

ПРИМЕНЕНИЕ ОБУЧАЮЩИХ ПРОГРАММ-ТРЕНАЖЕРОВ В ОБУЧЕНИИ ПРОГРАММИРОВАНИЮ

Томский государственный педагогический университет

В соответствии с примерной рабочей программой общего образования по информатике и ИКТ, «одним из важнейших понятий курса информатики и информационных технологий основной школы является понятие алгоритма» [1]. Соответственно обучение алгоритмизации и программированию, т.е. способам реализации алгоритмов на ЭВМ, является одним из основных компонентов школьного курса информатики.

При этом, по оценке ряда специалистов, ситуация, сложившаяся вокруг преподавания программирования в школе, является проблемной. «Нынешнее положение курса алгоритмизации и программирования в школе можно назвать незавидным. Тенденция к его сокращению вплоть до полного вытеснения явно прослеживается не только в периодически публикуемых программах различных авторов, но и в отдельных инструктивно-методических документах» [2].

Обратим внимание и на содержательный аспект. В настоящее время существуют три наиболее распространенных подхода к преподаванию программирования:

- преподавание программирования как теоретической дисциплины без привязки к конкретным языкам и системам;
- преподавание на основе специально разработанного языка, ориентированного на обучение основным навыкам программирования;
- изучение одного или нескольких языков программирования, широко используемых при решении научных и хозяйственных задач (такие языки можно назвать стандартными).

Наиболее приемлемым для общеобразовательной школы, где курс информатики преподается в 8–11 классах, является сочетание первого и третьего подходов – обучение теоретическим основам программирования на базе стандартного языка. При этом необязательно вдаваться в глубины языка. Учащиеся, которых он заинтересует, могут сделать это и сами. Наибольшее внимание следует уделить переходу от алгоритмических структур к их программной реализации на языке программирования [3]. При этом все же следует как можно больше внимания уделять именно самим алгоритмам, в этом случае ученику в будущем будет легче перейти с одного языка программирования на другой.

Проанализируем государственные нормативные документы [4–6]. Федеральный базисный учебный план для образовательных учреждений РФ отводит 105 часов для обязательного изучения информатики и информационных технологий на ступени основного общего образования и 70 часов – на ступени полного общего образования. На алгоритмизацию и программирование отводится около 19 часов. В случае профильного изучения информатики ситуация значительно улучшается. Федеральный базисный учебный план отводит 280 часов для обязательного изучения информатики и информационных технологий на ступени среднего общего образования, то есть по 140 учебных часов на каждый год обучения. Как мы видим, на изучение раздела программирования отводится недостаточное количество времени, это приводит к тому, что изучение некоторых тем проходит поверхностно, а некоторые исключаются вовсе.

Следует обратить внимание и на затруднение в освоении алгоритмизации и программирования у значительной части учащихся. Данная тема и с точки зрения учителей, и с точки зрения учеников является сложнейшей в рамках учебного предмета. К сожалению, необходимо отметить и низкий уровень квалификации многих учителей информатики, что не способствует качественному освоению предмета.

С нашей точки зрения, учитывая приведенные выше факторы, представляется актуальной задача совершенствования методики преподавания программирования в средней школе, а возможно, и в высших учебных заведениях. Одной из возможностей повышения качества усвоения материала является использование обучающих программ-тренажеров. Их использование позволит значительно сократить время изучения и закрепления нового материала, тем самым ускорить процесс обучения. Учащиеся смогут наглядно разобрать алгоритм столько раз, сколько им потребуется для понимания материала, при этом тренажер может использоваться как на занятиях, так и самостоятельно учащимися. Важную роль в методике обучения программированию следует отводить самостоятельной работе учеников, так как только самостоятельная разработка алгоритмов и программ долж-

ным образом способствует развитию алгоритмического мышления и закреплению необходимых навыков. Тренажер, с нашей точки зрения, должен наглядно демонстрировать процесс работы изучаемых алгоритмов. Важным моментом является возможность размещения тренажеров на web-страницах. Данная технология позволяет обеспечить общедоступность создаваемых средств. Технологически это приводит к необходимости реализации тренажеров в виде web-приложений, то есть наборов скриптов, выполняющихся либо на стороне сервера, либо на стороне клиента. Программы-тренажеры, являющиеся web-приложениями и демонстрирующие процесс работы алгоритмов, назовем web-визуализаторами.

