МИНОБРНАУКИ РОССИИ

ГОСУДАРСТВЕННОЕ ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО ПРОФЕССИОНАЛЬНОГО ОБРАЗОВАНИЯ

“ВОРОНЕЖСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ”

Факультет *романо-германской филологии*

Кафедра *французской филологии*

*Теория и методика преподавания иностранных языков и культур*

Курсовая работа

*Языки программирования. История и современное состояние*

Студентка \_\_\_\_\_\_\_\_\_\_ Суховеркова А.А.

Руководитель \_\_\_\_\_\_\_\_\_\_ Донина О.В.

Воронеж 2018

# Содержание

Введение

1. Языки программирования
   1. История развития языков программирования
2. Современные языки программирования
   1. Си. Его разновидности
   2. PASCAL
   3. BASIC
3. Заключение
4. Список использованной литературы

## Введение

На современном этапе развития компьютерных технологий невозможно представить высококвалифицированного специалиста, не владеющего информационными технологиями. Поскольку деятельность человека в значительной степени зависит от степени владения информацией, а также способности эффективно ее использовать. Для свободной ориентации в информационных потоках современный специалист любого профиля должен уметь использовать, получать и обрабатывать информацию, прежде всего с помощью компьютеров, телекоммуникаций, а также других новейших средств связи, в том числе и уметь обращаться с языками программирования.

Актуальность данной темы обусловлена тем, что стремительное развитие компьютерных технологий определил процесс появления новых разнообразных знаковых систем для записи алгоритмов – языков программирования.

Объектом исследования стали языки программирования и история развития языков программирования.

Целью курсовой работы является изучение языков программирования и их развития, обзор современных языков программирования.

Цели исследования:

1. Просмотреть общее сведения и уровни языков программирования.

2. Просмотреть историю развития языков программирования.

3. Сделать обзор современных языков программирования

Задачи исследования:

1. Ознакомление с языками программирования.

2. Рассмотрение истории развития языков программирования.

3. Обзор современных языков программирования.

В первой главе рассматриваются общие сведения о языках программирования и их история развития.

Во второй главе рассматривается обзор современных языков программирования

## Языки программирования

Язык программирования — формальный язык, предназначенный для записи компьютерных программ. Язык программирования определяет набор лексических, синтаксических и семантических правил, определяющих внешний вид программы и действия, которые выполнит исполнитель (обычно — ЭВМ) под её управлением.

Языки программирования являются искусственными языками. От естественных языков они отличаются ограниченным числом “слов” и очень строгими правилами записи команд (операторов). Поэтому при применении их по назначению они не допускают свободного толкования выражений, характерного для естественного языка.

Можно определить ряд требований к языкам программирования и классифицировать эти языки по их особенностям

Основные требования, предъявляемые к языкам программирования:

|  |
| --- |
| Единство - использование одних и тех же символов для обозначения одних и тех же или родственных понятий в разных частях алгоритма. Количество этих символов должно быть по возможности минимальным; |
| Наглядность - использование в языке по возможности уже существующих символов, хорошо известных и понятных как программистам, так и пользователям ЭВМ; |
| Гибкость - возможность относительно удобного, несложного описания распространенных приемов математических вычислений с помощью имеющегося в языке ограниченного набора изобразительных средств; |
| Модульность - возможность описания сложных алгоритмов в виде совокупности простых модулей, которые могут быть составлены отдельно и использованы в различных сложных алгоритмах. |
| Однозначность - недвусмысленность записи любого алгоритма. Отсутствие ее могло бы привести к неправильным ответам при решении задач. |

На данный момент в мире существует несколько сотен постоянно используемых языков программирования. Для каждого из этих языков есть своя область применения.

Любой алгоритм – это последовательность предписаний, выполнив которые можно за конечное число шагов перейти от исходных данных к результату. В зависимости от степени детализации предписаний обычно определяется уровень языка программирования — чем меньше детализация, тем выше уровень языка.

