МОРДОВСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ ИМЕНИ Н.П. ОГАРЕВА

**ФАКУЛЬТЕТ ЭКОНОМИЧЕСКИЙ**

**КАФЕДРА ИНФОРМАЦИОННЫХ СИСТЕМ В ЭКОНОМИКЕ И УПРАВЛЕНИИ**

КУРСОВАЯ РАБОТА

Тема: **Системы программирования**

Оценка\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

САРАНСК 2009

**Содержание**

Введение

1. Система программирования как неотъемлемая часть современных ЭВМ

1.1 Определение системы программирования

1.2 Классификация систем программирования

1.3 Средства создания программ

2. Современные системы программирования

2.1 Microsoft Visual Basic

2.2 Turbo Pascal

2.3 C++ Builder

2.4 Symantec Café

3. Практическая часть

Заключение

Список использованных источников

**Введение**

В пятидесятые годы двадцатого века с появлением компьютеров на электронных лампах началось бурное развитие систем программирования. К сегодняшнему дню насчитывают несколько поколений систем программирования. Каждое из последующих поколений по своей функциональной мощности качественно отличается от предыдущего. С появлением персональных компьютеров системы стали составными частями интегрированных сред разработки. Появились системы, применяемые в различных офисных программах. В настоящее время системы программирования применяются в самых различных областях человеческой деятельности, таких как научные вычисления, системное программирование, обработка информации, искусственный интеллект, издательская деятельность, удаленная обработка информации, описание документов.

С течением времени одни системы развивались, приобретали новые черты и остались востребованы, другие утратили свою актуальность и сегодня представляют в лучшем случае чисто теоретический интерес.

Цель работы - описать современные системы программирования.

**1. Системы программирования как неотъемлемая часть ЭВМ**

**1.1 Определение системы программирования**

Неотъемлемая часть современных ЭВМ – системы программного обеспечения, являющиеся логическим продолжением логических средств ЭВМ, расширяющим возможности аппаратуры и сферу их использования. Система программного обеспечения, являясь посредником между человеком и техническими устройствами машины, автоматизирует выполнение тех или иных функций в зависимости от профиля специалистов и режимов их взаимодействия с ЭВМ. Основное назначение программного обеспечения – повышение эффективности труда пользователя, а также увеличение пропускной способности ЭВМ посредством сокращения времени и затрат на подготовку и выполнение программ. Программное обеспечение ЭВМ можно подразделить на общее и специальное программное обеспечение.

Общее программное обеспечение реализует функции, связанные с работой ЭВМ, и включает в себя системы программирования, операционные системы, комплекс программ технического обслуживания. Специальное программное обеспечение включает в себя пакеты прикладных программ, которые проблемно ориентированы на решение вполне определенного класса задач.

**Системой программирования** называется комплекс программ, предназначенный для автоматизации программирования задач на ЭВМ (2, 569). Система программирования освобождает проблемного пользователя или прикладного программиста от необходимости написания программ решения своих задач на неудобном для него языке машинных команд, и предоставляют им возможность использовать специальные языки более высокого уровня. Для каждого из таких языков, называемых входными или исходными, система программирования имеет программу, осуществляющую автоматический перевод (трансляцию) текстов программы с входного языка на язык машины. Обычно система программирования содержит описания применяемых языков программирования, программы-трансляторы с этих языков, а также развитую библиотеку стандартных подпрограмм. Важно различать язык программирования и реализацию языка.

**Язык программирования** – это набор правил, определяющих систему записей, составляющих программу, синтаксис и семантику используемых грамматических конструкций. **Реализация языка** – это системная программа, которая переводит (преобразует) записи на языке высокого уровня в последовательность машинных команд.

**1.2 Классификация систем программирования**

По набору входных языков различают системы программирования одно- и многоязыковые. Отличительная черта многоязыковых систем состоит в том, что отдельные части программы можно составлять на разных языках и с помощью специальных обрабатывающих программ объединять их в готовую для исполнения на ЭВМ программу.

По структуре, уровню формализации входного языка и целевому назначению различают системы программирования машинно-ориентированные и машинно-независимые.

