**Доклад**

по

Функционално програмиране

Тема: Парадигми на програмиране (Programming paradigms)

I - Парадигми за програмиране

1. Същност
2. Източници

II - Функционални езици

1. Чисти функционални езици
2. Нечисти функционални езици
3. Источници

III - Входно/Изходни операции от различен тип – конзола, база от данни, файл, мрежа

1. Конзола
2. База от данни
3. Файл
4. Мрежа
5. Източници

IV – Състояние (State) на програмата

1. Глобално състояние
2. Локално състояние
3. Източници

Изготвил: Ваня Ванева 12а

**I - Парадигми за програмиране**

1. **Същност**

Парадигмата на програмиране,

парадигма за програмиране или

програмна парадигма представлява фундаменталния стил на програмиране. Основните програмни парадигми са:

* обектно ориентирано – една програмна система се моделира като набор от обекти, които взаимодеистжат помежду си.
* императивно – парадигма в програмирането, спрямо която една програма се състои от редица команди, които определят какво и в какъв ред компютърът трябва да направи.
* функционално – стил за изграждането на структурата и елементите на програми, който третира като изчисляжа оценката на математически функции.
* декларативно – стил на изграждане на структурата и елементите на компютърните програми - който изразява логиката на изчислението, без да описва неговия контролен поток.

1. **Източници**

* <https://en.wikipedia.org/wiki/Declarative_programming>
* <https://bg.wikipedia.org/wiki/%D0%9F%D0%B0%D1%80%D0%B0%D0%B4%D0%B8%D0%B3%D0%BC%D0%B0_%D0%BD%D0%B0_%D0%BF%D1%80%D0%BE%D0%B3%D1%80%D0%B0%D0%BC%D0%B8%D1%80%D0%B0%D0%BD%D0%B5>

**II - Функционални езици**

1. **Чисти функционални езици**

* Haskell (Ха̀скъл) е функционален език за програмиране. Той е полиморфично статично-типизиран, „мързелив“ (нищо не се прави, докато не се наложи), Haskell е базиран на ламбда смятане, следователно ламбда се използва за негово лого. Поради близостта си с математиката и богатството от математически библиотеки езикът е широко използван за решаване на кратки математически задачи.
* Лисп е съвкупност от програмни езици и данни. Въз основа на него са произлезли много диалекти, най-известните от които са Scheme и Common Lisp. Лисп е вторият език за програмиране от високо ниво в историята. Програмите на Лисп се представят като структури от типове данни на самия език, сред които списъците играят главна роля.

1. **Нечисти функционални езици**

* Scheme е функционален език за програмиране, диалект на LISP. Характерен е със своя минималистичен синтаксис. Подходящ е за първоначално запознаване с функционалното програмиране.
* Програмният език R е система за статистически анализи и графики. R е едновременно и програмен език и софтуерна среда. Този език се смята за диалект на езика S.

1. **Источници**

<https://bg.wikipedia.org/wiki/%D0%9A%D0%B0%D1%82%D0%B5%D0%B3%D0%BE%D1%80%D0%B8%D1%8F:%D0%A4%D1%83%D0%BD%D0%BA%D1%86%D0%B8%D0%BE%D0%BD%D0%B0%D0%BB%D0%BD%D0%B8_%D0%B5%D0%B7%D0%B8%D1%86%D0%B8_%D0%B7%D0%B0_%D0%BF%D1%80%D0%BE%D0%B3%D1%80%D0%B0%D0%BC%D0%B8%D1%80%D0%B0%D0%BD%D0%B5>

**III - Входно/Изходни операции от различен тип – конзола, база от данни, файл, мрежа**

1. **Конзола**

Функцията за вход/изход на конзолата приема вход от стандартни входни устройства и изчислява и дава изход на стандартно изходно устройство. Клавиатурата е стандартно входно устройство, а мониторът е стандартно изходно устройство.



1. **База от данни**

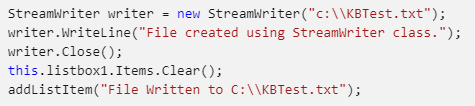
Базата данни за вход/изход е система, съдържаща данни за потреблението на всеки продукт/услуга от определени големи фирми, които я опотребяват.



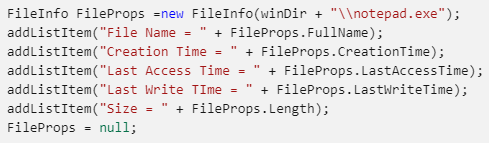
1. **Файл**

Входящият поток се използва за четене на данни от файл (операция за четене), а изходният поток се използва за запис във файла (операция за запис).

Write a text file:

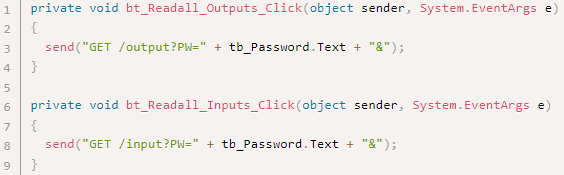


View file information:



1. **Мрежа**

I/O (Input/Output) е система за обработка на информация, предназначена да изпраща и получава данни от компютърен хардуерен компонент, устройство или мрежа. Данните могат да се изпращат между устройства по мрежа.



1. **Източници**

<https://www.includehelp.com/cpp-tutorial/console-input-output-operations-methods.aspx>

<https://www.tutorialspoint.com/csharp/csharp_file_io.htm>

<https://www.routledge.com/InputOutput-Databases-Uses-in-Business-and-Government/Gould/p/book/9780815350361>

<https://www.trentonsystems.com/blog/what-is-i/o>

<https://www.wut.de/e-5763w-29-apus-000.php>

<https://learn.microsoft.com/en-us/troubleshoot/developer/visualstudio/csharp/language-compilers/file-io-operation>

**IV – Състояние (State) на програмата**

1. **Глобално състояние**

Глобалното състояние е „декларирано“ извън компютъра и е достъпно навсякъде.

1. **Локално състояние**

Локалното състояние е „декларирано“ на компютъра е достъпно само на компютъра.

1. **Източници**

<https://www.codecademy.com/forum_questions/51a8eb6b899f415fab00139e>