

оказывают не только консультационную помощь разработчикам тестов, но и при необходимости осуществляют независимую экспертизу и сертификацию присланных тестов.

Таким образом, только при условии коллегиального построения и квалифицированной экспертизы единого банка тестовых заданий, а также их сертификации на уровне вуза или ЦОКТПО МГУП, тесты могут стать действительно эффективным средством повышения качества образовательного процесса, в том числе и дистанционного.

#### *Литература:*

1. Васильев В.И. Требования к программно-дидактическим тестовым материалам и технологиям компьютерного тестирования [Текст] / В. И. Васильев, А. А. Киринок, Т. Н. Тягунова. М.: Изд-во МГУП, 2005/ 27 с.
2. Аскеров Э.М. Технология реализации автоматизированной системы контроля знаний [Электронный ресурс] / Э.М. Аскеров, М.А. Емелин, И.Д. Рудинский, Н. А. Строилов // Конгресс конференций «Информационные технологии в образовании ИТО-2006». Режим доступа: <http://ito.edu.ru/2006/Moscow/VI/VI-0-6323.html>
3. Цветков В. Я. Многоуровневые тестирующие конструкции [Электронный ресурс] / В. Я. Цветков, В. П. Кулагин, Т. В. Булгакова // Вопр. Интернет-образования. № 10. Режим доступа: [http://vio.fio.ru/vio\\_10/resource/Print/art\\_1\\_27.htm](http://vio.fio.ru/vio_10/resource/Print/art_1_27.htm)
4. Шишов С.Е., Кальней В.А. Школа: мониторинг качества образования // Педагогическое общество России. М., 2000. С. 15.

**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 AVL-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 AVL-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 отличия 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 AVL-деревя 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 FibonacciTree.class. В обеих программах каскадная балансировка представлена пошагово со всеми промежуточными положениями узлов дерева.

Предложенные программы иллюстрируют не 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 не 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.

### **Юшкова Е.Г. СОЗДАНИЕ ПРЕЗЕНТАЦИИ ПО ИЗУЧАЕМОЙ ТЕМЕ ИЛИ ТЕМЕ КУРСОВОЙ РАБОТЫ НА АНГЛИЙСКОМ ЯЗЫКЕ.**

*elena\_mil2000@mail.ru*

*Сыктывкарский государственный университет*

Сегодня мир претерпевает глобальные перемены. Изменились и его временные и пространственные границы. Планета приобрела характер всеобщего «офиса», где каждый может «заглянуть» в соседний кабинет, поздороваться, обменяться мнениями, заключить контракт, проанализировать ситуацию в различных сферах деятельности, а также решить личные проблемы, т.е. находиться в режиме «on-line» или готовности для нового качественного личностного изменения. Это касается и сферы образования. На сегодняшний момент формируется единая образовательная среда, много говорится об эффективности использования компьютерных программ и сети Интернет при обучении иностранному языку. На самом деле, в век глобализации образовательного пространства, а также международного сотрудничества специалистов различных стран, иностранный язык выступает средством общения в их профессиональной деятельности. Современные средства связи с партнерами, доступ к информационным ресурсам сети Интернет предполагают достаточно свободное владение, как компьютерными технологиями, так и иностранным языком. Наш век - век информации, неотъемлемой ценностью которого является сама информация, так как путь к успеху во многих областях зависит от способности быстро получать ее, умение работать с ней не только на родном языке.

Таким образом, важной особенностью современного этапа социально – экономического развития общества является формирование особой области производства, характеризующейся не только развитием материальной базы, но и системой специфических технологий, которые принято называть информационными или информационно- коммуникационными. В настоящее время происходит накопление информационного и образовательного потенциала в школах и в вузах, и только небольшое количество наиболее продвинутых в этой области школ и вузов принимает участие в межрегиональных и международных образовательных программах.