

ИСТОРИЯ РАЗВИТИЯ ЯЗЫКОВ ПРОГРАММИРОВАНИЯ

А. В. ШИЛО¹, Б. А. ЖЕЛЕЗКО²

¹ – студент 2 курса, ФЭФ, группы СФФ-5, Белорусского государственного экономического университета

² – научный руководитель, кандидат технических наук, профессор кафедры информационных технологий, Белорусского государственного экономического университета, Минск, 220672, Партизанский пр., 26, тел. (8017) 249-19-81, e-mail: zhelezko_b@bseu.minsk.by

Аннотация. В данной работе кратко рассмотрена история развития языков программирования: Basic, Fortran, Cobol, PL/1, C, C++, Pascal, Forth, Prolog, Ada, Modula-2, Oberon, Oberon-2, Java.

Ключевые слова: языки программирования, операционная система, компилятор.

1. РАЗВИТИЕ ПОКОЛЕНИЙ ЯЗЫКОВ ПРОГРАММИРОВАНИЯ

Языки программирования по своим возможностям и времени создания принято делить на несколько поколений. Каждое из последующих поколений по своей функциональной мощности качественно отличается от предыдущего. К сегодняшнему дню насчитывается пять поколений языков программирования.

В первое поколение входят языки, созданные в 40-50 гг., когда компьютеры только появились на свет. В то время программы писались в машинных кодах, то есть каждая компьютерная команда вводилась в ЭВМ в двоичном виде. Это требовало огромных усилий по набору цифровых текстов и приводило к множеству трудноуловимых ошибок. Конечно, ни о каких мало-мальски больших проектах речи идти не могло. Ситуация качественно изменилась в середине 50-х годов, когда был написан первый ассемблер, что само по себе можно считать подвигом, учитывая довольно сложную логику программы. Хотя этот ассемблер был неполноценным в сегодняшнем понимании, но он позволял задавать названия команд в символическом виде и указывать числа не только в двоичном, но и в десятичном или шестнадцатеричном формате, что существенно облегчало работу программистов.

Языки первого поколения продолжают использоваться и сегодня, хотя в значительно меньшем объеме. Чаще всего приходится писать программы в машинных кодах для новых микропроцессоров, для которых еще

не разработаны компиляторы, поддерживающие требуемый набор команд.

Расцвет второго поколения языков программирования пришелся на конец 50-х - начало 60-х годов. Был создан символический ассемблер, позволявший писать программы без привязки к конкретным адресам памяти. В него было введено понятие переменной, и он по сути стал первым настоящим (хоть и машинно-ориентированным) языком программирования со своим компилятором. Скорость создания и эффективность программ резко возросли. Ассемблеры активно применяются в настоящее время, как правило, для создания программ, максимально использующих возможности аппаратуры – различных драйверов, модулей состыковки с нестандартным оборудованием и т. д. В некоторых областях, например, в машинной графике, на ассемблере пишутся библиотеки, эффективно реализующие стандартные алгоритмы обработки изображений.

Третье поколение языков программирования принято относить к 60-м годам. В это время родились языки, которые называют универсальными языками высокого уровня, с их помощью можно решать задачи из любых областей. Это общеизвестные Фортран, Кобол, Алгол и др. Такие качества новых языков, как относительная простота, независимость от конкретного компьютера и возможность использования мощных синтаксических конструкций позволили резко повысить производительность труда программистов.

Практически все современные коммерческие продукты, рассчитанные на

массовый рынок, написаны на языках 3-го поколения.