По этому критерию возможно выделить следующие уровни языков программирования:

|  |  |  |
| --- | --- | --- |
| машинные | машинно- ориентированные (ассемблеры) | машинно- независимые (языки высокого уровня) |

Машинно-ориентированные языки и машинные языки — это языки низкого уровня. Они требуют указания мелких деталей процесса обработки данных. В то время как языки высокого уровня имитируют естественные языки, используя отдельные слова разговорного языка и общепринятые математические символы. Такие языки, несомненно, более удобны для человека.

Если язык программирования ориентирован на определенный тип процессора и учитывает его особенности, то он называется языком программирования низкого уровня. Здесь “низкий уровень” не значит “плохой”. Имеется в виду, что операторы языка близки к машинному коду и ориентированы на конкретные команды процессора.

При программировании на машинном языке программист может держать под своим контролем каждую команду и каждую ячейку памяти, использовать все существующие возможности имеющихся машинных операций. Но процесс написания программы на машинном языке очень утомительный и трудоемкий. Программа получается громоздкой, труднообозримой, ее трудно отлаживать, изменять и развивать.

Поэтому в случае, когда нужно иметь эффективную программу, в максимальной степени учитывающую специфику конкретного компьютера, вместо машинных языков используют близкие к ним машинно-ориентированные языки (ассемблеры).

Язык ассемблера (англ. assembly language) — машинно-ориентированный язык программирования низкого уровня. Его команды прямо соответствуют отдельным командам машины или их последовательностям. Этот язык используется для представления в удобочитаемой форме программ, записанных в машинном коде.

Язык ассемблера позволяет программисту пользоваться текстовыми мнемоническими (то есть легко запоминаемыми человеком) кодами, по своему усмотрению присваивать символические имена регистрам компьютера и памяти, а также задавать удобные для себя способы адресации.

С помощью языков низкого уровня создаются очень эффективные и компактные программы, потому что разработчик получает доступ ко всем возможностям процессора. В то же время, при этом требуется очень хорошо понимать устройство компьютера, а окончательная программа не может быть перенесена на компьютер с другим типом процессора. Подобные языки, как правило, применяют для написания небольших системных приложений, драйверов устройств, когда важнейшими требованиями становятся компактность, быстродействие и возможность прямого доступа к аппаратным ресурсам. В некоторых областях, например, в машинной графике, на языке ассемблера пишутся библиотеки, эффективно реализующие алгоритмы обработки изображений, требующие интенсивных вычислений.

Следовательно, программы, написанные на языке ассемблера, требуют значительно меньшего объема памяти и времени выполнения. Знание программистом языка ассемблера и машинного кода дает ему понимание архитектуры машины. Несмотря на то, что большинство специалистов в области программного обеспечения разрабатывают программы на языках высокого уровня, наиболее мощное и эффективное программное обеспечение полностью или частично написано на языке ассемблера.

Языки высокого уровня - были разработаны для того, чтобы освободить программиста от учета технических особенностей конкретных компьютеров. Уровень языка характеризуется степенью его близости к естественному, человеческому языку. Машинный язык не похож на человеческий, он крайне беден в своих изобразительных средствах. Средства записи программ на языках высокого уровня более выразительны и привычны для человека.

Важным преимуществом языков высокого уровня является их универсальность, независимость от ЭВМ. Программа, написанная на таком языке, может выполняться на разных машинах.

Таким образом, можно сформулировать основные преимущества языков высокого уровня перед машинными:

* алфавит языка высокого уровня значительно шире алфавита машинного языка, что повышает наглядность текста программы;
* набор операций, допустимых для использования, не зависит от набора машинных операций, а выбирается из соображений удобства формулирования алгоритмов решения задач определенного класса;
* формат предложений достаточно удобен и гибок для использования, что позволяет нам с помощью одного предложения задать достаточно содержательный этап обработки данных;
* требуемые операции задаются с помощью общепринятых математических обозначений;
* данным в языках высокого уровня присваиваются индивидуальные имена;
* в языке может быть предусмотрен гораздо более широкий набор типов данных по сравнению с набором машинных типов данных.

