

ПРИНЦИПЫ СОЗДАНИЯ, СТРУКТУРА И СОДЕРЖАНИЕ АДАПТИРУЕМОГО ПРОФЕССИОНАЛЬНОГО ОБРАЗОВАНИЯ В ОБЛАСТИ ПРОГРАММИРОВАНИЯ И БАЗ ДАННЫХ

В.В. Плещев

В статье описываются принципы и средства адаптации образования в области программирования и баз данных. Дается сравнительный анализ средств программирования (Visual Basic, Delphi, C#, C++, Builder, Visual C++, Visual J++) и систем управления базами данных (Access, SQL-Server, Oracle, Visual FoxPro). Приводится классификация учебного материала по специализации, уровням изучения (начальный, основной и углубленный) и по тематическим модулям и темам.

Обычно выделяются три типа образования: базовое (начальное, среднее и высшее) профессиональное, дополнительное (институты повышения квалификации кадров) и самообразование. Базовое образование отвечает за освоение накопленных знаний и опыта, а дополнительное - за сопровождение настоящего. Все вместе они вместе составляют понятие непрерывного образования. Формы обучения могут быть самыми различными, например: очная, заочная, дистанционная, сокращенная.

Адаптация профессионального образования позволяет эффективно развивать способности и профессиональные навыки каждого студента с учетом его индивидуальных особенностей, исходного и результирующего уровней подготовки.

Отметим основные принципы адаптации образования:

- систематичность и последовательность в работе со студентом на максимально доступном для него уровне;
- развитие самостоятельности и творчества у студента, умения эффективной работы с технической литературой и с электронными средствами обучения и контроля знаний;
- интенсивное развитие способных студентов;

Учебными заведениями разработаны и внедрены различные приемы в организации такого учебного процесса, например:

- 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.

Что касается Государственных образовательных стандартов профессионального образования по специальностям (ГОС), то можно отметить следующее.

- Они задают только примерное, минимально необходимое и укрупненное содержание дисциплин. Например, для специальности "Прикладная информатика (по областям)", не указываются конкретные изучаемые языки и средства программирования, СУБД, CASE-средства.

- Для дополнительного образования или самообразования требования к содержанию дисциплин часто определяются индивидуально образовательными стандартами учебного заведения (ОСУЗ) или даже самими обучающимися (при самообразовании).

Таким образом, имеется достаточный простор при формировании детального содержания и учебных программ дисциплин и, следовательно, широкие возможности (в особенности при дополнительном образовании, повышении квалификации, самообразовании) по индивидуализации обучения.

Структура и содержание подготовки должны задаваться методическим обеспечением. Исходя из необходимости адаптации образования, сформулируем дополнительные **принципы разработки обеспечения.**

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

Современность. Отражение текущего состояния изучаемой области.

Классификация. Декомпозиция учебного материала по специализации (например, по видам программных средств) и по уровням изучения и освоения.

ПРИНЦИПЫ СОЗДАНИЯ, СТРУКТУРА И СОДЕРЖАНИЕ АДАПТИРУЕМОГО ПРОФЕССИОНАЛЬНОГО ОБРАЗОВАНИЯ В ОБЛАСТИ ПРОГРАММИРОВАНИЯ И БАЗ ДАННЫХ

Модульность. Описание групп взаимосвязанных тем выделяется в отдельные тематические модули (блоки). Путем комбинации учебных модулей, тем и уровней изучения можно создать индивидуальные образовательные траектории, ориентированные на различные формы обучения, аудиторию и на отдельных студентов. В этом и заключается сущность *адаптации образования*.

Самодостаточность (кейс-обеспечение). Все, что нужно для изучения и контроля (включая электронное тестирование) полученных знаний и навыков (в соответствии с требованиями ГОС и ОСУЗ) содержится в этом обеспечении. Наличие компакт-дисков с демонстрационными примерами, программами, иллюстрациями и другими дополнительными материалами.

Рассмотрим применение этих принципов на примере разработки учебников [14, 15] и учебных пособий [10-12].

