ВОЗМОЖНА ЛИ МИГРАЦИЯ К ПАРАДИГМЕ ФУНКЦИОНАЛЬНОГО ПРОГРАММИРОВАНИЯ?

Атрощенко Н.А.

Атрощенко Натэлла Александровна – старший преподаватель,
кафедра экономической информатики, инженерно-экономический факультет,
Белорусский государственный университет информатики и радиоэлектроники,
г. Минск, Республика Беларусь

Аннотация: *в статье анализируются вопросы целесообразности перехода от императивного подхода к вычислениям, характерным для объектно-ориентированного программирования, к парадигме функционального программирования и декларативному принципу вычислений.*
Ключевые слова: *функциональное программирование, императивное программирование, декларативный подход, объектно-ориентированное программирование.*

IS MIGRATION TO A FUNCTIONAL PROGRAMMING PARADIGM POSSIBLE?

Atroshchenko N.A.

Atroshchenko Natella Alexandrovna – Senior Lecturer,
DEPARTMENT OF ECONOMIC INFORMATICS, FACULTY OF ENGINEERING AND ECONOMICS,
BELORUSIAN STATE UNIVERSITY OF INFOMATICS AND RADIOELECTRONICS,
MINSK, REPUBLIC OF BELARUS

Abstract: *the article analyzes the feasibility of the transition from an imperative approach to computations, characteristic of object-oriented programming, to the paradigm of functional programming and the declarative principle of computation*
Keywords: *functional programming, imperative programming, declarative approach, object-oriented programming.*

Ещё сравнительно недавно парадигма объектно-ориентированного программирования (ООП) считалась единственно надёжной, понятной и хорошо продуманной основой для производящихся в большом количестве в мире многоуровневых распределённых приложений и проектов. Инкапсуляция, полиморфизм, наследование – его незыблемые столпы, органичные и интуитивно понятные даже любому новичку в мире объектно-ориентированного программирования, словно подсказаны самой природой. После изучаемого в школе процедурного программирования, приходя в мир ООП, всем приходилось расставаться с начальными навыками построения программ, где код - лишь последовательность команд. Едва–едва обосновавшемуся в мире ООП программисту сегодня опять надо перестраиваться и внедряться в мир декларативного подхода и функционального программирования (ФП). Вполне можно принять утверждение, что функциональное программирование – это не только отдельная парадигма, основанная на использовании чистых функций (pure functions), без применения использования изменяемых данных (mutable data), общего состояния (shared state), и побочных эффектов (side-effects), но и особый способ мышления в программировании. Известно, что любой язык программирования зиждется на определённом наборе абстракций. В ФП мыслить придётся на так называемом уровне «абстракций абстракций». Элегантные конструкции, созданные из строительных блоков – функций высшего порядка [4] – это мощный инструмент для уменьшения количества кода с возможностью творить собственные архитектурные решения и даже паттерны. Для хорошего понимания сути ФП нужно уяснить суть декларативного подхода без прописывания пошагового выполнения команд. Также важной частью является понимание процесса обёртывание каждого действия и параметра в функции, а также построение цепочек вызываемых одной за другой функций типа f1.f2.f3 или комбинации вида f1(f2(f3())). Функции становятся базовыми единицами языка, такими, как числа или строки.

Проблемы многозадачности приводят необходимости применения монад и функторов, функциональном способе выстраивания вычислений в цепочки и возможности обернуть данные в некоторый контейнер или контекст.

Замена обработки конструкции if-else на монаду Either, Null – исключение на монаду Maybe [4], объединение функций в цепочку (chaining), использование для этого каррирования (currying) и функций высшего порядка – это лишь некоторые изменения программного кода в сторону ФП, где уже на сегодня есть достаточное количество схожих инструментов. Функциональное программирование значительно улучшает подход к написанию кода, вне зависимости от предстоящих задач и языка программирования. Сегодня практически все языки реализовали или собираются реализовать парадигму ФП. Уже

сравнительно давно Java и C++11 успешно используют лямбда исчисления. Очевидные плюсы использования функционального программирования при написании кода рано или поздно заставляют искать новые пути в этой парадигме для оптимизации приложений. К этим плюсам, как правило, относят значительное уменьшение объёма кода, его лучшую читабельность, более лёгкую поддержку и проверку кода, меньшую вероятность отслеживания ошибок и их возникновение [1]. Кроме этого парадигма декларативного программирования весьма привлекательна более эффективной и удобной организацией модульного тестирования, когда полностью исключается возможность побочных эффектов, например, связанных с возможностью использования одной и той же внешней переменной сразу несколькими функциями. «Чистое тестирование» даёт возможность простой проверки каждой функции при разных наборах аргументов. Возможность избежать побочных эффектов на стадии написания кода делает созданный программный продукт более надёжным, конфигурируемым, избавляет от трудноотслеживаемых ошибок [2]. Нельзя не согласиться, что это уже немало. Однако, есть серьёзные недостатки, часто не очевидные на первый взгляд, связанные с непредсказуемым порядком вызова функций, невозможностью вмешаться в процесс на стадии итераций и в связи с этим проблемой постоянной необходимости отслеживания проверки наличия и освобождения динамической памяти. Вопрос и полном переходе с ООП на ФП, а также с императивного программирования на декларативное остаётся открытым. Пока ведутся «учёные споры», эти парадигмы создания программного кода пока неплохо уживаются на практике вместе, хотя далеко не всегда являются взаимозаменяемыми. Главный ориентир - стандарты ISO 9001 [2], которые устанавливают критерии системы менеджмента качества программного кода. Только время и опыт подтвердят состоятельность новых подходов для языков программирования высшего порядка в мире IT.

Список литературы / References

1. *Атрощенко Н.А*. Качество программного продукта в ракурсе современных практических требований // Проблемы науки, 2018. № 11 (52). С. 51-54.
2. *Атрощенко Н.А*. О концепциях выбора технических средств для разработки веб-проекта // Наука и образование сегодня, 2016. № 9 (10). С. 22-24.
3. International Organization for Standardization, ISO = Международная организация по стандартизации: Официальный сайт. [Электронный ресурс]. Режим доступа: http://WWW.ISO.ORG/ (дата обращения:11.10.2021).
4. *Филд А., Харрисон П.* Функциональное программирование = Functional Programming // М.: Мир, 1993. С. 637.