В.В. Плецев. Анализ и формирование содержания предмета изучения и методического обеспечения...

<table>
<thead>
<tr>
<th>Классы</th>
<th>Количество предложенных вопросов</th>
<th>Учащихся в классах</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>1</td>
<td>2</td>
</tr>
<tr>
<td>экспериментальные</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>контрольные</td>
<td>2</td>
<td>5</td>
</tr>
</tbody>
</table>

среднюю и выборочное среднее квадратичное отклонение.

В контрольной группе распределение учащихся по количеству правильных ответов должно описыва-ться кривой нормального распределения. В этом случае мы получим значение критерия $X^2$ меньше критического значения статистики $X^2_{крит} = 14.07$, взятого для уровня значимости $\alpha = 0.05$.

В экспериментальной группе распределение учащихся по количеству правильных ответов должно соответствовать кривой, отличной от нормального распределения. В этом случае мы получим значение критерия $X^2$ больше критического значения статистики $X^2_{крит} = 14.07$, взятого для уровня значимости $\alpha = 0.05$.

Полученные результаты показывают (таблица), что для уровня значимости $\alpha = 0.05$ в экспериментальной группе $X^2_{крит} = 27.6$, в контрольной группе $X^2_{крит} = 9.4$ при критическом значении статистики $X^2_{крит} = 14.07$. Это говорит (с достоверностью 95 %) о том, что изменения в представлениях учащихся о строении и схемах нашего мира обусловлены не слу-

чайными факторами, а носят закономерный характер.

Из анализа проведенного исследования можно заключить, что изучение интегрированных естественно-научных курсов в которых органично сочетаются теория и лабораторные работы комплексного характера, приводит к формированию следующих положений мировоззренческого характера:

- окружающая природа, человек, космос представляют собой единое целое;
- все объекты и процессы природы находятся во взаимосвязи (элемент – система);
- природа – источник всех наших знаний;
- фундаментальные законы природы существуют независимо от познающего и изменяющего ее человека.

Таким образом, можно заключить, что курс есте-

ствознания в старших классах благодаря своему интегративному характеру, органичному сочетанию теории и лабораторных работ способствует эффективному формированию современного мировоззрения учащихся.

Литература
2. Берестнева О.Г., Уразаев А.М., Муратова Е.А. и др. Математические методы в психологии. Томск, 2001.

УДК 378; 002:681.3

В.В. Плецев

АНАЛИЗ И ФОРМИРОВАНИЕ СОДЕРЖАНИЯ ПРЕДМЕТА ИЗУЧЕНИЯ И МЕТОДИЧЕСКОГО ОБЕСПЕЧЕНИЯ АДАПТИВНОГО ОБРАЗОВАНИЯ В ОБЛАСТИ ПРОГРАММИРОВАНИЯ

Уральский государственный экономический университет

При формировании содержания образования в области программирования (языки и средства про-

GRAMMирования, СУБД, CASE-средства) за основу взят государственный образовательный стандарт по специальности «Прикладная информатика (по областям)». Изучение ведется в рамках взаимосвязанных курсов «Информатика и программировании», «Высо-

КОУРОВНЕНЫЕ МЕТОДЫ ИНФОРМАТИКИ И ПРОГРАММИР-

ОВАНИЯ», «БАЗЫ ДАННЫХ», «РАЗРАБОТКА И СТАНДАРТИЗАЦИЯ ПРОГРАММНЫХ СРЕДСТВ И ИНФОРМАЦИОННЫХ ТЕХНОЛОГИЙ».

Анализ предмета изучения

Для определения содержания образования нужно проанализировать состояние этой области. Сформулируем кратко результаты такого анализа по основным разделам этой области. Ввиду массового применения Windows рассматривались средства, работающие под Windows. В перспективе можно рассмотреть средства на платформе Linux.

Visual Basic широко используется для интеграции офисных пакетов и различных типов СУБД.