*Машинно-ориентированные* системы программирования имеют входной язык, наборы операторов и изобразительные средства которых существенно зависят от особенностей ЭВМ (внутреннего языка, структуры памяти и т.д.). Машинно-ориентированные системы позволяют использовать все возможности и особенности машинно-зависимых языков:

* высокое качество создаваемых программ;
* возможность использования конкретных аппаратных ресурсов;
* предсказуемость объектного кода и заказов памяти;
* для составления эффективных программ необходимо знать систему команд и особенности функционирования данной ЭВМ;
* трудоемкость процесса составления программ (особенно на машинных языках и ЯСК), плохо защищенного от появления ошибок;
* низкая скорость программирования;
* невозможность непосредственного использования программ, составленных на этих языках, на ЭВМ других типов.

*Машинно-независимые* системы программирования – это средство описания алгоритмов решения задач и информации, подлежащей обработке. Они удобны в использовании для широкого круга пользователей и не требуют от них знания особенностей организации функционирования ЭВМ. В таких системах программы, составляемые языках, имеющих название высокоуровневых языков программирования, представляют собой последовательности операторов, структурированные согласно правилам рассматривания языка (задачи, сегменты, блоки и т.д.). Операторы языка описывают действия, которые должна выполнять система после трансляции программы на машинном языке. Таким образом, командные последовательности (процедуры, подпрограммы), часто используемые в машинных программах, представлены в высокоуровневых языках отдельными операторами. Программист получил возможность не расписывать в деталях вычислительный процесс на уровне машинных команд, а сосредоточиться на основных особенностях алгоритма.

**1.3 Средства создания программ**

В самом общем случае для создания программы на выбранном языке программирования нужно иметь следующие компоненты.

1. *Текстовый редактор*. Так как текст программы записывается с помощью ключевых слов, обычно происходящих от слов английского языка, и набора стандартных символов для записи всевозможных операций, то формировать этот текст можно в любом редакторе, получая в итоге текстовый файл с исходным текстом программы. Лучше использовать специализированные редакторы, которые ориентированы на конкретный язык программирования и позволяют в процессе ввода текста выделять ключевые слова и идентификаторы разными цветами и шрифтами. Подобные редакторы созданы для всех популярных языков и дополнительно могут автоматически проверять правильность синтаксиса программы непосредственно во время ее ввода.

2. Исходный текст с помощью *программы-компилятора* переводится в машинный код. Исходный текст программы состоит, как правило, из нескольких модулей (файлов с исходными текстами). Каждый модуль компилируется в отдельный файл с *объектным кодом*, которые затем требуется объединить в одно целое. Кроме того, системы программирования, как правило, включают в себя библиотеки стандартных подпрограмм (имеют расширение .LIB). Стандартные подпрограммы имеют единую форму обращения, что создает возможности автоматического включения таких подпрограмм в вызывающую программу и настройки их параметров.

3. Объектный код модулей и подключенные к нему стандартные функции обрабатывает специальная программа – *редактор связей*. Данная программа объединяет объектные коды с учетом требований операционной системы и формирует на выходе работоспособное приложение – *исполнимый код* для конкретной платформы. Исполнимый код это законченная программа, которую можно запустить на любом компьютер, где установлена операционная система, для которой эта программа создавалась. Как правило, итоговый файл имеет расширение .exe или .com.

4. В современных системах программирования имеется еще один компонент – *отладчик*, который позволяет анализировать работу программы во время ее исполнения. С его помощью можно последовательно выполнять отдельные операторы исходного текста последовательно, наблюдая при этом, как меняются значения различных переменных.

5. В последние несколько лет в программировании (особенно для операционной среды Windows) наметился так называемый визуальный подход. Этот процесс автоматизирован в *средах быстрого проектирования*. При этом используются готовые визуальные компоненты, свойства и поведение которых настраиваются с помощью специальных редакторов. Таким образом, происходит переход от языков программирования системного уровня к языкам сценариев.

1. **Современные системы программирования**

**2.1 Microsoft Visual Basic**

**Microsoft Visual Basic** — средство разработки программного обеспечения, разрабатываемое корпорацией Microsoft и включающее язык программирования и среду разработки. Язык Visual Basic унаследовал дух, стиль и отчасти синтаксис своего предка — языка Бейсик, у которого есть немало диалектов. В то же время Visual Basic сочетает в себе процедуры и элементы объектно-ориентированных и компонентно-ориентированных языков программирования. Среда разработки VB включает инструменты для визуального конструирования пользовательского интерфейса.

