Доклад

По Функционално програмиране

Тема: Анонимни функции

Изготвил: Валентин Кьосев 12 ,,а‘‘ клас

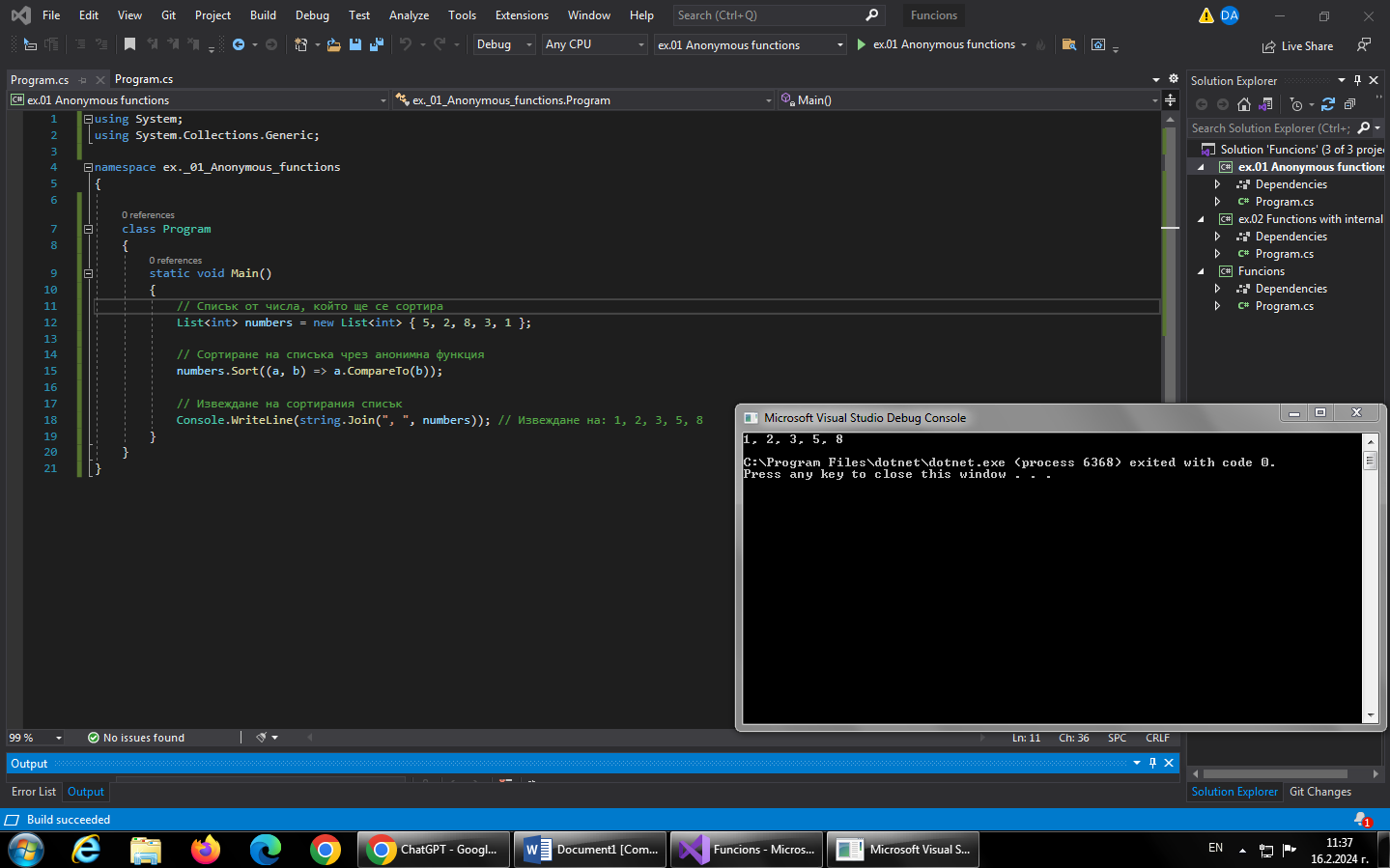
1. Анонимни функции. Примери
2. Функции с вътрешно състояние. Примери
3. **Анонимни функции**

