

Windows Forms като потребителски интерфейс



Учителски екип

Обучение за ИТ кариера

<https://it-kariera.mon.bg/e-learning/>



Съдържание

1. Използване на Windows Forms (Winforms) за презентационен слой
2. Разделяне на приложението на проекти според слоя



Какво е Winforms?

- Winforms е графичен потребителски интерфейс
 - Използва контроли за създаване на интерфейса
 - Чудесен визуален интерфейс за нашето приложение
- Благодарение на Winforms ние ще дадем добър и удобен външен вид на нашето CRUD приложение

Form1

Name

Price

Stock

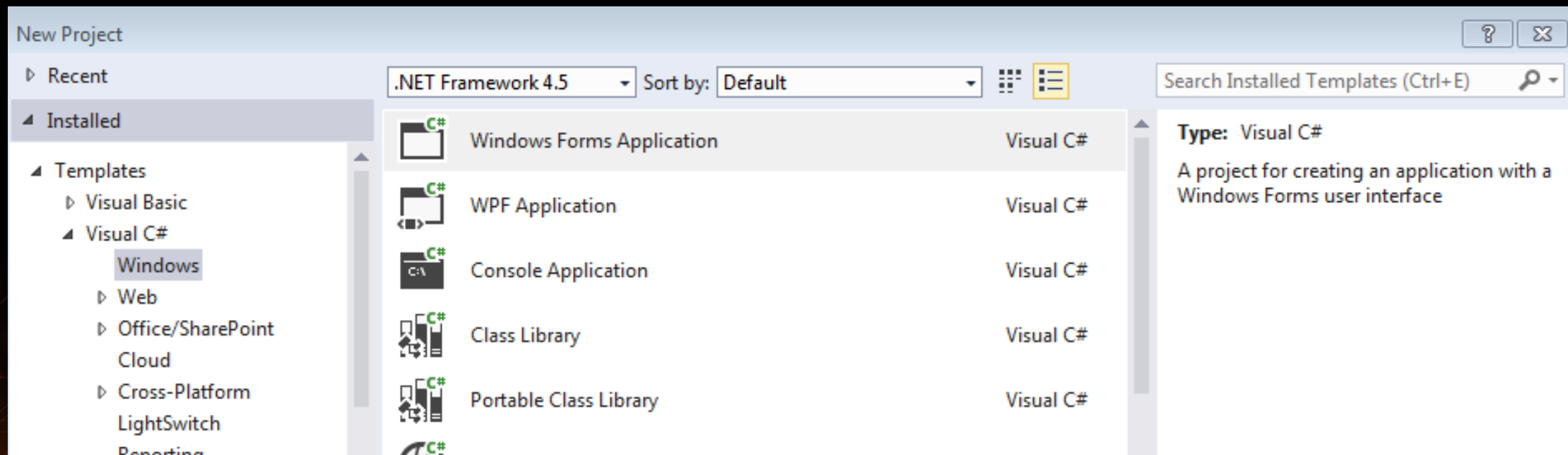
	Id	Name	Price	Stock
▶	2	Test222	100,00	2
	4	Test333	1000,00	8
	6	test	100,00	200

Структура на проекта ни
Изграден върху Winforms

Какви проекти трябва да имаме?