Visual Basic считается хорошим средством быстрой разработки прототипов программы, для разработки приложений баз данных и вообще для компонентного способа создания программ, работающих под управлением операционных систем семейства Microsoft Windows.

Первое признание серьёзными разработчиками Visual Basic получил после выхода версии 3 — VB3. Окончательное признание как полноценного средства программирования для Windows — при выходе версии 5 — VB5. Версию VB6, входящую в состав Microsoft Visual Studio 6.0, стала по-настоящему зрелым и функционально богатым продуктом. После этого разработчики из Microsoft существенно изменили направление развития данной технологии.

Visual Basic .NET не позволяет программировать по-старому, ибо, по сути, является совершенно другим языком, таким же, как и любой другой язык программирования для платформы .NET. Индивидуальность языка, так же как и его преимущества (простота, скромность создания программ, лёгкость использования готовых компонент) при использовании в среде .NET не имеют такого значения, как раньше — всё сосредоточено на возможностях самой системы .NET, на её библиотеке классов. Поэтому сегодня нужно говорить о классическом Visual Basic, его диалектах Visual Basic for Applications (VBA) и Visual Basic Scripting Edition (VBScript) и о языке для платформы (4, 467) .NET — Visual Basic .NET. Основные разновидности Visual Basic:

1) Классический Visual Basic (версии 5-6).Этот язык очень сильно привязан к своей среде разработки и к операционной системе Windows, являясь исключительно инструментом написания Windows-приложений. Привязка к среде заключается в том, что существует большое количество средств, предназначенных для помощи и удобства программирования: встроенный отладчик, просмотр переменных и структур данных на лету, окно отладки, всплывающая подсказка при наборе текста программы (Intellisense). Все эти преимущества делают бесполезным и даже невозможным использование Visual Basic вне среды разработки, например в обычном текстовом редакторе.

2) Visual Basic for Applications (VBA) Это средство программирования, практически ничем не отличающееся от классического Visual Basic, которое предназначено для написания макросов и других прикладных программ для конкретных приложений. Наибольшую популярность получил благодаря своему использованию в пакете Microsoft Office. Широкое распространение Visual Basic for Applications в сочетании с изначально недостаточным вниманием к вопросам безопасности привело к широкому распространению макровирусов.

3) Visual Basic Scripting Edition (VBScript).Скриптовый язык, являющийся несколько усечённой версией обычного Visual Basic. Используется в основном для автоматизации администрирования систем Windows, а также для создания страниц ASP и сценариев для Internet Explorer.

Достоинства VB:

* Высокая скорость создания приложений с графическим интерфейсом для MS Windows.
* Простой синтаксис, позволяющий очень быстро освоить язык.
* Возможность как компиляции в машинный код, так и интерпретации во время отладки.

Недостатки:

* Поддержка операционных систем только семейства Windows и Mac OS X (Исключение — VB1 for DOS).
* Отсутствие механизма наследования объектов. Существующие в языке наследование, позволяет наследовать только интерфейсы объектов, а не их самих. Таким образом, в унаследованном классе должны быть явно переписаны все функции базового класса. Также в унаследованном классе невозможно добавление каких-либо методов, присущих только данному классу, то есть если абстрактный базовый класс содержит только два метода, то и производный класс содержит только два метода, не более и не менее того.
* Требует установленных DLL для работы программы.

# 2.2 Turbo Pascal

**Turbo Pascal** — Интегрированная среда разработки программного обеспечения для платформ DOS и Windows 3.x и язык программирования в этой среде, диалект языка Паскаль от фирмы Borland.

Товарный знак Borland Pascal был зарезервирован для дорогих вариантов поставки (с бо́льшим количеством библиотек и исходным кодом стандартной библиотеки), оригинальная дешёвая и широко известная версия продавалась как Turbo Pascal. Название Borland Pascal также используется в более широком смысле — как неофициальное название версии языка Паскаль от фирмы Borland.(1, 4)

Turbo Pascal — это среда разработки для языка программирования Паскаль. Используемый в Turbo Pascal диалект базировался на более раннем UCSD Pascal, получившем распространение, в первую очередь, на компьютерах серии Apple II. Компилирующая компонента Turbo Pascal была основана на компиляторе Blue Label Pascal, первоначально созданном в 1981 году Андерсом Хейлсбергом для операционной системы NasSys микрокомпьютера Nascom. Позднее он был переписан как Compass Pascal для операционной системы CP/M, затем как Turbo Pascal для DOS и CP/M. Одна из версий Turbo Pascal была доступна под Apple Macintosh примерно с 1986 года, но её разработка прекратилась примерно в 1992 году (5, 134).

