

Таким образом, разработан универсальный программный комплекс «Thermogram Analyser» регистрации и обработки экспериментальных данных различных видов термического анализа.

Создан метод идентификации пиков слабых тепловых эффектов на термических кривых в простом термическом анализе.

Путем анализа большого количества экспериментальных термограмм удалось подобрать функции для математического описания пиков тепловых эффектов.

#### ЛИТЕРАТУРА

1. Аносов В. Я., Погодин С. А. Основные начала физико-химического анализа. М., 1947. С. 863.
2. Егунов В. П. Введение в термический анализ. Самара, 1996. С. 270.
3. Шумаков П. В., Фаронов В. В. Delphi 4. Руководство разработчика баз данных. М., 1999. С. 560.
4. Альберг Дж., Альберг Дж., Нилсон Э., Уолш Дж. Теория сплайнов и их приложения. М., 1972. С. 123.
5. Стечкин С. Б., Субботин Ю. Н. Сплайны в вычислительной математике. М., 1976. С. 56.

**Владимир Васильевич ПЛЕЩЕВ** —

доцент кафедры «Информационные системы в экономике»  
экономического факультета Уральского государственного  
экономического университета,  
кандидат технических наук, доцент

## Дифференцированное и интегрированное преподавание программирования и баз данных: предмет изучения, принципы формирования методического обеспечения

УДК 378.016/004

**АННОТАЦИЯ.** Перечисляются характерные особенности адаптивного образования в области программирования. Предлагаются результаты сравнительного анализа современных средств программирования (языки программирования, СУБД, CASE-средства). Приводится экспертная оценка возможностей этих средств и быстроты их освоения. Формулируются принципы создания соответствующего методического обеспечения: системность, дифференцирование по специализации программных средств и уровням их изучения, самостоятельность (кейс-обеспечение), развитие, модульность.

*The author enumerates characteristic features of adaptive training in the field of programming, offers the results of the comparative analysis of modern means of programming (programming languages, SCDB, CASE — means). The expert estimation of opportunities of these means and speed of their development is given. Principles of creation of corresponding methodical provision are formulated: integration, differentiation according to specialization of software and levels of their studying, self-sufficiency (a case-provision), development, modality.*



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

Отметим основные особенности такого подхода:

— 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;

— 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.

Под интеграцией преподавания имеется в виду учет взаимосвязей между смежными дисциплинами, исключение дублирования, согласованность и унификация понятий, ориентация на отраслевую специфику специальности.

При формировании содержания образования в области программирования и баз данных (языки и средства программирования, СУБД, CASE-средства) за основу взят государственный образовательный стандарт 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».

### АНАЛИЗ ПРЕДМЕТА ИЗУЧЕНИЯ

Для определения содержания образования нужно проанализировать состояние этой области. Сформулируем кратко результаты такого анализа по основным разделам этой области. В виду 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. Visual Basic имеет простую и удобную среду RAD, допускает работу с различными типами СУБД. Позволяет создавать Web-страницы совместно с HTML и ASP. 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.

*Delphi* является мощной системой программирования. Программы, написанные на Delphi, не требуется снабжать дополнительными библиотеками. В Delphi имеется собственная 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). 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 созда-



вать 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. Delphi рекомендуется для быстрой разработки приложений.

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 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++ *Builder* есть некий «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, мощная библиотека MFC. Правка кода в режиме отладки и последующее его выполнение без полной перекомпиляции и прерывания отладочной сессии. Имеется отличная, хорошо структурированная полная 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 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 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 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 Mic-



Microsoft Visual J++ не поддерживает стандарт визуального Java-программирования JavaBeans. Рекомендуется для разработки Internet-приложений.

**СУБД Access.** 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. 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 исполнимого 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) обрабатываемой информации.

**СУБД SQL Server.** 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 удаленным базам данных.

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.** 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-средства), 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. 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: VPwin (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 (McDonnell Douglas).



