Лабораторные работы пропедевтического курса просты и наглядны. Они имеют цель познакомить студентов с эмпирическим основанием ньютоновской механики, а также способствуют формированию навыков индуктивного обобщения экспериментальных данных с целью выявления эмпирических закономерностей в проводимом эксперименте. Особенность виртуальных работ, размещенных в системе в том, что студенты могут выполнять в любое время, даже дома. Отчет о работе остается в системе, и преподаватель, проверяя его, определяет баллы за выполненную работу.

В ходе выполнения лабораторных работ и решения задач физического практикума формируется *теоретическое мышление и познавательная самостоятельность* студента.

Закрепить теоретические знания и получить практические навыки решения типичных задач школьного курса можно с использованием мультимедийных обучающих систем (МОС). МОС - обучающие программы, созданные на основе гипермедиа, предоставляющие обучающемуся самостоятельный выбор траектории обучения, темпа работы, обеспечивающие разноуровневое обучение. Обучающие программы можно рекомендовать из Интернет-ресурсов, например, веб-сайт: «Единая Коллекция цифровых образовательных ресурсов для учреждений общего и начального профессионального образования» http://school-collection.edu.ru/about/.

Существенная роль при изучении физики принадлежала и принадлежит контролю усвоения знаний. Внедрение электронных средств контроля интенсифицирует и улучшает качество работы преподавателей. Компьютерные системы тестирования обладают объективностью и стимулируют самостоятельную работу студентов. Мы используем возможности системы поддержки учебного процесса EDUCON, которая создает и хранит портфолио каждого обучающегося: все сданные им работы, все оценки и комментарии преподавателя к работам, все сообщения в форуме. Все отметки по курсу хранятся в сводной ведомости. Вопросы тестов сохраняются в базе данных и могут повторно использоваться в одном или разных курсах. На прохождение теста может быть дано несколько попыток. Возможно, установить определенный интервал времени на работу с тестом.

В последние годы наблюдается ощутимое снижение уровня физико-математической подготовки выпускников полной средней школы. Теоретическое мышление, культура физико-математического мышления большинства абитуриентов слабо развито. Например, у студентов первого курса вызывает подчас затруднение сформулировать элементарное обобщающее утверждение.

Дополнительные возможности при сложившихся обстоятельствах открывает применение компьютерной техники. В условиях информационной предметной среды по физике для достижения качества обучения студентов технического вуза необходимо, чтобы компьютерные средства применялись систематически, и их применение служило задаче обеспечения индивидуального подхода в обучении.

Литература

1. Семенова, Н.Г. Мультимедийные педагогические средства в системе общедидактических методов обучения [Текст]/Н.Г.Семенова // Вестник оренбургского государственного университета.-2005.-№2.-С.95-103
2. Толстик, А.М. Роль компьютерного эксперимента в физическом образовании [Текст]/ А.М. Толстик// Физическое образование в вузах.-2002.-Т.8.-№2.-С.94-102
3. Филатова, Л.О. развитие преемственности школьного и вузовского образования в условиях введения профильного обучения в старшем звене средней школы. [Текст]/ Л.О.Филатова –М.: Издательство: Бином. Лаборатория знаний -2005.-С.192.

Попов А.А.

ДИНАМИЧЕСКИЕ СХЕМЫ ДЛЯ ИЛЛЮСТРАЦИИ ОСНОВНЫХ ОПЕРАЦИЙ С ДВОИЧНЫМИ ДЕРЕВЬЯМИ

apopov@vvoi.ru
Марийский государственный университет
г. Йошкар-Ола

При чтении лекций по программированию возникает трудности в изложении некоторых алгоритмов. В докладе [1] предложено иллюстрировать такие алгоритмы с помощью так называемых динамических схем. Такая схема иллюстрирует алгоритм, каждому шагу которого ставится в соответствие одна статическая схема. Набор статических схем, сменяющих друга на одном поле, определяет динамическую схему для заданного алгоритма.

В этом докладе рассматриваются динамические схемы для иллюстрации алгоритмов, реализующих основные операции с деревьями. Каждая схема представлена в виде байт-кода, т.е. файла с расширением .class, который образуется при компиляции Java-приложения. Такие файлы запускаются на выполнение из командной строки при условии, что на компьютере установлена виртуальная Java-машина.

Изучение таких динамических структур данных, как деревья, начинается, как правило, с идеально сбалансированного дерева [2]. В программе TreeIdealWrite.class представлен процесс формирования идеально сбалансированного дерева по схеме: корень - левое поддерево - правое поддерево. Узлы дерева представлены в виде узких прямоугольников, соединенных между собой ломаными линиями. Информация о количестве

ключей, которые будут занесены в правое поддерево, располагается в стеке. В процессе формирования дерева на экран выводится текущая информация из стека. При изучении деревьев интерес представляют различные варианты обхода дерева. В данной программе реализованы шесть вариантов обхода, соответствующие возможным перестановкам в схеме: корень - левое поддерево - правое поддерево. Сравнительный анализ вариантов обхода позволяет легко понять, что собой представляют восходящий, нисходящий и фланговый обходы, как слева направо, так и справа налево.

