**Доклад**

**по**

**Функционално програмиране**

**Тема: Въведение във функционалното програмиране**

1. **Какво означава функционално програмиране?**
2. **Концепции за функционално програмиране в C#**
3. **Функции**
4. **Чисти функции**
5. **Функции като стойности на функция**
6. **Рекурсия**
7. **Рекурсивна реализация на цикли. Примерен код на C#**
8. **Опашкова рекурсия. Примерен код на C#**
9. **Рекурсивно обхождане на списък. Примерен код на C#**
10. **Източници**

Изготвил: Ваня Ванева 12а

1. **Какво означава функционално програмиране?**

Функционалното програмиране (ФП) е парадигма за програмиране, в която се акцентира върху използването на функции като основен строителен елемент. Програмите в тази парадигма се разглеждат като изчислителни процеси, базирани на математически функции, без странични ефекти и изменяеми състояния.

*Функционалното програмиране в C# включва използване на ламбда функции, Higher-Order функции и чисти функции, като подчертава акцента върху функциите като основни строителни блокове.*

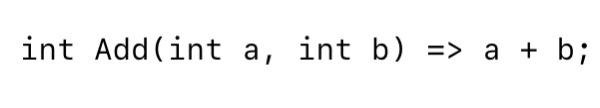
1. **Концепции за функционално програмиране в C#**

* **Ламбда функции:** Въведени са като изрази, позволявайки компактно изразяване на функции.
* **LINQ (Language Integrated Query):** Предоставя декларативен начин за работа с данни.
* **Higher-Order функции:** Възможността да подаваме функции като аргументи и да връщаме функции от други функции.

1. **Функции**

Функциите са базови строителни блокове във функционалното програмиране, приемащи входни данни и връщащи резултат.

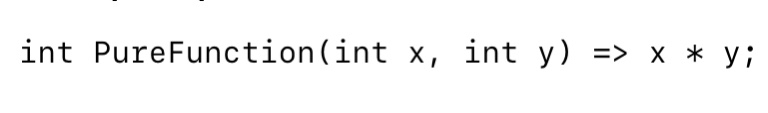
Примерен код на C#:



1. **Чисти функции**

Чистите функции нямат странични ефекти и зависят само от подадените аргументи за вход.

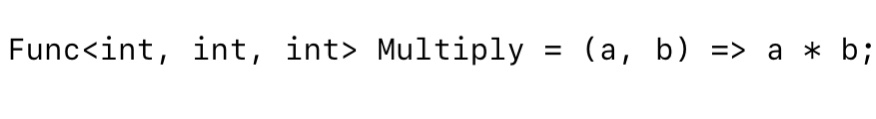
Примерен код на C#:



1. **Функции като стойности на функция**

Възможността да подаваме функции като параметри и да връщаме функции като резултат.

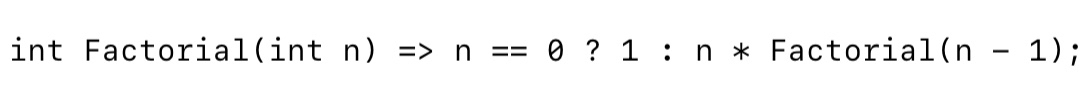
Примерен код на C#:



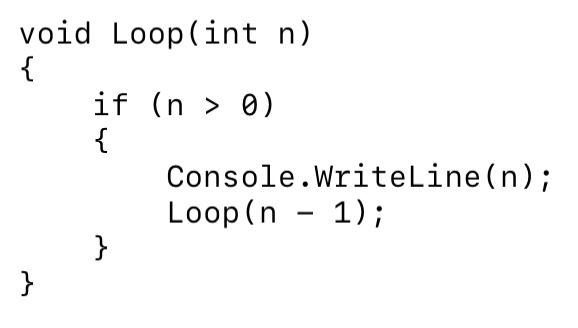
1. **Рекурсия**

Използване на функции, които се извикват сами върху себе си, за решаване на проблеми.

