

Образование призвано формировать систему личностных ценностей, адекватных ценностям общечеловеческим: жизнь – высшее благо, благополучие – целенаправленный труд над собой и в обществе, богатство – соотнесение желаемого и реально-рационально-потребного, благородство – проявление сопричастности, сопереживания, возможное только в результате осознания себя частью окружающего мира во всех его проявлениях. Таким образом, интеграция экологического и экономического образования, создает возможности для формирования уникального комплексного образовательного средства в виде эколого-экономического образования.

Литература

1. Амэнд А. Ф. Дидактические основы непрерывного эколого-экономического общего образования: Дисс. в виде науч. докл. ... д-ра пед. наук. – Екатеринбург, 1997.

2. Амэнд А. Ф. Теория и практика непрерывного эколого-экономического образования: Монография. – Челябинск: Издательство Челябинского государственного педагогического университета «Факел», 1996.

3. Концепция модернизации российского образования на период до 2010 года // Приложение к журналу «Директор школы», 2002, № 1.

4. Поляков В. А., Сасова И. А. Непрерывное экономическое образование молодежи // Педагогика, 1994, № 4.

5. Реймерс Н. Ф. Экология: Теории, законы, правила, принципы и гипотезы. – М., 1994.

6. Рябинина Н. П. Теория и практика подготовки будущего учителя к осуществлению эколого-экономического образования школьников: Дис. ... д-ра пед. наук. – М., 1998.

7. Экономическое образование в школе: Материалы Международного семинара. – СПб.: Образование, 1995.

УДК 681.3 (045)
ББК 435.236

КЛАССИФИКАЦИЯ И АНАЛИЗ СОДЕРЖАНИЯ ОБРАЗОВАНИЯ В ОБЛАСТИ ПРОГРАММИРОВАНИЯ И БАЗ ДАННЫХ

В. В. Плещев

Классификация содержания образования производится по двум критериям: уровням изучения и специализации программных средств.

Выделим три основных уровня изучения и их примерное соответствие типам учебных заведений:

1. Начальный (ознакомительный) уровень – студенты колледжей (техникумов) по некомпьютерным специальностям.

2. Основной уровень – студенты вузов и институтов повышения квалификации кадров по не компьютерным специальностям и студенты колледжей (техникумов) по компьютерным специальностям.

3. Углубленный уровень – студенты вузов и институтов повышения квалификации кадров по компьютерным специальностям.

Для определения и научного обоснования структуры и содержания подготовки специалистов в указанной области нужно проанализировать современное состояние этой области. Сформулируем кратко результаты такого анализа по основным разделам этой области.

Среды быстрой разработки приложений (RAD). Большинство современных языков программирования имеют собственные среды быстрой разработки приложений. Для языков *Visual Basic*, *C#* такие среды являются частью самих языков программирования. Для языков *Object Pascal*, *C++* среды *RAD* достаточно автономны и имеют собственные наименования *Delphi*, *C++ Builder*, *Visual C++* соответственно. Современные среды программирования автоматизируют все процессы жизненного цикла программных средств (ЖС ПС).

Под системой или средством программирования обычно понимается сам язык программирования и его среда *RAD*. Имя системы программирования определяется именем среды *RAD*.

Visual Basic [5; 14] широко используется для интеграции офисных пакетов и различных типов СУБД. *Visual Basic* имеет простую и удобную среду *RAD*, допускает работу с различными типами СУБД. Позволяет создавать *Web*-страницы совместно с *HTML* [14; 20] и *ASP* [14]. Приложения на *Visual Basic* имеют более низкую скорость выполнения и требуют подключения *DLL*-файлов. *Visual Basic* поддерживается платформой *Microsoft .NET* при создании корпоративных решений. *Visual Basic* рекомендуется для быстрой разработки простых и недорогих приложений и *Internet*-приложений (средствами *VBScript* – подмножество *Visual Basic*).