Динамическая схема, соответствующая алгоритму классической задачи поиска по дереву с включением проиллюстрирована в программе SearchTree.class. В ней случайным образом генерируется 35 ключей, принадлежащих отрезку [1, 20], т.е. заранее известно, что последовательность содержит повторяющиеся ключи. В данной задаче для каждого нового ключа или формируется новый узел согласно определению дерева поиска, или счетчик одинаковых ключей ранее созданного узла увеличивается на единицу. Одновременно с увеличением счетчика изменяется цвет прямоугольника. В этой же программе реализован другой вариант формирования дерева поиска. На каждом шаге проверяется критерий сбалансированности, т.е. для каждого узла контролируется разность высот левого и правого поддеревьев. Если указанная разность по модулю больше единицы, то производится балансировка дерева относительно найденного узла. Те узлы, которые после балансировки займут новые положения, отмечаются другим цветом. Для полученного АВЛ-дерева реализованы два варианта обхода, при которых ключи дерева выводятся или в порядке возрастания, или в порядке убывания. Сбалансированное и несбалансированное деревья можно сравнивать друг с другом, нажимая клавишу Home. При каждом нажатии клавиши появления обоих деревьев чередуются на одном поле. Для определения плотности заполнения можно вывести на экран все вакантные места дерева.

Для получения АВЛ-дерева запрограммированы 4 варианта балансировки: однократный LL-поворот, однократный RR-поворот, двукратный LR-поворот, двукратный RL-поворот. Однако ни все случаи могут быть проиллюстрированы на двадцати разных ключах. В программе WriteTree.class генерируются 35 различных ключей. В результате при рассмотрении любой комбинации ключей, как правило, реализуются все 4 варианта балансировки. С другой стороны, на 35 ключах отличие сбалансированного и несбалансированного деревьев более явно выражено. В этой программе кроме двух вариантов обхода дерева реализованы два варианта удаления дерева. На каждом шаге удаляется по одному листу дерева, начиная с правого поддерева, или, начиная с левого поддерева. В программе WriteTreeVar.class собраны некоторые специальные варианты расположения ключей, для которых балансировки производятся относительно корня или относительно узлов, близких к корню. В этом случае несбалансированные деревья явно неоднородны, а балансировки при формировании АВЛ-дерева происходят более часто.

В программе DeleteNodeTree.class представлена динамическая схема, с помощью которой иллюстрируется процесс удаления любого узла дерева. Первоначально формируется сбалансированное дерево поиска, в котором можно выбрать любой узел для удаления. При этом предусмотрен механизм движения по дереву влево, вправо и вверх. Удаление заданного узла в дереве происходит более сложным образом, чем в списке, хотя случай удаления, как в списке, здесь тоже есть. Наибольший интерес представляет удаление такого узла, у которого есть одновременно и левое и правое поддерево. Программа показывает, содержимое, какого узла копируется на место удаляемого. Причем реализованы два варианта, когда на замену идет содержимое крайнего левого узла в правом поддереве, или крайнего правого узла в левом поддереве. Переход от одного варианта к другому происходит после нажатия одной клавиши.

Процесс удаления узла дерева с последующей балансировкой оставшихся узлов реализован в программе DeleteNodeTreeNew.class. Несмотря на схожесть процессов балансировки при добавлении узла и при удалении, последний процесс сложнее. Балансировка при удалении может быть каскадной, т.е. по отношению к нескольким узлам по пути от удаляемого узла до корня. Хорошим примером каскадной балансировки является удаление крайней правой вершины из дерева Фибоначчи[2]. Данный пример проиллюстрирован в специальной программе FibonachiTree.class. В обеих программах каскадная балансировка представлена пошагово со всеми промежуточными положениями узлов дерева.

Предложенные программы иллюстрируют ни столько сложные вопросы, хотя некоторые из них сложны для понимания, а, сколько неудобные при изложении на лекциях. Причем программы оформлены в двух вариантах, как приложения, с целью иллюстрации лекций, и как апплеты, с целью их размещения в Интернете для самостоятельной работы студентов.

В заключении отметим, что далеко ни все разделы курса программирования требуют иллюстрации, но иметь динамические схемы для сложных и интересных алгоритмов просто необходимо, особенно для студентов, специализирующихся в области программирования.

Литература

1. Попов А.А. Динамические схемы для иллюстрации лекций по программированию. Вестник Московского городского педагогического университета. Серия "Информатика и информатизация образования", - Москва-Йошкар-Ола: 2008, № 1(11), с. 105-107.
2. Вирт Н. Алгоритмы и структуры данных, - СПб, 2005.