В 1982 году Филипп Кан приобрёл компилятор у Андерса Хейлсберга и перебрался из Парижа в Калифорнию, где основал компанию Borland.

Когда в 1983 году появилась первая версия Turbo Pascal, такой тип среды разработки был относительно новым. Во время дебюта на американском рынке, Turbo Pascal продавался по цене в 49,99 долл. Помимо привлекательной цены, встроенный компилятор Паскаля также был очень высокого качества. Приставка «Turbo» намекала как на скорость компиляции, так и на скорость производимого им исполняемого кода. Turbo Pascal создавал машинный код за один проход, без шага компоновки.

После рекламной кампании за первый месяц поступило заказов на 150 тыс. долларов — так много, что местные банки отказывались оплачивать чеки и кредитные карточки, подозревая мошенничество.

За первые два года было продано не менее 300 тысяч копий компилятора, что превзошло объём продаж всех прочих языков для микрокомпьютеров.

Для того времени это была потрясающая среда разработки. Она была проста и интуитивно понятна, с хорошо организованным меню. Ранние версии использовали раскладку горячих клавиш WordStar. В более поздних версиях появилась возможность быстро получить определение ключевого слова языка, просто поставив курсор на ключевое слово и нажав клавишу справки. Справочные статьи часто включали примеры кода, использующего данное ключевое слово. Это позволяло неопытным программистам изучать Паскаль даже без помощи книг, используя лишь среду разработки. В поставку входило большое количество исходных текстов демонстрационных и прикладных программ. В их числе были даже шахматы.

Среда позволяла легко встраивать в код на Паскале вставки на языке ассемблера. Пользователь имел возможность проходить программу шаг за шагом; при переходе на ассемблерный блок это также работало. В любой момент пользователь мог добавить переменную или регистр в удобно расположенное окно для наблюдения за ними. При построчной отладке программ, использующих графические режимы IBM PC, происходило корректное переключение между графическим режимом программы и текстовым режимом среды разработки.

Помимо всего этого, имелось средство профилирования. Книги, включённые в поставку Borland Pascal, давали детальное описание языка ассемблера Intel вплоть до указания количества тактовых циклов, необходимых для выполнения каждой инструкции. В общем и целом, система давала превосходные возможности для оптимизации кода; пользователю не требовалось пользоваться чем-либо кроме среды разработки. Всё было сделано так идеально, что даже школьник мог этим пользоваться. Эти качества позволили версии Паскаля от Borland стать стандартом языка Паскаль де-факто.

С начала 1990-х TP/BP используется в университетах для изучения фундаментальных концепций программирования.

Вероятно, разработка Microsoft Pascal была прекращена из-за конкуренции с высоким качеством и небольшой ценой Turbo Pascal. Другая версия гласит, что Borland заключил соглашение с Microsoft на прекращение разработки Turbo BASIC (среды разработки для BASIC, ответвившейся от Turbo Pascal), если Microsoft прекратит разработку Microsoft Pascal. Некоторое время Microsoft выпускал QuickPascal, который был почти 100%-совместим с Turbo Pascal.

В течение нескольких лет Borland улучшал не только среду разработки, но и язык. В версии 5.5 в него были введены передовые возможности объектно-ориентированного программирования. Последней выпущенной версией была версия 7. Borland Pascal 7 включал в себя среду разработки и компиляторы для создания программ под DOS, под DOS с расширителем DOS и Windows 3.x, в то время как Turbo Pascal 7 мог создавать только обычные DOS-программы.

С 1995 года в Borland прекратили разработку Turbo Pascal и предложили в качестве замены среду разработки Delphi. Новая версия языка подверглась изменению (в особенности ООП), и языку вернулось изначальное название, закреплённое разработчиками Apple Object Pascal. Старая объектная модель Turbo Pascal и соответствующий синтаксис поддерживался как устаревший, использование обеих объектных моделей одновременно в одной и той же программе не поддерживается.

Достоинства Turbo Pascal:

Удобная среда разработки, включающая функциональный отладчик, доступный в любой момент.