Для того чтобы применение web-визуализаторов было действительно эффективным, необходимо сформировать их комплект исходя из предметного наполнения и целей изучения информатики в средней школе. Рассмотрим предметное наполнение раздела «Алгоритмизация и программирование» на базовом уровне. В раздел входят такие темы [1]:

- Алгоритм. Свойства алгоритма. Способы записи алгоритмов.
- Исполнители алгоритмов (назначение, среда, режим работы, система команд).
- Компьютер как формальный исполнитель алгоритмов (программ).
- Алгоритмические конструкции: следование, ветвление, повторение.
- Разбиение задачи на подзадачи, вспомогательный алгоритм.
- Алгоритмы работы с величинами: типы данных, ввод и вывод данных.
- Языки программирования, их классификация.
- Правила представления данных.
- Правила записи основных операторов: ввода, вывода, присваивания, ветвления, цикла и т.д.
- Правила записи программы.
- Обработываемые объекты.

Исходя из перечисленных выше тем следует, что учащиеся должны овладеть первоначальными навыками программирования на языке высокого уровня, что включает в себя способность разрабатывать алгоритмы линейной структуры, использовать операторы ветвления, выбора, циклов, организации подпрограмм, в том числе рекурсивных. Ученик должен уметь использовать простые и составные типы данных: целочисленные, вещественные, символьные, массивы, записи. Так как при изучении некоторых тем целесообразно использовать тренажеры, мы предлагаем следующие web-визуализаторы:

1. Исполнение линейного алгоритма.
2. Условный оператор.

3. Циклы с предусловием.
4. Цикл с постусловием.
5. Цикл с параметром.
6. Суммирование элементов массива.
7. Поиск минимального и максимального элементов в массиве.
8. Простые сортировки массива.
9. Поиск подстроки.
10. Представление множества в ЭВМ.
11. Организация рекурсии.
12. Ввод, вывод в файл.

Для профильного курса информатики, ориентированного на углубленное изучение программирования, можно расширить предметное наполнение следующими темами [7]:

- Деревья. Обход дерева. Переборные алгоритмы.
- Матрицы. Работа с числами, матрицами, строками, списками.
- Инварианты, индуктивные доказательства.
- Генерация псевдослучайных последовательностей.

В этой связи список визуализаторов необходимо дополнить, например, тренажерами по следующим темам:

1. Умножение матриц.
2. Улучшенные сортировки массива.
3. Внешние сортировки, сортировки файла.
4. Работа стека и очереди.

Данный список не является исчерпывающим. Web-визуализаторы могут быть полезны и при обучении программированию в вузе. В качестве примера рассмотрим подготовку будущих учителей информатики. К блоку дисциплин по данной специальности, непосредственно связанных с программированием, можно отнести дисциплины: программирование, объектно-ориентированное программирование в среде Delphi; практикум по решению задач; теория алгоритмов, теоретические основы информатики; численные методы, основы искусственного интеллекта; компьютерное моделирование. Существует ряд тем, которые вызывают наибольшие затруднения у студентов, что зачастую связано с трудностью наглядной демонстрации. Именно в этом случае и целесообразно использование программ-тренажеров. Например, в рамках дисциплины «Теоретические основы информатики» можно успешно применять следующие визуализаторы:

1. Поиск минимального остова.
2. Поиск в глубину и ширину.
3. Поиск кратчайшего пути в графе.
4. Реализация конечного автомата.

