ДОКЛАД

По Функционално програмиране

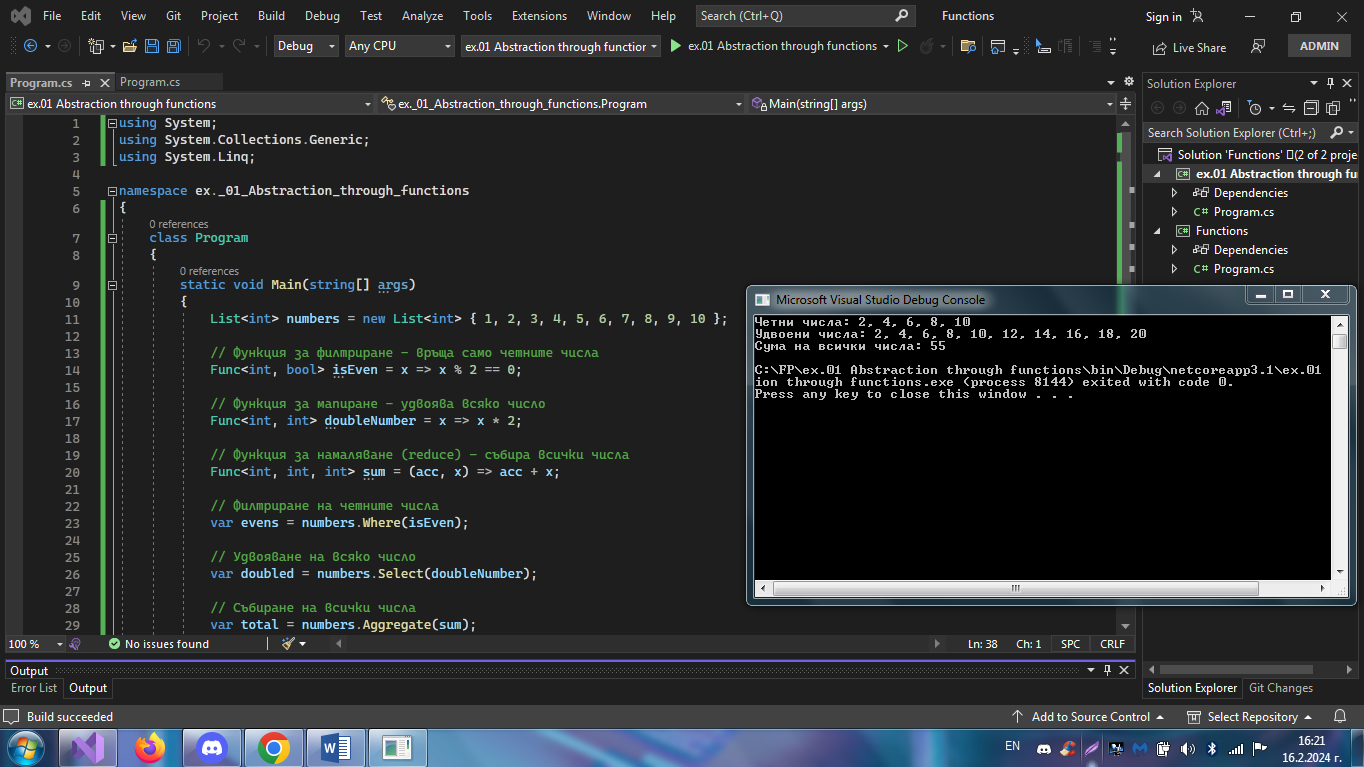
Тема: Абстракции чрез функции

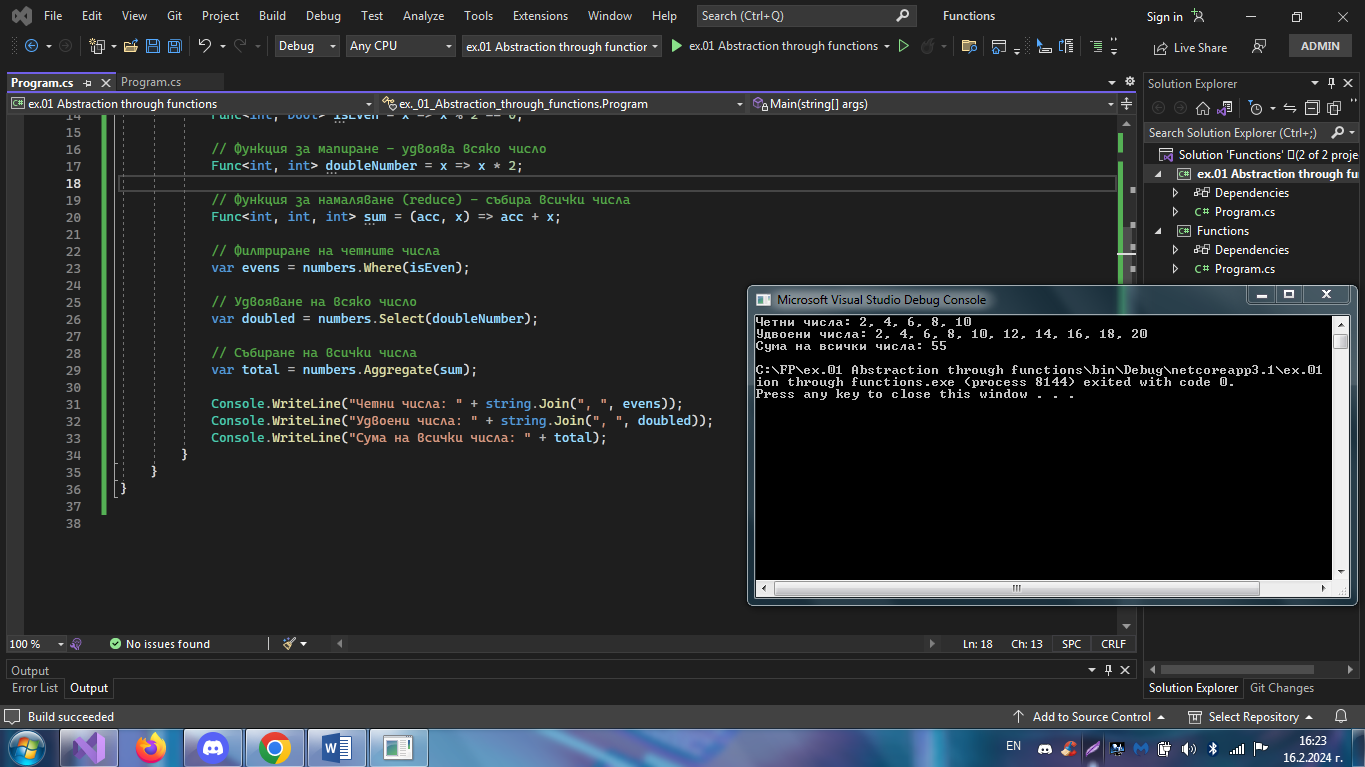
Изготвил: Валентин Кьосев 12 ,,а‘‘ клас

1. Абстракции чрез функции
2. Изчисления върху списъци
3. **Абстракции чрез функции**

* Абстракция чрез функции в C# се отнася до използването на функции (методи) за създаване на абстракции, които позволяват скриване на детайлите на реализацията и предоставят по-високо ниво на абстракция за определена функционалност или задача.
* В основата си, абстракцията чрез функции се основава на принципите на обектно-ориентираното програмиране, където функциите (методите) служат за създаване на интерфейси, които определят поведението на обекти, без да е необходимо да се познава подробността на тяхната реализация.
* Чрез абстракцията на функции, програмистите могат да разделят сложни задачи на по-малки, по-лесни за управление подзадачи, които могат да бъдат изпълнени от отделни функции. Това не само прави кода по-четим и поддържаем, но също така позволява по-голяма гъвкавост и преизползваемост на кода.
* Примери за абстракции чрез функции включват използването на методи с параметри, обектно-ориентирани конструкции като класове и интерфейси, анонимни функции, делегати и други концепции, които позволяват създаването на абстракции в програмирането.