С начала 70-х годов по настоящее время тянется период языков четвертого поколения. После первых восторгов по поводу безграничных способностей ЭВМ стали более ясны возможности существующих языков программирования. Несмотря на рождение новых технологий (ООП, визуальное программирование, CASE-методологии, системный анализ), процесс создания больших программных комплексов оказался очень трудоемкой задачей, так как для реализации крупных проектов требовался более глобальный подход к решаемым задачам, чем предлагали имевшиеся средства разработки. Языки 4-го поколения частично снимали эту проблему. Целью их создания было, в первую очередь, стремление к увеличению скорости разработки проектов, снижению числа ошибок и повышению общей надежности работы больших программных комплексов, возможность быстрого и легкого внесения изменений в готовые проекты, упрощение самих языков для конечного пользователя, активное внедрение технологий визуальной разработки и т. д. Все средства разработки 4-го поколения имеют мощные интегрированные оболочки и обладают простым и удобным пользовательским интерфейсом. Они чаще всего используются для проектирования баз данных и работы с ними (встроенные языки СУБД), что объясняется возможностью формализации всех понятий, используемых при построении реляционных баз данных. Языки 4-го поколения активно применяются в различных специализированных областях, где высоких результатов можно добиться, используя не универсальные, а проблемно-ориентированные языки, оперирующие конкретными понятиями узкой предметной области. Как правило, в эти языки встраиваются мощные примитивы, позволяющие одним оператором описать такую функциональность, для реализации которой на языках младших поколений потребовалось бы написать тысячи строк кода.

Рождение языков пятого поколения относится к настоящему времени. Довольно неожиданно вокруг самого названия (пятое поколение) разгорелись жаркие споры. Возникло несколько программистских

"школ", представители каждой из которых имеют свое мнение о том, какие средства разработки считать языками 5-го поколения, а какие нет. Считают, что к системам 5-го поколения можно отнести не только новые мощные языки, но и системы создания программ, ориентированные на непрограммиста. Подобные системы отличаются стремлением предоставить конечному пользователю-неспециалисту богатые возможности создания прикладных программ с помощью визуальных средств разработки без знания программирования.

В целях обеспечения возможности создания программ, решающих самые разные задачи, разработчики стараются использовать богатый многолетний опыт программной индустрии. Системы 5-го поколения имеют открытую архитектуру и нередко поддерживают большое количество продуктов третьих фирм, предоставляя пользователю возможность интеграции с готовыми решениями для различных областей. Это могут быть всевозможные визуальные редакторы, генераторы отчетов, стандартные библиотеки, удобные Мастера (Wizards) быстрого создания типовых приложений, CASE-системы, средства интеграции с базами данных и т. п. Чем больше приложений удастся объединить в одном пакете, тем большими возможностями он будет обладать.

Таким образом, определенная часть компьютерных экспертов считают продукты последнего поколения уже не языками, а средствами разработки, прикладными пакетами, не имеющими к процессу создания программ с помощью языков программирования никакого отношения. Проектирование программы происходит в специализированном визуальном редакторе, и работа с исходными текстами отсутствует.

Однако значительно большая группа специалистов считает, что языки 5-го поколения являются именно языками программирования, требующими от разработчика соответствующей квалификации и умения составлять программы вручную. Это, как правило, узкоспециализированные языки, предоставляющие программисту мощные высокоуровневые возможности обработки информации из конкретной области знаний. К языкам 5-го поколения относят также интегрированные с базами знаний и

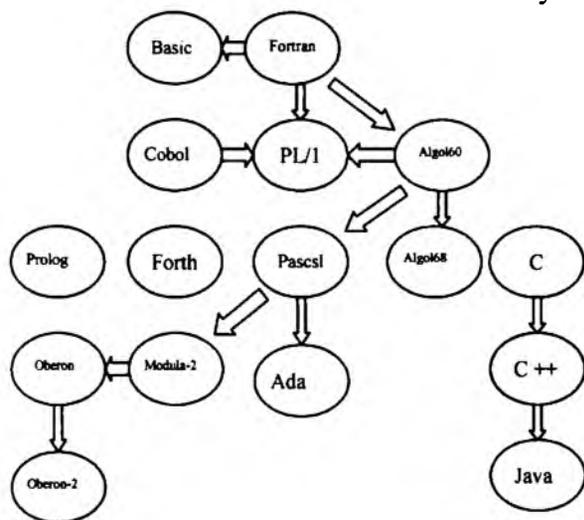
экспертными системами программные комплексы с собственными языками программирования.

Наиболее актуальными для систем 5-го поколения станут достижения в таких областях, как логическое программирование (пролог -подобные языки и машины вывода), исследовательское программирование (когда проект слишком сложен и неясен, а средства разработки позволяют быстро создать шаблон программы и включать в него работающие кусочки с постепенным приближением к конечному результату), использование для создания программ естественных языков, технологии управления базами знаний, методы обработки и анализа текстовой информации (энциклопедии, Web-страницы, документы) с возможностью смыслового поиска и т. д.

