

- посіб. для студ. спеціальності 151 «Автоматизація та комп'ютерно-інтегровані технології», освітньо-професійна програма «Автоматизація та комп'ютерно-інтегровані технології кібер-енергетичних систем»; уклад.: О. Й. Штіфзон, П. В. Новіков, В.П. Бунь. – Електронні текстові дані (1 файл: 2,2 Мбайт). – Київ : КПІ ім. Ігоря Сікорського, 2020. – 144 с.
2. Електричні апарати: підручник / [Бржезицький В.О., Зелінський В.Ц., Лежнюк П.Д., Рубаненко О.Є.]. – Херсон: ОЛДІ-ПЛЮС, 2016. – 602 с.
 3. Методичні вказівки вибір двигунів за потужністю та пускозахисної апаратури при розрахунку електроприводів. / Ю.М. Хандола, М.С. Сорокін, О.Ю. Назаренко, В.В. Гузенко. – Харків: ХНТУСГ, 2019. – 34 с.
 4. Теоретичні основи електротехніки: Частина 1. Електричні кола постійного та змінного струму. Чотириполюсники [Електронний ресурс]: навч. посіб. для студ. спеціальності 141 «Електроенергетика, електротехніка та електромеханіка»/ КПІ ім. Ігоря Сікорського; уклад.: Ю. В. Перетятко, А. А. Щерба– Електронні текстові дані (1 файл: 21.7 Мбайт). – Київ: КПІ ім. Ігоря Сікорського, 2021. – 115 с.
 5. Перехідні процеси в системах електропостачання [Текст]: підруч. для ВНЗ/ Г. Г. Півняк, І. В. Жежеленко, Ю. А. Папаїка, Л. І. Несен; за ред. Г. Г. Півняка. – Дніпро: НГУ, 2016. – 600 с.

СЕКЦІЯ XI. КОМП'ЮТЕРНА ТА ПРОГРАМНА ІНЖЕНЕРІЯ

FUNCTIONAL PROGRAMMING AND EXAMPLES OF ITS USE

Лозяк Антон Петрович

здобувач вищої освіти інституту інженерії та інформаційних технологій
Київський національний університет технологій та дизайну, Україна

Науковий керівник: Любимова Наталія Володимирівна

старша викладачка кафедри філології та перекладу
Київський національний університет технологій та дизайну, Україна

Functional programming - is a programming paradigm based on the use of functions as the basic building block of a program.

In functional programming, data is considered immutable, and functions are considered pure, i.e., those that have no side effects and always return the same result for given input parameters. Functional programming allows you to create more reliable and robust programs that are easier to test and maintain."[1]

History of functional programming:

The basis of functional programming is the command lambda calculus. It was developed in the 1930s for functional implementation, definition, and recursion.[2]

Lisp was the first functional programming language. McCarthy developed it in 1960.[2]

In the late 70s, researchers at the University of Edinburgh defined ml (a meta-language).[2]

In the early 80s, the hope language added algebraic data types for recursion and equations.[2]

In 2004, he introduced "Scala" in the field of functional language innovation.[2]

Features of functional programming:

The key features of functional programming include the use of pure functions, decomposable data, recursion, and deferred computation. Functional programming also supports the creation of functions and the manipulation of higher-level functions. With these features, other programmers also benefit from many other features such as functional programming, a high level of abstraction, simplicity and ease of syntax, and the ability to execute code in parallel, such as increased program reliability and stability, improved parallelization, and the ability to write readable and modular code. All these features make functional programming a powerful and flexible tool for software development."[1]

Basic concepts of functional programming:

The basic concept of functional programming is a set of techniques and principles that allow you to write cleaner, more flexible, and scalable programs. Using these concepts can help improve the quality of your code and simplify the software development process. Let's list the main ones:

A pure function is a function that has no side effects and always returns the same result

for a given input parameter. These functions are more reliable and easier to test."[1]

Data immutability-in functional programming, data is considered immutable, and data is changed by creating a new object. This avoids errors associated with changing the same data in different parts of the program.[1]

Recursive functional programming uses functions that call themselves. Recursion simplifies and makes it easier to define complex operations, such as tree spanning or factorial calculation.[1]

Functional data types-in functional programming, a data type is defined as a set of values rather than a set of operations. These data types provide a higher level of abstraction and generalization, making it easier to write code."[1]

Function configuration allows you to create a new function from an existing function. This is code reuse, which is one of the basic principles of programming.[1]

