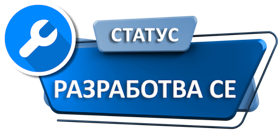
# Упражнение: GUI приложения и Windows Forms

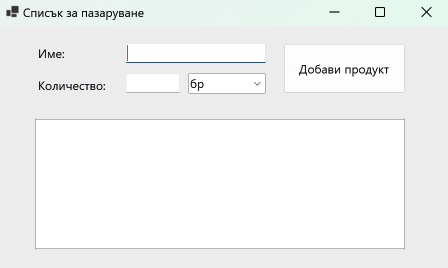
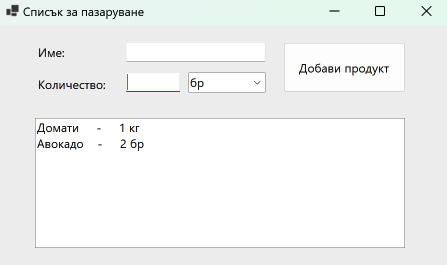
Състояние на настоящия учебен материал:



## Списък за пазаруване

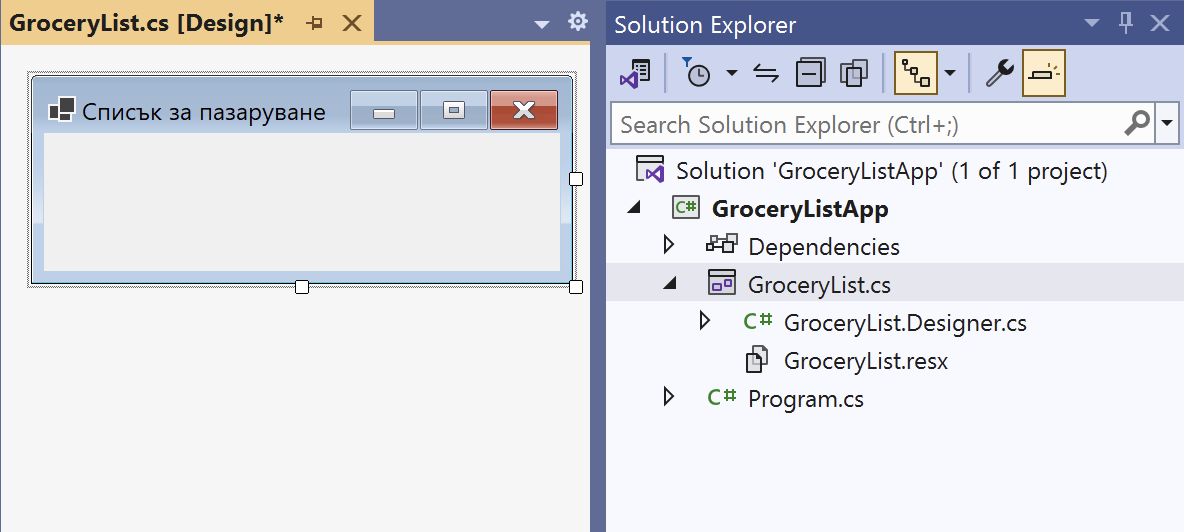
Създайте **Windows Forms** приложение, което да добавя **продукти** за пазаруване и да ги показва в **списък**. Всеки продукт има **име**, **количество** и **мерна единица** (грам, литър, брой…). При натискане на бутон за добавяне, продуктът трябва да се добави и покаже в списъка за пазаруване.

