

участники выступают с новыми идеями, презентациями, предложениями по заданной теме. Далее происходит поиск самых интересных идей и их общее обсуждение.

К интерактивным методам обучения на уроке также относят мастер-классы, построение шкалы мнений, дерево решений.

Все активные и интерактивные методы обучения призваны решать главную задачу – научить ребенка учиться. То есть истина не должна преподноситься «на блюдечке». Гораздо важнее развивать критическое мышление, основанное на анализе ситуации, самостоятельном поиске информации, построению логической цепочки и принятию взвешенного и аргументированного решения.

УДК 004.432

Купцова В. Ю.

ДАТАОРИЕНТИРОВАННЫЕ ЯЗЫКИ ПРОГРАММИРОВАНИЯ

*Белорусский национальный технический университет,
г. Минск, Республика Беларусь*

Научный руководитель: ст. преподаватель Астапчик Н. И.

Сегодня принято объединять данные в «объекты», которые инкапсулируют данные, связанные с одной сущностью, и методы, которые обновляют эту сущность некоторым необязательным образом. Мы называем это объектно-ориентированным программированием. Язык, который делает это особенно удобным, называется объектно-ориентированным языком.

Альтернативой является размещение данных в «таблицах», которые содержат данные для многих похожих объектов, и процедуры, которые обрабатывают всю таблицу в тесном цикле. Это может быть более эффективным из-за того, как работает кэш процессора, при условии, что нам не нужно много специализаций по разрушению конвейеров. Чтобы отличить

этот стиль от объектно-ориентированного программирования, некоторые люди решили назвать его ориентированным на данные. Язык, который делает это особенно удобным, можно назвать ориентированным на данные языком.

Отличительной чертой ориентированных на данные языков является богатство структуры их типов. Это может включать в себя различные разновидности структурированных данных (массивы, деревья, множества, отношения), абстрактные типы, полиморфизм, наследование, вычисления над типами и т. д.

Языки ориентированы на работу с одним определенным типом данных. Например, APL работает с матрицами и векторами, Snobol обрабатывает строки, SETL выполняет операции над множествами. Существуют также такие языки, как K и J. Рассмотрим эти языки более подробно.

1. J – функциональный язык. Но язык не обычный, не классический функциональный. Единица данных – n -мерный массив, а не список. J является интерпретатором, с динамической типизацией.

Выражения строятся из глаголов, имен существительных, наречий, союзов и прочего. В J кроме мерности массива, есть тип его элементов. Т.е. элементы всегда однотипные. Числовых типов несколько: логический (0, 1), целый, целый с любым количеством знаков, рациональный, действительный, комплексный. Также вместе с числовыми типами есть специальные числа: минус бесконечность, бесконечность, неопределенность. Динамическая типизация J замечательно подходит для математических операций – он на каждой операции сам определит, какие в нем числа.

2. K – язык программирования, оптимизированный для работы с массивами, модификаторами действий. Среди особенностей также отсутствие циклов, зависимые переменные и K-tree. Реализован для Windows, Solaris, Linux. Изначально язык программирования K был разработан Артуром Уитни. В настоящее время

мя 32-разрядная версия языка К реализована для windows, linux или MacOS X для бесплатного некоммерческого применения.

Особое развитие также получили языки для работы с базами данных: 3GL, PL/SQL, FoxPro.

Под третьим поколением (3GL) первоначально понимались все языки более высокого уровня, чем ассемблер. Главной отличительной чертой языков третьего поколения стала независимость от аппаратного обеспечения, то есть выражение алгоритма в форме, не зависящей от конкретных характеристик машины, на которой он будет исполняться. Код, написанный на языке третьего поколения, перед исполнением транслируется либо непосредственно в машинные команды, либо в код на ассемблере и затем уже ассемблируется. При компиляции, в отличие от предыдущих поколений, уже нет соответствия один к одному между инструкциями программы и генерируемым кодом.

Стала широко использоваться интерпретация программ – при этом инструкции программы не преобразуются в машинный код, а исполняются непосредственно одна за другой. Независимость от «железа» достигается за счет использования интерпретатора, скомпилированного под конкретную аппаратную платформу. Таким образом, широкое распространение получили языки, связанные с работой с базами данных, так как они обладают большим количеством преимуществ перед объектно-ориентированными. Отличительной чертой датаориентированных языков является богатство структуры их типов, однако языки обладают одним значительным недостатком – они ориентированы на работу с определенным типом данных.