Посуттила в редакцию 11.08.2008

Литература

1. Примерная программа основного общего образования по информатике и ИТ – http://window.edu.ru/window_catalog/pdf2txt?p_id=14196
2. Ильясов А.Н. К вопросу о роли и месте программирования в школьной информатике – <http://www.rusedu.info/PrintArticle101.html>
3. Лапчик М.П. и др. Методика преподавания информатики: Учеб. пособие для студ. пед. вузов / Под общей ред. М.П. Лапчика. М., 2003.
4. Федеральный компонент государственного стандарта основного общего образования – <http://www.ed.gov.ru/ob-edu/noc/rub/standart/p1/1287/>
5. Федеральный компонент государственного стандарта основного полного образования – <http://www.ed.gov.ru/ob-edu/noc/rub/standart/p2/1288/>
6. Базисный учебный план – <http://www.ed.gov.ru/ob-edu/noc/rub/standart/>
7. Примерная программа среднего (полного) общего образования по информатике и ИТ, профильный уровень – http://window.edu.ru/window_catalog/pdf2txt?p_id=14239

УДК 378. 02: 372.8

Н.Ф. Долганова, А.Н. Стась

ОСНОВНЫЕ ДИДАКТИЧЕСКИЕ ПРИНЦИПЫ ПОСТРОЕНИЯ ДИСЦИПЛИНЫ «ЭЛЕМЕНТЫ ВЫЧИСЛИТЕЛЬНОЙ ГЕОМЕТРИИ» В УСЛОВИЯХ ПЕДАГОГИЧЕСКОГО ВУЗА

Томский государственный педагогический университет

Одной из особенностей подготовки студентов педагогических вузов в современных условиях является стремление специалистов как можно шире раскрыть потенциал использования компьютерных технологий в образовании. Поэтому возникает необходимость не только знакомить студентов педагогических вузов с возможностями применения ЭВМ в образовании, но и формировать мотивацию для повышения уровня работы с компьютером. Необходимо создавать условия не только для совершенствования пользовательских навыков, но и для формирования умений разрабатывать собственные программные средства.

На наш взгляд, для этого следует в первую очередь усиливать фундаментальную математическую подготовку студентов, обучающихся по специальности «Информатика», а в области информационных технологий – студентов, обучающихся математике. В данной ситуации уместно предложить ввести новую учебную дисциплину «Вычислительная геометрия».

Однако стандартами педагогического образования не заложено изучение спецкурса «Вычислительная геометрия» [1, 2]. В то же время вуз может организовать изучение данной дисциплины за счет национально-регионального компонента, устанавливаемого вузом. Именно такой возможностью воспользовалась кафедра информатики ФМФ ТГПУ, введя в учебный план специальности 030100 дисциплину «Элементы вычислительной геометрии» [3]. Что касается открывающейся на ФМФ ТГПУ с 2008 г. специальности 230200 «Информационные

системы», ориентированной на подготовку разработчиков педагогических программных средств, то для нее государственным стандартом предусмотрена дисциплина «Компьютерная геометрия и графика» [4]. В рамках этой дисциплины предусмотрено изучение основ вычислительной геометрии.

Целью дисциплины «Вычислительная геометрия» является изучение и освоение базовых понятий, моделей, методов, структур данных и алгоритмов, применяемых при решении задач вычислительной геометрии, с упором на задачи, возникающие в процессе разработки педагогических программных средств (ППС). Данный курс направлен на помощь в формировании межпредметных связей между математикой и информатикой, приобщение студентов к исследовательской деятельности, формирование их творческой активности, он является элементом информационной подготовки [5, с. 40–45].

С особенностями преподавания данной дисциплины в педагогическом университете и с подробным содержанием разделов дисциплины можно ознакомиться в предыдущих работах авторов и в примерной рабочей программе по дисциплине «Вычислительная геометрия» [6; 7; 8, с. 117–120; 9, с. 466–467], в данной же работе мы обратим внимание на общие методические принципы построения дисциплины «Вычислительная геометрия».

Как уже отмечалось ранее, курс вычислительной геометрии в основном включен в учебный план технических вузов и классических университетов и его преподавание ведется на протяжении