* Контекстная справочная система, по которой можно изучать язык без обращения к сторонним источникам.
* Высокая скорость компиляции, высокая скорость выполнения откомпилированных программ.
* Встроенная возможность использовать вставки на языке ассемблера.

Недостатки:

* Компилятор рассчитан на реальный режим DOS, применение которого сходит на нет. Однако в последних версиях компилятора и среды введена поддержка защищённого режима вместе с соответствующим отладчиком (TD).
* В модуле CRT имеется ошибка (некорректный подсчёт количества циклов для функции delay, не рассчитанный на быстрые процессоры, процессоры с переменной частотой и многозадачные среды), из-за которой при запуске программы на компьютерах с тактовой частотой более 200 MHz сразу происходило аварийное завершение с сообщением «Runtime error 200 at…». Существуют разные варианты исправления модуля CRT. В варианте Клауса Хартнегга ошибка 200 не возникает, но длительность Delay на быстрых компьютерах меньше желаемой, и эта проблема по незнанию иногда тоже считается недостатком Turbo Pascal.

**2.3 C++ Builder**

**C++ Builder** - программный продукт, инструмент быстрой разработки приложений (RAD), интегрированная среда программирования (IDE), система, используемая программистами для разработки программного обеспечения на языке C++.

C++ Builder объединяет в себе комплекс объектных библиотек (STL, VCL, CLX, MFC и др.), компилятор, отладчик, редактор кода и многие другие компоненты. Цикл разработки аналогичен Delphi. Большинство компонентов, разработанных в Delphi, можно использовать и в C++ Builder без модификации, но, к сожалению, обратное утверждение не верно.

C++ Builder содержит инструменты, которые при помощи drag-and-drop действительно делают разработку визуальной, упрощает программирование благодаря встроенному WYSIWYG - редактору интерфейса и пр.

C++ Builder первоначально создавалась только для платформы Microsoft Windows. Поздние версии, содержащие Кроссплатформенную компонентную библиотеку Borland, основанную на Qt , поддерживают и Windows и Linux. (8, 991)

В 2003 Borland выпустила *C++ BuilderX* (CBX), написанный при помощи той же инфраструктуры, что и JBuilder, который при этом был мало похож на C++ Builder или Delphi. Этот продукт предназначался для разработки больших программ для крупных предприятий, но коммерческого успеха не достиг. В конце 2004 года Borland объявила, что продолжит развитие классического C++ Builder и объединит его со средой разработки Delphi, прекратив, таким образом, разработку C++ BuilderX.

Спустя примерно год после этого объявления, Borland выпустила *Borland Developer Studio 2006*, который включал в себя *Borland C++ Builder 2006*, предлагавший улучшенное управление конфигурацией и отладкой. *Borland Developer Studio 2006* — единственный полноценный комплект, содержащий Delphi, C++Builder и C#Builder.

В 2007 CodeGear выпустила *C++ Builder 2007*, в котором реализовала полную поддержку API Microsoft Windows Vista, увеличила полноту соответствия стандарту ANSI C++, увеличила скорость компиляции и сборки до 500 %, включила поддержку MSBuild, архитектур баз данных DBX4 и «VCL для Web», поддеживающий AJAX. Поддержка API Microsoft Windows Vista включила в себя приложения, изначально оформленные в стиле Vista, и естественную поддержку VCL для Aero и Vista Desktop. *CodeGear RAD Studio 2007* содержит *C++ Builder 2007* и *Delphi*. Также в 2007 CodeGear «воскресила» марку «Turbo» и выпустила две «Turbo» версии C++ Builder: Turbo C++ Professional и Turbo C++ Explorer (бесплатный), основанных на *Borland C++ Builder 2006*.

В конце 2008 года компания CodeGear выпустила новую версию RAD Studio, в которую вошли Delphi 2009 и С++ Builder 2009.

Следующая версия, CodeGear C++Builder (кодовое имя «Commodore»), будет обладать поддержкой x86-64 и возможностью создавать нативный x86-64 код.