**Абстракция чрез функции в C# - Пример**

****

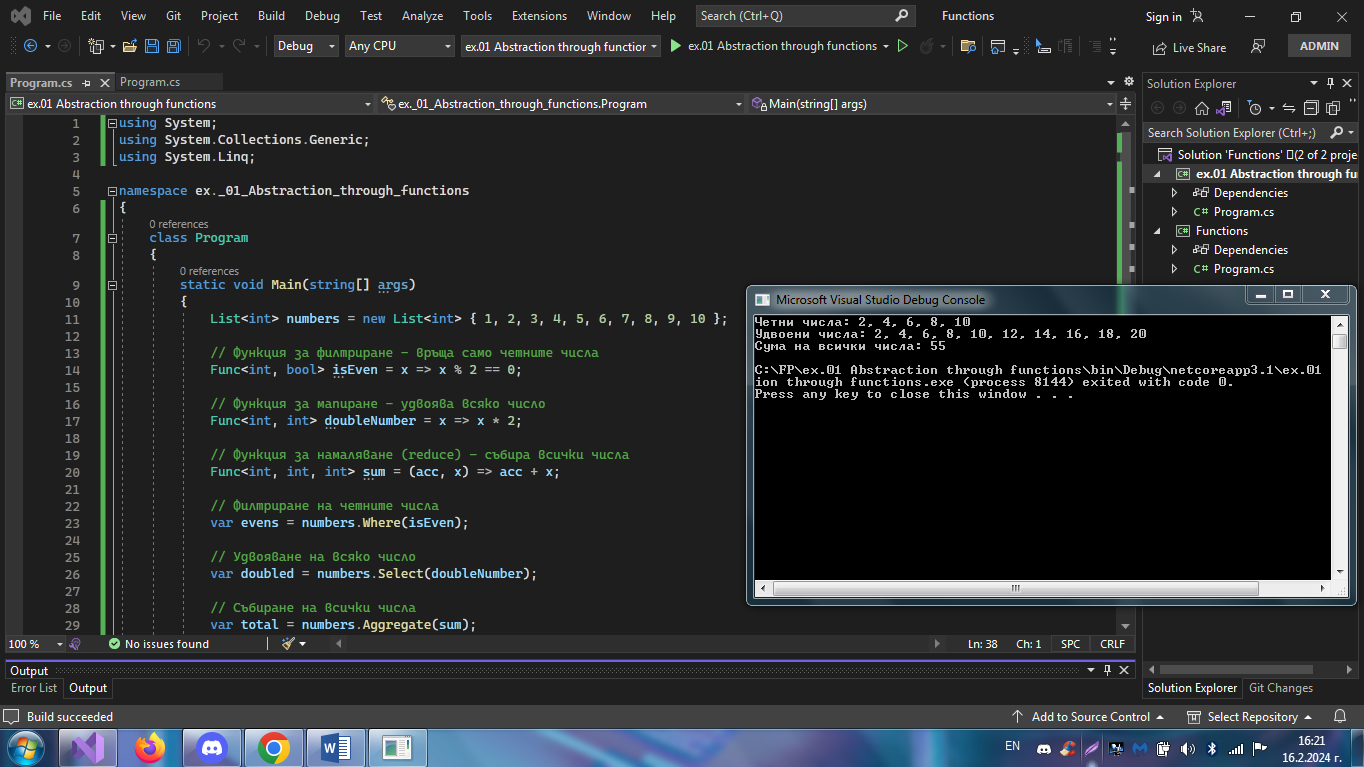
****

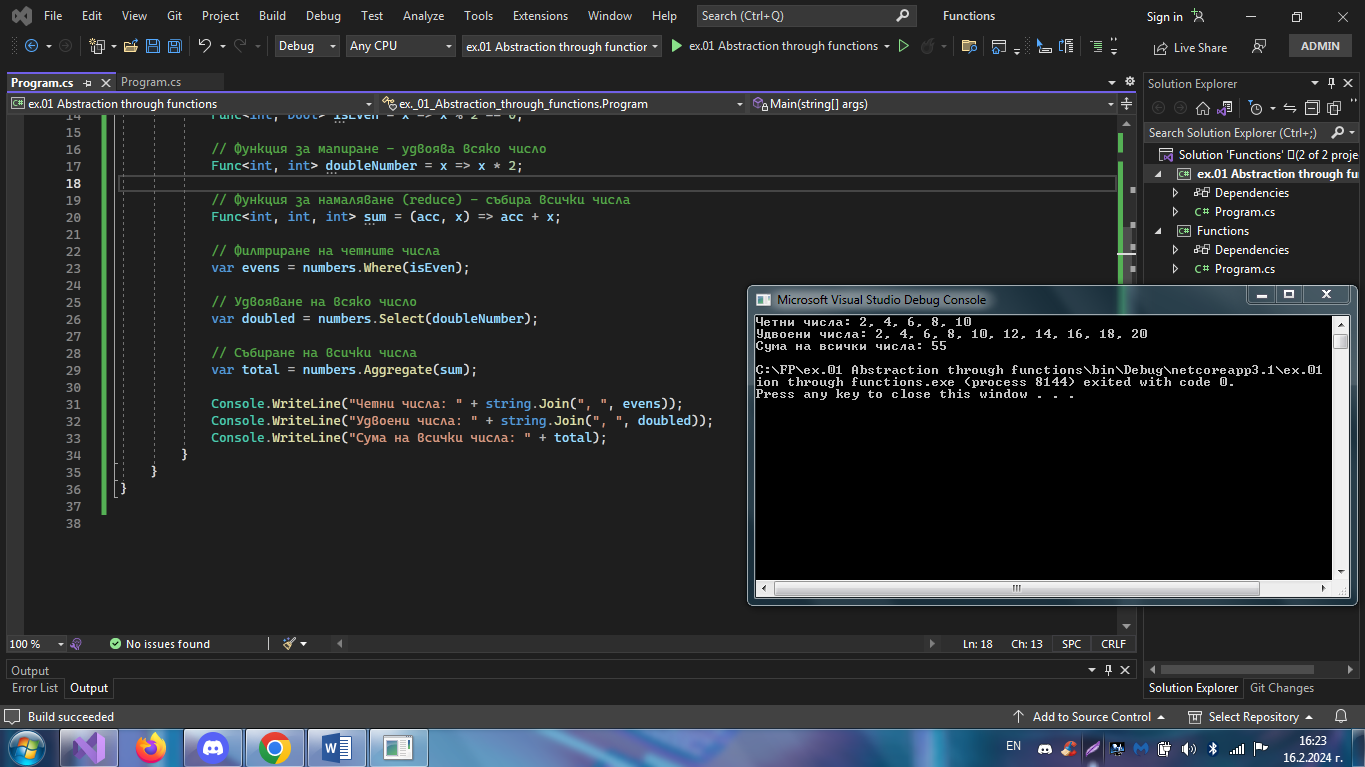
Този пример използва анонимни функции `(Func<>)` за дефиниране на абстракция върху различни операции, които могат да се прилагат върху колекция от числа. Така, чрез преизползване на тези анонимни функции, можем лесно да приложим различни операции върху същата колекция от числа, като не се налага да пишем повтарящ се код за всяка операция.

