

Язык формирует наш способ мышления и определяет то, о чем мы можем мыслить.

Б.Л. Ворф

Программное обеспечение

Metod-kopilka.ru
Метод-копилка.ру

Кузнецов Максим Евгеньевич
Учитель информатики

МОУ СОШ 3, Мурманская обл., ЗАТО г.Североморск, пгт.Росляково

Программное обеспечение (software) -

– это набор команд, управляющих работой компьютера.

Функции программного обеспечения:

- управлять компьютерными ресурсами организации;
- обеспечивать пользователя всеми инструментами, необходимыми для извлечения пользы из этих ресурсов;
- выполнять роль посредника между организациями и хранимой информацией.

Программа (program) -

– это набор команд для компьютера.

Процесс создания или написания программ называется *программированием*.

Люди, которые специализируются на этом виде деятельности – *программистами*.

Синонимом слову "программа" является термин "*приложение*" (*application*).

запустить программу или *запустить на выполнение*

ОСНОВНЫЕ ТИПЫ ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ



Системное программное обеспечение (system software)

– это набор программ, которые управляют компонентами компьютера (процессор, коммуникационные и периферийные устройства).

Системные программисты

Прикладное программное обеспечение (application software)

– программы, написанные для пользователей или самими пользователями (для задания компьютеру конкретной работы).

Прикладные программисты

Основные типы программного обеспечения



Системное программное обеспечение

координирует работу различных компонентов компьютера и играет роль посредника между прикладными программами и аппаратным обеспечением

Системное программное обеспечение, которое управляет работой компьютера, называется операционной системой (ОС)

ФУНКЦИИ ОПЕРАЦИОННОЙ СИСТЕМЫ

- *распределяет (allocates)* и *назначает (assigns)* использование ресурсов компьютера
- *планирует (schedules)* использование ресурсов компьютера и время исполнения задач
- осуществляет текущий *контроль (monitoring)* работы компьютера

Структура операционной системы:

- **Ядро** – переводит команды с языка программ на язык «машинных кодов», понятный компьютеру.
- **Драйверы** – программы, управляющие устройствами.
- **Интерфейс** – оболочка, с помощью которой пользователь общается с компьютером.

Особенности Windows

- Многозадачность.
- Единый программный интерфейс.
- Единый интерфейс пользователя.
- Графический интерфейс пользователя.
- Единый аппаратно-программный интерфейс.

Графические операционные системы:

Выпущенная в сентябре 1995 г. система **Windows 95** стала первой графической операционной системой для компьютеров IBM PC.

Все следующие версии операционных систем Windows (98, NT, ME, 2000, XP) являются графическими.

Операционные системы компании Microsoft:

- DOS (Disk Operating System). Интерфейс – командная строка. Все команды приходилось набирать вручную, в командной строке ОС.
- Windows 3.1 и 3.11. – первый графический интерфейс. Хотя многие не считали эту систему операционной, а лишь системой, расширяющей возможности DOS.
У всех дальнейших операционных систем интерфейс графический.
- Windows-95.
- Windows-98 / NT / ME / XP / Vista.
- Windows 2000 / 2003 Server

Альтернативные ОС:

Словом UNIX обозначается не одна операционная система, а целое семейство ОС.

Важное – совместимость, переносимость, настраиваемость и, самое главное, стабильность.

От мощного древа UNIX отпочковался и целый ряд «свободных» операционных систем:

Linux, FreeBSD, NetBSD и OpenBSD.

Также существуют и другие альтернативные операционные системы.

Прикладное программное обеспечение:

Основной функцией прикладного программного обеспечения является выполнение задач, поставленных конечными пользователями.

Для разработки прикладного ПО может быть использовано множество языков программирования.

