Смысл программирования

Часто людям приходится делать что-то, а потом повторять те же действия через промежуток времени. Когда человек первый раз сталкивается с задачей, то обдумывает последовательность действий (алгоритм) для ее решения. Придумав удачный алгоритм, человек его запоминает, и последующее выполнение похожих задач происходит уже «на автомате». Когда мы действуем согласно какой-либо инструкции, то являемся просто исполнителями.

Но компьютер может быть лучшим исполнителем, чем человек, хотя бы за счет высокой скорости выполнения действий. У компьютера тоже есть память, куда можно записать последовательность действий (алгоритм) для решения той или иной задачи? Машина будет следовать хранящимся в нем инструкциям раз за разом и быстро выполнять их.

При этом надо понимать, что запрограммировать компьютер (т.е. записать в него алгоритм) все равно надо. И делает это человек. Человек разрабатывает последовательность действий для решения задачи и записывает их в память машины. Сам по себе компьютер ничего не знает, он просто железо и делает лишь то, что записано в его памяти. Придумывание алгоритма для решения задачи достаточно трудоемкий процесс. Однако выгоды, получаемые от его реализации с помощью компьютера, перекрывают эти затраты.

Определение компьютерной программы

Предположим, что поместить в память компьютера инструкции для решения задач не проблема. Но поймет ли их компьютер? Он не «говорит» на естественных для нас языках. Для него нужны инструкции на понятном ему языке. Комплекс инструкций для решения определенной группы задач, записанный на понятном компьютеру языке является ничем иным как компьютерной программой.

Определение и история языков программирования

Так какой же язык понятен компьютеру, как следует выражаться в его память, чтобы он потом делал то, что мы хотим. Компьютер – это электронное вычислительное устройство. Вычислительное! Он работает с числами, складывает, вычитает, сравнивает. Больше ни с чем. Но как же? Ведь мы привыкли обрабатывать на компьютере не только числовую информацию, но также текстовую и графическую. Поэтому может сложиться впечатление, что компьютер работает не только с числами. Фокус заключается в том, что любую информацию, в том числе текстовую и графическую можно закодировать числами. А чтобы

запрограммировать какие-то действия с нечисловой информацией можно манипулировать именно числами, в которых она представлена в закодированном виде.

Первые программы для ЭВМ писались именно в числах. Это сложно для человека. Представьте, что все, что вы хотите сказать, нужно сказать исключительно числами. Тут дело усложнялось еще тем, что компьютеры проще создавать таким образом, чтобы они считали в двоичной системе счисления. Записи программ получались слишком длинными. Для их сокращения пользовались восьмеричной и шестнадцатеричной системами счисления. Запись программы числами называется машинным языком программирования.

Программировать работу компьютера в машинных кодах трудно, т.к. думать числами неестественно для человека. Мы привыкли думать словами. А что если сопоставить часто используемым группам чисел слова, а затем написать программу перевода слов на числа. Т.е. теперь программист может описать алгоритм в словах, затем передать его специальной программе-переводчику (транслятору), который преобразует словесный алгоритм в машинный код, понятный компьютеру. И человеку хорошо и компьютеру понятно. От человека требовалось только создать этот самый транслятор. Человечество регулярно придумывает различные языки программирования и пишет под них трансляторы. Но сначала появились ассемблеры.

Чуть позже программисты стали замечать, что в принципе программирование сводится к вводу и выводу данных, выбору той или иной ветки выполнения программы и повторения одних и тех же действий определенное количество раз. Но самое главное, некоторые части кода программы много раз используются в ней в разных местах. Так пришли к выводу о том, что программа должна представлять собой структуру из обособленных частей. Появилось структурное программирование.

Программистская мысль не спала и структурным программированием не ограничилась. Начали появляться объектно-ориентированные, логические, функциональные и другие способы программирования. Так в объектно-ориентированном программировании основной идеей стала аналогия с реальным миром, где есть объекты, имеющие свойства, умеющие что-то делать сами и подвергающиеся изменениям извне.

Решение поставленной задачи при этом происходит путем взаимодействия описанных объектов.

Следует понимать, что конкретный язык программирования может содержать в себе несколько концепций (парадигм) программирования. Например, быть структурным и объектно-ориентированным одновременно. Языков множество, парадигм (принципов программирования) единицы.

Информация взята с сайта http://pas1.ru