Следовательно, языки высокого уровня в значительной мере являются машинно-независимыми. Они облегчают работу программиста и повышают надежность создаваемых программ.

Языки высокого уровня делятся на:

* процедурные;
* логические;
* объектно-ориентированные.

Процедурные языки предназначены для однозначного описания алгоритмов. При решении задачи процедурные языки требуют в той или иной форме явно записать процедуру ее решения.

Первым шагом в развитии процедурных языков программирования было появление проблемно-ориентированных языков. В этом названии нашел отражение тот факт, что при их разработке идут не от «машины», а «от задачи»: в языке стремятся максимально полно учесть специфику класса задач, для решения которых его предполагается использовать.

Разнообразие классов задач привело к тому, что на сегодняшний день разработано несколько сотен алгоритмических языков. Правда, широкое распространение и международное признание получили лишь 10-15 языков. Среди них в первую очередь следует отметить: Fortran и Algol - языки, предназначенные для решения научно-технических задач, Cobol – для решения экономических задач, Basic – для решения небольших вычислительных задач в диалоговом режиме. В принципе каждый из этих языков можно использовать для решения задач не своего класса. Однако, как правило, применение оказывается не удобным.

Логические языки- (Prolog, Lisp, Mercury, KLO и др.) ориентированы на формализованное и систематическое описание задачи с тем, чтобы решение следовало из составленного описания. В этих языках указывается что дано и что требуется получить. При этом поиск решения задачи возлагается непосредственно на ЭВМ.

Объектно-ориентированные языки (Objective Caml, C++, Java, Object Pascal, и др.). Руководящая идея объектно-ориентированных языков заключается в стремлении связать данные с обрабатывающими эти данные процедурами в единое целое - объект.

## История развития языков программирования

* Программа – алгоритм, записанный на языке программирования.
* Программа – последовательность операторов языка.
* Языки программирования – искусственные языки, строго формализованные; существует правила записи операторов языка – синтаксис языка.

1. Машинный язык (40-50 годы XX в.).

Программы на машинном языке – очень длинные последовательности единиц и нулей, являлись машинно зависимыми, т.е. для каждой ЭВМ необходимо было составлять свою программу.

1. Ассемблер (начало 50-ых годов XX в.).

Вместо 0 и 1 программисты теперь могли пользоваться операторами (MOV, ADD, SUB и т.д.), которые похожи на английские слова. Программы на ассемблере также являются машинно-зависимыми. Для преобразования в машинный код использовался компилятор (спец. программа – переводчик в машинный код).

1. Первые языки программирования высокого уровня.

С середины 50-ых гг. XX в. начали создавать первые языки программирования высокого уровня (high-level language). Эти языки были Машино независимыми (не привязаны к опред. типу ЭВМ). Но для каждого языка были разработаны собственные компиляторы.

Примеры таких языков: FORTRAN (FORmula TRANslator; 1954) предназначен для научных и технических расчетов; COBOL (1959) был предназначен в основном для коммерческих приложений (обрабатывал большие объемы нечисловых данных) – Common Business-Oriented Language); язык BASIC (Beginner’s All Purpose Instuction Code – универсальный язык символьных инструкций для начинающих) (1964 г.)

1. Алгоритмические языки программирования.

С начала 80-ых г. XX в. начали создаваться языки программирования, которые позволили перейти к структурному программированию (использование операторов ветвления, выбора, цикла и практически отказ от частого использования операторов перехода (goto). К этим языкам относятся: язык Pascal (назван его создателем Никлаусом Виртом в честь великого физика Блеза Паскаля; 1970); язык Си, позволяющий быстро и эффективно создавать программный код (1971)

1. Языки объектно-ориентированного программирования

(90-ые г. XX в.). В основу этих языков положены программные объекты, которые объединяют данные и методы их обработки. В этих языках сохранялся алгоритмический стиль программирования. Для них были разработаны интегрированные среды программирования, позволяющие визуально конструировать графический интерфейс приложений:

язык С++ (1983) - продолжение алгоритм. языка Си;

язык Object Pascal (1989) был создан на основе языка Pascal. После создания среды программирования – Delphi (1995);

язык Visual Basic(1991) был создан корпорацией Microsoft на основе языка Qbasic (1975) для разработки приложений с графическим интерфейсом в среде ОС Windows.