Таблица 2.3 – Краткие сведения о версиях продукта

|  |  |
| --- | --- |
| Год | Версия |
| 1997 | 1 |
| 1998 | 3 |
| 1999 | 4 |
| 2000 | 5 |
| 2002 | 6 |
| 2003 | X |
| 2005 | 2006 |
| 2007 | 2007 |
| Сентябрь 2008 | 2009 |
| 25 августа 2009 | 2010 |

**2.4 Symantec Café**

Язык Java является принципиально новым языком программирования, созданным компанией Sun Microsystems для создания многоплатформных приложений (applications и applets) для страниц "всемирной паутины" сети Internet. Язык Java может быть назван упрощенным вариантом C++, без усложненных конструкций и дополнительных возможностей. Java предлагает широкие возможности объектно-ориентированного программирования и повторного использования кода.

Symantec Cafe является первой интегрированной средой визуальной разработки для создания приложений (applications и applets) для страниц "всемирной паутины" сети Internet (3, 265).

Symantec Cafe интегрирует комплект разработчика Java Development Kit компании Sun Microsystems в популярную многооконную среду визуальной разработки, созданную компанией Symantec для создания приложений для Windows 95 и Windows NT. Symantec Cafe предлагает полнофункциональную систему управления проектами, а также мощные инструменты редактирования и просмотра кода, что обеспечивает резкое увеличение эффективности разработки приложений на языке Java для сети Internet. Приложения, созданные с помощью Symantec Cafe могут затем встраиваться в документы HTML и выполняться на различных платформах при использовании Java-соместимых программ просмотра, таких как Netscape Navigator.

Symantec Cafe позволяет разрабатывать приложения на языке Java, которые могут затем встраиваться в страницы всемирной паутины для обеспечения более высокой функциональности, чем существующие HTML-страницы. Java-компилятор генерирует байткод, который может затем встраиваться в HTML-определения страниц всемирной паутины. Наиболее популярные программы просмотра в сети Internet, такие Netscape Navigator, включают встроенный интерпретатор Java-байткода, позволяющий выполнять Java-приложения на компьютере пользователя во время просмотра страницы Internet, содержащей это Java-приложение.

Это дает возможность включать в Internet страницу программное обеспечение, что предлагать пользователю гораздо более богатые возможности, по сравнению с просто текстом или статической графикой. Например, существует возможность включить новый тип данных и назначить соответствующий ей Java-байткод, предназначенный специально для обработки этого типа информации на клиентской машине. Кроме того, в этом случае Java-приложение запускается на клиентской машине, что позволяет снижать загрузку web-сервера. В результате достигается более высокая функциональность и производительность при просмотре сетей Internet. Cafe позволяет разрабатывать любые виды многоплатформенных приложений (applets and applications). Сокращенное приложение (applets) представляет собой ограниченная версия полнофункционального Java-приложения (applications), предназначенного для работы с web-документами. Например, сокращенное приложение не имеет доступа к файлам на клиентском компьютере. Такой подход предназначен, с одной стороны, для обеспечения целостности созданных Java-приложений при загрузке их из Internet, а с другой - для того, чтобы избежать случайной потери информации на клиентской машине вследствие работы загруженного из Internet приложения. Полнофункциональные Java-приложения более похожи на стандартные программы, за исключением того, что они многоплатформенны и могут запускаться под Windows, Macintosh и Unix. Основные возможности Symantec Café:

1. Cafe выполняет "на лету" грамматический разбор Java-код и создает репозиторий информации о Java-приложениях и Java-библиотеках классов. Это позволяет пользователю наглядно иерархию классов Java-приложения, лучше понять стандартные классы Java и классы Java-приложений.
2. Class Editor позволяет просматривать исходный текст на языке Java, а также просматривать/редактирования методы, данные и классы. Class Editor позволяет разработчику работать с объектно-ориентированными частями Java-программы в противоположность работы с исходными текстами.
3. ProjectExpress, "Wizard"-подобный инструмент, позволяющий быстро создавать проекты вокруг набора Java-программ и использовать преимущества Cafe с минимальными затратами.
4. AppExpress, "Wizard"-подобный инструмент, помогающий начать работу разработчикам, не знакомым с языком Java. AppExpress автоматически создает полнофункциональные Java- приложения, которые могут быть доработаны позже с помощью дополнительных возможностей Cafe.
5. В Cafe входит профессиональный редактор для разработчиков на языке Java, который поддерживает цветовое выделение синтаксических конструкций и ключевых символов, а также включает интегрированный макро-язык для расширения функциональных возможностей редактирования. Кроме того, редактор способен быстро переходить в любую часть описания в Java-приложении или Java-библиотек классов, резко повышая производительность труда разработчиков.
6. Cafe включает полностью интегрированный комплект разработчика Java Development Kit (JDK) компании Sun, с графической поддержкой опций и параметров Java-компилятора, интерпретатора и отладчика. Кроме того, Cafe поддерживает управление вложенными проектами, а также возможность построения Java-приложений, как сокращенных, так и полнофункциональных, непосредственно из среды разработчика.