Языки 5-го поколения, ориентированные на конкретные области применения, уже в ближайшее время могут завоевать самую широкую популярность. Наиболее вероятно это для продуктов, позволяющих создавать приложения для работы с базами данных, области информатики, наиболее успешно поддающейся формализации. Наглядное подтверждение этому – тенденции развития практически всех известных СУБД корпоративного уровня.

2. ЯЗЫКИ ПРОГРАММИРОВАНИЯ

Рисунок.



Иерархия языков программирования

2.1. BASIC

Джон Кемени и Томас Курц, сотрудники математического факультета Дармутского колледжа, написали простой и легкий в

изучении язык, который назвали Бейсик (Basic Beginner's All-purpose Symbolic Instruction Code - универсальный код символических инструкций для начинающих).

В конце 1975 г. Был создан первый интерпретатор Крошки-Бейсика, состоящий примерно из 20 страниц восьмеричного кода. Его создали два программиста-любителя Дик Уиппл и Джон Арнольд. В США выпускался журнал "Dr. Dobb's Journal of computer Calisthenics & Orthodonta: Running Light without Overbyte.", посвященный исключительно Крошке-Бейсику. В это же время изготовитель "Альтаира" фирма MITS (Micro Instrumentation and Telemetry Systems) начала продавать собственную версию языка Бейсик. Эту версию в 1975 г. написал первокурсник Гарвардского университета Бил Гейтс и его друг Пол Аллен, молодой программист фирмы "Хониуэл".

Эпоха Крошки-Бейсика завершилась в конце 70-х годов. Микрокомпьютеры выпускались теперь в полностью собранном виде, изготовители встраивали интерпретаторы версий Бейсика непосредственно в ПЗУ компьютера. Вскоре наибольшую популярность завоевал М-Бейсик.

Характерные черты языка:

- диалоговый режим работы;
- нумерация строк;
- вещественный и символьный типы данных,
- управляющие конструкции;
- все переменные являются глобальными,
- наличие массивов.

2.2. FORTRAN

В 1954 г. группа разработчиков фирмы IBM под руководством Джона Бекуса приступила к разработке языка программирования. Местом рождения Фортрана стала штаб-квартира фирмы IBM в Нью-Йорке. Несмотря на это группа разработчиков продолжала работу над языком. Работа над компилятором языка оказалась не простой и продолжалась более двух лет. В апреле 1957 г. компилятор языка Фортран был готов для использования владельцами машины IBM-704. Вначале Фортран не хотели признавать, однако по сравнению со своими предшественниками

Фортран был весьма прост для обучения и использования. Через пять лет Фортран использовался на шести различных моделях компьютеров фирмы IBM. Новый язык не был лишен недостатков, устранение которых привело к появлению Фортрана 2, который позволял присоединять программы на ассемблере. В 1958 г. была создана версия Фортран 3, а версия Фортран 4, расширившая возможности языка, появилась в 1962 г. Фортран оказался первым широко распространенным коммерческим языком.

2.3. SOVOL

В 1959 г. состоялась конференция по языкам программирования. Ее возглавил Чарльз Филлипс, директор Института исследований систем обработки данных министерства обороны США. Эта конференция получила название CODASYL (Conference on Data System Languages - конференция по языкам систем обработки данных). Причиной созыва конференции послужило создание единого языка для деловых приложений.

К середине осени новый язык обрел определенный вид и имя. Он был назван Кобол (Common Business Oriented Language - универсальный язык, предназначенный для бизнеса).

Популярность Кобола возросла после публикации в 1962г. его новой версии. Язык был удобен для сферы бизнеса, машинно-независим и легок для восприятия. Кобол - программа состоит из четырех самостоятельных разделов, следующих в строго в определенном порядке: раздел идентификаций, раздел оборудования, раздел данных, раздел процедур.

Раздел идентификаций задает название программы и содержит всю справочную информацию о ней. В разделе оборудования приводятся параметры ЭВМ, для которой предназначена исходная программа на Коболе. Раздел данных описывает обрабатываемые данные. Раздел процедур включает операторы, определяющие действия над обрабатываемыми данными.