### Пример

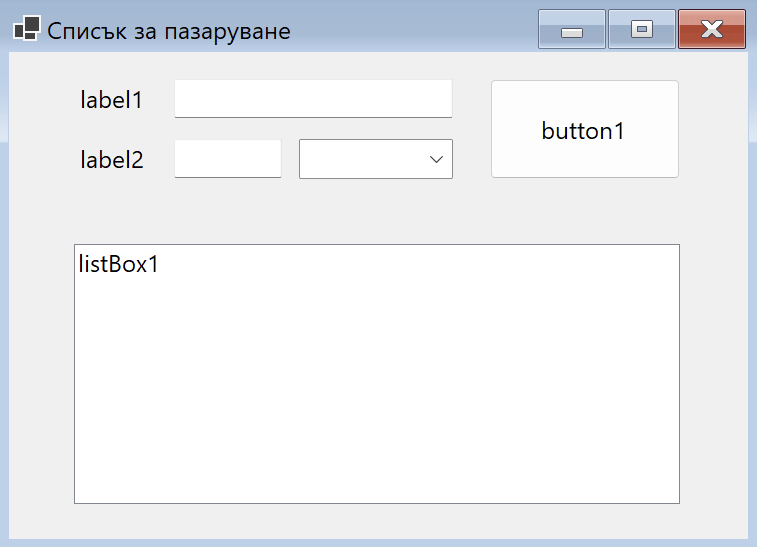


### Насоки

1. Създаваме нов **Windows Forms** **проект** и му задаваме **подходящо име**, например "**GroceryListApp**". Променяме и **името на формата**, за да описва нейното предназначение, например "**GroceryList**". Следва да променим и **заглавието на формата** – "**Списък за пазаруване**".



1. Добавяме необходимите **контроли** – **две текстови полета** (TextBox), **два надписа** (Label), **един падащ списък** (ComboBox), **един бутон** (Button) и **един списък** (ListBox).

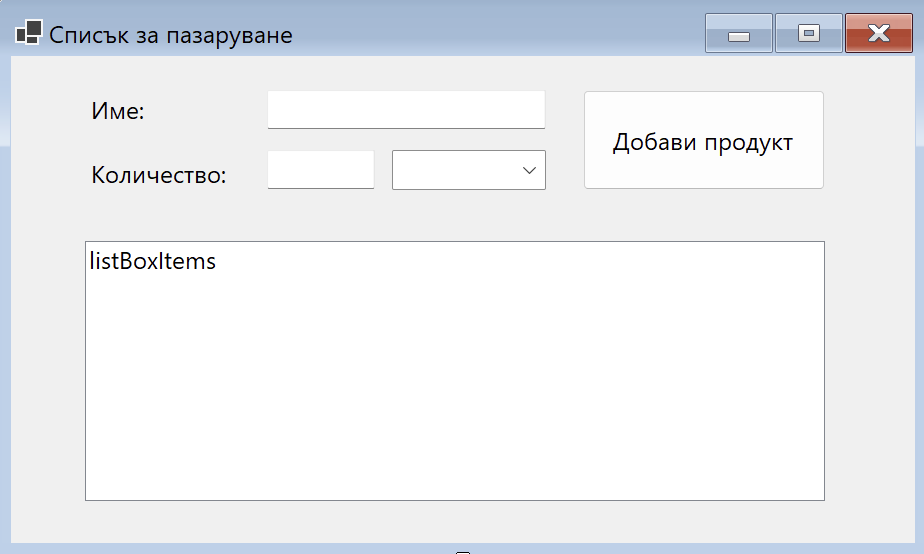


1. Променяме **имената** на контролите:

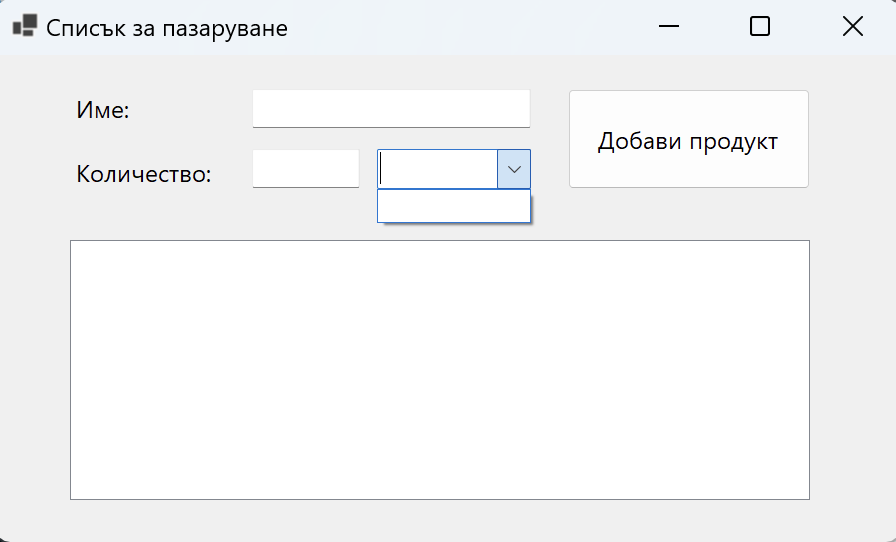
* label1 🡪 labelItemName
* label2 🡪 labelItemAmount
* textBox1 🡪 textBoxItemName
* textBox2 🡪 textBoxItemAmount
* comboBox 🡪 comboBoxItemUnits
* button1 🡪 buttonAddItem
* listBox1 🡪 listBoxItems