C++ используется как самостоятельно, так и в средах RAD C++ Builder и Visual C++. Язык допускает эффективное использование практически всех возможностей компьютеров и операционных систем. Реализует кроссплатформенные разработки для различных типов архитектур и процессоров. В Visual Studio.NET язык C++ был расширен для обеспечения полной поддержки программирования под Microsoft.NET Framework. Любой класс, написанный на C++, становится доступен любому другому языку программирования, ориентированному на Microsoft.NET Framework. Отсюда вытекает межязыковое наследование классов. Язык отличается сложностью и необходимостью запоминания методов работы с каждым объектом.

C++ Builder есть не менее «гибрид» Delphi и C++: использует ту же среду RAD и библиотеку VCL, что и Delphi. C++ Builder имеет гораздо лучшую объектную модель. Существует возможность совместного использования компонентов (классов) из библиотек VCL и фундаментальных классов фирмы Microsoft (MFC). Библиотека VCL написана на Delphi, а не на C++, что увеличивает, по сравнению с Delphi, размер исполняемых файлов.

Microsoft Visual C++ обладает самыми большими возможностями, имеет очень хороший компилятор C++, отличный отладчик, мощную библиотеку MFC. Правка кода в режиме отладки и последующее его выполнение без полной перекомпиляции и перезаписи отладочной сессии. Имеется отличная, хорошо структурированная полная справочная система. Имеются кроссплатформенные средства. Возможна разработка Web-приложений, сервисов NT, DLL и статических библиотек, консольных приложений. Среда разработки ограничивается простым GU builder и средствами автоматического создания результатов его работы с программным кодом.

Visual C++, C++, C++ Builder рекомендуются для разработки сложных, дорогостоящих прикладных и общеиспользуюемых приложений.

C#-язык совмещает простоту Visual Basic с мощью C++. Сама фирма Microsoft отмечает следующие преимущества: синтаксис, аналогичный C++ и столь же мощный, как C++, но прост и безопасен; полное взаимодействие с сервисами COM; полная поддержка COM и платформы.NET Framework; контроль типов; расширенные метаданные, позволяющие создавать новые типы; поддержка XML. Программа на языке C# (и на других языках, поддерживающих .NET) транслируется в промежуточный код MSIL (Microsoft Intermediate Language), который в процессе выполнения приложения заменяется в памяти машины на машинный код, оптимизированный для данной платформы. Это обеспечивает перекомпилирование приложений без перетрансляции на различные платформы с использованием библиотек.NET Framework.


СУБД Access. Идеальная по простоте и удобству среда разработки. В качестве недостатков можно от-
метить отсутствие возможности формирования исполняемого файла, медленная работа с базами данных больших объемов, отсутствие средств администрирования. Рекомендуется использовать для разработки простых приложений и персональных баз данных с ограниченным объемом (несколько сотен тысяч записей) информации.

**СУБД Visual FoxPro.** СУБД имеет собственный язык программирования с возможностью создания исполняемого EXE-файла. Недостатки СУБД те же, что и у СУБД Access, плюс наличие ошибок при визуальном связывании таблиц. Рекомендуется использовать для разработки достаточно сложных приложений с ограниченным объемом (не более миллиона записей) работаемой информации.

**СУБД SQL Server.** Полнофункциональная серверная СУБД с прекрасными средствами защиты, администрирования, архивирования и восстановления баз данных, с мощным языком запросов Transact-SQL. Обеспечивает налаженное и компактное хранение и высокопроизводительную обработку баз данных больших объемов. SQL Server поддерживает тиражирование данных, параллельную обработку, отличается простой установки и использования. Пользователь компьютера-клиента с помощью сетевых средств своей операционной системы может устанавливать связи с компьютером-сервером, где установлен SQL Server. На компьютерах-клиентах с помощью персональных СУБД (Access, Visual FoxPro) или языков программирования (Visual Basic, Delphi, C++ Builder, Visual C++) через ODBC осуществляется доступ к удаленным базам данных.