Текст программы записывается с помощью основных символов, образующих алфавит языка.

2.4. PL/1

В 1961 г. IBM, чтобы сохранить первенство на рынке компьютеров, занялась изготовлением "Системы-360". Эти машины должны были удовлетворить в равной степени потребности как в аналитических вычислениях, так и в обработке данных в сфере бизнеса, и в специальных приложениях. Компьютер с такими возможностями требовал мощного языка. За полгода до окончания работы над аппаратным обеспечением IBM начала заниматься языком.

Был создан "комитет по разработке передового языка". Комитет состоял из представителей фирм "Lockheed", "Union Carbide", "Standard Oil" из Калифорнии и специалисты из отделов программирования фирмы IBM. Комитет возглавил Джордж Рэдин. Они приступили к работе в октябре 1963г, и к февралю 1964 г. спецификации языка были завершены.

Этот язык был назван PL/1 (от programming language one) работа над компилятором была поручена лаборатории фирмы IBM в Англии. Многие черты Фортрана, Кобола и Алгола нашли отражение в PL/1. Критики отмечали, что язык наделен слишком многими качествами, что привело к повтору недостатков предыдущих языков [1].

2.5. C

Язык Си был разработан и реализован Д. Ритчи в 1972г. в фирме Bell Labs, использовался в самых различных приложениях, в основном под управлением операционной системы UNIX. Он представлял собой дальнейшее развитие языка Би (B), который основывался на созданном в Кембриджском университете языке BCPL (от Basic Combined Programming Language- базовый комбинированный язык программирования), который в свою очередь был потомком Алгола-60.

В 1977г. была начата работа по созданию машинно-независимой версии транслятора с языка Си с целью облегчить его перенос в новые окружения. Результатом этого явилось появление совместимых по входному языку трансляторов с языка Си для 15 различных типов ЭВМ.

В ходе работ по созданию мобильного транслятора с языка Си сам он был переработан для повышения мобильности написанных на нем программ [2].

2.6. C ++

Первые версии языка программирования Си++ (тогда он назывался "Си с классами") были разработаны в начале 80-х годов Бьярном Страуструпом, сотрудником знаменитой AT&T Bell Labs, где ранее были разработаны операционная система UNIX и язык программирования Си. По признанию самого автора языка, Си++ никогда не разрабатывался на бумаге. Проектирование, реализация и документирование новых возможностей происходили фактически одновременно. Единственной целью разработки было создание языка, на котором было бы удобно программировать автору и его друзьям. За основу был взят популярный в среде профессиональных разработчиков язык программирования Си. Первыми средствами, которыми был расширен Си, стали средства поддержки абстракций данных и объектно-ориентированного программирования.

Первыми средствами, которыми был расширен Си, стали средства поддержки абстракций данных и объектно-ориентированного программирования. Как это принято в AT&T, описание нового языка не было опубликовано сразу. Первыми его пользователями стали сами сотрудники Bell Labs.

С 1985 года в язык были введены новые возможности: множественное и виртуальное наследование, шаблоны функций и классов, обработка исключительных ситуаций. Кардинально изменена семантика совместного использования оператора new, изменен синтаксис для вложенных классов. С момента опубликования и до настоящего момента язык постоянно совершенствовался и расширялся. Важным этапом в его развитии стала публикация в 1990 году подробного и достаточно строгого описания языка. Фактически одновременно с этим началась стандартизация языка. Инициатором стандартизации выступил не автор языка. Более того, Страуструп всегда довольно прохладно относился к попытке его полной стандартизации и выступал за реализации, в которых базовые возможности языка расширялись бы средствами и библиотеками, характерными только для данной реализации.

