АВТОНОМНАЯ НЕКОММЕРЧЕСКАЯ ОРГАНИЗАЦИЯ

ВЫСШЕГО ОБРАЗОВАНИЯ

БЕЛГОРОДСКИЙ УНИВЕРСИТЕТ

КООПЕРАЦИИ ЭКОНОМИКИ И ПРАВА

КАФЕДРА ИНФОРМАЦИОННЫХ СИСТЕМ И ТЕХНОЛОГИЙ

Доклад

ПО ДИСЦИПЛИНЕ «ИНФОРМАЦИОННЫЕ ТЕХНОЛОГИИ»

на тему: " Классификация языков программирования реферат"

Выполнил студент

факультета ТДиИТ

группы ПИН-81

Лопин А.С.

Проверил:

к.т.н., доц. УШАКОВА Н.Н.

Белгород 2019

Оглавление:

Введение………………………………………………………………………...3

## Классификация языков программирования…………………………………..4

### Машинный язык……………………………………………………………….5

## Поколения языков программирования………………………………………...………….………...6

Заключение……………………………………………………….……………..7

Список используемой литературы………………………………………….…8

**Введение:**

Общеизвестно, что информационные технологии являются одной из наиболее быстро развивающихся областей современной жизни. Новые технологии, проекты, названия и аббревиатуры появляются едва ли не каждый день.

Прогресс компьютерных технологий определил процесс появления новых разнообразных знаковых систем для записи алгоритмов – языков программирования. Смысл появления такого языка – оснащенный набор вычислительных формул дополнительной информации, превращает данный набор в алгоритм.

Язык программирования служит двум связанным между собой целям: он дает программисту аппарат для задания действий, которые должны быть выполнены, и формирует концепции, которыми пользуется программист, размышляя о том, что делать.  Первой цели идеально отвечает язык, который настолько "близок к машине", что всеми основными машинными аспектами можно легко и просто оперировать достаточно очевидным для программиста образом. Второй цели идеально отвечает язык, который настолько "близок к решаемой задаче", чтобы концепции ее решения можно было выражать прямо и коротко.

Связь между языком, на котором мы думаем и программируем, и задачами и решениями, которые мы можем представлять в своем воображении, очень близка. По этой причине ограничивать свойства языка только целями исключения ошибок программиста в лучшем случае опасно. Как и в случае с естественными языками, есть огромная польза быть, по крайней мере, двуязычным. Язык предоставляет программисту набор концептуальных инструментов, если они не отвечают задаче, то их просто игнорируют. Например, серьезные ограничения концепции указателя заставляют программиста применять вектора и целую арифметику, чтобы реализовать структуры, указатели и т.п. Хорошее проектирование и отсутствие ошибок не может гарантироваться исключительно за счет языковых средств.

## Классификация языков программирования

## Языки программирования низкого уровня

Языки программирования низкого уровня (машинно-ориентированные) (low-level language) – это языки, наборы операторов и изобразительные средства которых существенно зависят от особенностей ЭВМ (внутреннего языка, структуры памяти и т.д.). Эти языки программирования предназначены для определенного типа ЭВМ и отражают его внутренний машинный код. Особенности машинно-ориентированных языков:

* высокое качество создаваемых программ (компактность и скорость  выполнения);
* 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;
* низкая скорость программирования;
* невозможность непосредственного использования программ, составленных на этих языках, на ЭВМ других типов.

Машинно-ориентированные языки по степени автоматического программирования подразделяются на классы.

### Машинный язык

Самый простой язык. Представляет собой попросту процессорные инструкции в виде последовательностей из многоразрядных чисел, которые хранятся в компьютере представленными в виде единиц и нулей.

Машинный язык (computer (machine) language)– язык программирования, элементами которого являются команды компьютера, характеризующиеся:

* количеством операндов в команде;
* назначением информации, задаваемой в операндах;
* набором операций, которые может выполнить компьютер и др.

Каждому действию, инструкцию о выполнении которого может получить процессор, соответствует определенное число. Программа, написанная на машинном языке, представляет собой серию таких чисел с вкраплениями других чисел, являющих собой или параметры, контролирующие действия, или данные, на основе которых эти действия выполняются. Машинные языки различны для разных типов компьютеров. Поэтому говорят, что машинные языки машинно-зависимы. Машинные языки неудобны для человека и тяжело им воспринимаются. Программирование на них слишком медленно и утомительно. Это языки самого нижнего уровня. Считается, чем ближе к машине, тем ниже уровень языка.

## Поколения языков программирования

Языки программирования по своим возможностям и времени создания принято делить на несколько поколений (Generation Language, GL). Каждое из последующих поколений по своей функциональной мощности качественно отличается от предыдущего. К сегодняшнему дню насчитывается пять поколений ЯП.

В первое поколение (1GL) 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 вместе с ее операндами вводилась в ЭВМ в двоичном виде. Это требовало огромных усилий по набору цифровых текстов и приводило к множеству трудноуловимых ошибок. Ситуация качественно изменилась в середине 50-х годов, когда был написан первый ассемблер. 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 (2GL) 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.

**Заключение:**

Изобретение языка программирования позволило нам общаться с машиной, понимать её. Но если мы обратим внимание на темпы роста и развития новейших технологий в области программирования, то можно предположить, что в ближайшем будущем, человеческие познания в этой сфере, помогут произвести на свет языки, умеющие принимать, обрабатывать и передавать информации в виде мысли, слова, звука или жеста. Так и хочется назвать это детище компьютеризированного будущего: «языки программирования "высочайшего" уровня». Возможно, концепция решения этого вопроса проста, а ближайшее будущее этого проекта уже не за горами.

Размышляя над этим, хочется верить в прогресс науки и техники, в высоко - компьютеризированное будущее человечества, как единственного существа на планете, пусть и не использующего один, определенный разговорный язык, но способного так быстро прогрессировать и развивать свой интеллект, что и перехода от многоязыковой системы к всеобщему пониманию долго ждать не придется.

Целью данного обзора была попытка получить представление обо всем многообразии существующих языков программирования. Среди программистов часто бытует мнение о «всеобщей применимости» того или иного языка (C, C++, Pascal и т.п.). Это мнение возникает по нескольким причинам: недостаток информации, привычка, инертность мышления. Мы попытались слегка компенсировать первый фактор. По поводу остальных можем лишь сказать, что настоящий профессионал должен постоянно стремиться повышать свою профессиональную квалификацию. Разумеется, прежде чем приниматься использовать новый язык, нужно внимательно изучить все его особенности, включая наличии эффективной реализации, возможности взаимодействия с существующими модулями и т.п., и только после этого принимать решение. Но, тем не менее, одно другого стоит.

**Список литературы:**

**https://www.referat911.ru/Programmirovanie-i-kompyutery**