Новая версия SQL Server под названием Yukon содержит NET Framework Common Language Runtime внутри ядра СУБД. SQL Server не имеет собственных средств разработки приложений, выполняется только под Windows. Обычно СУБД SQL Server используется для организации сервера баз данных, к которому организован доступ с клиентских машин средствами других СУБД или языков программирования. SQL Server рекомендуется использовать для создания очень больших централизованных или распределенных баз данных коллективного использования для средних и крупных предприятий.

**СУБД Oracle.** Современная универсальная полнофункциональная система управления реляционной базой данных, работающая в различных операционных средах и на различных типах компьютеров. Oracle представляет собой целый мир, включающий методы и средства по проектированию (CASE-средств), созданию и использованию распределенных баз данных, языки запросов (SQL, SQL*PLUS, PL/SQL), генераторы обработки и оперативного анализа данных (Oracle Express OLAP), конструкторы и мастера форм (Oracle Form Builder), запросов, отчетов (Oracle Report Builder) и приложений (Oracle Developer, Oracle Project Builder) и др. Oracle отличается сложностью и большой стоимостью приобретения. Рекомендуется использовать для очень больших централизованных или распределенных баз данных крупных предприятий.

**Классификация CASE-средств.** Рассмотрим классификацию по различным критериям.

1. Ориентация на этапы жизненного цикла:
   - средства анализа для построения и анализа моделей предметной области: BPwin (Logic Works), Design/IDEF (Meta Software);
   - средства анализа и проектирования для создания проектных спецификаций: CASE, Анализтикс (Макро-Проект), Vantage Team Builder (Cayenne), Silverrun (Silverrun Technologies), PRO IV (McDonnel Douglas);
   - средства разработки приложений: Delphi (Borland), PowerBuilder (SyBase), Developer/2000 (ORACLE), New Era (Informix), SQL Windows (Centura), Uniface (Compware), JAM (JYACC).

2. Функциональная полнота:
   - автоматизация отдельных этапов жизненного цикла: Erwin (Logic Works), CASE, Анализтикс (Макро-Проект), Silverrun (Silverrun Technologies), S-Designer (SPD);
   - степень зависимости от СУБД:
     - независимые, поддерживающие несколько форматов данных через ODBC: S-Designer (SPD), Powersoft, ERwin (Logic Works), Silverrun (Computer Systems Adviser Inc.);
     - встроенные в СУБД: Designer/2000 (ORACLE).

3. Тип используемой модели:
   - структурные, основанные на методах структурного и модульного программирования: Vantage Team Builder (Cayenne);
   - объектно-ориентированные Rational Rose (Rational Software), Object Team (Cayenne);

В табл. приводятся примерные экспертные оценки (по десятибалльной системе) возможностей и быстроты их освоения (оценки перечисляются в ячейке таблицы через запятую) для средств разработки приложений: Visual Basic (VB), Delphi (Del), C++ Builder (CB), Visual C++ (VC), Access (Acc), SQL Server (Sql), Oracle (Ora), Visual FoxPro (Fox). Присвоение этих оценок определяет значение рейтинга.

Для более точной оценки программного средства (ПС) можно использовать методику балльной оценки.

1. Составляется дерево требуемых показателей качества ПС.
2. Каждой терминальной вершине экспертом присваивается оценка по десятибалльной системе. Если оцениваемый показатель отсутствует в ПС, то устанавливается нулевая оценка.
### Экспертная оценка основных средств разработки приложений