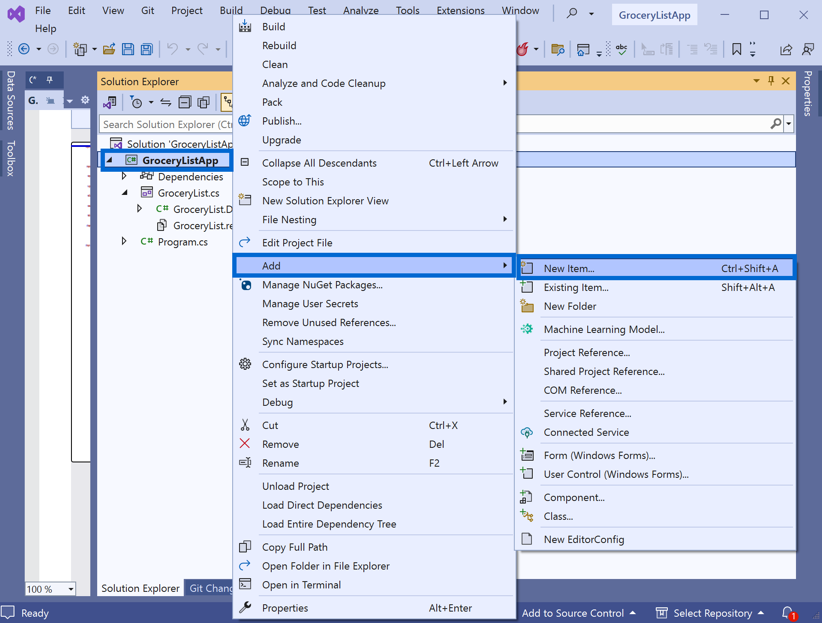
1. Променяме **текста** на контролите:

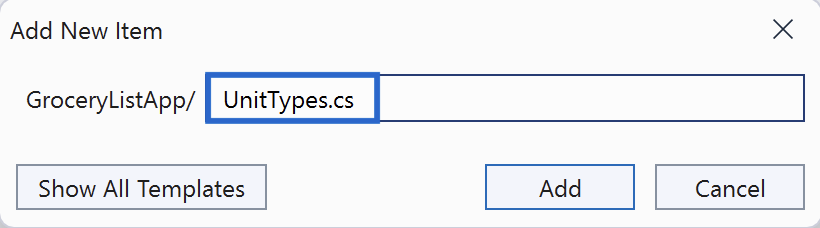
* label1 🡪 Име:
* label2 🡪 Количество:
* button1 🡪 Добави продукт



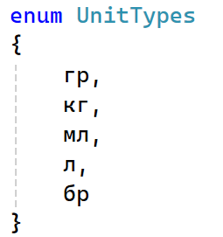
1. Пускаме приложението с **[Ctrl+F5]**. Трябва да тръгне **без грешки** и с **празни стойности**.



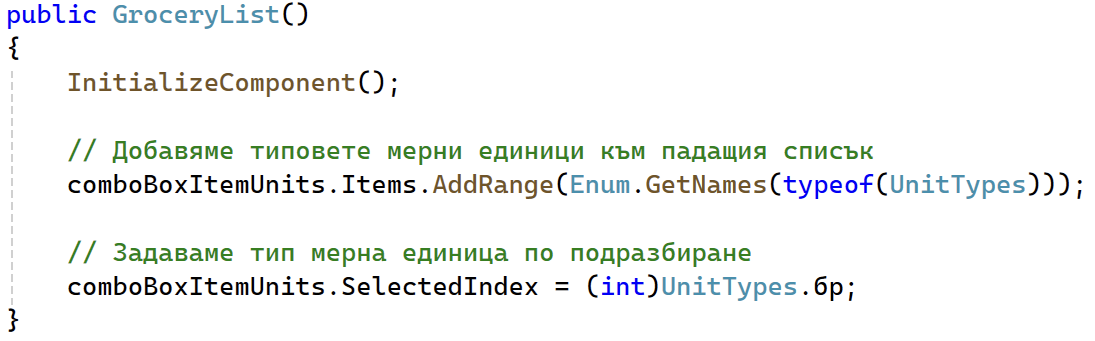
1. Добавяме **опции** за **падащия списък** (ComboBox). Създаваме нов файл с подходящо име, например "**UnitTypes**", който ще пази **енумерация** от **мерни** **единици**.