Принцип системности

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

Принцип современности

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 Auriga) 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 исходного 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 использо-

вание 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 мощностью 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 поддерж-

ка 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 WFS (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 ограни-

ПРИНЦИПЫ СОЗДАНИЯ, СТРУКТУРА И СОДЕРЖАНИЕ АДАПТИРУЕМОГО ПРОФЕССИОНАЛЬНОГО ОБРАЗОВАНИЯ В ОБЛАСТИ ПРОГРАММИРОВАНИЯ И БАЗ ДАННЫХ

ченным 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 маломощными компьютерами.

СУБД SQL Server [9, 11]. Полнофункциональная серверная СУБД с прекрасными средствами защиты, администрирования, архивирования и восстановления базы данных, с мощным языком запросов Trans-act-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 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 [11]. Современная универсальная полнофункциональная СУБД, 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. 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

Рассмотрим классификацию по различным критериям.

1. Ориентация на этапы жизненного цикла
 - Средства анализа для построения и анализа моделей предметной области: BPwin (Logic Works) [12, 21], Design/DEF (Meta Software).
 - Средства анализа и проектирования для создания проектных спецификаций (CASE. Аналитик (МакроПрожект), Vantage Team Builder (Cayenne), Silverrun (Silverrun Technologies), PRO-IV (McDonnell Douglas).
 - Средства разработки приложений: Delphi (Borland), PowerBuilder (SyBase), Developer/2000 (ORACLE), New Era (Informix), SQL Windows (Centura), Unifase (Compuware), JAM (JYACC).
2. Функциональная полнота
 - Автоматизация отдельных этапов жизненного цикла: Erwin (Logic Works) [12, 21], 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) [12, 22], Object Team (Cayenne).

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

В следующей таблице 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). Произведение этих оценок определяет значение рейтинга.

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

Принцип классификации

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 не компьютерным 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.

Укрупненно сформулируем результаты классификации.

Таблица

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 (SQL)	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 (OLAP)		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

ПРИНЦИПЫ СОЗДАНИЯ, СТРУКТУРА И СОДЕРЖАНИЕ АДАПТИРУЕМОГО ПРОФЕССИОНАЛЬНОГО ОБРАЗОВАНИЯ В ОБЛАСТИ ПРОГРАММИРОВАНИЯ И БАЗ ДАННЫХ

По специализации программных средств были выбраны все выше описанные средства, кроме C# (средство только формируется и пока мало используется) и Java, Visual J++ (узкая специализация - Internet-приложения).

При наличии большого числа ПС возникает проблема выбора наиболее эффективных ПС в некоторой предметной области (в заданной системе показателей качества). Для решения этой проблемы можно использовать **методику балловой оценки ПС**.

1. Составляется дерево требуемых показателей качества ПС.

2. Каждой терминальной вершине экспертом присваивается оценка по десятибалльной шкале. Если оцениваемая показатель отсутствует в ПС, то указывается нулевая оценка.

3. Каждому исходному узлу дерева присваивается оценка, равная средней оценке ее подчиненных узлов.

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

В оценке ПС могут участвовать несколько независимых экспертов и их оценки усредняются. Выбирается и изучается ПС с наибольшим рейтингом.

Приведем основные показатели ПС.

- **Категория 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, строителей 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.

- **Фирма-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** выборки, просмотра, загрузки, индексации, обновления, чтения и записи с произвольной выборкой, генерации запроса, отчета и др.

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 1п1егп1-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-средства (BPwin, ERwin, Rational Rose и др.).

Принцип модульности

Весь учебный материал разбит на следующие тематические блоки и темы по каждому блоку (в порядке их изучения).

Блок изучения основ теории программирования (дисциплина "Информатика и программирование") [14].

- Предмет и содержание дисциплины. Алгоритмы. Свойства, типы и формы записей алгоритмов.

- Языки программирования. Классификация, метаязыки.

- Объектно-ориентированное программирование и его свойства.

- Этапы проектирования программного средства (ПС). Состав и порядок разработки проектной документации.

- Типы программ. Классификация программного обеспечения.

- Информация, данные и их характеристики. Определение, свойства, количество информации и единицы ее измерения. Типы данных.

- Системы счисления. Определение и правило перевода из одной системы счисления в другую. Представление чисел в оперативной памяти.