Объединенный ANSI-ISO (ANSI X3J16; ISO WG21/N0836) комитет начал

функционировать в конце 1989 года. Целью его работы является создание единого стандарта для языка Си++ и его библиотечных средств. За основу проекта стандарта было взято описание языка, данное в 1990 году. В работе объединенного комитета значительное место занимает изучение возможных изменений текста проекта стандарта, а также уточнение различных правил языка. Позволим себе напомнить, что непосредственный предшественник Си++ - язык Си прошел успешно процесс стандартизации. Работа по его стандартизации завершилась в 1989 году, и стандартизованный вариант сейчас известен под именем ANSI Си. Работа по стандартизации Си++ осложнялась тем, что язык долгое время был открыт для расширений. Си++ стал довольно громоздким языком (сопоставимым разве что с языком Ada), и ни один человек сейчас не в состоянии точно помнить все его детали и тонкости. С момента начала стандартизации несколько изменилась и сама идеология C++. Изначально автор отвергал возможность использования в языке средств динамического определения типов, однако в текущем проекте стандарта такие средства имеются. Изначально планировалось, что окончательная редакция проекта стандарта будет опубликована в 1994 году. Эти сроки были безнадежно провалены. Согласно расписанию работы комитета по стандартизации, проект стандарта принятый в апреле 1995 года, был предложен для публичного обсуждения и сделан доступен пользователям Internet.

2.7. PASCAL

Язык программирования Pascal был создан Никлаусом Виртом, и назван в честь французского философа и математика XVIIв. Блеза Паскаля. В то время Вирт был профессором информатики в Федеральном техническом университете в Швейцарии и нуждался в языке, с помощью которого можно было обучать студентов навыкам программирования.

Концепция Паскаля была разработана Н. Виртом примерно в 1970 году и Паскаль быстро получил широкое распространение благодаря легкости его изучения, наглядности составленных на нем текстов программ. Поскольку Паскаль послужил основой для разработки других языков

программирования, таких как Ада и Модуль-2, и поскольку многие языки содержат аналогичные Паскалю структуры, знание Паскаля является солидной базой для изучения других языков программирования.

Язык Паскаль стараниями Андерса Хейлсберга превратился в мощную профессиональную систему программирования Turbo Pascal. После просуществовавшей сравнительно недолго и не получившей широкое распространения версии 1.0 в середине 1984 года появляется версия 2.0. Распространение которой пошло стремительными темпами. К осени 1985 года появляется версия 3.0, отличающаяся от версии 2.0 следующими особенностями: компилятор и редактор стали работать существенно быстрее, появилась возможность передачи параметров в программу с помощью команды RUN, стал возможным вызов MS-DOS из программы, стала удобной работа с файлами. С начала 1988 года начинает распространяться версия 4.0. Здесь Турбо Паскаль представлен в совершенно новой форме. Не только становится еще более быстрым компилятором, но и появляется совершенно новое программное окружение. Появилась возможность разбиения программы на части (модули), компилируемые по отдельности. К осени 1988 года появилась версия 5.0 с еще более развитым программным окружением. Здесь заслуживает внимания прежде всего встроенный отладчик. Турбо Паскаль может использоваться в большинстве существующих для персональных компьютеров операционных систем.

Появление инструментальных средств Borland Pascal with Objects и Delphi для разработки программ в среде Windows лишний раз показала какие по истине неисчерпаемые возможности таит в себе Паскаль. Borland Pascal и используемый в Delphi язык Object Pascal основываются на Turbo Pascal и развивают его идеи.

Delphi - компилятор языка Pascal. Delphi 1 был первым инструментарием разработки Windows приложений, объединившим в себе оптимизирующий компилятор, визуальную среду программирования и мощные возможности работы с базами данных.

Годом позже Delphi 2 предложил все то же, но на новом уровне современной 32-битной операционной системы Windows 95 и Windows NT [3].

2.8. FORTH

Язык программирования Форт был разработан Чарльзом Муром. Первоначально язык назывался FOURTH, однако на ЭВМ, на которой он работал, символьные имена могли иметь только пять букв. Так язык стал называться FORTH. Несмотря на конкуренцию других языков программирования, в частности языка Си, Форт мало-помалу стал завоевывать популярность, особенно при решении задач управления сложными объектами в реальном времени.

Характерные черты языка Форт:

- стек - единственная структура данных;
- основной тип данных - целый;
- простой синтаксис и компактная запись;
- малая потребность в ресурсах;
- быстрая интерпретация на основе машинного кода;
- отсутствие контроля за переменными.