Delphi является мощной системой программирования [3; 7; 15; 18]. Простота, скорость и эффективность объясняют популярность *Delphi*. *Delphi* имеет один из самых быстрых компиляторов, порождающий неплохой объектный код. Программы, написанные на *Delphi*, не требуется снабжать дополнительными библиотеками. В *Delphi* имеется собственная мощная и высокопроизводительная СУБД *Interbase*. В версии *Delphi 7 Studio* (проект *Aurora*) появились новые возможности (термин *Delphi* используется уже в качестве языка программирования, а не только как название среды *RAD*). *CASE*-инструмент *ModelMaker*, базирующийся на стандартах языка проектирования объектно-ориентированных приложений *UML (Unified Modeling Language)* и совместимый с *Rational Rose* [22]. Инструмент моделирования *Bold for Delphi*, реализующий архитектуру *MDA (Model Driver Architecture)*. Данная архитектура позволяет оперировать целыми бизнес-объектами. Изменение модели производится без изменения ис-

ходного кода. Средство *IntraWeb* позволяет создавать *Web*-системы по аналогии создания обычных форм и приложений. Мощный генератор отчетов *Rave Reports Borland Edition*. Среда разработки поддерживает синтаксис языков *HTML*, *XML*, *C++*, *C#*. *Delphi* имеет средства для интеграции *Windows*-, *Linux*- и *NET*-приложений. Новый проект *Galileo* позволит создавать многоплатформенные приложения и будет поддерживать языки *Delphi*, *C#* и *Visual Basic*. *NET*. Недостатком можно считать статическое присоединение библиотеки визуальных компонентов (*VCL*) и компонентов к исполняемому файлу.

Delphi рекомендуется для быстрой разработки приложений.

C++ [16] 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 кросс-платформенные 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 межъязыковое наследование классов. Язык отличается сложностью и необходимостью запоминания методов работы с каждым объектом.

C [4] - подмножество языка программирования *C++*. Практически не используется в настоящее время.

C++ Builder [1; 15] есть некий «гибрид» *Delphi* и *C++* [16]: использует ту же среду *RAD* и библиотеку *VCL*, что и *Delphi*. *C++ Builder* имеет гораздо лучшую объектную модель. Директивы препроцессора предоставляют лучшие возможности управления кодом. 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++ [17] обладает самыми большими возможностями; имеет очень хороший компилятор *C++*; отличный отладчик, мощная библиотека *MFC*. Правка кода в режиме отладки и последующее его выполнение без полной перекомпиляции и прерывания отладочной сессии. Технология подсказок, работающая быстрее; полноценный браузер классов и менеджер исходного кода. Имеется отличная, хорошо структурированная полная (более гигабайта) справочная система. Имеются кросс-платформенные средства. Возможна разработка *Web*-приложений, сервисов *NT*, *DLL* и статических библиотек, консольных приложений. Среда разработки ограничивается простым *GUIbuilder* и средствами автоматического связывания результатов его работы с программным кодом.

Visual C++, *C++*, *C++ Builder* рекомендуются для разработки сложных, дорогостоящих и высококачественных прикладных и общесистемных 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; расширяемые метаданные, позволяющие создавать новые типы; поддержка *XML*. Программа на языке *C#* (и на других языках, поддерживаемых *.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 переносимость приложений без перетрансляции на различные платформы с использованием универсальных библиотек. *.NET Framework*.

Microsoft Visual J++ имеет среду *RAD*, аналогичную среде *Visual Basic*. Позволяет создавать *ActiveX*-элементы, которые можно использовать в *Visual Basic*, *Visual C++*, *Delphi*, *C++ Builder*. Допускается использование *ActiveX/COM*-компонентов и *Windows-specific* функций, что позволяет реализовать, например, доступ к *ADO*, преобразование *JavaBeans* в *ActiveX*-компоненты, классы поддержки *DHTML*, компоненты работы с базами данных. Сам язык *Java* имеет следующие преимущества: безопасность, легкость интеграции в *Internet/Intranet*, универсальность, простота, высокая скорость исполнения промежуточного байт-кода, автоматическая «сборка мусора», отсутствие указателей, что позволяет избежать трудно обнаруживаемых ошибок и несанкционированного доступа к оперативной памяти. Однако, *Microsoft Visual J++* использует библиотеку *WFC (Window Foundation Classes)*, что привязывает ее к *Windows* и не поддерживает стандарт визуального *Java*-программирования *JavaBeans*. *Visual J++* рекомендуется для разработки *Internet*-приложений.