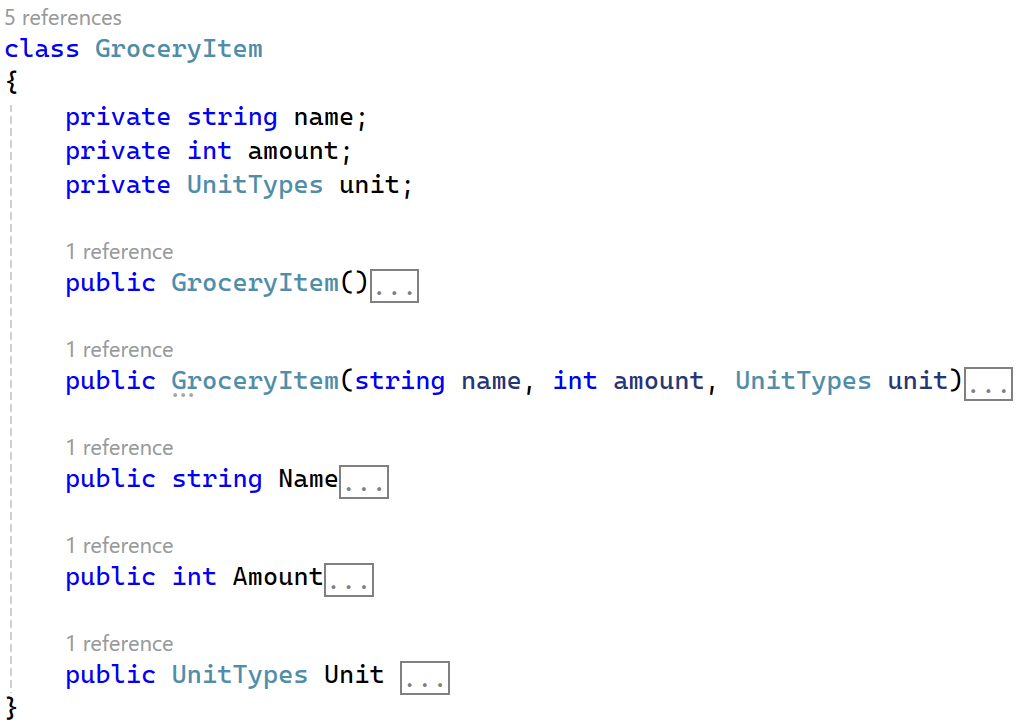
1. Изреждаме възможните **опции**.



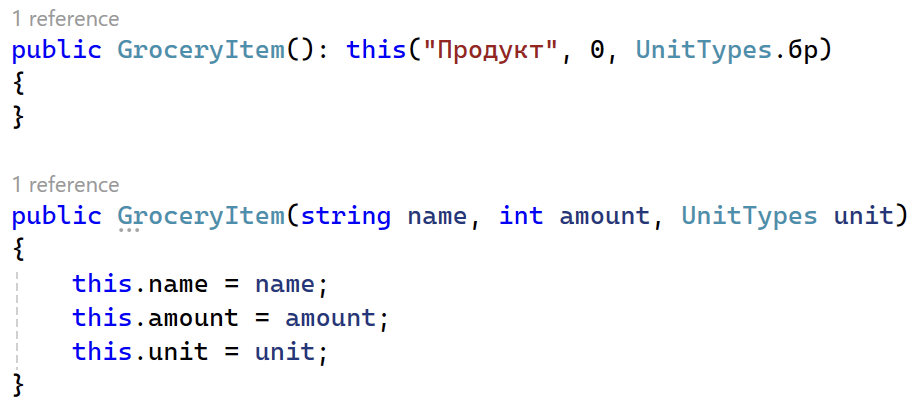
1. Добавяме **опциите** за мерна единица да се **визуализират** при **стартиране** на приложението. Във файла **GroceryList.cs**:



