

## ЯЗЫКИ ПРОГРАММИРОВАНИЯ

Е.В. Сулимчик<sup>1</sup>, Н.А. Кочетова<sup>2</sup>

<sup>1</sup> – студентка I курса, факультета ФЭУТ, группы ДГХ. Белорусского государственного экономического университета

<sup>2</sup> – научный руководитель, ассистент кафедры информационных технологий, Белорусского государственного экономического университета, Минск, 2200672, Партизанский пр., 26. тел. (8017) 249-19-81

*Аннотация.* В докладе рассматриваются основные понятия языков программирования (алфавит, синтаксис, семантика) и общие характеристики (мощность, уровень и концептуальная целостность). Подробно описана история создания и классификация языков на начальной стадии развития программирования - в 60-е и 70-е годы двадцатого столетия.

*Ключевые слова:* алгоритм, числовой код, компьютер.

### 1. История развития языков программирования.

В середине 50-х, когда вычислительная техника прочно укоренилась в университетах и научно-исследовательских лабораториях США и Европы, наступило время стремительного прогресса в области программирования. Посредниками между программистами и машинами стали языки программирования нового типа. От языка АССЕМБЛЕР они отличались большей гибкостью и возможностью использования конструкций, подобных предложениям. Появление этих языков ознаменовало первый шаг на пути создания программ научно-исследовательских лабораторий и финансовых отделов.

Двумя яркими представителями нового направления были сотрудники математического факультета Дартмутского колледжа Томас Курц и Джон Кемени.

Когда они только начинали своё сотрудничество, Дартмутский колледж даже не имел своего компьютера. Благодаря усилиям двух молодых учёных всего за 10 лет колледж не только стал обладателем прекрасного вычислительного центра, но и заслужил честь называться родиной языка БЕЙСИК

В первое время Кемени и Курц были вынуждены работать со своими математическими программами где придется. Следует отметить, что они не могли самостоятельно работать на компьютере. Виной тому была система под названием «система пакетной обработки» (batch processing).

Помимо этого, самые первые программы для машины IBM-704 приходилось писать на языке АССЕМБЛЕР, который по существу мало чем отличался от программирования в двоичном коде. Если программа начинала вести себя «странно», то единственным способом поиска ошибки было изучение так называемого домна памяти – длинной бумажной ленты, содержащей таинственные числовые коды [1].

### 2. Основные понятия языков программирования

Язык программирования – множество текстов некоторого алфавита, удовлетворяющих правилам синтаксиса и задающих порядок вычислений в соответствии с правилами семантики.

Алфавит языка программирования – набор символов, включающий буквы, цифры и специальные знаки.

Синтаксис языка программирования – совокупность правил записи, которым должна удовлетворять любая программа, включает также правила ввода текстов программ в ЭВМ.

Семантика языка программирования – правила определяющие, какие операции, и в какой последовательности должна выполнить ЭВМ, работая по программе [2].

### 3. Общая характеристика языков программирования.

Язык программирования не просто средство описания алгоритмов. На основе системы понятий языка человек может обдумывать свои задачи, а с помощью нотации может выразить свои соображения

вычисление функций, аргументами которой могут быть другие функции. Лисп - функциональный язык в отличие от большинства других языков, которые являются императивными. Целью создания Лиспа было использование его в системах автоматического доказательства теорем. Поэтому он называется также языком искусственного интеллекта. К этой группе языков относятся также разработанные позже и вобравшие в себя многие идеи Лиспа языки Пролог, Плэнер.

Еще одним специализированным языком, созданным в начале 60-х годов, является Снобол. Основной тип данных в Сноболе - строка символов. Он содержит высокоуровневые средства для обработки строк, в основе реализации которых лежат нормальные алгоритмы А.А. Маркова. Поэтому он относится к марковским языкам. Области применения Снобола являются конструирование компиляторов, символическая математика, обработка текстов, перевод с естественных языков и др.