1. **Изчисления върху списъци**

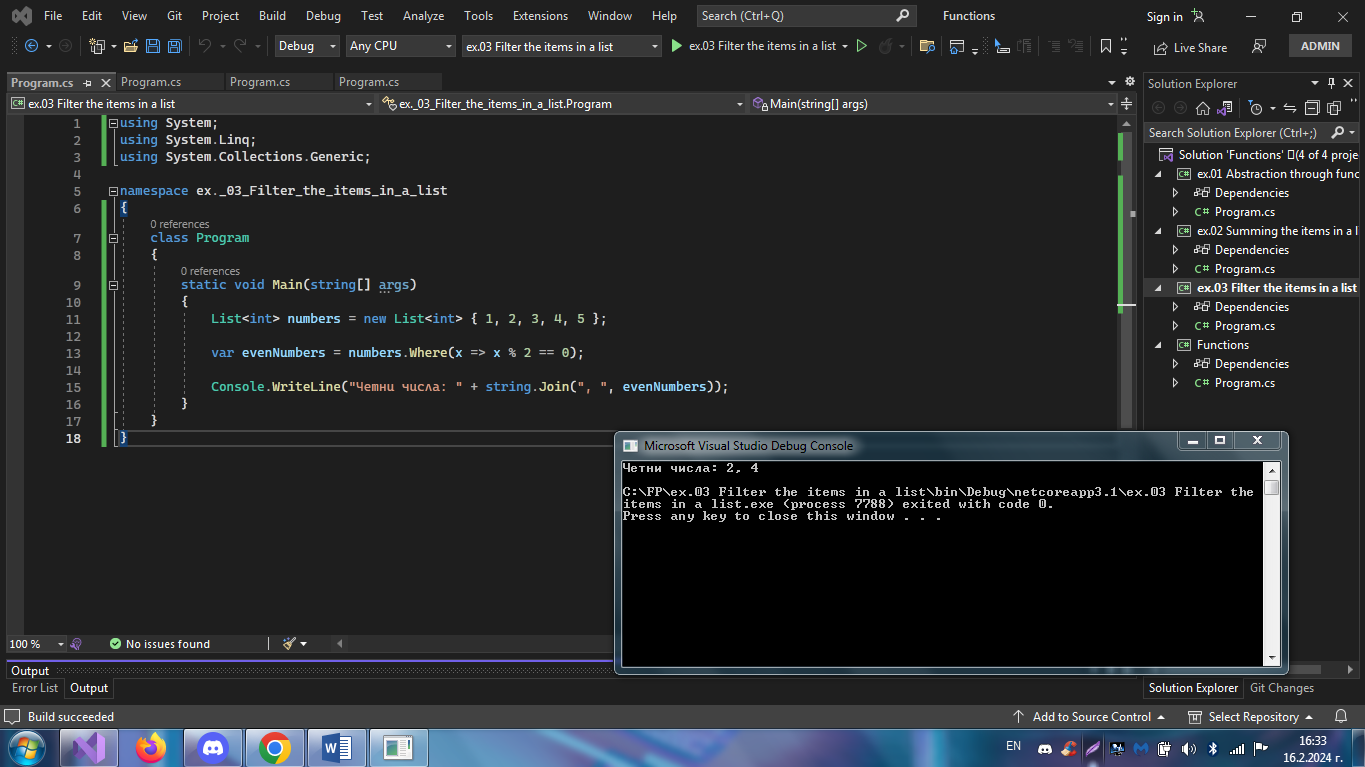
* В C# изчисленията върху списъци (например сумиране, филтриране, мапиране и други операции) могат да бъдат извършвани по различни начини. Един от най-удобните начини за това е използването на LINQ (Language Integrated Query) - езикът за заявки, вграден в C#.