<table>
<thead>
<tr>
<th>Оцениваемое средство</th>
<th>VB</th>
<th>Del</th>
<th>CB</th>
<th>VC</th>
<th>Acc</th>
<th>Sql</th>
<th>Ora</th>
<th>Fox</th>
</tr>
</thead>
<tbody>
<tr>
<td>Команды управления</td>
<td>7,7</td>
<td>8,6</td>
<td>8,6</td>
<td>9,5</td>
<td>2,9</td>
<td>5,5</td>
<td>6,9</td>
<td></td>
</tr>
<tr>
<td>Мастера форм</td>
<td>5,8</td>
<td>6,8</td>
<td>6,8</td>
<td>2,7</td>
<td>7,9</td>
<td>8,8</td>
<td>6,7</td>
<td></td>
</tr>
<tr>
<td>Мастера запросов и представлений</td>
<td>4,6</td>
<td>5,5</td>
<td>5,5</td>
<td>9,9</td>
<td>9,7</td>
<td>7,5</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Мастера отчетов</td>
<td>5,9</td>
<td>3,9</td>
<td>3,9</td>
<td>8,9</td>
<td>9,8</td>
<td>7,7</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Конструкторы форм</td>
<td>5,4</td>
<td>6,5</td>
<td>6,5</td>
<td>2,2</td>
<td>8,8</td>
<td>9,6</td>
<td>6,7</td>
<td></td>
</tr>
<tr>
<td>Конструкторы запросов и представлений</td>
<td>6,6</td>
<td>5,5</td>
<td>5,5</td>
<td>9,9</td>
<td>5,6</td>
<td>9,7</td>
<td>7,4</td>
<td></td>
</tr>
<tr>
<td>Конструкторы отчетов</td>
<td>4,5</td>
<td>8,7</td>
<td>8,7</td>
<td>7,9</td>
<td>9,8</td>
<td>7,7</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Команды формирования запросов и представлений (SQL)</td>
<td>5,4</td>
<td>6,5</td>
<td>6,5</td>
<td>7,5</td>
<td>7,8</td>
<td>8,7</td>
<td>9,6</td>
<td>6,5</td>
</tr>
<tr>
<td>Работа с файлами</td>
<td>7,8</td>
<td>9,9</td>
<td>8,7</td>
<td>9,6</td>
<td>5,6</td>
<td>7,6</td>
<td>6,7</td>
<td></td>
</tr>
<tr>
<td>Работа с базами данных (БД)</td>
<td>9,6</td>
<td>8,7</td>
<td>8,6</td>
<td>9,7</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Оперативная обработка (OLAP)</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>7,8</td>
<td>7,7</td>
<td></td>
<td>9,7</td>
</tr>
<tr>
<td>Обработка сбояных ситуаций</td>
<td>4,8</td>
<td>8,7</td>
<td>8,6</td>
<td>9,5</td>
<td>7,6</td>
<td>8,6</td>
<td>4,4</td>
<td></td>
</tr>
<tr>
<td>Администрирование БД</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>4,9</td>
<td>8,8</td>
<td>9,8</td>
<td>3,5</td>
</tr>
<tr>
<td>Копирование и восстановление БД</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>7,7</td>
<td>8,8</td>
<td>8,6</td>
<td>9,4</td>
</tr>
<tr>
<td>Взаимодействие с другими приложениями</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>5,7</td>
<td>6,7</td>
<td>7,5</td>
<td>7,6</td>
</tr>
<tr>
<td>Приближенная средняя оценка (без учета нулевых оценок)</td>
<td>6,7</td>
<td>7,7</td>
<td>7,6</td>
<td>7,5</td>
<td>6,8</td>
<td>8,7</td>
<td>6,7</td>
<td></td>
</tr>
<tr>
<td>Итоговый рейтинг (с учетом дробной части средних оценок)</td>
<td>37</td>
<td>46</td>
<td>42</td>
<td>35</td>
<td>52</td>
<td>45</td>
<td>54</td>
<td>34</td>
</tr>
</tbody>
</table>

3. Каждому исходному узлу дерева присваивает- ся оценка, равная средней оценке ее подчиненных узлов.

4. Полученная оценка для корневого узла дает нам интегрированную оценку (итоговый рейтинг) всего ПС.