Программы на языке Форт (FORTh, редуцированное от fourth - четвертый) предельно кратки и занимают совсем немного места в памяти. Программы на Форте работают значительно быстрее, чем программы на Бейсике, но одновременно это затрудняет их чтение и сопровождение. Форт, с точки зрения автора, это язык программирования четвертого поколения. Идея создания данного языка принадлежит Чарльзу Муру, который разработал его в конце 60-х - начале 70-х годов как персональное средство повышения производительности труда. Форт стал более широко применяться в задачах управления после того, как Мур использовал его для реализации программы, предназначенной для управления радиотелескопом Аризонской обсерватории.

И, наконец, язык АПЛ - мощный и высокоуровневый язык для решения научных задач. Основным типом данных является массив; имеется огромное множество разнообразных средств для описания действий с массивами. Этот язык интерактивный: он позволяет программисту вмешиваться в процесс трансляции и исполнения, внося на любой стадии исправления и изменения, в отличие от пакетных языков, которые не предоставляют таких возможностей.

Конец 60-х - начало 70-х - период завершения развития средств описания действий и интенсификация развития средств описания данных. Идеи Алгол - 60 доведены до совершенства в языке Алгол - 68, который оказал значительное влияние на языки следующего поколения [3].

Среди языков, при создании которых ставилась цель улучшить Алгол, следует отметить Паскаль. Данный язык получил свое название в честь французского математика 17в. Блеза Паскаля. Паскаль считается важнейшим инструментом для обучения методам структурного программирования и с 1983 года введен в учебные курсы во всех средних школах США для учащихся, которые специализируются в области информатики. Он был разработан в конце 60-х швейцарским ученым Никлаусом Виртом. Язык требует от программиста определения всех переменных в отдельной секции, расположенной в начале программы. В Паскаль-программах сравнительно немного ошибок и их проще понять и исправить программисту, не являющемуся автором программы. Это делает Паскаль весьма подходящим для создания больших программ. Однако для коротких программ Паскаль может оказаться излишне громоздким. В 1963г. он был объявлен официальным языком программирования для учащихся средних школ, которые намерены специализироваться в области вычислительной техники и программирования в американских университетах.

Независимо от Паскаля разработан язык Симула-67, который относится к языкам моделирования, т.е. используется как средство для имитационного моделирования сложных систем. Симула-67 содержит в качестве подмножества Алгол-60. Но что более важно и что поставило этот язык впереди своего времени, он содержит также новую конструкцию - класс. Под классом понимается специальное средство описания типа данных, которое позволяет специфицировать как структуру типа, так и последовательность действий, которые предназначены для его обработки. Однако этот язык не располагал таким богатым набором типов данных, как Паскаль, что не позволило воспользоваться всеми преимуществами понятия класса.

эффективные программы, генерируемые компилятором) [2].

## 5. Пути развития и классификация языков программирования.

Первый язык программирования высокого уровня ФОРТРАН (FORTRAN, FORMula TRANslator – переводчик формул) был разработан фирмой ИБМ в 1955г. С этого времени развитие языков программирования характеризуется двумя основными тенденциями: совершенствование средств описания алгоритмов и данных. Фортран включал самые элементарные средства, многие из них непосредственно отражали возможности ЭВМ того времени. ФОРТРАН специально предназначен для работы с формулами, используемых в математике и других научно-технических дисциплинах. Однако овладение им представляло трудность даже для инженеров [2].

Однако “дартмутский дуэт” стал искать методы упрощения использования языка программирования. Первым шагом была замена пакетного режима системой разделения времени (time sharing). Следующая идея заключалась в создании языка программирования, который легко бы осваивали студенты. Таким языком стал BASIC (Beginner All-Purpose Symbolic Code). Язык BASIC действительно универсален. Бейсик - это первый компьютерный язык, который изучают школьники и другие начинающие программисты. Он был разработан в середине 60-х. В следующее десятилетие он завоевал всеобщее признание вследствие своей компактности и пригодности для первых персональных компьютеров с их ограниченным объемом памяти. Его стали использовать как встроенный язык микрокомпьютеров. Бейсик сравнительно несложен для изучения и хорошо подходит для разработки коротких и простых программ. В то же время большие программы на этом языке могут оказаться неуклюжими и плохо организованными.

