

УДК 004.588

**МЕТОДИКА ФОРМИРОВАНИЯ ЭТАЛОННЫХ НАБОРОВ
ВХОДНЫХ И ВЫХОДНЫХ ДАННЫХ ДЛЯ АНАЛИЗА
РЕЗУЛЬТАТОВ ВЫПОЛНЕНИЯ ЗАДАНИЙ С НЕРАЗРЕШИМЫМ
МНОЖЕСТВОМ ПРАВИЛЬНЫХ ОТВЕТОВ**

О.Е. Вашенков, А.В. Лямин

В работе представлена структура системы AcademicNT, роль и функции виртуальных лабораторий и методы разработки эталонных наборов для качественной оценки результатов виртуального эксперимента. Исследование проводилось для виртуальной лаборатории по информатике. Такой подход может быть использован для любых виртуальных лабораторий.

Ключевые слова: виртуальная лаборатория, информационно-образовательная среда, электронное обучение.

Введение

В рамках информационно-образовательной среды виртуальные лаборатории необходимы для формирования и проверки креативных навыков. В отличие от электронных тестов, задания виртуальных лабораторий имеют неразрешимое множество пра-

вильных ответов. Для таких заданий нельзя описать правильный ответ конечным множеством или формальным выражением, например, в виде регулярных выражений. Электронные задания, выполненные по технологии виртуальных лабораторий, используются в различных дисциплинах, позволяя проводить эксперименты и контролировать методику их выполнения [1, 2]. Ярким примером использования данной технологии стал многостилевой редактор кода, который предназначен для проверки навыков реализации алгоритмов с возможностью кодирования на наиболее распространенных языках программирования [2–4].

Следует отличать виртуальные лаборатории от тренажеров и интерактивных демонстраций. Виртуальная лаборатория позволяет организовать автоматическую проверку. Тренажеры и демонстрации не предназначены для проверки заданий средствами информационно-образовательной среды.

Взаимодействие модулей системы AcademicNT

Компоненты виртуальной лаборатории активно взаимодействуют с модулями информационно-образовательной среды AcademicNT. При запросе задания клиенту передается интерактивный элемент лаборатории – апплет. После взаимодействия с апплетом пользователь отправляет ответ на сервер с помощью управляющих элементов HTML-страницы. При этом используется язык сценариев JavaScript для получения данных из апплета и заполнения HTML-формы. Взаимодействие между клиентом и сервером организовано по протоколу HTTP, что накладывает ограничение на символы, используемые в ответе пользователя. Поэтому задача апплета – корректное экранирование ответа пользователя перед отправкой. Сервер перенаправляет HTTP-запрос веб-приложению, выполненному по технологии Java Servlet. Сервлет организует вызов хранимой процедуры на стороне системы управления базами данных (СУБД) и передает ей ответ пользователя. Процедура вызывает команду на стороне сервлета для отправки XML-сообщения на проверяющий сервер. Сообщение содержит ответ пользователя и эталонные данные. Вместе с командой на проверку сервлету передается ряд дополнительных переменных и список процедур, которые требуется вызвать после проверки. Проверяющий сервер осуществляет проверку для каждой пары эталонных наборов и возвращает результат сервлету в форме XML-сообщения. Взаимодействие между сервлетом и проверяющим сервером происходит по протоколу RLCP (Remote Laboratory Control Protocol – протокол управления удаленной лабораторией) [4]. После получения ответа сервлет разбирает XML-ответ и последовательно вызывает хранимые процедуры для занесения результатов проверки. Связи между модулями представлены на рис. 1.