- Организация и типы файлов.

- Организация человеко-машинного интерфейса. Принципы разработки интерфейса. Адаптация и ее типы.

Блок изучения систем программирования (дисциплины "Информатика и программирование" и "Высокоуровневые методы информатики и программирования").

Visual Basic [5, 14]

- Объявление данных, выражения, внутренние подпрограммы.

- Организация вычислений. Операции. Ввод-вывод, файлы.

- Вызов внешних процедур и функций.

- Управление обработкой. Установка среды обработки ошибок.

- Среда разработки. Основные классы, объекты, свойства, события и методы. Размещение объектов. Модули и классы.

- Разработка интерфейса. Меню. Панель инструментов, диалоги.

- Графические средства. Работа с мышью. Рисование.

- Клавиатура, печать. Имитация нажатий клавиш.

- Создание баз данных, запросов и форм.

- Формирование отчетов средствами Crystal Reports, DataReports).

- Дополнительные возможности Visual Basic. ActiveX-элементы. Динамически присоединяемые библиотеки. Функции API. Технология OLE. Директивы компиляции. Параметры настройки приложения. Динамическое создание объектов. Создание справочной системы.

- Создание Web-страниц средствами VBScript. Основные средства создания Web-страниц. Создание ASP-файла. Основы создания сценариев.

Object Pascal, Delphi [3, 7, 15, 18]

- Типы данных. Константы. Переменные. Структурированные типы данных. Указатели. Процедурные указатели.

- Операторы, операции, процедуры, функции, модули, классы, компоненты, методы, свойства и события, директивы компиляции.

- Обработка исключительных ситуаций.

- Типы файлов. Файлы Паскаля. Объектный подход к файлам. Файлы Windows. Обслуживание внешних файлов и каталогов.

- Динамические структуры данных.

- Общие сведения. Окна. Панели компонентов и их настройка. Архив объектов. Характеристика компонентов. Размещение и настройка компонентов на форме.

- Глобальные объекты. Ввод, редактирование и вывод текста. Списки, кнопки, контейнеры, счетчики.

- Графика, меню, структурированные компоненты, системные компоненты, стандартные диалоги.

- Работа с базами данных, формы, отчеты, анализ данных.

- Взаимодействие приложений. Создание динамически подключаемых библиотек, новых компонентов, ActiveX-элементов и пакетов. Многопоточные приложения. Технологии ADO, COM/DCOM, OLE, CORBA, MIDAS, INTERNET, ODBC.

C, C++, C++ Builder, Visual C++ [1, 4, 15-17]

- Основные возможности. Элементы, константы. Типы данных. Массивы. Структуры. Объединения. Перечисления. Предпроцессор. Директивы. Функции. Консольные приложения. Классы памяти.

- Операции, операторы, файлы, базы данных.

- Объектно-ориентированное программирование. Классы, методы и события.

Блок изучения основ теории проектирования баз данных (дисциплина "Базы данных") [8, 11].

- История создания СУБД. Недостатки файловых систем.

- Основные понятия и определения. Требования к БД. СУБД, ее функции, связь с другими программными компонентами. Функциональная полнота СУБД, классификация БД. Принципы проектирования БД.

ПРИНЦИПЫ СОЗДАНИЯ, СТРУКТУРА И СОДЕРЖАНИЕ АДАПТИРУЕМОГО ПРОФЕССИОНАЛЬНОГО ОБРАЗОВАНИЯ В ОБЛАСТИ ПРОГРАММИРОВАНИЯ И БАЗ ДАННЫХ

- Данные. Источники данных. Определение предметной области, объекта, экземпляра объекта, атрибута, значения данного, концептуальной модели, ключевого элемента, записи данных, файла, плоского файла. Уровни независимости, модели данных и их взаимосвязи.

- Администрирование БД. Жизненный цикл БД. Технологические процедуры создания и ведения баз данных.

- Объекты, атрибуты, схемы и подсхемы, СУБД. Определение, назначение, функции объектов, наборов объектов, двумерных файлов, отношений, строк, идентификаторов, первичных и вторичных ключей, инвертированных списков, элементарных данных, схем, подсхем, экземпляра схемы. Формы изображения моделей.