В начале 60-х годов появилось много языков, каждый из которых до сих пор представляет значительное явление в программировании. Поэтому были предприняты различные попытки создать универсальный язык программирования. Ни одна из них не увенчалась полным успехом. Среди десятка наиболее широко

используемых языков каждый ориентирован на решение специфических задач.

Алгол-60 (Algol, от ALGOrithmic Language - алгоритмический язык) впервые предоставил для описания алгоритмов согласованное и единообразное множество конструкций, включающее явные операторы управления для предоставления последовательного выполнения операторов, выбора и итерации, а также достаточно мощный и удобный механизм подпрограмм. Стало возможным разрабатывать программы методом сверху вниз с помощью последовательного уточнения абстрактных действий.

Кобол (Cobol, от Common Business Oriented Language), созданный в 1960г. объединенным комитетом производителей и пользователей компьютеров, был задуман как основной язык для массовой обработки данных в сферах управления и бизнеса. В Коболе все данные описываются в отдельной секции, которые не совпадают с секцией команд. Это позволяет использовать совместно одни и те же описания данных в различных программах. Язык Кобол заслуживает внимания за развитие такой важной части языков программирования как средства описания и обработки файлов. Однако и Фортран, и Алгол - 60, и Кобол не были в полном смысле универсальными. Первые два были предназначены в основном для расчетных задач, а последний - для экономических [1].

Первым действительно универсальным языком программирования был ПЛИ. Объединивший многие возможности и средства языков Фортран, Алгол - 60 и Кобол. Он предоставляет практически все необходимые средства описания алгоритмов, хотя и не столь развитые, как в современных языках, средства описания и обработки таких сложных типов данных, как массивы и структуры, а также простых типов данных.

Параллельно с развитием универсальных языков создавались и специализированные языки, в которых за счет сужения области их использования удавалось построить отдельные высокоуровневые механизмы для описания и данных, и действий. Так в языке Лисп (LISP, от LIST Processing - обработка списков) допускается лишь один тип данных - список и один базовый механизм выполнения - рекурсивные функции. Программа на Лиспе представляет собой

по поводу решения задач. Любой язык, хороший или плохой, ограничивает программиста в выборе способов решения. В плохих языках трудно или даже невозможно выразить некоторые понятия; хороший предоставляет программисту такие возможности, которые облегчают разработку программы. Сравнить языки и способы реализации в них семантических структур позволяют характеристики и свойства языков программирования.

#### **4. Характеристики и свойства языков программирования.**

Основными характеристиками, позволяющими сравнивать языки программирования и выбирать наилучшие для решения той или иной задачи, являются мощность, уровень и концептуальная целостность.

Мощность языка характеризуется количеством и разнообразием задач, алгоритмы решения которых можно записать, используя этот язык. Очевидно, самым мощным является машинный язык. Любую задачу, запрограммированную на каком-либо языке можно запрограммировать и на машинном языке.

Уровень языка характеризуется сложностью решения задач с помощью этого языка. Чем проще записывается решение задач, чем более непосредственно реализуются сложные операции и понятия, чем меньше объем получаемых программ, тем выше уровень языка.

Концептуальная целостность языка характеризуется свойствами совокупности понятий, служащих для описания этого языка, и включает три взаимосвязанных аспекта: экономию, ортогональность и единообразие понятий. Экономия понятий предполагает достижение максимальной мощности языка с помощью минимального числа понятий. Ортогональность понятий означает, что между понятиями не должно быть взаимного влияния. Так, если какое-либо понятие используется в различных контекстах, то правила использования должны быть одни и те же. Единообразие понятий требует согласованного, единого подхода к описанию и использованию всех понятий.

Обычно чем меньше мощность языка (то есть, чем уже область его применения), тем выше его уровень. По этой причине наряду с

универсальными языками разрабатываются и специализированные в некоторой конкретной области. Конечно, чем мощнее язык, тем труднее обеспечить концептуальную целостность; в то же время высокий уровень языка непосредственно связан с концептуальной целостностью.

Перечисленные характеристики языков программирования определяют наличие или отсутствие надежности, удобочитаемости, полноты, гибкости, простоты. Эти свойства позволяют более детально сравнивать языки.