**Сумиране на елементите в списък – Пример**

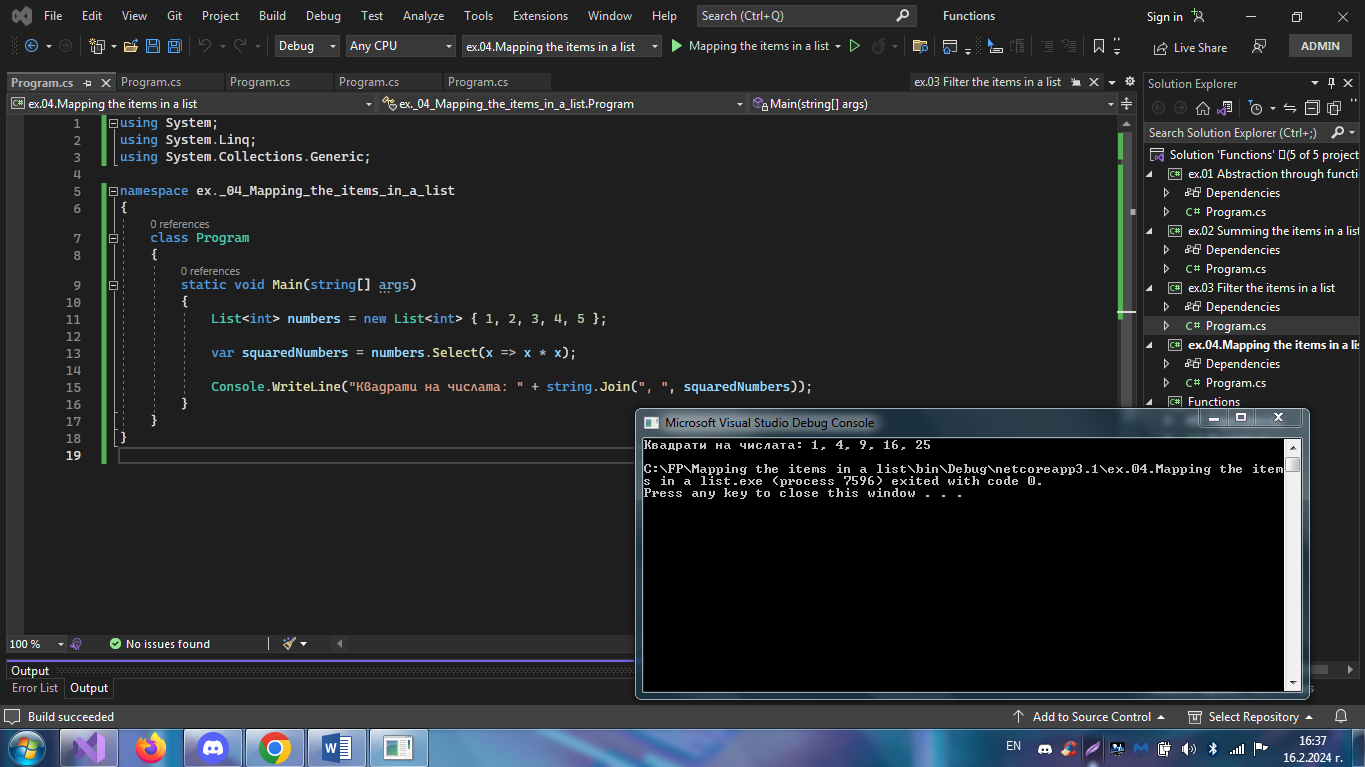
****

****

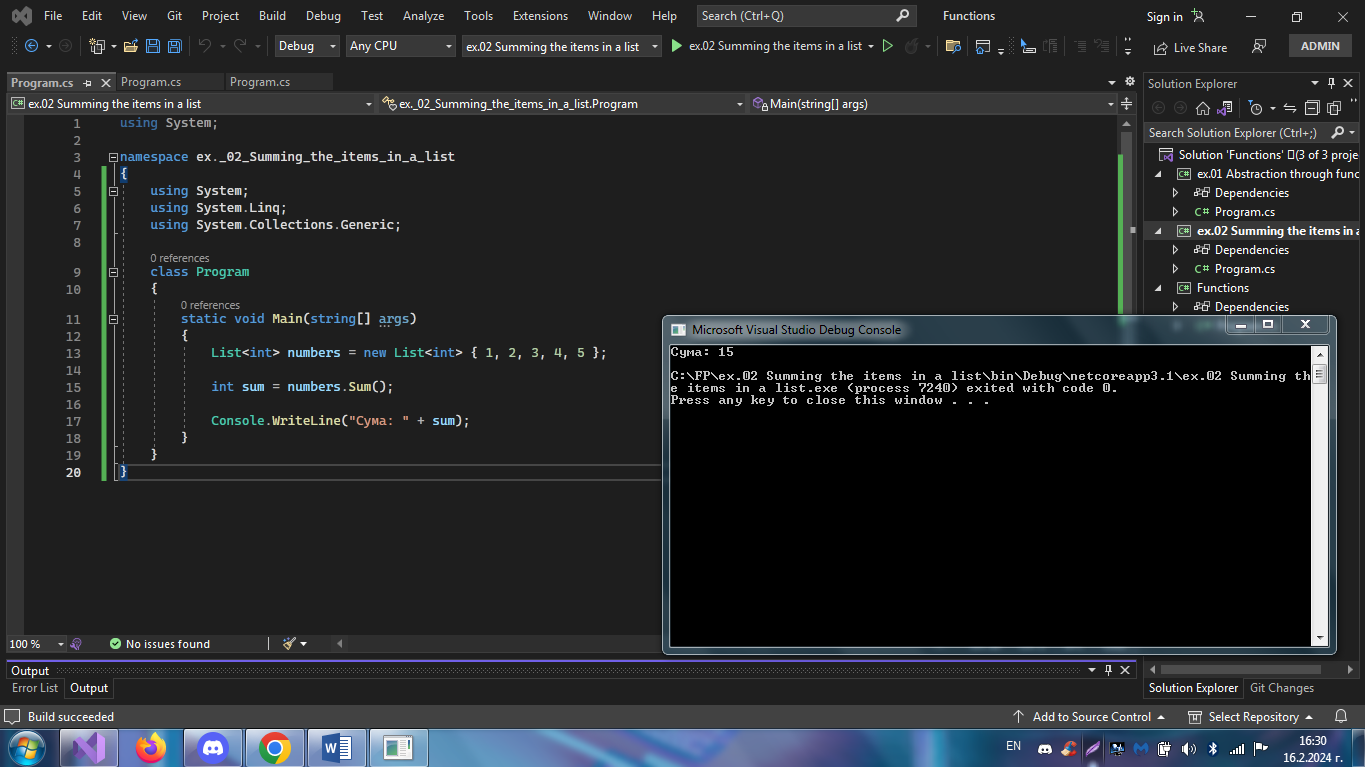
**Филтриране на елементите в списък – Пример**

****

**Мапиране на елементите в списък – Пример**

****

**Намиране на минимален/максимален елемент в списък – Пример**

****

**Използвана литература/Източници**

<https://bg.wikipedia.org/wiki/%D0%90%D0%B1%D1%81%D1%82%D1%80%D0%B0%D0%BA%D1%86%D0%B8%D1%8F_(%D0%BA%D0%BE%D0%BC%D0%BF%D1%8E%D1%82%D1%8A%D1%80%D0%BD%D0%B8_%D0%BD%D0%B0%D1%83%D0%BA%D0%B8)>

<https://www.shu.bg/wp-content/uploads/teachers/storage/83/41274940.pdf>

<https://learn.fmi.uni-sofia.bg/pluginfile.php/452367/mod_resource/content/0/%D0%97%D0%94-%D0%A4%D1%83%D0%BD%D0%BA%D1%86%D0%B8%D0%BE%D0%BD%D0%B0%D0%BB%D0%BD%D0%BE%20%D0%BF%D1%80%D0%BE%D0%B3%D1%80%D0%B0%D0%BC%D0%B8%D1%80%D0%B0%D0%BD%D0%B5-20.pdf>

<https://vinss.files.wordpress.com/2010/11/19-1-funkcionalnoprogramiranerevisedbynisheva.pdf>