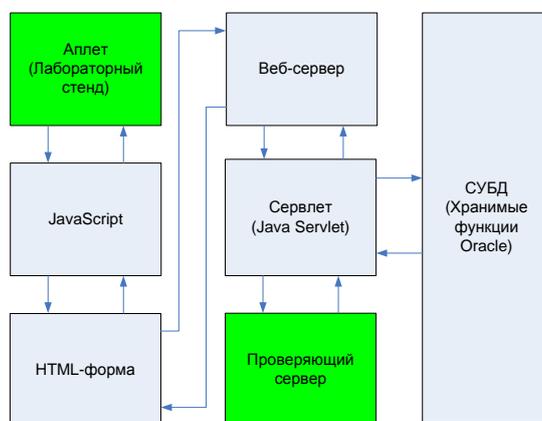


Рис. 1. Схема взаимодействия модулей информационно-образовательной среды

Технология виртуальных лабораторий

В общем случае ответ на задание виртуальной лаборатории содержит описание некоторой системы с определенным набором входов и выходов. Эта система конструируется в соответствии с заданием на специализированном виртуальном стенде из объектов и отношений предметной области. Знание о правильном ответе может быть представлено в виде пар эталонных наборов входных и выходных данных. Для оценки правильности ответа необходимо получить реакцию описанной системы на заданном наборе входных данных и сравнить ее с соответствующим эталонным набором выходных данных. Совпадение реакции системы с эталонным набором выходных данных будет свидетельствовать о правильности ответа. Для получения реакции системы, описанной в ответе, на заданном наборе входных данных используется специализированная виртуальная машина. Организация взаимодействия виртуального стенда и виртуальной машины, входящих в состав лаборатории, формирование заданий, эталонных наборов входных и выходных данных, протоколирование выполненных действий и полученных результатов – функции информационно-образовательной среды.

Для удобства разработки проверяющих серверов лабораторий был создан каркас, в котором организовано взаимодействие с системой AcademicNT, обработка RLCP-запросов и формирование RLCP-ответов. Основными требованиями для авторов виртуальных лабораторий является разработка лабораторного стенда (апплета), алгоритмов проверки ответа и XML-описаний сценария, заданий и установки в формате AcademicRM. От лабораторного стенда требуется только поддержка метода для получения ответа испытуемого в текстовом виде.

Использование каркаса для разработки проверяющего сервера позволяет абстрагироваться от низкоуровневой сетевой передачи и протокола RLCP, фокусируя внимание только на алгоритмах проверки.

С помощью эталонных наборов автор описывает соответствие между заданными входными и выходными данными. Задача проверяющего сервера – установить эталонные входные данные перед выполнением работы, а после выполнения сверить эталон с данными, полученными в ходе автоматического эксперимента. Эталонные наборы должны быть описаны для каждого задания отдельной виртуальной лабораторной работы. Автор может разработать один и более наборов. Итоговая оценка за задание устанавливается системой как процент успешно пройденных наборов.

Эталонные наборы представляются в XML-описании как текстовые данные. Например, если требуется перечислить последовательность целых чисел, набор может выглядеть следующим образом: 0 1 12 15 3 4 5 8. Программный модуль проверяющего сервера получает эту строку, а ее интерпретация – задача автора лаборатории. Например, в виртуальной лаборатории «Многостилевой редактор кода» допустимыми являются следующие эталонные наборы:

- целочисленный массив: $a=[0, 1, 2, 3, 4, 5]$;
- строковая переменная: $b="answer"$;
- вещественная переменная с допустимой ошибкой – 10%: $c\{10\% \}="1.2E+3"$.

Задача проверяющего сервера – корректно разобрать входные данные и произвести сравнение эталонных выходных данных со значениями, полученными в результате выполнения программы испытуемого.

Разработка эталонных наборов

Виртуальная лаборатория по информатике была разработана для проведения единого государственного экзамена по информатике в компьютерной форме. Цель работы

– проверка навыка испытуемых в разработке алгоритмов решения стандартных задач. Далее будут рассмотрены особенности данной виртуальной лаборатории и методика формирования эталонных данных. Сложности в составлении эталонных наборов заключаются в выборе необходимых и достаточных данных для полноценной проверки работы. Например, при наличии одного эталонного набора оценка работы сводится к двоичному «правильно»/«неправильно», для более детальной дифференциации оценки требуется провести анализ влияния входных данных на выполнение программы, разработанной испытуемым.

В качестве примера рассмотрим типовые задачи по информатике – поиск минимума и сортировка массива. Первая задача звучит следующим образом: найти индекс последнего минимального элемента в целочисленном массиве из десяти элементов. На рис. 2 представлен интерфейс виртуальной лаборатории по информатике с решением задачи и использованием стиля кодирования, аналогичного языку С.