1. Языки программирования для компьютерных сетей.

В 90-ые годы XX в. в связи с бурным развитием Интернета были созданы языки, обеспечивающие межплатформенную совместимость. На подключенных к Интернету компьютерах с различными ОС (Windows, Linux, Mac OS и др.) могли выполняться одни и те же программы. Исходная программа компилируется в промежуточный код, который исполняется на компьютере встроенной в браузер виртуальной машиной:

язык Java - объектно-ориентированный язык был разработан фирмой Sun Microsystems для создания сетевого программного обеспечения (1995);

язык JavaScript – язык сценариев Web-страниц (компания Netscape). (1995)

1. Языки программирования на платформе .NET.

Интегрированная среда программирования Visual Studio .Net, разработанная корпорацией Microsoft, позволяет создавать приложения на различных языках объектно-ориентированного программирования, в том числе:

на языке Visual Basic .Net ( на основе Visual Basic) - 2003 г.;

на языке Visual C# (С-шарп) – на основе языков С++ и J – 2003 г.;

на языке Visual J# (J-шарп) – на основе Java и JavaScript – 2003 г.

## Современные языки программирования

Алгоритмический язык представляет собой один из способов записи алгоритма. Язык программирования строго формализован, то есть все команды записываются по правилам и отступления от этих правил не допускаются. При записи команд нельзя изменить ни одного знака - возникает ошибка.

Каждый язык имеет свой алфавит – набор символов, которые можно использовать при записи программ на этом языке. Разные версии одного и того же языка могут немного различаться алфавитом.

Программа, написанная на языке программирования, состоит из команд, задающих последовательность действий. Эти действия выполняются над некоторыми объектами. Объектами могут быть числа, текстовые строки, переменные и другие. Языки отличаются друг от друга множеством допустимых объектов и набором операций, которые можно выполнять над этими объектами.

Первым значительным шагом представляется переход к языку ассемблера. Не очень заметный, казалось бы, шаг — переход к символическому кодированию машинных команд — имел на самом деле огромное значение.

Программисту не надо было больше вникать в хитроумные способы кодирования команд на аппаратном уровне. Более того, зачастую одинаковые по сути команды кодировались совершенно различным образом в зависимости от своих параметров.

Вместе с тем, переход к новому языку таил в себе и некоторые отрицательные стороны. Впервые в истории развития программирования появились два представления программы: в исходных текстах и в откомпилированном виде. Сначала, пока ассемблеры только транслировали мнемоники в машинные коды, одно легко переводилось в другое и обратно, но затем по мере появления таких возможностей, как метки и макросы, дизассемблирование становилось все более и более трудным делом. К концу ассемблерной эры возможность автоматической трансляции в обе стороны была утеряна окончательно. В связи с этим было разработано большой количество специальных программ-дизассемблеров, осуществляющих обратное преобразования, однако в большинстве случаев они с трудом могут разделить код и данные. Кроме того, вся логическая информация теряется безвозвратно.

Универсальные языки.

Иногда их делят на

* процедурно-ориентированные
* объектно-ориентированные

В настоящее время граница между этими видами стерлась. Эти языки используются чаще всего для решения самых разнообразных задач. И хотя каждый из языков имеет свои особенности, что делает его наиболее эффективными для решения определенного вида задач, но в принципе для решения любой задачи можно выбирать любой язык программирования.