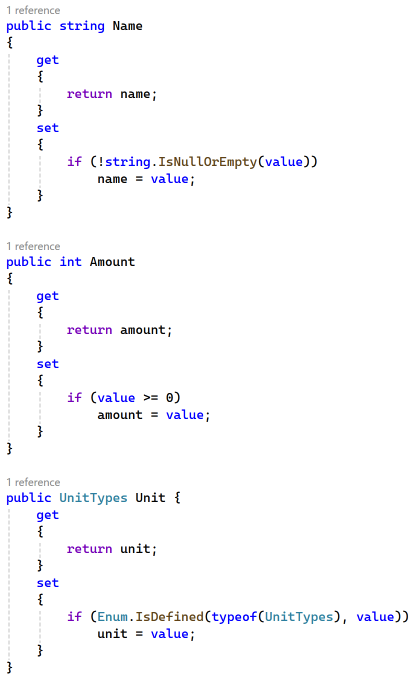
1. Създаваме клас **GroceryItem**, който ще съдържа всички свойства за продукт – **име**, **количество**, **тип мерна единица**.



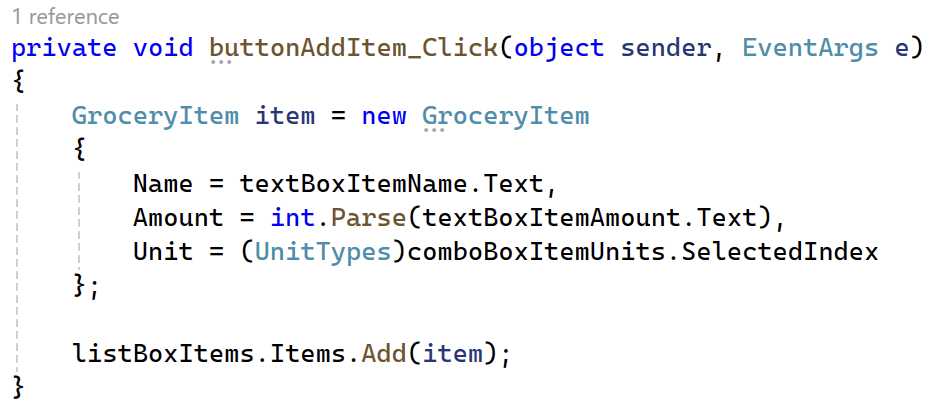
1. Създаваме **конструкторите** на класа – един **по подразбиране** и един, който **взима** подадените **стойности**.



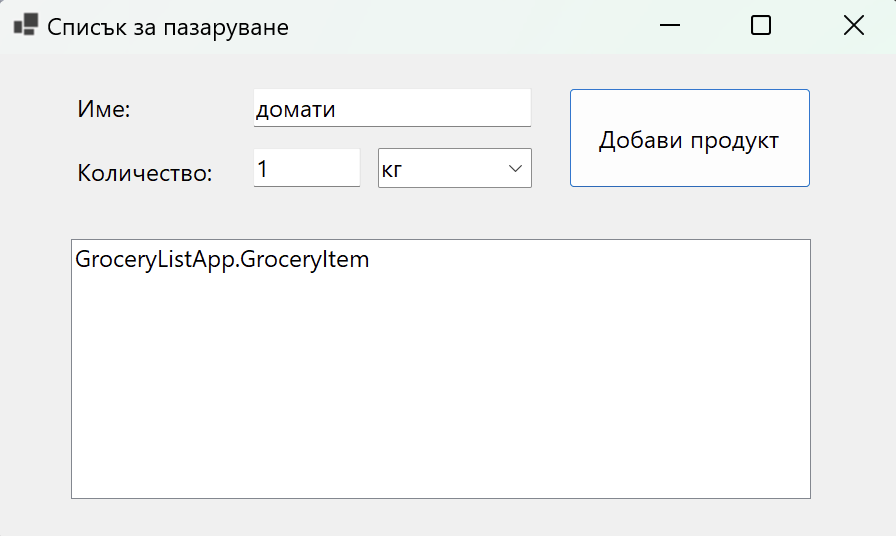
1. Имплементираме публичните **пропъртита** с **гетъри** и **сетъри**.



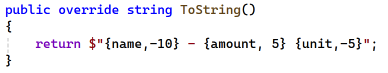
1. Добавяме **метод-обработчик** на бутона с име **buttonAddItem\_Click** на събитието **Click** на бутона за добавяне на продукт.
2. Записваме **подадените данни** към **нов продукт** и го **добавяме** към **списъка** **с продукти**.