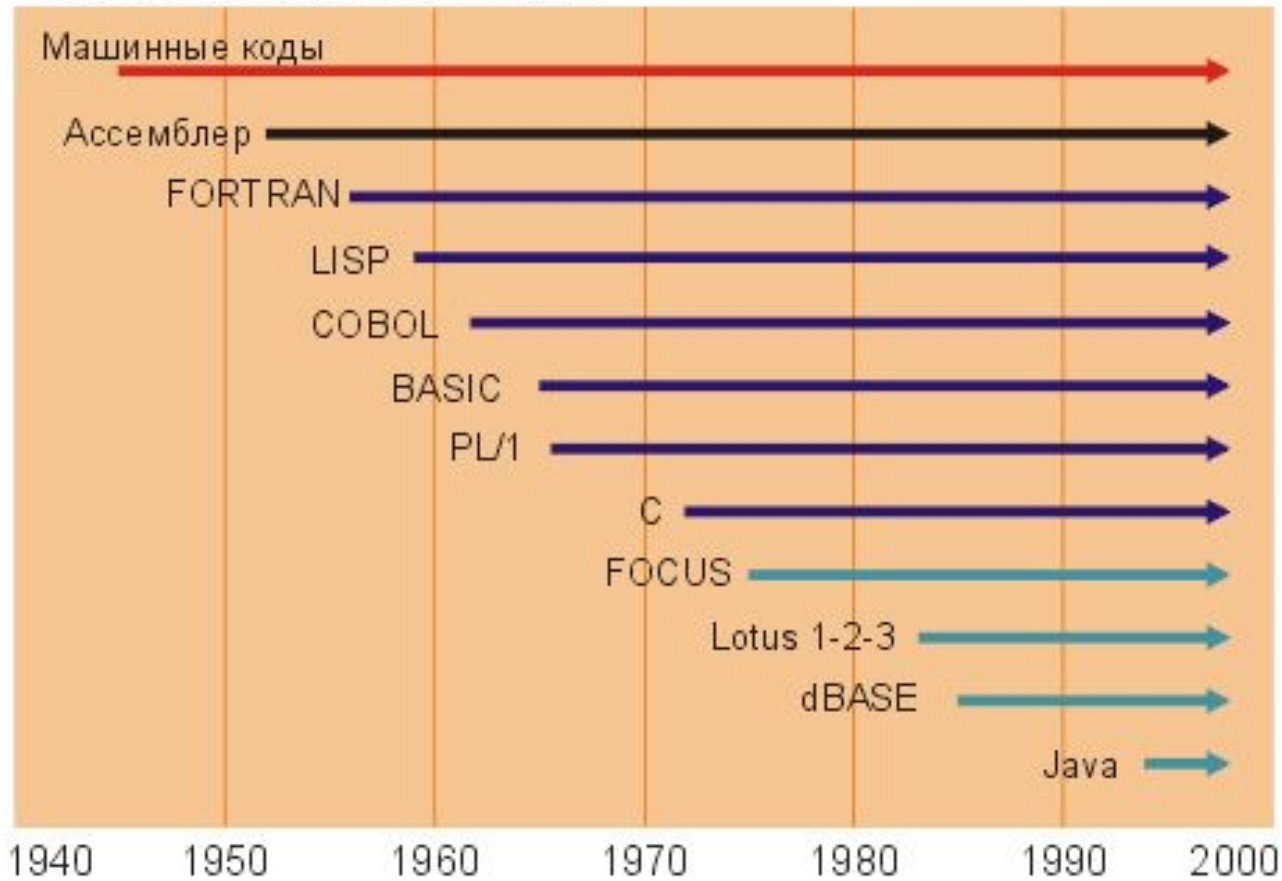
Каждый из этих языков имеет свои сильные и слабые стороны.

ПОКОЛЕНИЯ ЯЗЫКОВ ПРОГРАММИРОВАНИЯ

А. Поколения



В. Языки программирования



ПОПУЛЯРНЫЕ ЯЗЫКИ ПРОГРАММИРОВАНИЯ:

```
AR 5,3
```

Ассемблер

```
READ (5, 100) ID, QUANT, PRICE  
TOTAL = QUANT * PRICE
```

FORTRAN

```
MULTIPLY QUANT-SOLID BY UNIT-PRICE GIVING SALES-TOTAL.
```

COBOL

Ассемблер

Многие программисты продолжают использовать Ассемблер, так как этот язык программирования дает им полный контроль над аппаратным обеспечением компьютера и генерирует очень эффективный исполняемый код.

Как и машинные коды, **Ассемблер** разрабатывается под определенные типы компьютеров и микропроцессоров. Несмотря на описанные преимущества, программирование на этом языке требует больших затрат времени, он труден для изучения и понимания; программы, написанные на Ассемблере трудно отлаживать.

Ассемблер сегодня используется в основном для написания системного ПО.

FORTRAN

FORTRAN – FORMula TRANslator был спроектирован в 1956 году в основном для инженеров, математиков и ученых, которые имеют в основном дело с формулами и проблемами, ориентированными на вычисления.

На FORTRANе можно довольно просто описывать сложные вычисления, манипулировать массивами и распечатывать выходные множества чисел. Хотя на этом языке было написано немало бизнес-приложений, он не очень подходит для частых операций ввода-вывода и работы со списками.

FORTRAN сравнительно легко осваивается, но его синтаксис очень требователен к точности ввода операторов, что вызывает частые ошибки и делает сложной отладку программ.

COBOL

COBOL – Common Business-Oriented Language
– общий язык для приложений, ориентированных на бизнес и коммерцию.

COBOL был разработан в начале 60-х годов, для того чтобы было легче писать программы для бизнеса, которые могли использоваться в таких видах деятельности как обработка заказов, ведение бухгалтерии, планирование производства и т.д.

Данные, которыми оперирует COBOL – это, прежде всего, записи, файлы, таблицы и списки. Программы, написанные на языке COBOL, относительно понятны даже неспециалистам.

BASIC

BASIC (Beginners All-purpose Symbolic Instruction Code) был создан в 1964 году для обучения студентов колледжей по использованию компьютеров.

Это простой язык, который недавно подошел к такому уровню развития, при котором его можно использовать даже для построения больших систем высокой производительности. Слабая сторона BASICa – то что он выполняет все задачи одинаково, без оптимизации кода. Хотя BASIC прост и привлекателен, ему очень не хватает формальной структуры, которая делает язык, подобный C и C++, более подходящим для больших проектов.