СУБД Access [19]. Идеальная по простоте и удобству среда разработки. В качестве недостатков можно отметить отсутствие возможности формирования исполнимого файла, медленная работа с базами данных больших объемов, отсутствие средств администрирования. *Access* рекомендуется использовать для разработки простых приложений и персональных баз данных с ограниченным объемом (несколько сотен тысяч записей) информации для небольших предприятий.

СУБД Visual FoxPro [6; 10; 11; 13]. СУБД имеет собственный язык программирования с возможностью создания исполнимого *EXE*-файла. Недостатки СУБД те же, что и у СУБД *Access*, плюс наличие ошибок при визуальном связывании таблиц. *Visual FoxPro* рекомендуется использовать для разработки достаточно сложных приложений с ограниченным объемом (не более миллиона записей) обрабатываемой информации для небольших и средних предприятий. Несмотря на неопределенное будущее и на архаичность, изучение этой СУБД целесообразно из-за наличия большого количества сопровождаемых, ранее разработанных, баз данных и приложений на небольших предприятиях с малоомощными компьютерами.

СУБД SQL Server [9; 11]. Полнофункциональная серверная СУБД с прекрасными средствами защиты, администрирования, архивирования и восстановления базы данных, с мощным языком запросов *Transact-SQL*. Обеспечивает надежное и компактное хранение и высокопроизводительную обработку баз данных больших объемов. *SQL Server* поддерживает: тиражирование данных, параллельную обработку, отличается простотой управления и использования. Пользователь компьютера-клиента с помощью сетевых средств своей операционной системы может устанавливать связь 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 базам данных, размещенным на сервере.

Новая версии *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 [11]. Современная универсальная полнофункциональная система управления реляционной базой данных, работающая в различных операционных средах (*Windows NT, Unix, Linux, Solaris, OS/2, SVM* и др.) и на различных типах компьютеров. *Oracle* представляет собой целый мир, включающий методы и средства по проектированию (*CASE-технологии*), 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 (*Report Builder*) и приложений (*Oracle Developer, Oracle Project Builder*) и др. Если пользователь имеет опыт работы с языками программирования и с другими СУБД, особенно с СУБД *SQL Server*, то ему можно достаточно быстро освоить основы работы с СУБД *Oracle*, так как многие методы проектирования баз данных, приемы и средства работы с СУБД достаточно аналогичны (*SQL, Transact SQL и PL/SQL, SQL Server Enterprise Manager и Oracle Enterprise Manager*, хранимые процедуры, триггеры, генераторы форм и отчетов и др.). *Oracle* отличается сложностью и большой стоимостью приобретения. *Oracle* рекомендуется использовать для очень больших централизованных или распределенных баз данных крупных предприятий.

CASE-средства. Обычно к *CASE-средствам* относят программное средство, автоматизирующее совокупность процессов ЖЦПС.

CASE-средство содержит следующие компоненты:

- Репозиторий. Он должен обеспечивать хранение версий проекта и его отдельных компонентов, синхронизацию поступления информации от различных разработчиков при групповой разработке, контроль метаданных на полноту и непротиворечивость.

- Графические средства анализа и проектирования, обеспечивающие создание и редактирование иерархически связанных диаграмм.

- Средства разработки приложений, включая системы программирования и генераторы кодов, конфигурационного управления, документирования, тестирования, управления проектом, реинжиниринга.

Классификация CASE-средств 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. Аналитик (МакроПрожект), *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 (Compuware)*, *JAM (ЯУАСС)*.

2. Функциональная полнота

- Автоматизация отдельных этапов жизненного цикла: *Erwin (Logic Works)*, CASE. Аналитик (МакроПрожект), *Silverrun (Silverrun Technologies)*, *S-Designer (SPD)*.

- Интегрированные системы, поддерживающие весь жизненный цикл: *Vantage Team Builder (Cayenne)*, *Designer/2000* с системой *Developer/2000 (ORACLE)*.

3. 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)*.

4. Тип используемой модели

- Структурные, основанные на методах структурного и модульного программирования: *Vantage Team Builder (Cayenne)*.

- Объектно-ориентированные *Rational Rose (Rational Software)*, *Object Team (Cayenne)*.