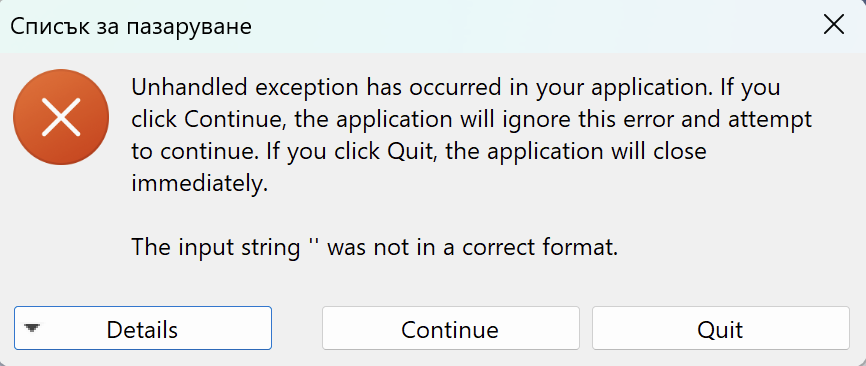
1. Сега **добавянето работи**, но визуализирането на продукти все още не.



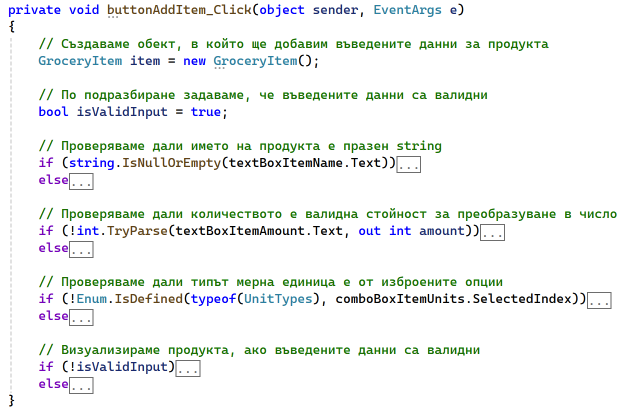
Променяме това като **пренапишем** (override) метода **ToString()** на класа **GroceryItem**.



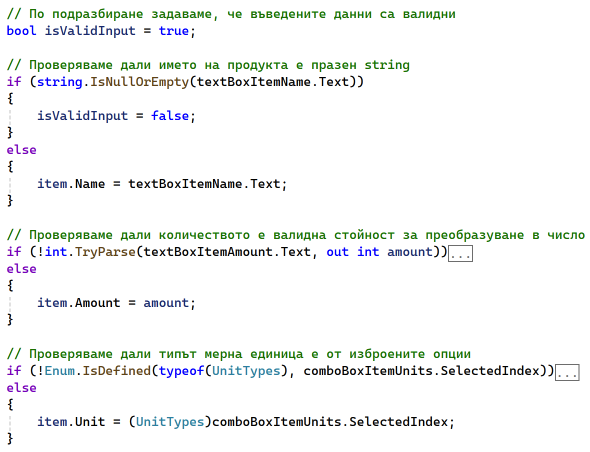
1. Ако се опитаме да добавим **празен** **продукт** или с **невалидни** **стойности**, програмата ще ни покаже **грешка**.



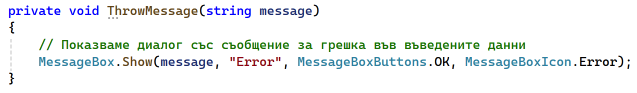
1. Добавяме **проверки**, за да не позволим на програмата да записва **празни** или **невалидни** **данни**.



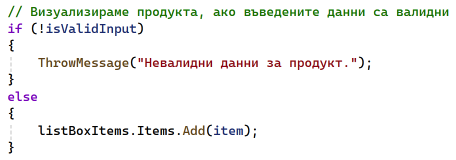
1. На всяка **проверка**, която **не е** каквото **програмата очаква**, задаваме **isValidInput = false;**



1. Ако въведените данни не са валидни, показваме диалог със съобщение за грешка.



1. Ако всички данни са **валидни**, може да **добавим** новия **продукт** към **списъка** за пазаруване.



1. Нашето приложение работи **успешно**.