Язык Форт использовался для математического обеспечения корабля многоцелевого использования типа Шаттл, спутников Земли, для разработки телеигр, при создании мультфильмов Stars Wars и т.д. В 1976г. Комитет международного астрономического союза принял Форт в качестве стандартного языка программирования. Позднее Форт применялся для создания экспертных систем, систем искусственного зрения, автоматизации анализа крови и кардиологического контроля.

В США создано общество пользователей Форты (FIG, FORTH Interesting Group) [4].

2.9. PROLOG

Язык программирования Пролог (PROgramming LOGic) появился в 1972 г., Он был разработан Аланом Колмери из университета в Лумини (Марсель). Пролог в отличие от своих "сверстников", продолжительное время развивался, применялся и обсуждался в сравнительно узком кругу исследователей, работающих в области искусственного интеллекта.

Алан Колмери, автор языка Пролог, начал работы над полноценной компьютерной реализацией трудов Ковальского с 1972 года

во французском университете Марсель-Экс. Он составил алгоритм формального способа интерпретации процесса логического вывода и разработал систему автоматического доказательства теорем, которая была написана на Фортране. Она-то и послужила прообразом Пролога. Этот язык был выбран для решения задач в различных областях, включающих:

- математическую логику;
- решение абстрактных задач;
- понимание естественного языка;
- автоматизацию проектирования;
- символьное решение уравнений;
- анализ биохимических структур.

Пролог является диалоговым языком. Программа на Прологе состоит из множества утверждений, каждое из которых является либо фактом о заданной информации, либо правилом, указывающим, как решение связано с заданными фактами или каким образом его можно из них вывести. В отличие от подавляющего большинства других языков Пролог обычно рассматривается в одном контексте с понятием “логическое программирование”. Фактически Пролог является не процедурным, а декларативным языком. Человек лишь описывает структуру задачи, а Пролог сам ищет решение [5].

2.10. ADA

Язык Ада был разработан при финансовой поддержке Министерства обороны США. В 1978 году был объявлен конкурс на язык программирования, который бы отвечал следующим требованиям:

- возможность использовать неоднородные структуры;
- разделение памяти;
- реализация неявных функций преобразования типов.

Это конкурс завершился появлением в июле 1980г. спецификаций языка Ада. Этот язык был разработан международным коллективом программистов фирмы Honeywell и ее французского филиала Cii-Bull, которым руководил Ж. Ишбиа. Язык был назван в честь Августы Ады Лайвейс, считающейся первой программисткой, дочери английского поэта Байрона.

Окончательная версия спецификаций языка Ада появилась в феврале 1983г. Она послужила основой для нового стандарта

Министерства обороны США и стандарта Американского национального института стандартов (ANSI). Ada95 - эта версия является первой в мире объектно-ориентированной системой программирования, на которую был введен международный стандарт. Впоследствии МО решило использовать “Ada”, как свой внутренний сертифицированный знак [2].

2.11. MODULA-2

Язык программирования Модула-2 был создан Н. Виртом в 1979г и впервые реализован на мини-ЭВМ PDP-11. В 70-х гг. Паскаль получил широкое признание у пользователей ЭВМ и преподавателей, однако первоначально он был разработан для обучения программированию и имел множество недостатков как язык разработки программного обеспечения. В Модуле-2 эти недостатки были устранены, но при этом сохранены логическая структура и характерные черты его предшественника. Кроме того, в Модуле-2 были введены новые мощные языковые средства. В мае 1986г. в ВЦ СОАН СССР состоялся первый Всесоюзный семинар по Модуле-2 и инструментальным системам на его основе.

Язык программирования Модула-2 относится к машинно-независимым языкам. Н. Вирт применил Модулу-2 для написания полной операционной системы для мини-ЭВМ Lilith. Характерной чертой Модулы-2 является отдельная компиляция, позволяющая разрабатывать и хранить в библиотеках программы, которые можно использовать повторно [6].

2.12. OBERON

Язык Оберон был создан в 1987 году Никлаусом Виртом, профессором Института компьютерных систем Федерального технического университета (ETH, Цюрих, Швейцария), автором языков Паскаль и Модула-2. Язык носит имя спутника планеты Уран. Оберон отличается от Модулы-2 отсутствием многих необязательных конструкций; добавлены средства объектно-ориентированного программирования - расширяемые записи.