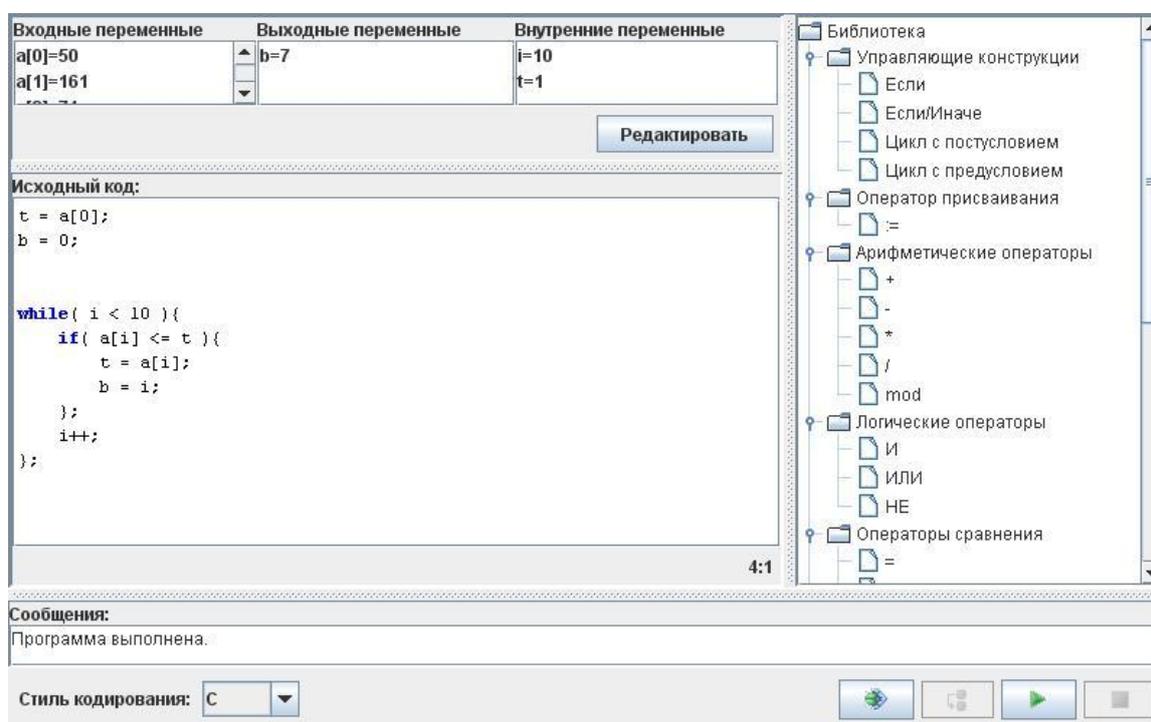


Рис. 2. Решение задачи о поиске минимума в массиве

В приведенных ниже примерах переменная a принадлежит набору входных данных, а b – эталонным выходным данным. В самом простом случае проверка может быть организована на основе одного набора:

$a=[10, 10, 10, 10, 10, 1, 10, 10, 10, 10]$ $b=5$

В таком случае не учитывается вариант, когда программа некорректно обрабатывает граничные случаи. Такие случаи могут быть связаны с ошибками в операторах управления (условия, циклы). В данной программе таких операторов два.

В инварианте цикла может быть допущена ошибка в условии или при инициализации переменной цикла. В таком случае может возникнуть исключение времени выполнения (выход за границы массива) или пропуск крайних элементов. Исключение времени выполнения в данной работе может быть обнаружено при локальной отладке программы в среде редактора кода. Пропуск крайних элементов найти сложнее, и программа в таком случае может считаться «частично» рабочей. Для проверки можно добавить два дополнительных набора:

a=[0, 1, 2, 3, 4, 5, 6, 7, 8, 9] b=0
 a=[1, 2, 3, 4, 5, 6, 7, 8, 9, 0] b=9

По заданию требовалось найти последний минимальный элемент. Ошибка в операторе условия может привести к нахождению первого из минимальных элементов. Для проверки этого условия требуется добавить следующий набор:

a=[0, 1, 2, 3, 4, 5, 6, 7, 8, 0] b=9

Таким образом, вместо двоичной оценки было получено четыре критерия для проверки.