- Средства разработки приложений: Delphi (Borland), PowerBuilder (SyBase), Developer/2000 (ORACLE), New Era (Informix), SQL Windows (Centura), Uniface (Compuware), JAM (JYACC).
- 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 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.

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
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



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

1. Составляется дерево требуемых показателей качества ПС.
2. Каждой терминальной вершине экспертом присваивается оценка по десятибалльной системе. 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 (итоговый рейтинг) всего ПС.

В оценке ПС могут участвовать несколько независимых экспертов и их оценки усредняются. 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.

На формирование содержания предмета изучения влияют варианты разработки приложений:

1. Использование исключительно средств (если они имеются) СУБД (Oracle, Visual FoxPro). *Достоинства*: 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. *Достоинства*: 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 пакетов прикладных программ (ППП), легко настраиваемых на тип СУБД.



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

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

### ПРИНЦИПЫ ФОРМИРОВАНИЯ АДАПТИВНОГО МЕТОДИЧЕСКОГО ОБЕСПЕЧЕНИЯ

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

*Интеграция* Учет взаимосвязей между указанными дисциплинами и специфики специальности, исключение дублирования учебного материала. Согласованность и унификация понятий, правил изложения и оформления учебного материала.

*Самодостаточность* (кейс-обеспечение). Все, что нужно для изучения и контроля (включая электронное тестирование) 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.*

— *Системы программирования.* Visual Basic, Object Pascal, Delphi, C, C++, C++ Builder, Visual C++.

— *Основы теории проектирования баз данных.*

— *СУБД Access, Visual FoxPro, SQL Server, Oracle.*

— *Разработка приложений.* CASE-средства VPwin, ERwin, Rational Rose.

*Дифференцирование*. Декомпозиция учебного материала по специализации (видам программных средств) и по уровням изучения и освоения. Укрупненно сформулируем результаты дифференцирования. По специализации программных средств были выбраны все выше описанные средства, кроме C# (средство только формируется и пока мало используется) и Java, Visual J++ (достаточно узкая специализация). Примерное соответствие уровней изучения типам учебных заведений и содержанию учебного материала было принято следующим:

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

2. Основной уровень (студенты Вузов и институтов повышения квалификации кадров по не компьютерным специальностям и студенты колледжей по компьютерным специальностям.) — 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.

3. Углубленный уровень (студенты Вузов и институтов повышения квалификации кадров по компьютерным специальностям.) — 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).

### 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 были применены в учебном процессе в нескольких высших и средних учебных заведениях с различными формами обучения и показали свою эффективность.

### ЛИТЕРАТУРА

1. 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, 2002.
2. Плещев В. В. Высокоуровневые методы информатики и программирования. Delphi 5, С++ Builder 5, Visual С++ с примерами и упражнениями: Учебник. 2-е изд., испр. и доп. (реком. УМО Минобразования РФ). Екатеринбург, 2002.
3. Плещев В. В. Базы данных. Visual FoxPro, Access, SQL Server, Oracle: Учеб. пособие. Екатеринбург, 2002.
4. Плещев В.В. Разработка и стандартизация программных средств и информационных технологий: Учеб. пособие. Екатеринбург, 2003.

**Сергей Александрович БАРДАСОВ** —

*доцент кафедры экономики и управления собственностью,  
кандидат физико-математических наук*

## **Предпочтительность метода равных частот относительно метода равных интервалов при построении вариационных рядов**

УДК 519.21(075.8)

**АННОТАЦИЯ.** Доказано, что группы, образованные методом равных частот, по сравнению с группами, образованными методом равных интервалов, имеют большее значение функции максимального правдоподобия.

*Is proved, that the groups formed by a method of equal frequencies, in comparison with groups formed by a method of equal intervals, have the greater value of function of the maximal plausibility.*

Пусть имеется  $N$  объектов, которые характеризуются ранжированными в порядке возрастания численными значениями  $x_1, x_2, \dots, x_N$  некоторого признака.