* Анонимните функции са форма на вложена функция, в която е позволен достъпа до променливи в обхвата на съдържащата функция. Ето защо анонимните функции трябва да бъдат имплементирани чрез използването на затваряне. За разлика от именуваните вложени функции, те не могат да бъдат рекурсивни без помощта на т.нар. оператор на неподвижната точка (т.нар. анонимна неподвижна точка или анонимна рекурсия).

**Приложение**

* Анонимните функции могат да се използват за съдържащи функционалност, които не се нуждаят от наименуване и имат възможност за използване за кратък период от време. Някои първични примери включват затваряне и обработване.
* Употребата на анонимните функции е въпрос на стил. Използването им никога не е единственият начин за решаване на проблем: всяка анонимна функция може да бъде дефинирана като именувана функция и наричана по име. Някои програмисти използват анонимните функции за капсулиране на специфични еднократни кодове, за да не се усложнява основния код с много едноредови нормални функции.
* В някои програмни езици, анонимните функции често се имплементират за всяка специфична цел като свързващи събития за обратно извикване, или за представяне на функциите за конкретни стойности, което може да е по-ефикасно, по-четливо, и по-малко податливо на грешки от използваните по общи именувани функции.

**Анонимна функция в C# - Пример**

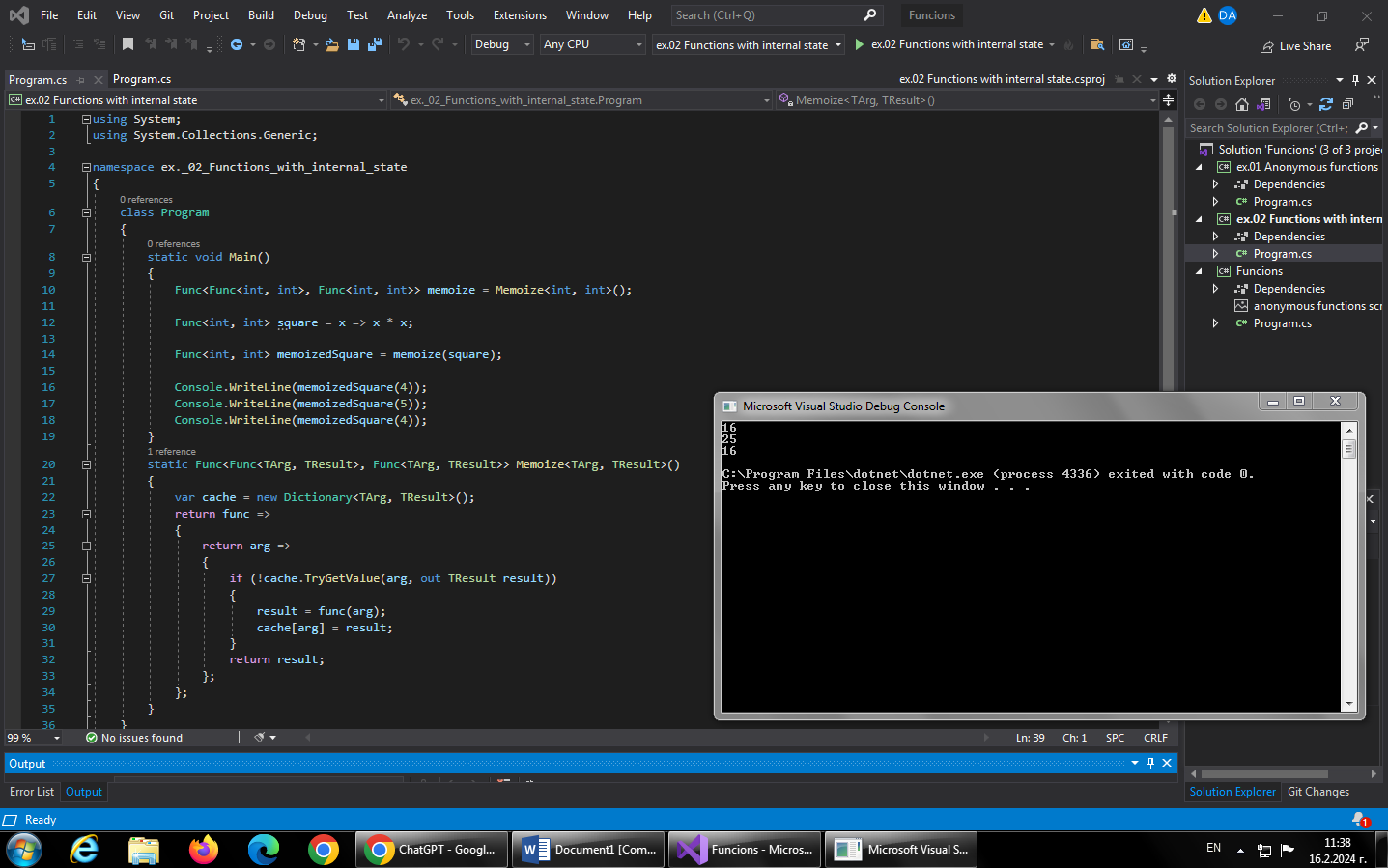
****

Тук предаваме анонимна функция чрез lambda израз `(a, b) => a.CompareTo(b)` . Този lambda израз приема два аргумента ` a` и `b` (две числа) и сравнява тяхната стойност чрез метода `CompareTo()` , който е част от типа` int`. Тази анонимна функция се използва от метода `Sort()` за сортиране на списъка `numbers`.

1. **Функции с вътрешно състояние**