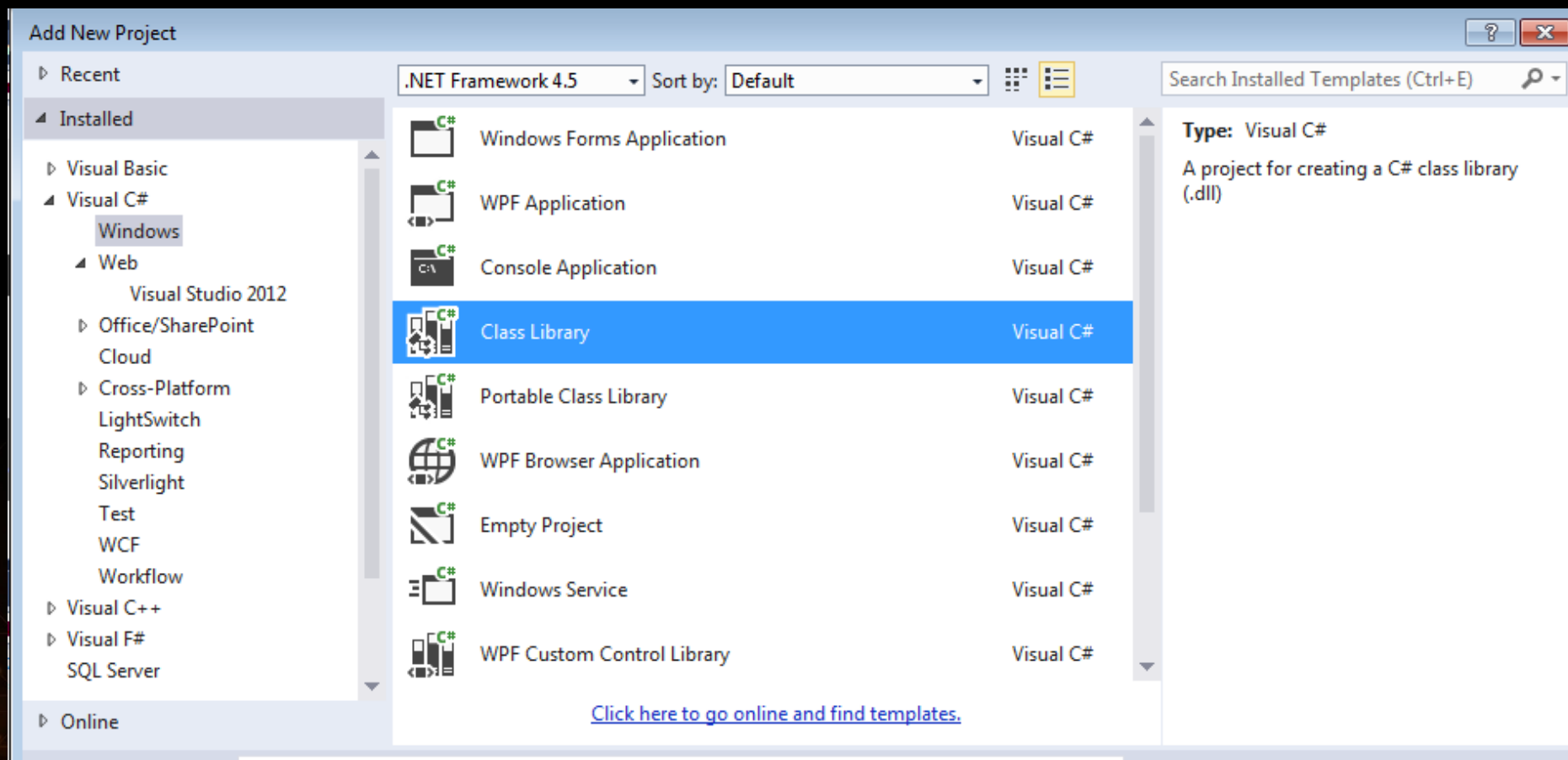
- Windows Forms Application

- Това ще е проекта ни, който ще действа като презентационен слой



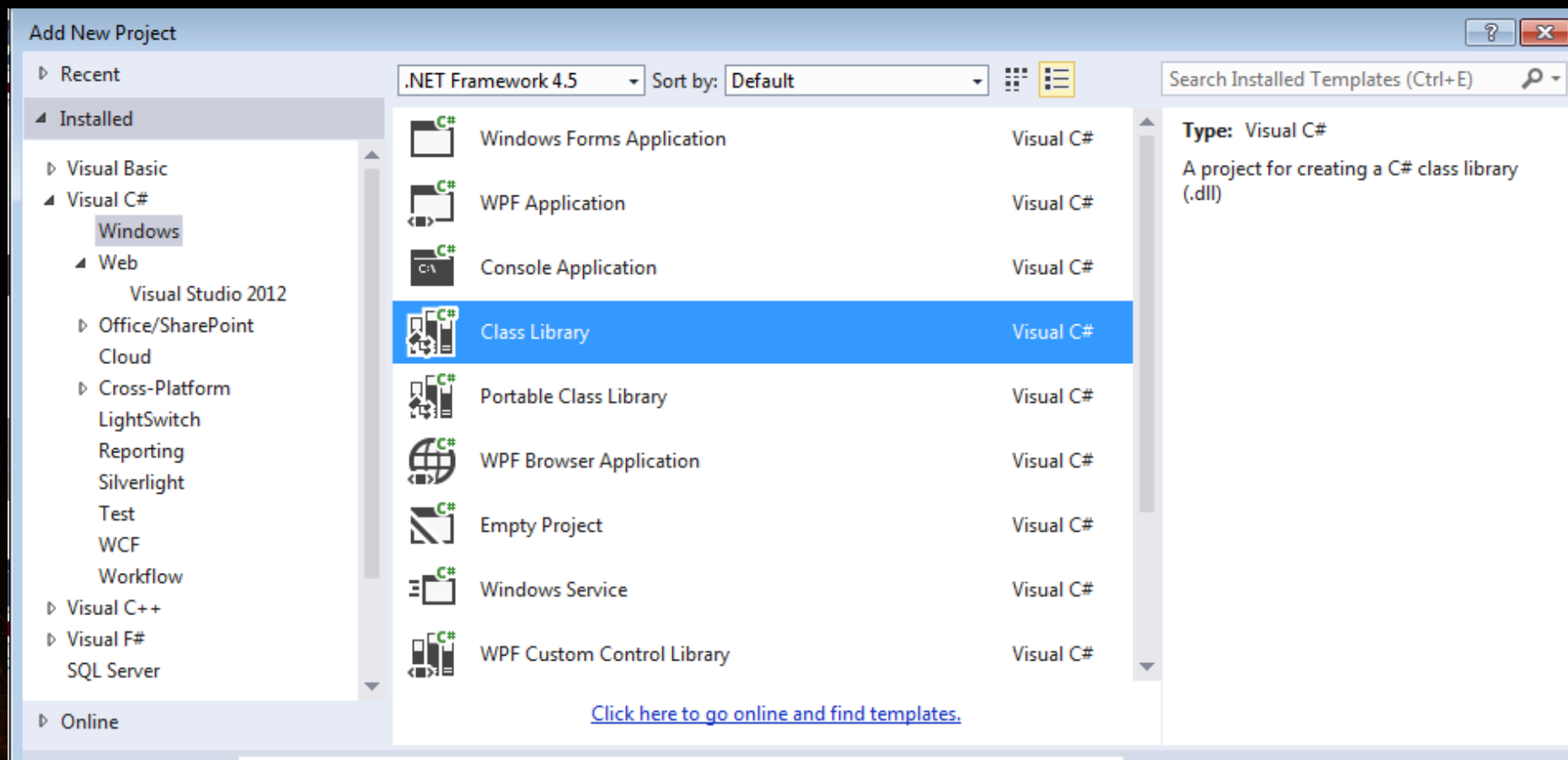
Какви проекти трябва да имаме? (2)

- **Data** - това ще е проекта ни, който ще бъде слоен за данни. При създаването му от Visual C# -> Windows трябва да изберем **Class library**