В оценке ПС могут участвовать несколько независимых экспертов, и их оценки усредняются. При этом все основные положения ПС:

- категории пользователей: программисты, администраторы БД, квалифицированные пользователи, разрабатывающие приложения; конечные пользователи;

- удобство и простота использования: понятные процедуры установки ПС, удобный и унифицированный интерфейс пользователя, простота и полнота команд работы с базой, наличие и удобство оперативной справочной системы и документации;

- качество и возможности средств разработки: мастеров, экспертов, конструкторов, построителей таблиц, форм, запросов, представлений, отчетов;

- модель представления данных: реляционная, объектно-ориентированная и др.;

- качество средств защиты: контроль на уникальность записей по первичному ключу, контроль целостности связей, проверка значений полей;

- качество коммуникационных средств: поддержка различных сетевых протоколов и интерфейсов (SQL, ODBC, IDAPI, SAA), наличие средств групповой работы (языки программирования, администрирование разработкой, разграничение полномочий, защита от несанкционированного доступа), возможность реорганизации БД;

- физические разработчик: солидность, время существования, опыт, специализация, доступность, наличие «горячей линии», размер и финансовое положение;

- стоимость и технические характеристики: общие параметры, ограничения, типы данных, типы запросов, многопользовательская работа;

- производительность: выборка, просмотр, загрузки, индексации, обновления, чтения и записи с произвольной выборкой, генерации запроса, отчета и др.

На формирование содержания предмета изучения влияют варианты разработки приложений:

1. Использование исключительно средств (если они имеются) СУБД (Oracle, Visual FoxPro). Достоинство: высокая эффективность обработки данных. Недостатки: трудно сменить в приложении тип СУБД, сложность и трудоемкость освоения разнообразных средств СУБД. Этот вариант рекомендуется для больших предприятий, использующих один тип СУБД Oracle или SQL. Достоинства: целостность, возможность смены в приложении типа СУБД, использование уже освоенных пользователем языков программирования, упрощение процесса освоения.
СУБД. Недостатки: более низкая эффективность обработки данных. Этот вариант рекомендуется для предприятий с ограниченным персоналом на обслуживание баз данных с различными типами СУБД и для предприятий-разработчиков пакетов прикладных программ (ППП), легко настраивающих на тип СУБД.

3. Комбинированый двух вариантов. Для каждого приложения или его части выбирают наиболее подходящий вариант.

Таким образом, каждый из вариантов разработки приложений и средств программирования СУБД имеет свои преимущества, недостатки и области эффективного применения. Это обосновывает необходимость их изучения и освоения.

Особенности адаптивного образования

Адаптивный подход позволяет эффективно развивать способности и профессиональные навыки каждого студента с учетом его индивидуальных особенностей, исходного и результирующего уровня подготовки.

Отметим основные особенности такого подхода:
– систематичность и последовательность в работе со студентом на максимально доступном для него уровне;
– развитие самостоятельности и творческого уровня студента, умения эффективной работы с технической литературой и электронными средствами обучения и контроля знаний;
– интенсивное развитие способностей студентов;
– разработка индивидуальных программ обучения и самостоятельного планирования;
– контроль хода обучения с возможной корректировкой программы обучения;
– создание адаптируемого учебно-методического обеспечения для самостоятельной работы;
– гибкое и оптимальное сочетание индивидуальных и групповых форм учебной работы.

Принципы формирования адаптивного методического обеспечения

Содержание предмета изучения должно задаваться методическим обеспечением. Исходя из необходимости адаптации процесса обучения, сформулируем основные принципы разработки такого обеспечения.

Системность. Учет взаимосвязей между участвующими дисциплинами, исключение дублирования учебного материала. Согласованность и унификация понятий, правил изложения и оформления учебного материала.

Самостоятельность (кейс-обучение). Все, что нужно для изучения и контроля (включая электронное тестиранние) полученных знаний и навыков, содержится в этом обеспечении.

Развитие. Оперативное изменение и дополнение учебного материала с учетом современного состояния изучаемой области.

Модульность. Описание групп взаимосвязанных тем выделяется в отдельные тематические модули (блоки):
– основы теории программирования;
– основы теории проектирования баз данных;
– СУБД Access, Visual FoxPro, SQL Server, Oracle;
– разработка приложений. CASE-средства BPwin, ERwin, Rational Rose.