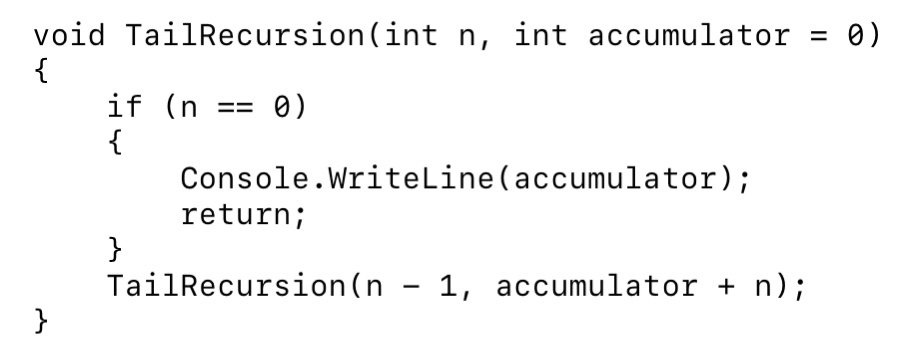
Примерен код на C#:



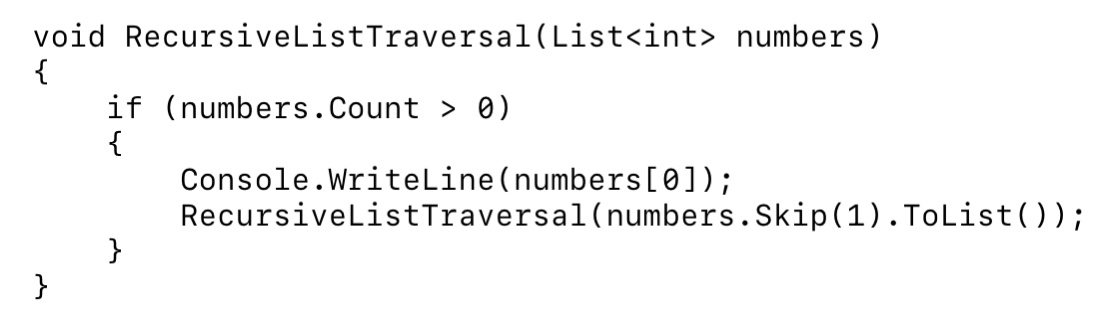
1. **Рекурсивна реализация на цикли. Примерен код на C#**

****

1. **Опашкова рекурсия. Примерен код на C#**

****

1. **Рекурсивно обхождане на списък. Примерен код на C#**

****

1. **Източници**

<https://bg.wikipedia.org/wiki/%D0%A4%D1%83%D0%BD%D0%BA%D1%86%D0%B8%D0%BE%D0%BD%D0%B0%D0%BB%D0%BD%D0%BE_%D0%BF%D1%80%D0%BE%D0%B3%D1%80%D0%B0%D0%BC%D0%B8%D1%80%D0%B0%D0%BD%D0%B5>

<https://softuni.bg/blog/important-programming-concepts>

<https://functionalprogrammingcsharp.com/functional-features-of-c-sharp>

<https://functionalprogrammingcsharp.com/getting-started-with-functional-programming-in-c-sharp>

<https://www.thoughtco.com/introduction-to-functions-in-c-958367#:~:text=David%20Bolton%20is%20a%20software,%2C%20BAE%20Systems%2C%20and%20LCH.&text=In%20C%23%2C%20a%20function%20is,do%20not%20exist%20by%20themselves>.

<https://www.telerik.com/blogs/functional-programming-csharp-brief-consideration#:~:text=Pure%20functions%20are%20a%20subset,state%20or%20other%20side%20effects>.

<https://functionalprogrammingcsharp.com/pure-functions>

<https://www.telerik.com/blogs/functions-as-data-functional-programming-in-c>

<https://bg.wikipedia.org/wiki/%D0%A0%D0%B5%D0%BA%D1%83%D1%80%D1%81%D0%B8%D1%8F>