Парадигмы программирования

Что такое парадигма вообще? Можно сказать, что это определенный взгляд на явления окружающего мира и представление о возможных действиях с ними. В программировании под парадигмой принято понимать обобщение о том, как должна быть организована работа программы.

Среди прочего выделяют такие парадигмы программирования как директивное (структурное), объектно-ориентированное и декларативное (функционально-логическое). Многие языки поддерживают несколько парадигм программирования. С другой стороны, есть языки ориентированные исключительно на реализацию одной парадигмы.

Структурное программирование

Некоторые представители: Fortran, Pascal, C.

Директивная программа предписывает, как достичь результата, пошагово описывая действия. Поэтому такое программирование является достаточно легким для понимания.

В структурном программировании от входных данных полностью зависит последовательность выполнения команд.

В директивном программировании в свое время возникла концепция локализации части кода в так называемые подпрограммы (функции, методы), с последующим их вызовом из

разных мест основной программы. При вызове в подпрограмму могут передаваться какие-либо данные в виде аргументов; а подпрограмма, в свою очередь, может возвращать в главную программу результат (т.е. полученные в ходе ее выполнения данные).

Функциональное и логическое программирование

Представители функциональных языков: List, Haskell.

Представитель логических языков: Prolog.

Декларативная программа заявляет (декларирует), что должно быть достигнуто в качестве цели. Важным является точная формулировка задачи. Программист не задает алгоритм для ее решения.

Функциональное программирование основано на математическом понятии функции, которая не изменяет свое окружение; это отличие функционального программирования от функций в структурных языках. Функциональная программа состоит из совокупности определений функций, которые в свою очередь представляют собой вызовы других функций и предложений, управляющих последовательностью вызовов. Каждая функция возвращает некоторое значение в вызвавшую его функцию, вычисление которой после этого продолжается; этот процесс повторяется до тех пор, пока не будет достигнут результат.

В логическом программировании программы выражены в виде формул математической логики, и решение задачи достигается путем вывода логических следствий из них.

Объектно-ориентированное программирование

Представители объектно-ориентированных языков: C++, Java, Python.

Особое внимание уделяется данным, которые представляются в программе в виде объектов. Объекты взаимодействуют между собой с помощью механизма передачи сообщений. Задача программиста - реализовать такие объекты, при взаимодействии которых можно будет получать желаемый результат.

ООП призвано решать более сложные и объемные задачи по сравнению с директивным программированием.

В основе ООП лежат такие понятия как наследование, полиморфизм и инкапсуляция.

Инкапсуляция предполагает, что малозначащие детали объекта скрыты. Объект, получая какую-либо команду, сам «знает» как ее обработать исходя из того, к какому классу он принадлежит.

Все объекты являются экземплярами классов, которые по отношению друг к другу могут выступать в роли родитель-потомок. Дочерние классы наследуют свойства родительского. В случае, когда 100% наследование не требуется, выручает так называемый полиморфизм, который предполагает переопределение методов родительского класса в дочерних классах.