* Функциите с вътрешно състояние са такива, които запомнят състоянието си между повиквания. Това означава, че те не само приемат входни данни и връщат резултат, както обикновените функции, но и запазват информация за предходни извиквания и използват тази информация при следващите извиквания.
* В някои езици за програмиране, като например Python, функциите могат да имат вътрешно състояние чрез използване на глобални променливи или чрез затваряне на стойностите на променливи (closures). В други езици като JavaScript, функциите по подразбиране са затворени и могат да съхраняват локални променливи, които остават достъпни след като функцията е приключила с изпълнението си.
* Функциите с вътрешно състояние са често използвани за съхранение на информация, която трябва да остане съхранена между различни извиквания на функцията, като например броячи, кеширани данни или обекти със състояние. Те могат да бъдат полезни за поддържане на състояние или за избягване на необходимостта от глобални променливи, които могат да доведат до нежелани странични ефекти и затрудняват разбирането на програмата.

**Функция с вътрешно състояние в C# - Пример**

****

Тук функцията `Memoize` създава и връща затворен lambda израз, който има достъп до кеша за резултатите `cache` и използва този кеш, за да оптимизира изчисленията. Когато функцията се извика с аргумент, първо се проверява дали резултатът за този аргумент вече е кеширан. Ако е, резултатът се връща директно от кеша. Ако не е, изчислява се резултата чрез подадената оригинална функция и той се кешира за бъдещи извиквания. Така се подобрява ефективността на функцията, като се избягва повтарянето на изчисления за едни и същи аргументи.

Използвана литература/Източници

<https://bg.wikipedia.org/wiki/%D0%90%D0%BD%D0%BE%D0%BD%D0%B8%D0%BC%D0%BD%D0%B0_%D1%84%D1%83%D0%BD%D0%BA%D1%86%D0%B8%D1%8F>

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

<https://elixirschool.com/bg/lessons/basics/functions>