Cafe включает мощный AppExpress, простой и удобный в использовании генератор приложений, автоматически создающий полнофункциональные Java-приложения, которые можно модифицировать и развивать.

Для создания Java приложения необходимо запустить AppExpress из меню "Tools", указать тип приложения в поле "Java Applet", определить каталог для создания приложения и нажать кнопку "Finish". Это все, что необходимо сделать. Проект приложения на языке Java автоматически загрузится средой Cafe.

Чтобы построить и запустить Java-приложение, нужно выбрать команду "Run" из меню "Project". Cafe попросит подтвердить необходимость построения проекта. Выберите "Yes" и Java приложение будет построено. Созданное приложение доступно для расширения и модификации.

Cafe имеет удобный "Wizard"-подобный инструмент ProjectExpress, позволяющий легко создавать новые проекты. Cafe позволяет просто и быстро импортировать уже существующий Java-код или проект в Cafe с минимальными затратами. Используя ProjectExpress, можно определить тип проекта Java или С/C++, затем добавить указание на файлы с исходным текстом и проект автоматически будет создан и загружен в Cafe.

Cafe поддерживает вложенную организацию проектов, что значительно сокращает затраты времени на организацию и управление созданием приложений для Internet. Cafe Project Manager может управлять проектами с различными опциями и вершинами без необходимости загрузки или выгрузки того или иного проекта.

Когда открывается или создается Java проект в Cafe, включается уникальный модуль грамматического разбора текста в фоновом режиме, автоматически анализирующий исходный код на языке Java и создающий репозиторий информации о Java приложениях и любых библиотеках классов Java, используемых в приложениях. Этот репозиторий используется для просмотра и управления всеми классами Java в приложении.

Входящая в Cafe программа просмотра классов Class Editor позволяет быстро проектировать и перемещаться по исходному коду Java-приложения за счет мощных встроенных инструментов навигации. Class Editor позволяет отказаться от работы с индивидуальными индивидуальными классами Java и работать напрямую с определениями и членами классов Java.

Class Editor позволяет быстро перемещаться к любому участку кода Java приложения. Достаточно ввести или указать нужное имя класса и/или члена, и в соответствующем окне появится его исходный код. Этот же метод действует при определении принадлежности того или иного класса или метода. Class Editor позволяет также быстро создавать новые классы и добавлять Новые члены классов.

Cafe предлагает разработчикам профессиональный редактор, поддерживающий выделение цветом синтаксических конструкций и ключевых слов. Кроме того, редактор включает макроязык для расширения возможностей редактирования и может перемещаться к любому участку описания Java-кода в Java-приложении или Java-классах.

**3. Практическая часть**

1. Постановка задачи

Разработать программу с диалоговым окном для вычисления предельной ошибки выборки в случае механического отбора:

,



N – численность генеральной совокупности;

n – численность выборки;

w – выборочная доля;

t – коэффициент доверия, для которого задать значения t =3 при p = 0,997, t = 2 при p = 0,954, t = 1 при p = 0,683.

В программе использовать проверку корректности исходных данных.

2. Разработка проекта

1. на Листе 1 в диапазон ячеек А1:Е1 вводим заголовки таблицы исходных данных и полученного результата.
2. Входим в Режим конструктора (панель инструментов Элементы управления). Создаем элемент управления Кнопка: Ввод.
3. Двойным нажатием на кнопку Ввод, входим в редактор VB. Создаем процедуру обработки нажатия на кнопку Ввод:

Private Sub CommandButton1\_Click()

UserForm1.Show

End Sub

1. Выберем команду Insert| UserForm. В проекте появится новая форма.

Добавляем на форму три объекта Label (Метка), три объекта TextBox (Текстовое поле) и группу из трех объектов OptionButton (Переключатель) для ввода исходных данных.