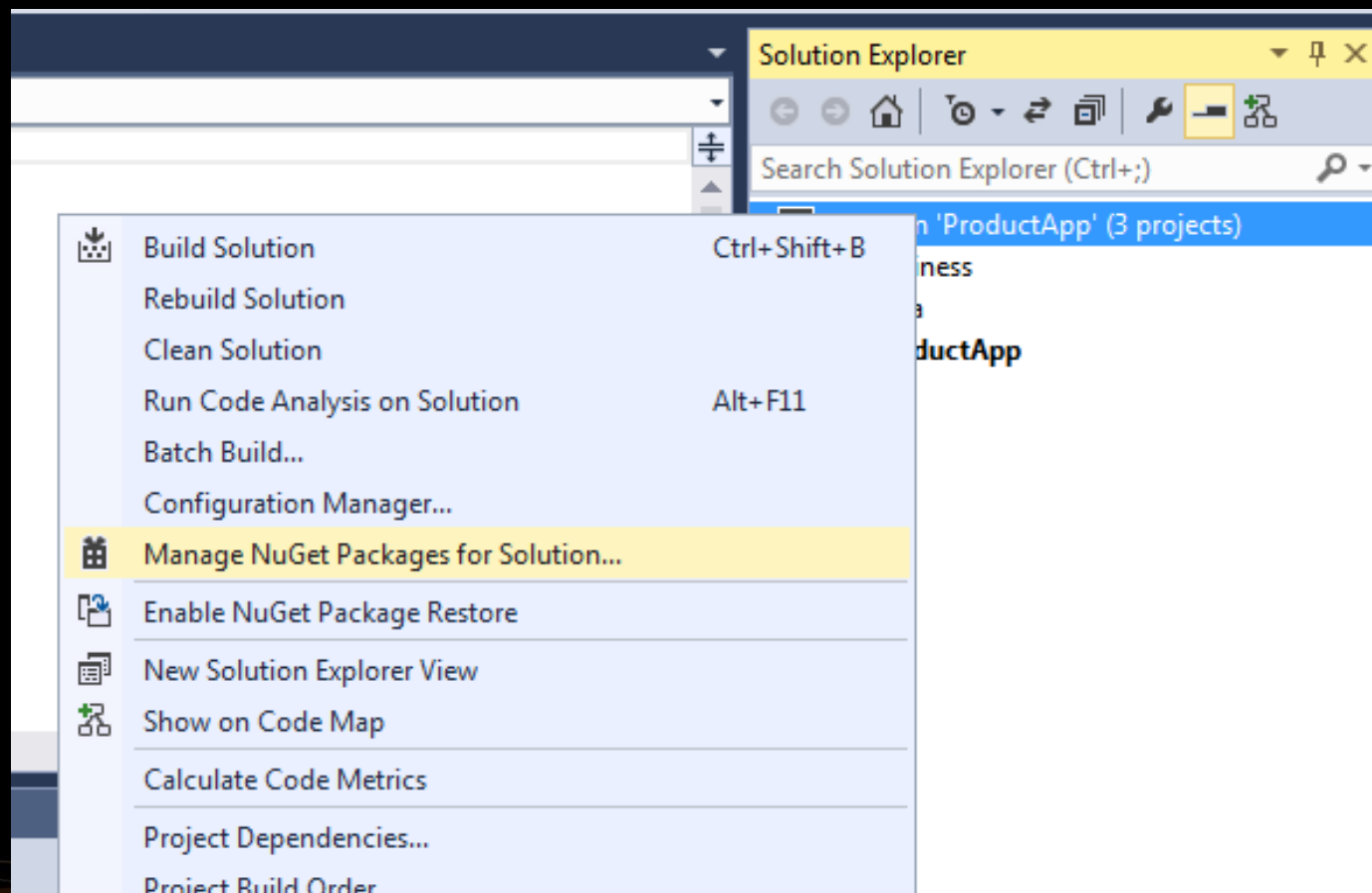
Какви проекти трябва да имаме? (3)

- **Business** - това ще е проекта ни, който ще бъде слоен за услуги. При създаването му от Visual C# -> Windows трябва да изберем **Class library**



Добавяне на EntityFramework

- Цъкнете с десен бутон върху solution-а и изберете Manage NuGet Packages for Solution и инсталирайте EntityFramework. Изберете инсталация за всичките проекти в solution-а.