- Иерархические структуры. Основные понятия. Иерархические файлы. Сцепленные ключи. Исходные и порожденные записи.

- Сетевые структуры. Определение и состав сетевых структур. Сетевая модель CODASYL. Перевод сетевой модели в иерархическую. Наборы CODASYL их структура. Язык описания и манипулирования данных типа CODASYL.

- Реляционные структуры. Реляционные модели и СУБД, сравнительный анализ. Таблица и ее свойства. Перевод иерархической и сетевой структур в реляционную. Сцепленные ключи. Манипулирование отношениями. Алгебра отношений и исчисление отношений, сравнительный анализ, язык запросов SQL. Нормализация и денормализация отношений. Транзакции, целостность базы данных.

- Канонические структуры данных. Определение канонической схемы, устранение избыточности, объединение представлений пользователей. Процесс конструирования.

- Типы баз данных. Постреляционные, многомерные, объектно-ориентированные, документальные, фактографические, гипертекстовые, мультимедийные, объектно-ориентированные, распределенные, БД, XML-серверы, хранилища данных.

- Организация процессов обработки данных. Ограничения целостности. OLAP - технология. Защита, архивация, восстановление, реорганизация, реструктуризация баз данных.

Блок изучения СУБД (дисциплина "Базы данных"). Приведем основные изучаемые темы по каждой СУБД.

СУБД Access [11, 19]

- Создание и корректировка базы. Оболочка, меню, настройка панелей. Диалоговые средства создания и корректировки, сорти-

ровки и связывания таблиц. Поиск, фильтры. Целостность базы.

- Создание экранных форм. Типы мастеров. Конструктор форм.

- Создание запросов. Типы мастеров по созданию запросов. Понятие базового и динамического запросов. Конструктор запросов и правила работы с ним. Итоговые операции по группе. Запросы на создание и корректировку таблиц.

- Создание отчетов. Типы мастеров. Конструктор отчетов.

- Создание меню и панелей пользователя. Создание кнопочной формы, меню базы и формы. Настройка параметров запуска.

- Создание макросов и модулей.

- Проекты, защита, восстановление, обмен с внешними данными.

СУБД Visual FoxPro [6,10,11,13]

- Создание и корректировка базы. Типы, структура и правила записи команд, типы файлов. Проект приложения, оболочка, меню, настройка панелей, файл конфигурации. Файлы, рабочие области, псевдонимы. Индексы и их классификация, управление индексами. Совместная обработка таблиц. Целостность базы и средства ее достижения. Командные и диалоговые средства создания баз данных.

- Основные функции и команды структурного программирования, обработка событий, поддержка отладки и выявление ошибок, программное выполнение, функции файлов низкого уровня.

- Создание экранных форм. Мастера. Конструктор форм.

- Создание запросов и представлений. Конструктор и командный язык запросов и представлений. Итоговые операции по группе.

- Создание отчетов. Типы мастеров. Конструктор отчетов.

- Создание окон, меню и панелей пользователя. Командные и диалоговые средства создания меню. Функции работы с окнами и меню. Обмен данными с внешними системами.

СУБД SQL Server [9, 11]

- Создание и настройка базы данных и ее таблиц. Создание и настройка представлений. Языки запросов SQL, SQL*PLUS, Trans-act-SQL. Создание хранимых процедур, триггеров. Формирование правил контроля вводимых значений. Формирование стандартных значений.

- Администрирование. Настройка параметров. Системные базы данных и таблицы. Тестирование и сжатие баз данных. Создание резервных копий и восстановление баз дан-

ных. Использование службы выполнения расписаний SQL Server Agent. Защита данных.

- Реплики, обмен данными с внешними системами. ODBC.

СУБД Oracle [11]

Изучаемые темы примерно аналогичны темам для SQL Server с добавлением тем по изучению средств разработки приложений и PL/SQL.

Блок разработки приложений [2, 12, 13, 21, 22] (дисциплина "Разработка и стандартизация программных средств и информационных технологий") завершает изучение и освоение систем программирования и СУБД.