Оберон - это самый простой универсальный язык. При этом, продолжая традицию Паскаля и Модулы-2, он обеспечивает строгий контроль на этапе

трансляции, способствуя созданию надежных программ.

2.13. OBERON-2

В 1992 году были приняты расширения языка Оберон, предложенные Х. Мёссенбёком. Расширенный язык получил название Оберон-2. Основное нововведение - связанные с типами процедуры. Сейчас Оберон-2 является фактическим стандартом языка.

Летом 1993 года в лондонском пригороде Croydon в отеле "Oakwood" состоялась конференция разработчиков Оберон-компиляторов и программистов, на которой были согласованы требования к составу библиотечных модулей, сопровождающих реализации Оберона. Принятый документ известен под названием "Oakwood guidelines" ("Дубовые требования").

Первоначально Оберон возник в рамках проекта создания операционной системы для однопользовательской рабочей станции Ceres. Эта ОС, также носящая имя Оберон, была написана (на Обероне) Н.Виртом и Юргом Гуткнехтом.

Впоследствии эта Оберон-система была перенесена на множество платформ: Amiga, Linux, Macintosh, Unix (DECstation, HP station, RS600, Silicon Graphics, SparcStation), PowerMac, Windows. Сейчас это семейство оригинальных Оберон-систем известно под названием Oberon V4.

2.14. JAVA

Начало 1996 года ознаменовано появлением нового языка программирования Java. На домашней странице WWW Consortium Java была внесена в список так называемых Мобильных Кодов - одного из перспективных направлений развития технологии World Wide Web. И вот в конце 1996 года на Западе начался бум Java, который к моменту проведения выставки Unix-Expo'96 докатился и до нашей страны.

Согласно истории технология Java (Кофе) родилась из проекта Oak (Дуб), основной целью которого была разработка объектно-ориентированных средств описания и коммуникации различного рода электронных устройств. Из-за неудачи этого проекта 1994 году опыт, накопленный в рамках его реализации, было решено применить к продуктам ориентированным на применение

в Internet. С апреля 1995 года по сети свободно распространяется HotJava - интерфейс просмотра страниц World Wide Web для платформ Sun. Буквально через месяц Netscape Communication - законодатель моды в разработке программ-интерфейсов Internet, покупает лицензию на Java . В настоящее время HotJava реализована не только для SunOS и Solaris, но и для многих других Unix-платформ и Windows NT. Кроме HotJava мобильный код Java может интерпретироваться и второй версией программы Netscape Navigator для всех систем кроме Windows 3.x.

Система программирования на Java позволяет компилировать программы для компьютерной платформы, на которой она стоит в том же ключе как и любая другая, например, C или C++. В этом случае главными отличиями Java-программ, которые называются Java-applications, является использование библиотеки Java-классов, которые обеспечивают разработку безопасных, распределенных систем. При этом утверждается, что язык позволяет делать гораздо меньше ошибок при разработке программ. Главным при этом является тот факт, что в Java напрочь отсутствует адресная арифметика.

Гораздо более интересным является разработка мобильных Java байт-кодов, которые в терминах Java-технологии называются applets [7].

ЛИТЕРАТУРА

1. Фролов Г.Д. Практический курс программирования на языке PL-1. - Издательство "Наука", 1983 г.
2. Языки программирования Ада, Си, Паскаль. Под ред. Фьюэра А.Р. - Издательство "Радио и связь", 1989 г.
3. Фаронов В.В. Turbo Pascal 7.0. - Издательство "Нолидж", 1997 г.
4. Романовский И.В. Начальный курс программирования на языке Форт. - Изд. "Финансы и Статистика", 1990г.
5. У. Клоксин, К. Меллиш. Программирование на языке Пролог. - Издательство "Мир", 1987 г.
6. Э. Непли, Р. Платт. Программирование на языке Модула-2. - Издательство "Радио и связь", 1989 г.
7. Д. Флэнэген. JAVA in a nutshell. - Издательство "BHV", 1997 г.