Pascal

Названный в честь Блеза Паскаля, французского математика и философа, язык программирования **Pascal** был разработан швейцарским профессором компьютерных наук Никлаусом Виртом в конце 60-х.

Программирование на Pascal стало популярным благодаря высокоэффективному выходному коду, т.е. программы, написанные на нем занимают немного места в памяти. Благодаря наличию возможностей обработки сложных массивов данных и набору простых, но мощных команд, Pascal применяют в основном для обучения будущих профессионалов навыкам программирования.

C и C++

Язык C был разработан в начале 70-х в AT&T Bell Labs. На сегодня эти языки являются фактически единственным выбором для построения операционных систем и сложных приложений, таких как электронные таблицы, компиляторы, сетевые утилиты, коммерческие приложения и проч. Операционные системы UNIX, Linux и Windows большей частью написаны на C, даже компилятор C написан на C. Эти мощные и сложные языки генерируют быстрый и эффективный код. Программист, владеющий C, имеет полный контроль над средой разработки и может заставить компьютер делать практически все, что ему нужно.

Как и ожидалось, эти языки очень сложны, и для того, чтобы овладеть навыками программирования на C, одного желания мало – нужен талант, как и в случае работы с Ассемблером и машинными кодами.

PL/1

PL/1 (Programming Language 1) был разработан компанией IBM в 1964 году. Это очень мощный, но и очень сложный многоцелевой язык программирования. Он пригоден для проектирования как научных, так и бизнес-приложений и даже операционных систем (огромная операционная система MULTICS была написана на PL/1). IBM, бывшая в то время безоговорочным лидером в мире компьютеров, провозгласила, что PL/1 стратегически необходим всем компаниям.

Однако, попытка создать универсальный язык провалилась – программисты не желали переучиваться с COBOLa и FORTRANa на абсолютно новый язык, а коммерческие компании не хотели тратить миллионы долларов на перепроектирование своих приложений, написанных на COBOLe и FORTRANE.

Ada

Другую попытку создать компьютерный эсперанто предприняло в 1980 году Министерство обороны США. Отдав привилегию разрабатывать язык, названный **ADA**, международному сообществу, правительство начало обсуждать заключение контрактов на его использование. Язык предполагалось использовать прежде всего для разработки программного обеспечения систем вооружений. Предполагалось, что он будет одинаково эффективен для различных аппаратных платформ. Специалисты, работавшие в правительстве США, надеялись, что именно ADA станет удачной заменой языкам программирования, применявшихся в бизнесе. Но этого не случилось по тем же причинам, что и с PL/1 – природа языка ADA весьма сложная. Несмотря на привлекательность в некоторых нюансах, в решении других проблем ADA сильно ограничен.

LISP u Prolog

LISP был создан в конце 50-х годов математиком Джоном Маккарти (John McCarthy), который ориентировал этот язык на формирование из вводимых символов, которые в свою очередь являлись операторами, переменными и значениями данных, значащих списков. LISP больше подходит для манипулирования такими списками, чем для решения громоздких числовых задач. **Prolog**, разработанный в 1970 году, также подходил для этих целей. Отличается же Prolog от LISP в основном тем, что с первым можно работать на обычных компьютерах, а второй работает лучше на машинах, специально созданных для работы с LISP.

СРЕДСТВА РАЗРАБОТКИ ЧЕТВЕРТОГО ПОКОЛЕНИЯ

Ориентированы на пользователя

Ориентированы на профессионалов



Приложения для ПК

- Excel
- Access
- Word
- Power Point

Языки запросов/генераторы отчетов

- Easytrieve
- Intellect
- Query-by-Example
- Inquire

Графические языки

- Systat
- SAS Graph
- Harvard Graphics

Генераторы приложений

- FOCUS
- DMS
- SAS
- Mapper
- Ideal
- Natural
- CSP

Пакеты прикладных программ

- MSA Payroll
- Maxicalc
- AVP
- Sales/Use Tax
- AMAPS

Языки очень высокого уровня

- APL
- Nomad

ОБЪЕКТНО-ОРИЕНТИРОВАННОЕ ПРОГРАММИРОВАНИЕ

В объектно-ориентированном программировании данные и предназначенные для их обработки процедуры соединяются в один *объект (object)*.

Объект, таким образом, представляет собой комбинацию данных и программного кода. Вместо передачи данных процедурам, программа посылает объекту сообщение (*message*) выполнить процедуру, уже встроенную в него. (В объектно-ориентированных языках программирования процедуры называются *методами [methods]*).

То же сообщение может быть послано множеству других объектов, но каждый объект будет обрабатывать сообщение отлично от другого.

Технология Java

Java – это объектно-ориентированный язык программирования, комбинирующий данные и определенные функции обработки этих данных. Java сконструирован таким образом, что программист может создавать **апплеты** (*applets*) – крохотные Java-программки, каждая из которых выполняет одну небольшую функцию.