- Комбинированные, поддерживающие одновременно обе модели: *Designer/2000 (ORACLE)*.

В учебный материал целесообразно включить описание наиболее доступных и популярных CASE-средств *Rational Rose 2002* [12; 22], *BPwin 4.0* [12; 21] и *ERwin 4.0* [12; 21].

Экспертная оценка программных средств

В следующей таблице приводятся примерные экспертные оценки (по десятибалльной системе) возможностей и скорости 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)*. Произведение этих оценок определяет значение рейтинга, который учитывает возможность программного средства и скорость его освоения.

Экспертная оценка основных средств разработки приложений

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

Выводы. Таким образом, каждое из средств программирования и СУБД имеют свои преимущества, недостатки и области эффективного применения. Это обосновывает необходимость их изучения и освоения.

При формировании содержания подготовки специалистов в указанной области за основу взят ГОС по специальности «Прикладная информатика (по областям)». Изучение ведется в рамках взаимосвязанных курсов «Информатика и программирование», «Высокоуровневые методы информатики и программирования», «Базы данных», «Разработка и стандартизация программных средств и информационных технологий».

Структура и содержание подготовки должны задаваться в методическом обеспечении. Исходя из необходимости индивидуализации процесса обучения и классификация содержания, сформулируем основные принципы разработки такого обеспечения.

Принцип классификации. Декомпозиция учебного материала по специализации (видам программных средств) и по уровням изучения и освоения. По специализации программных средств были выбраны все выше описанные средства, кроме С# (средство только формируется и пока мало используется) и *Java*, *Visual J++* (узкая специализация – *Internet*-приложения).

Примерное соответствие уровней изучения содержанию учебного материала было принято следующим (соответствие уточняется при разработке рабочих учебных программ по соответствующим дисциплинам):

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

2. Основной уровень – дополнительные команды, функции, классы, объекты, методы, свойства, события; объектно-ориентированное программирование; визуальные средства в полном объеме, всевозможные конструкторы (таблиц, форм, диаграмм, запросов и отчетов), средства создания, использования и администрирования баз данных. Студенты создают приложения и *Internet*-приложения с использованием различных языков и средств программирования, СУБД и CASE-средств.

3. Углубленный уровень – командные средства создания и использования классов, объектов, свойств, методов, событий, баз данных, запросов, форм и других элементов и технологий (например, *ADO*, *COM/DCOM*, *OLE DB*, *ODBC*, *CORBA*, *MIDAS*, *OLE*, *OLAP*, *DLL*-библиотеки, *ActiveX*-элементы, *Internet*). Студенты создают сложные и объемные приложения и *Internet*-приложения с использованием средств разработки приложений (*Visual Basic*, *Delphi*, *C++ Builder*, *Visual C++*, *Assembler*, *VBScript*, *HTML*, *ASP*, *Dreamweaver*, *Crystal Reports* и др.), СУБД (*Visual FoxPro*, *Access*, *SQL Server*, *Oracle*) и CASE-средств (*BPwin*, *ERwin*, *Rational Rose* и др.).

Принцип самодостаточности (кейс-обеспечение). Все, что нужно для изучения и контроля (включая электронное тестирование) полученных знаний и навыков (в соответствии с требованиями ГОС) содержится в этом обеспечении.

Принцип современности. Отражение современного состояния изучаемой области.

Принцип системности. Учет взаимосвязей между указанными дисциплинами, исключение дублирования учебного материала. Согласованность и унификация понятий, правил изложения и оформления учебного материала.

Принцип модульности. Описания групп взаимосвязанных тем выделяются в отдельные тематические учебные модули (блоки):

- Основы теории программирования [14].
- Системы программирования [14; 15]. *Visual Basic* [5; 14], *Object Pascal*, *Delphi* [3; 7; 15; 18], *C*, *C++*, *C++ Builder*, *Visual C++* [1; 4; 15–17].
- Основы теории проектирования баз данных [8; 11].
- СУБД [11]. *Access* [11; 19], *Visual FoxPro* [6; 10; 11; 13], *SQL Server* [9; 11], *Oracle* [11].
- Разработка приложений [2; 12]. CASE-средства *BPwin*, *ERwin* [12; 21], *Rational Rose* [12; 22], *MacroFox* [13].