Lazy notation is another concept that allows you to calculate values only when you really need to run the program, which reduces resource consumption.[1]

Blinding is another important functional programming concept that makes it easier to combine functions or create new functions using existing ones.[1]

The main languages that are always in demand:

Java script-This is a language for the web, and since the 2010s, it has consistently ranked #1 in terms of code written on GitHub. Let's call it the "default" language. Every developer should know JavaScript at some point. It doesn't matter if you're a JavaScript Decommunist and always use it in your work, or if you just work on a basic level and come across it from time to time - in any case, you need to know its basic concepts and understand how it works."[3]

Python - Not everyone knows that Python was a language created before javascript. Python, although very interesting for its purity of syntax, remained in the shadows for a long time. In recent years, the popularity of Python has begun to grow. First of all, this was due to the fact that Western universities stopped teaching beginners complex languages. Instead, the educational institution offered students to learn Python, and it was widely used in various fields, including server-side development. This language can be a great tool to get started."[3]

SQL is not a Programming Language-it is a language used to work with databases, but every developer should know about it. Although developers need to know different programming languages, they can learn only 1 SQL and work only with databases - there is also a demand for such specialists.[1]

Haskell -is a pure functional programming language used to develop complex systems and algorithms. It is popular due to its expressiveness, power, and security. Haskell has many features for abstraction and composition of functions, as well as for lazy computing.[1]

Clojure -is a dynamic functional programming language that runs on the Java platform. It is popular for its simplicity and elegance. Clojure supports data immutability, higher-order functions, lazy computing, and has powerful mechanisms for working with data sequences.[1]

Scala is an object-oriented and functional programming language that runs on the Java platform. Scala is popular for its flexibility and the ability to use both functional and object-oriented programming styles. It has a powerful type system, supports lazy computing, and works with immutable data."[1]

F# -is a functional programming language developed by Microsoft for the .NET platform. It combines the features of functional and object-oriented programming and uses strong static typing.[1]

Erlang -is a functional programming language originally developed for creating

distributed systems. It supports parallel and distributed programming, has high fault tolerance and scalability.[1]

OCaml -is a functional programming language that uses static typing and supports object-oriented programming. It is widely used in academia and industry.[1]

Lisp -is one of the oldest functional programming languages, having been developed in 1958. Lisp is used to develop artificial intelligence, computer linguistics, and other fields.[1]

Examples of functional programming applications:

Functional programming concepts have long been used in various fields and industries and have led to many successes.

Here are some of the examples:

The Haskell programming language is used in the financial industry to create secure and reliable financial applications. For example, Standard Chartered Bank uses Haskell to create its transaction processing system.[1]

Spotify uses functional programming in their music streaming system. They use the Erlang programming language and its OTP (Open Telecom Platform) framework to create a highly efficient and fault-tolerant system."[1]

Jane Street Capital, a financial markets trading company, uses OCaml to create high-performance and secure trading systems.[1]

The Clojure programming language is used in the backend of various web services, such as Amazon Web Services and Walmart.[1]

Another example of functional programming in the real world is a part of Facebook's backend that was written in Haskell, and an online spam filter module uses a mixture of Haskell/C++ that processes up to a million messages per second.[1]

Advantages of functional programming:

Avoids unexplained problems and errors in the code.[3]

It is easier to test and perform unit testing and debug FP code.[3]

Parallel processing and parallelism.[3]

Hot deployable code and fault tolerance.[3]

Offers better modularity with shorter code.[3]

Improves developer productivity.[3]

Supports nested functions.[3]

Functional constructs like Lazy Map & Lists, etc.[3]

Allows efficient use of lambda calculus.[3]

Limitations of functional programming:

The functional programming paradigm is not simple, so it is difficult for a beginner to understand.[3]

Difficult to maintain because many objects evolve during coding.[3]

Requires a lot of mocking and careful customization of the environment.[3]

Reuse is very difficult and requires constant refactoring.[3]

Objects may not correctly represent the problem.[3]

Conclusion:

Functional Programming, or FP, is a way of looking at building software based on some basic defining principles

The concept of functional programming focuses on results, not processes

The goal of any FP language is to model mathematical functions

Some of the most famous functional programming languages are: 1) Haskell 2) SM 3) Clojure 4) Scala 5) Erlang 6) pure

A "pure function" is a function whose input is declared as an input and none of them should be hidden. The output is also reported as an output.