Имеются три варианта 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 при создании и использовании очень больших баз данных с длительным сроком работы, с хорошим процессом обучения персонала и сопровождения фирмой Oracle.

2. Использование средств универсальных языков программирования Delphi, C++ Builder, Visual C++ или Visual Basic. *Достоинство*: универсальность, можно легко сменить в приложении тип 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 универсальных программных средств (ППП), легко настраиваемых на тип СУБД.

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

В блоке предлагаются следующие основные темы.

- Особенности управления и планирования разработкой программных средств (ПС).

- Стандартизация жизненного цикла ПС. Уровни стандартизации, каскадная и спиральная модели. ГОСТ 19, 24, 34, РИСО/МЭК.

- Организация человеко-машинного интерфейса.

- Качество ПС. Методы достижения качества, сертификация и аттестация, стандарты качества, основные показатели качества.

- Конфигурационное управления версиями.

- Тестирование. Цели, этапы и формы тестирования программ, восходящее и нисходящее тестирование, стандарты по тестированию.

- Методология быстрой разработки приложений (RAD).

- Структурный подход к проектированию. Моделирование потоков данных, функциональное моделирование. Моделирование данных. Методологии IDEF0, IDEF1, IDEF3.

- Рынок программных продуктов.

- Общие характеристики, классификация и технология внедрения CASE-средств.

- Интеллектуализация вычислительных систем.

- Классификация и показатели оценки систем защиты ПС.

- Практическое освоение отдельных CASE-средств BPwin, ERwin [12, 21], Rational Rose [12, 22] и др.

Принцип самодостаточности

Учебный материал классифицирован по специализации и уровням изучения. Для контроля знаний студентов разработаны примеры, упражнения и задачи для программирования, темы практических, контрольных и курсовых работ, тесты для электронного тестирования, вопросы для зачетов и экзаменов. Дополнительный материал (примеры программ, контрольных и курсовых работ) содержится на компакт-дисках и на Web-страницах.

Заключение

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.

СПИСОК ЛИТЕРАТУРЫ

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.

ПРИНЦИПЫ СОЗДАНИЯ, СТРУКТУРА И СОДЕРЖАНИЕ АДАПТИРУЕМОГО
ПРОФЕССИОНАЛЬНОГО ОБРАЗОВАНИЯ В ОБЛАСТИ ПРОГРАММИРОВАНИЯ
И БАЗ ДАННЫХ

5. Евангелас П. Visual Basic 6. Руководство разработчика: Пер. с англ. ВНУ, 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 Л. Microsoft SQL Server 2000 для профессионалов. - СПб: Питер, 2001.
10. Плещев В.В. Основы программирования в СУБД FoxPro: Учеб. пособие - Екатеринбург: Изд-во Урал. гос. экон. ун-та, 1999.
11. Плещев В.В. Базы данных. Visual FoxPro, Access, SQL Server, Oracle: Учеб. пособие - Екатеринбург: Изд-во Урал. гос. экон. ун-та, 2002.
12. Плещев В.В. Разработка и стандартизация программных средств и информационных технологий: Учеб. пособие. - Екатеринбург: Изд-во Урал. гос. экон. ун-та, 2003.
13. Плещев В.В. Пакет прикладных программ автоматизации программирования в среде СУБД FoxPro. Библиотека и генератор программ супероболочки "MasgoFox". Учеб. пособие. - Екатеринбург: Изд-во Урал. гос. экон. ун-та, 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. 3-е изд., испр. и доп. (реком. УМО Минобразования РФ) - Екатеринбург: Изд-во Урал. гос. экон. ун-та, 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: Изд-во Урал. гос. экон. ун-та, 2002.
16. Страуструп Б. Язык программирования C++: Пер. с англ. 3-е изд. 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. Холцшлаг Молли Э. Использование HTML 4 (Специальное издание), 6-е изд.: Пер. с англ. Киев-СПб.: Вильямс, 2001.
21. Маклаков С.В. BPwin, ERwin. CASE-средства разработки информационных систем. - М.: ДИАЛОГ-МИФИ, 1999.
22. Уэнди Борс, Майкл Борс. UML Rational Rose. - М: ЛОРИ, 2000.