Надежность языка обеспечивает минимум ошибок при написании программ. Для этого язык должен быть таким, чтобы было трудно делать ошибки, не обнаруживаемые при компиляции. Язык должен защищать программиста от него самого, сделав трудным или даже невозможным появление некоторых ошибок.

Удобочитаемость языка - это свойство, обеспечивающее легкость восприятия программ человеком. Высокая степень удобочитаемости оказывается полезной с различных точек зрения. Во-первых, уменьшается сложность документирования. Во-вторых, удобочитаемость позволяет легче понимать программу. В-третьих, она позволяет легче сопровождать программу.

Полнота языка обеспечивает описание на языке решения задач определенной предметной области, а также с помощью средств языка процесс разработки программ.

Гибкость языка обеспечивает легкость выражения на языке необходимых для решения задач действий, предоставляет программисту достаточно возможностей для выражения всех операций в программе.

Простота языка обеспечивает легкость понимания семантики языковых конструкций и запоминания их синтаксиса.

При сравнении и выборе языков программирования следует учитывать еще два свойства языка, хотя они и не влияют непосредственно на процесс разработки программ. Это - мобильность и эффективность.

Мобильность языка обеспечивает независимость его от аппаратных средств, позволяет переносить программное обеспечение с машины на машину с относительной легкостью.

Эффективность языка обеспечивает эффективную реализацию языка (включая эффективную реализацию компилятора и

Объединение этих двух важных идей было осуществлено в языке Параллельный Паскаль. Развитие идеи заложенной в понятие класса, происходило и в рамках других языков. Например, язык Смолток основан на понятии объекта, содержащего понятийную и процедурную части. Объекты взаимодействуют между собой, посылая сообщения, представляющие собой требования выполнить те или иные процедуры над объектами-получателями сообщения. Этот язык принадлежит к классу объектно-ориентированных языков программирования. В языке Модуль введена конструкция "модуль", которая управляет доступом к объектам данных, ограничивая область вокруг объекта и операций над ним. Это существенное отличие от понятия класса стало основной идеей на последующих этапах развития языков программирования [2].

Параллельно с этим в различных языках программирования появились и другие важные идеи. Язык Евклид был разработан с целью обеспечить существенную поддержку верификации программ с помощью внесённых в текст программы утверждений о её выполнении. В языке LIS разработаны механизмы для отдельной компиляции модулей. Язык MESA позволил накопить опыт использования подпрограмм обработки исключений. Во многих языках вводилась конструкция параллельного программирования [2].

## 6. Заключение

Если 10 программистов попросить назвать самый лучший язык программирования, то, скорее всего, мы получим десяток разных ответов, ибо такого языка просто не существует. Теоретически для решения большинства задач программирования можно использовать любой язык. Однако на практике оказывается, что написать программу для конкретной задачи гораздо легче на одних языках, чем на других.

Каждый язык программирования имеет свой собственный набор ключевых слов, состоящих из слов, чисел букв или других символов. Ключевые слова соответствуют конкретным операциям или последовательностям операций, которые должен произвести компьютер. При выборе наиболее подходящего языка для решения той или иной задачи следует учитывать ряд

факторов, в том числе степень квалификации программиста и размер будущей программы.

Развитие средств описания данных и действий продолжается и в настоящее время. Однако с начала 80-х годов вышла на 1-ый план новая тенденция развития языков программирования – развитие средств, обеспечивающих автоматическое доказательство правильности программ. Такая цель требует пересмотра семантики и её выражение на языке многих широко используемых в настоящее время языковых возможностей. Кроме этого, необходимо наличие соответствующего программного обеспечения. Таким образом, можно уже говорить не о развитии языков, а о развитии языковых систем, включающих язык, транслятор, методы и процедуры автоматического доказательства правильности программ.

Впрочем, современные языки программирования – это тема для отдельного большого доклада.

## Литература

1. О Вудз, Д Фурман, С Е Роу и др. "Язык компьютера" М.: Мир 1989 г.
2. Калинин, Мацкевич "Универсальные языки программирования" //, - М.: "Радио и связь" 1994 г.