

ПЕРЕВАГИ ФУНКЦІОНАЛЬНОЇ ПАРАДИГМИ ПРИ РОЗРОБЦІ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ

В умовах постійної конкуренції, швидких змін у сфері інформаційних технологій, необхідності конкурувати на ринку, постійно виникають питання щодо використання тої чи іншої парадигми розробки програмного забезпечення для отримання максимальної вигоди та мінімізації часу і ресурсів, покладених на розробку програмного забезпечення.

Прийняття парадигми розробки програмного забезпечення несе за собою і прийняття всіх концепцій і методологій цього підходу. Використання лише кількох принципів обраної методики не гарантує якісно позитивних змін для програмного проекту. Лише розуміючи всі переваги і недоліки обраного способу розробки програмного забезпечення, а також оцінивши інші присутні в галузі методології проектування програмного забезпечення, можна прийняти рішення, яка з технологій найкраще підходить для задоволення потреб розробки програмного продукту.

Мови програмування, які використовуються в розробці промислових систем, не стоять на місці. Вводяться в обіг нові ідіоми, та й самі мови розширюються з ціллю додавання можливості оперувати абстракціями більш високого порядку. Використання абстракцій вищого порядку дозволяє приховати деталі реалізації, зробити компоненти програми більш універсальними, забезпечити більшу ймовірність їх повторного використання. Чи є щось спільне між цими нововведеннями, чи різні мови розвиваються незалежно і різноспрямовано? Загальний вектор розвитку мов простежується без особливих зусиль. З метою підвищення рівня абстракції від апаратури і наближення до проблемних областей все більша кількість концепцій, створених в рамках функціонального програмування, знаходять своє місце і в імперативних мовах.

Функціональна парадигма програмування, як і функціональні мови вже довгий час є джерелом «нових» ідей в популярних мовах програмування. Багато функціональних можливостей є присутніми у інших мовах та концепціях розробки програмного забезпечення. Однак, це лише розширення базової мови, а не використання іншої методології розробки. Окрім того, функціональні мови надають унікальні можливості розробникам програмного забезпечення. Завдяки тому, що у функціональну програмуванні кожен символ є константою, не існує побічного ефекту, який може виникнути під час виконання функції. Вона не може змінювати значення і стани, тим самим впливаючи на інші функції. Функції оперують лише значенням, яке повертають. Це значення може бути змінене лише при зміні аргументів, які подаються в функцію. Як наслідок, тестування, відладка та розпаралелення програмного коду проводяться дуже легко. Наявність апарату математичного перетворення, оптимізації, функцій вищого порядку, карринування, лінійні обчислення, абстрагування структур управління, та нескінчені (infinite) структури даних (корекурсія) надають функціональним мовам програмування значної переваги при правильному використанні. Володіння функціональною парадигмою в її «чистому» вигляді дозволяє ефективно застосовувати нові функціональні елементи сучасних мов для управління складністю розроблюваних систем, а також суттєво збагачує інструментарій, доступні для вирішення тієї чи іншої задачі.