Классификация. Декомпозиция учебного материала по специализации (видам программных средств) и по уровню изучения и освоения. Укрупненно сформулируем результаты классификации. По специализации программных средств были выбраны все вышеперечисленные средства, кроме C# (средство только формируется и пока мало используется) и Java, Visual J++ (достаточно узкая специализация). Примерное соответствие уровней изучения и уровней учебных заданий и содержания учебного материала будет принято следующим:

1. Начальный уровень (студенты колледжей по некомпьютерным специальностям) – основные команды, функции, классы, объекты, методы, свойства, события; визуальные мастера и построитель объектов. Студенты учатся составлять простые программы на Visual Basic и (или) Delphi, базы данных с СУБД Access.

2. Основной уровень (студенты вузов, институтов повышения квалификации кадров по некомпьютерным специальностям и студенты колледжей по компьютерным специальностям) – дополнительные команды, функции, классы, объекты, методы, свойства, события; объектно-ориентированное программирование; визуальные средства в полном объеме; все возможные конструкторы (таблиц, форм, диаграмм, запросов и отчетов); средства создания, использования и администрирования баз данных. Студенты создают приложения и Internet-приложения.

Заключение

Путем комбинации учебных модулей, тем и уровней изучения можно создать индивидуальные образовательные траектории, ориентированные на различные формы обучения, аудиторию и на отдельных студентов. Таким образом, реализуется принцип адаптации.

Для всего образовательного процесса в области программирования и баз данных в соответствии с указанными принципами были разработаны учебники [1, 2] и пособия [3, 4], в которых учебный материал классифицирован по специализации и уровням изучения. Для контроля знаний студентов были разработаны примеры, упражнения и задачи для программирования, темы практических, контрольных и курсовых работ, тесты для электронного тестирования, вопросы для зачетов и экзаменов.

Разработанное методическое и программное обеспечение было применено в учебном процессе в нескольких высших и средних учебных заведениях с различными формами обучения и показало свою эффективность.

Литература

УДК 378

О.Г. Берестнева*, О.В. Марухин†, А.М. Уразаев**

ИНФОРМАЦИОННАЯ ТЕХНОЛОГИЯ КОНТРОЛЯ КАЧЕСТВА ОБРАЗОВАНИЯ В ВЫСШЕЙ ШКОЛЕ

*Томский политехнический университет
**Томский государственный педагогический университет

Создание интеллектуальной системы компьютерного тестирования не дань моде, а необходимость, связанная с требованиями сокращения временных и стоимостных затрат на обучение субъектов образовательного процесса. Она должна обеспечивать упрощение процедур тестирования, тщательного анализа и удобного представления его результатов в различных проблемных областях.

Для эффективного анализа результатов тестирования вне зависимости от задачи, решаемой пользователем, особое значение приобретают методы визуализации, обеспечивающие для разных тестов единую форму представления конечной информации в виде отображений, адекватных зрительному восприятию человека и удобных для однозначного толкования полученных результатов. Так как результаты тестирования компонуются в виде числовых таблиц, то методы визуализации должны способствовать наглядному изображению этих таблиц в графическом виде.

В последнее время разработано достаточно большое количество интеллектуальных систем и методик для оценки качества подготовки специалистов. К числу этих систем относятся такие, например, как «Экспертные обучающие системы как инструмент педагогического общения» [1], целью которой является разработка модели экспертной обучающей системы: «Автоматизированная обучающая система по английскому языку с использованием мультимедиа и гипертекстовых учебных пособий» [2], целью которой стало создание автоматизированных обучающих компьютерных курсов английского языка. Следует указать также на систему «Компьютерного информационного обеспечения учебного процесса» [3], которая используется для создания системы компьютерных программ при автоматизации учебного процесса.

При сравнительном анализе приведенных выше систем можно отметить, что в качестве исходной информации в них используются среднестатистические данные, достоверность которых практически невозможна оценить для конкретных тестируемых групп и контролировать. Кроме того, в этих случаях дополнительно вводятся оценочные показатели, установленные, как правило, экспертным путем. Таким образом, эти используемые сегодня подходы к созданию систем имеют ряд недостатков, а именно: субъективизм, низкую достоверность исходной информации и, как следствие, низкое качество результатов оценки.

Рассматриваемый в настоящей работе подход, лишён этих недостатков, поскольку выводы основа-