Вторая задача звучит следующим образом: отсортировать целочисленный массив из десяти элементов по возрастанию. Решение задачи с использованием стиля языка C можно записать следующим образом:

```
i = 0; j = 0;
while( i < 10 ){
  j = j + 1;
  while( j < 10 ){
    if( b[j] < b[i] ){
      t = b[j];
      b[j] = b[i];
      b[i] = t;
    }
    j++;
  }
  i++;
}
```

Наиболее простой случай проверки можно описать одним набором:

a=[0, 1, 2, 4, 3, 5, 6, 7, 8, 9] b=[0, 1, 2, 3, 4, 5, 6, 7, 8, 9]

Данная программа содержит три оператора управления, следовательно, дополнительные эталонные наборы позволят обнаружить ошибки, связанные с этими операторами. Первый оператор – цикл с предусловием. Ошибка в инициализации переменной цикла или в операторе условия может привести к пропуску граничных элементов массива. Второй оператор также представляет собой цикл с предусловием, и инвариант цикла может сказаться на пропуске граничных элементов. Ошибка в записи третьего оператора условия может привести к обратному порядку сортировки или к лишним операциям обмена. Последнее сложно проверить, так как для таких простых программ разница во времени будет незначительной.

Для проверки инварианта первого цикла можно использовать следующий набор:

a=[1, 0, 2, 3, 4, 5, 6, 7, 8, 9] b=[0, 1, 2, 3, 4, 5, 6, 7, 8, 9]

Инвариант второго цикла влияет на обработку граничных элементов справа. Для проверки требуется следующий набор:

a=[0, 1, 2, 3, 9, 4, 5, 6, 7, 8] b=[0, 1, 2, 3, 4, 5, 6, 7, 8, 9]

Если автор лаборатории примет решение, что сортировка в порядке, обратном заданию, также является верной, можно добавить набор для проверки оператора условия:

a=[0, 1, 2, 3, 4, 5, 6, 7, 8, 9] b=[9, 8, 7, 6, 5, 4, 3, 2, 1, 0]

Таким образом, для дифференцирования оценки можно использовать четыре эталонных набора вместо одного для выявления не критических ошибок в алгоритмах.

Заключение

Одним из основных достоинств среды AcademicNT является наличие встроенного механизма для построения виртуальных лабораторий, которые необходимы для формирования и проверки навыков. В данной статье приведены схема взаимодействия модулей информационно-образовательной среды, принципы функционирования виртуальных лабораторий, требования к оформлению эталонных наборов данных и методика

разработки эталонных наборов для наиболее точного оценивания результатов виртуальных экспериментов.

Основные результаты статьи проиллюстрированы на примере виртуальной лаборатории «Многостилевой редактор кода», которая предназначена для проверки навыков реализации алгоритмов с возможностью кодирования на наиболее распространенных языках программирования.

Представленные в статье результаты прошли апробацию и продемонстрировали свою эффективность в ходе реализации следующих проектов:

- система дистанционного обучения СПбГУ ИТМО (<http://de.ifmo.ru>);
- система для проведения Интернет-олимпиад и экзаменов (<http://de.ifmo.ru/exam>).

Рассмотренный подход к разработке проверяющих наборов и виртуальная лаборатория используются в электронном курсе «Информатика». Использование дополнительных эталонных наборов позволило более точно оценить навыки студентов.

Литература

1. Вашенков О.Е., Лямин А.В., Тарлыков В.А. Оценивание результатов обучения в среде электронного учебно-методического комплекса по дисциплине «Когерентная оптика» // Оптика и образование – 2006: Сб. трудов конф. – СПб: СПбГУ ИТМО, 2006. – С. 70–71.
2. Васильев В.Н., Лисицына Л.С., Лямин А.В. Технология проведения ЕГЭ по информатике в компьютерной форме // Научно-технический вестник СПбГУ ИТМО. – 2007. – Выпуск 45. – С. 126–143.
3. Васильев В.Н., Лисицына Л.С., Лямин А.В. Сетевая технология проведения вступительных испытаний по информатике в режиме on-line // В сб.: Использование информационно-коммуникационных технологий в процессе оценки качества образования. – СПб, 2008. – С. 55–70.
4. Вашенков О.Е., Лямин А.В. Технология разработки виртуальных лабораторий в информационно-образовательной среде AcademicNT на примере работы по информатике // Проблемы разработки учебно-методического обеспечения перехода на двухуровневую систему в инженерном образовании: Материалы межвуз. н.-метод. конф. — М.: МИСиС, 2008. – С. 239–249.

Вашенков Олег Евгеньевич – Санкт-Петербургский государственный университет информационных технологий, механики и оптики, программист, vashenkov@cde.ifmo.ru

Лямин Андрей Владимирович – Санкт-Петербургский государственный университет информационных технологий, механики и оптики, кандидат технических наук, доцент, lyamin@mail.ifmo.ru