Добавляем на форму объект CommandButton (Кнопка) для подсчета предельной ошибки выборки в случае механического отбора.

1. Обрабатываем нажатие кнопки CommandButton1:

Private Sub CommandButton1\_Click()

If IsNumeric(TextBox1) And IsNumeric(TextBox2) And IsNumeric(TextBox3) Then

If Val(TextBox2) <= Val(TextBox1) Then

If (Val(TextBox3) >= 0) And (Val(TextBox3) <= 1) Then

j = 1

While Cells(j, 4) <> ""

j = j + 1

Wend

Cells(j, 1).Value = TextBox1

Cells(j, 2).Value = TextBox2

Cells(j, 3).Value = TextBox3

If OptionButton1.Value Then t = 1

If OptionButton2.Value Then t = 2

If OptionButton3.Value Then t = 3

Cells(j, 4).Value = Str(t)

ПОВ = t \* Sqr(TextBox3 \* (1 - TextBox3) / TextBox2)

MsgBox "Предельная ошибка выборки:" + Str(ПОВ), vbInformation + vbOKOnly, "Результат"

Cells(j, 5).Value = Str(ПОВ)

TextBox1 = ""

TextBox2 = ""

TextBox3 = ""

OptionButton1.Value = True

UserForm1.Hide

Else

MsgBox "Выборочная доля вышла за допустимый диапазон", vbCritical + vbOKOnly, "Внимание"

End If

Else

MsgBox "Численность выборки не должна превышать численность генеральной совокупности", vbCritical + vbOKOnly, "Внимание"

End If

Else

MsgBox "Ошибка ввода", vbCritical + vbOKOnly, UserForm1.Caption

End If

End Sub

Введенные данные и полученный результат помещаются в Excel-таблицу.

**Заключение**

Сегодня имеется немало систем программирования, выпускаемых различными фирмами и ориентированных на различные модели ПК и операционные системы. Наиболее популярны следующие визуальные среды быстрого проектирования:

1) Microsoft Visual Basic;

2) Turbo Pascal

3) C++ Builder

4) Symantec Café.

Из универсальных языков программирования сегодня наиболее популярны следующие:

Бейсик (Basic) – для освоения требует начальной подготовки (общеобразовательные школы)

Паскаль (Pascal) – требует специальной подготовки (школы с углубленным изучением предмета и общетехнические вузы)

Си++ (C++), Ява (Java) – требуют профессиональной подготовки (специализированные средние и высшие учебные заведения)

Именно эти системы и языки программирования в дальнейшем будут определять развитие информатики.

**Список использованных источников**

* 1. Зуев Е.А. прграммирование на языке Turbo Pascal 6.0, 7.0. – М.: Веста, Радио и связь, 1993, – 376 с.
  2. Информатика: Базовый курс/ Симонович С.В. и др. – СПб.: Питер, 2001. – 640 с.
  3. Моначов В. Язык программирования Java и среда NetBeans . – 2-е изд. – СПб.: БХВ-Петербург, 2009. – 720 с.
  4. Моргун А.Н. Справочник по Turbo Pascal для студентов. – М.: Диалектика, 2006. – 608 с.
  5. Сайлер Б., Споттс Д. Использование Visual Basic 6. Классическое издание. – М.: Вильямс, 2007. – 832 с.
  6. Сафронов И. Visual Basic в задачах и примерах. – СПб.: «БХВ-Петербург», 2008. – 400 с.
  7. Стефенс Д. Р*.* C++. Сборник рецептов. — КУДИЦ-ПРЕСС, 2007. — 624 с.
  8. Страуструп Б. Язык программирования С++ = The C++ Programming Language / Пер. с англ. — 3-е изд. — СПб.; М.: Невский диалект — Бином, 1999. — 991 с.
  9. Угринович Н. Информатика и информационные технологии. Учебник для 10 – 11 классов. 4-е изд. – М.: Бином. Лаборатория знаний, 2007. – 511с.
  10. Фаронов В.В. Turbo Pascal. Наиболее полное руководство. – СПб.: Питер, 2007. – 763 с.
  11. Хорстманн К.С., Корнелл Г. Java 2. Библиотека профессионала, том 1. Основы. – М.: Вильямс, 2008. – 816 с.
  12. Шилдт Г. Полный справочник по С++. – М.: «Вильямс», 2006. – 800 с.