Заключение

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

Для всего образовательного процесса в области программирования и баз данных в соответствии с указанными принципами были разработаны учебники [14; 15] и пособия [10–13], 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 (программы тестирования, CASE-средство *MacroFox* [13], примеры программ и др.) были применены в учебном процессе в нескольких высших и средних учебных заведениях с различными формами обучения и показали свою эффективность.

Литература

1. Архангельский А. Я. Программирование в *C++ Builder* 5. 2-е изд. – М.: БИНОМ, 2001.
2. Вендеров А. М. Проектирование программного обеспечения экономических информационных систем: Учебник. – М.: Финансы и статистика, 2002.
3. Гофман В. Э., Хомоненко А. Д. Работа с базами данных в *Delphi* 5. – СПб.: БХВ-Петербург, 2000.
4. Дейтел Х. М., Дейтел П. Дж. Как программировать на С: Пер. с англ. – М.: БИНОМ, 2000.
5. Евангелас П. *Visual Basic* 6. Руководство разработчика: Пер. с англ. – Санкт-Петербург: BHV, 2000.
6. Каратыгин С. А., Тихонов А. Ф., Тихонова Л. Н. *Visual FoxPro* 6. – М.: ЗАО «Издательство БИНОМ», 1999.
7. Козлов А. В. Программирование для Интернет в *Delphi* 5. – М.: БИНОМ, 2001.
8. Коннолли Томас и др. Базы данных: проектирование и сопровождение. Теория и практика, 2-е изд.: Пер. с англ.: Уч. пос. – М.: Издательский дом «Вильямс», 2000.

9. Мамаев Е., Шкорина А. *Microsoft SQL Server 2000* для профессионалов. – СПб: Питер, 2001.
10. Плещев В. В. Основы программирования в СУБД *FoxPro*: Учеб. пособие. – Екатеринбург: Издательство Уральского государственного экономического университета, 1999.
11. Плещев В. В. Базы данных. *Visual FoxPro, Access, SQL Server, Oracle*: Учеб. пособие. – Екатеринбург: Издательство Уральского государственного экономического университета, 2002.
12. Плещев В. В. Разработка и стандартизация программных средств и информационных технологий: Учеб. пособие. – Екатеринбург: Издательство Уральского государственного экономического университета, 2003.
13. Плещев В. В. Пакет прикладных программ автоматизации программирования в среде СУБД *FoxPro*. Библиотека и генератор программ супероболочки «*MacroFox*». Учеб. пособие. – Екатеринбург: Издательство Уральского государственного экономического университета, 1997.
14. Плещев В. В. Информатика и программирование. *Quick Basic* и *Visual Basic 6.0, VBScript, HTML, ASP, Dreamweaver, Crystal Reports* с примерами и упражнениями: Учебник. – Екатеринбург: Издательство Уральского государственного экономического университета, 2002.
15. Плещев В. В. Высокоуровневые методы информатики и программирования. *Delphi 5, C++ Builder 5, Visual C++* с примерами и упражнениями: Учебник. 2-е изд., испр. и доп. (реком. УМО Минобразования РФ). – Екатеринбург: Издательство Уральского государственного экономического университета, 2002.
16. *Страуструп Б.* Язык программирования *C++*: Пер. с англ. – М.: БИНОМ, 1999.
17. Секунов Н. Ю. Самоучитель *Visual C++*. – СПб.: БХВ-Санкт-Петербург, 2000.
18. Тейкейра С., Пачеко К. *Delphi 5*. Руководство разработчика: В 2 т.: Пер. с англ. – М.: Вильямс, 2000.
19. Михеева В. Д., Харитоновна И. А. *Microsoft Access 2002*. – СПб.: БХВ-Петербург, 2002.
20. Холцилаг Молли Э. Использование *HTML 4* (Специальное издание). Пер. с англ. – Киев-СПб.: Вильямс, 2001.
21. Маклаков С. В. *VPwin, ERwin*. CASE-средства разработки информационных систем. – М.: ДИАЛОГ-МИФИ, 1999.
22. Уэнди Богс, Майкл Богс. *UML Rational Rose*. – М.: ЛОРИ, 2000.