Среди универсальных языков программирования в настоящее время наиболее распространены:



## Си и его разновидности

Си [C] - Разработан Денисом Ритчи в начале 1970-х гг. Многоцелевой язык программирования высокого уровня. Используется на миниЭВМ и ПЭВМ. Является базовым языком операционной системы Unix, но также применяется и вне этой системы, для написания быстродействующих и эффективных программных продуктов, включая и операционные системы. Для IBM PC имеется ряд популярных версий языка Си, в том числе - Turbo C (фирмы Borland), Microsoft C и Quick C (фирмы Microsoft ), а также Zortech C (фирмы Symantec). Многие из указанных версий обеспечивают также работу с Си и Си++.

* Си++ [C++] - создан Бьярном Страустрапом на базе языка Си. Язык программирования высокого уровня. Является его расширенной версией, реализующей принципы объектно-ориентированного программирования. Используется для создания сложных программ. Для IBM PC наиболее популярной является система Turbo C++ фирмы Borland (США).
* C# (C Sharp) – “ Си Шарп ”: объектно-ориентированный яык программирования, о разработке которого в 2000 г. объявила фирма Microsoft . По характеру он напоминает языки C++ и Java и предназначен для разработчиков программ, использующих языки C и С++ для того, чтобы они могли более эффективно создавать Интернет-приложения. Указывается, что C # будет тесно интегрирован с языком XML.



## PASCAL

Паскаль [PASCAL - акроним с французского - Program Applique a la Selection et la Compilation Automatique de la Litterature] - разработан в конце 1960-х гг. Никлаусом Виртом. Процедурно-ориентированный язык программирования высокого уровня, первоначально для обучения программированию в университетах. Назван в честь французского математика XVII века Блеза Паскаля.

В своей начальной версии Паскаль имел довольно ограниченные возможности, поскольку предназначался для учебных целей, однако последующие его доработки позволили сделать его хорошим универсальным языком, широко используемым в том числе для написания больших и сложных программ. Существует ряд версий языка (например, ETH Pascal, USD Pascal, Turbo Pascal ) и систем программирования на этом языке для разных типов ЭВМ. Для IBM PC наиболее популярной является система Turbo Pascal фирмы Borland (США).

Delphi является «наследником» языка Паскаль; основные операторы в этих языках одинаковы. Но Delphi имеет средство для работы с различными графическими объектами (создания форм, кнопок, меню), а также для обработки сложных структур данных. Поэтому он очень популярен при разработке различных Windows- приложений.

## BASIC

Бейсик [BASIC - Beginner’s All-purpose Symbolic Instruction Code] - разработан в 1963 - 1964 гг. в Дартмутском колледже Томасом Куртом и Джоном Кемени. Язык программирования высокого уровня..

Первоначально предназначался для обучения программированию. Отличается простотой, легко усваивается начинающими программистами благодаря наличию упрощенных конструкций языка Фортран и встроенных математических функций, алгоритмов и операторов. Существует множество различных версий Бейсика, которые не полностью совместимы друг с другом. Некоторые реализации Бейсика включают средства обработки данных и наборов данных.

## Заключение

Изобретение языков программирования высшего уровня, их совершенствование и развитие, позволило человеку не только общаться с машиной и понимать ее, но и использовать ЭВМ для сложнейших расчетов.

Сегодня, любое среднее и крупное предприятие, имеет в своем штате программистов, обладающими знаниями программирования различными языками, которые редактируют, изменяют, и модифицируют программы используемыми сотрудниками предприятия.

## Список использованной литературы

* 1. Язык ассемблера (https://ru.wikipedia.org/wiki/Язык\_ассемблера)
  2. Язык программирования (https://ru.wikipedia.org/wiki/Язык\_программирования)
  3. C++,Turbo Pasckal,QBasik:Эволюция языков программирования http://langprog.far.ru/historylangprog.html
  4. Островский В.А. Информатика: учеб. для вузов. М.: Высшая школа, 2000.
  5. СИ. Язык программирования (https://ru.wikipedia.org/wiki/Си\_(язык\_программирования))