Добавяне на референции

- На проект **Business** изберете [Add]->References. Добавете референция към **Data**
- На проект **ProductApp** /проектът на Winforms/ изберете [Add] -> References. Добавете референция към **Business** и **Data**

Data/Model/Product.cs

- Създайте подпапка Model в Data, а в нея модел клас **Product.cs**

```
public class Product
{
    public int Id { get; set; }
    public string Name { get; set; }
    public decimal Price { get; set; }
    public int Stock { get; set; }
}
```

Data/ProductContext.cs

- Създайте **ProductContext** клас, който наследява **DbContext**. Ще има нужда да добавите и **using** директива:

```
using System.Data.Entity;

public class ProductContext : DbContext
{
    public ProductContext()
        : base("name=ProductContext")
    {
    }
    public DbSet<Product> Products { get; set; }
}
```

Business/ProductBusiness.cs

- Създаваме управляващо поле от клас **ProductContext**:

```
private ProductContext context;
```

- Създаваме методи, които извършват желаните операции, извиквайки свойството **Products** и методи върху него, познати ни от **LINQ**:

```
public List<Product> GetAll(){  
    using (productContext = new ProductContext())  
    {  
        return productContext.Products.ToList();  
    }  
}
```

- По аналогичен начин реализираме другите методи

Свързване на Winforms с другите слоеве

- В **App.config** се добавя познатия ни низ за връзка:

```
<connectionStrings>  
  <add name="ProductContext" connectionString="Data  
Source=.; Initial Catalog=ProductDb; Integrated Security=true"  
providerName="System.Data.SqlClient"/>  
</connectionStrings>
```

- Освен това трябва да създадем поле в **ProductForm** класа от клас **ProductBusiness**:

```
private ProductBusiness productBusiness = new ProductBusiness();
```

Компоненти във формата

- Добавете следните компоненти:
 - 3 броя **Label**: **lblName**, **lblPrice**, **lblStock** – етикети съответно за име, цена и бройки
 - 3 броя **TextBox**: **txtName**, **txtPrice**, **txtStock** – текстови кутии съответно за име, цена и бройки
 - 1 брой **DataGridView**: **dataGridView1** – табличка за визуализиране на данни
 - 4 бутона: **btnInsert**, **btnUpdate**, **btnSave**, **btnDelete**, като **btnUpdate** и **btnSave** трябва да бъдат един върху друг

Методи, които трябва да реализираме

- `private void UpdateGrid()` – обновява табличката
- `private void ClearTextBoxes()` – изчиства текстовите кутийки
- `private void ToggleSaveUpdate()` – разменя Update/Save
- `private void UpdateTextboxes(int id)` – задава данните за редактиране в текстовите кутийки
- `private void DisableSelect()` – забранява възможността за избиране на ред от табличката по време на редакцията
- `private void ResetSelect()` – позволява възможността за избиране на ред от табличката, деселектирайки досега избрания ред
- `private Product GetEditedProduct()` – връща обект от клас Product на базата на данните попълнени в текстовите кутийки

Методи, които трябва да реализираме

- `private void ProductForm_Load(object sender, EventArgs e)` – зарежда данни в табличката при зареждане на форма
- `private void btnInsert_Click(object sender, EventArgs e)` – предава за добавяне информация към бизнес логиката
- `private void btnUpdate_Click(object sender, EventArgs e)` – задава за редакция избрания ред в таблицата, ако има такъв
- `private void btnSave_Click(object sender, EventArgs e)` – запазва редакцията чрез подаване на информацията към бизнес логиката
- `private void btnDelete_Click(object sender, EventArgs e)` – изтрива избрания ред от табличката

Използване на приложението

Form1

Name

Price

Stock

	Id	Name	Price	Stock
▶	2	Test222	100,00	2
	4	Test333	1000,00	8
	6	test	100,00	200

Winforms като потребителски интерфейс



Въпроси?



Министерство на образованието и науката (МОН)

- Настоящият курс (презентации, примери, задачи, упражнения и др.) е разработен за нуждите на Национална програма "**Обучение за ИТ кариера**" на МОН за подготовка по професия "Приложен програмист"



Министерство
на образованието
и науката



Национална
програма
„Обучение за
ИТ кариера“

- Курсът е базиран на учебно съдържание и методика, предоставени от **фондация "Софтуерен университет"** и се разпространява под свободен лиценз **CC-BY-NC-SA**



SoftUni
Foundation

