след това бутон OK.

След добавянето на обръщение към библиотеката могат да се дефинират обекти от нейните класове като се използва ключовата дума **New**. Дефинирането на обект е всъщност дефиниране на променлива, но от тип клас. Ключовата дума **New** осигурява зареждане на класа в паметта и задава име на този клас. Например, ако трябва да дефинираме обект **O** от класа на нашата библиотека *Lib1*, трябва да напишем следния програмен ред:

Dim O As New Lib1.ShowMe

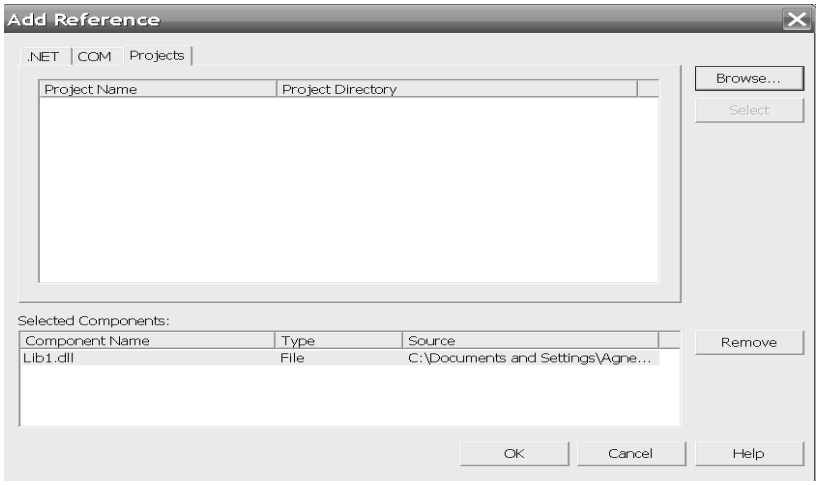
Понеже библиотеката е външен източник за нашия проект първо се цитира името на библиотеката, точка и името на класа от нея, който използваме.

Пример: Да се създаде нов проект с име *Program1*. В него да се добави обръщение към библиотеката от проект *Lib1*. Във формата да се добави бутон със заглавие "*Библиотека*", при натискането, на който се извиква функцията M() с текст "*Изикване на библиотека*".

1-ва стъпка: Създаване на проекта *Program1*.

2-ра стъпка: Добавяне обръщение към библиотеката от проект *Lib1*.

На фигурата е визуализиран прозореца *Add Reference* с вече добавено обръщение.



3-та: стъпка: Създаване на бутона и генериране на процедурата по натискането му.

4-та стъпка: Написване на следния код:

```
Private Sub Button1_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles Button1.Click
    Dim O As New Lib1.ShowMe
    O.M("Извикване на библиотека")
End Sub
```

5-та стъпка: Стартиране и теставне на проекта.

Задача 1: Да се създаде библиотека **S2.** Класът на библиотеката да се преименува на **Lica** и да съдържа следните функции пресмятащи лицата на равнинни фигури:

- STriang (a, ha) – връща лице на триъгълник по зададени страна и височина към нея.
- SRect(a, b) - връща лице на правоъгълник по зададени широчина и дължина.
- SParal(a, ha) - връща лице на успоредник по зададени страна и височина към нея.
- STrap(a, b, h) - връща лице на трапец по зададени основи и височина.
- SCirc(r) - връща лице на кръг по зададен радиус.

Да се създаде проект **Surface**, който използва библиотеката **S2**. Формата да е със заглавие *"Лица на равнинни фигури"*. В нея да се поставят следните бутони, съответно със заглавие и действие:

- "**Триъгълник**" (**btnSTriang**) – при натискането му чрез диалогови кутии се въвеждат страна и височина към нея за триъгилник и се пресмята лицето му като се извиква функцията **STriang**() от библиотеката. Резултата се извежда с диалогова кутия.
- "**Правоъгълник**" (**btnSrect**)- при натискането му чрез диалогови кутии се въвеждат широчината и дължината на правоъгълник и се пресмята лицето му като се извиква функцията **Srect**()от библиотеката. Резултата се извежда с диалогова кутия.
- "**Успоредник"** (**btnSParal**)- – при натискането му чрез диалогови кутии се въвеждат страна и височина към нея за успоредник и се пресмята лицето му като се извиква функцията **SParal**() от библиотеката. Резултата се извежда с диалогова кутия.
- "**Трапец"** (**btnSTrap**)- при натискането му чрез диалогови кутии се въвеждат основите и височината на трапец и се пресмята лицето му като се извиква функцията **STrap**() от библиотеката. Резултата се извежда с диалогова кутия.
- "**Кръг**" (**btnSCirc**)- при натискането му чрез диалогови кутии се въвеждат и се пресмята лицето му като се извиква функцията **SCirc**() от библиотеката. Резултата се извежда с диалогова кутия.

Задача 2: Като в задача 1 да се направи библиотека **V** за обеми на познати тела и да се използва в по подобен начин в проект.

Примерно решение на задача 1:

```
Public Class Lica
    Function STriang(ByVal a As Double, ByVal ha As Double) As Double
        STriang = a * ha / 2
    End Function
    Function SRect(ByVal a As Double, ByVal b As Double) As Double
        SRect = a * b
    End Function
    Function SParal(ByVal a As Double, ByVal ha As Double) As Double
        SParal = a * ha
    End Function
    Function STrap(ByVal a As Double, ByVal b As Double, ByVal h As Double) As Double
        STrap = (a + b) * h / 2
    End Function
    Function SCirc(ByVal r As Double) As Double
        SCirc = 3.14 * r * r
    End Function
End Class
Private Sub btnSTriang_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles btnSTriang.Click
        Dim STr As New S2.Lica
        Dim a, ha, S As Double
        a = CDbl(InputBox("Въведете страна", "Лице на триъгълник"))
        ha = CDbl(InputBox("Въведете височина", "Лице на триъгълник"))
        S = STr.STriang(a, ha)
        MsgBox("Лицето на триъгълника е: " & S)
    End Sub
    Private Sub btnSrect_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles btnSrect.Click
        Dim SR As New S2.Lica
        Dim a, b, S As Double
        a = CDbl(InputBox("Въведете широчина", "Лице на правоъгълник"))
        b = CDbl(InputBox("Въведете дължина", "Лице на правоъгълник"))
        S = SR.SRect(a, b)
        MsgBox("Лицето на правоъгълника е: " & S)
    End Sub
```

```
Private Sub btnSParal_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles btnSParal.Click
    Dim SP As New S2.Lica
    Dim a, ha, S As Double
    a = CDbl(InputBox("Въведете страна", "Лице на успоредник"))
    ha = CDbl(InputBox("Въведете височина", "Лице на успоредник"))
    S = SP.SParal(a, ha)
    MsgBox("Лицето на успоредника е: " & S)
End Sub

Private Sub btnSTrap_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles btnSTrap.Click
    Dim ST As New S2.Lica
    Dim a, b, h, S As Double
    a = CDbl(InputBox("Въведете дължината на малката основа", "Лице на трапец"))
    b = CDbl(InputBox("Въведете дължината на голямата основа", "Лице на трапец"))
    h = CDbl(InputBox("Въведете височина", "Лице на трапец"))
    S = ST.STrap(a, b, h)
    MsgBox("Лицето на трапеца е: " & S)
End Sub

Private Sub btnSCirc_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles btnSCirc.Click
    Dim SC As New S2.Lica
    Dim r, S As Double
    r = CDbl(InputBox("Въведете радиус", "Лице на кръг"))
    S = SC.SCirc(r)
    MsgBox("Лицето на кръга е: " & S)
```