

Новосибирский технологический институт  
Московского государственного университета  
дизайна и технологии (филиал)



Кафедра АиВТ

# ПРОГРАММНОЕ ОБЕСПЕЧЕНИЕ КОМПЬЮТЕРА



© Составитель: проф. Степанов Б.Ф.

Новосибирск 2009

Softinok.net

# Системное программное обеспечение



# История операционных систем

В истории операционных систем (software) важное место отводится ОС Unix.

Unix был разработан в середине 70-х годов прошлого века. В настоящее время Unix используется в корпоративной среде, а также нередко применяется в системах клиент-сервер сетей intranet. Однако Unix уступал Windows по доступности и богатству созданных на его платформе прикладных программ для персональных компьютеров.

Для устранения этих недостатков был разработан Linux — широкодоступная ОС, аналогичная Unix. Над проектом с середины 80-х работал Ричард Столман (Richard Stallman). Ядро Linux разработал финн Линус Торвальдс (Linus Torvalds). Дата его создания 1992 г. считается датой рождения Linux.

1981 г. — создана MS-DOS.

# Операционная система

- **Операционная система (ОС)** - комплекс программ, обеспечивающий управление компьютером, как единым целым, его взаимодействие с окружающей средой (человеком, прикладными программами, другими системами).
- ОС является главной частью системного программного обеспечения.
- ОС управляется командами.

# Операционная система

**Классические (несетевые) ОС должны выполнять следующие функции:**

- Планирование заданий и использования процессора, динамическая компоновка выполняемых программ,
- Обеспечение программ средствами коммуникации и синхронизации (обработка прерываний и обеспечение многозадачной работы)..
- Управление памятью.
- Управление файловой системой.
- Управление вводом-выводом.
- Обеспечение безопасности

# Операционная система

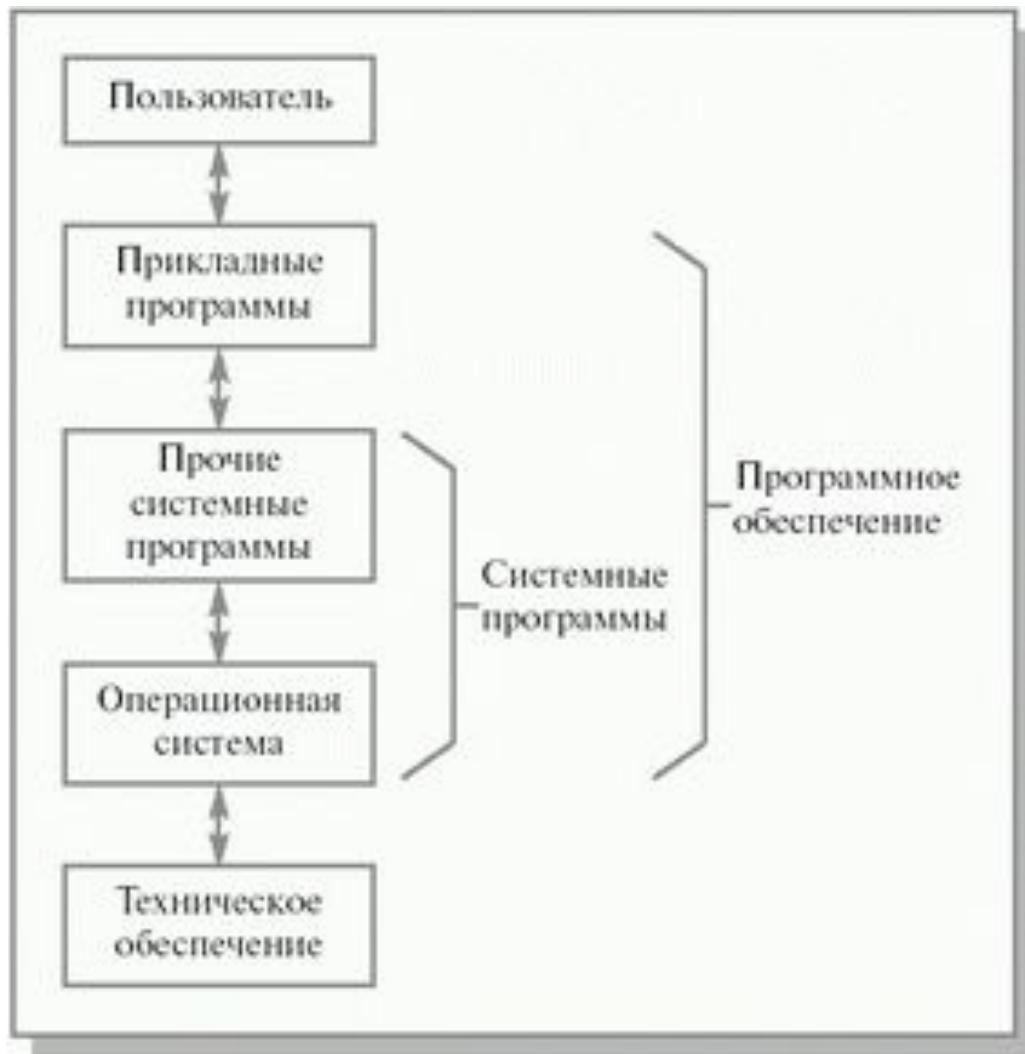
Операционные системы привязывают к процессорам, на основе которых разрабатываются компьютеры.

Для IBM-совместимых компьютеров различают операционные системы:

- однозадачные,
- многозадачные,
- сетевые
- системы реального времени.

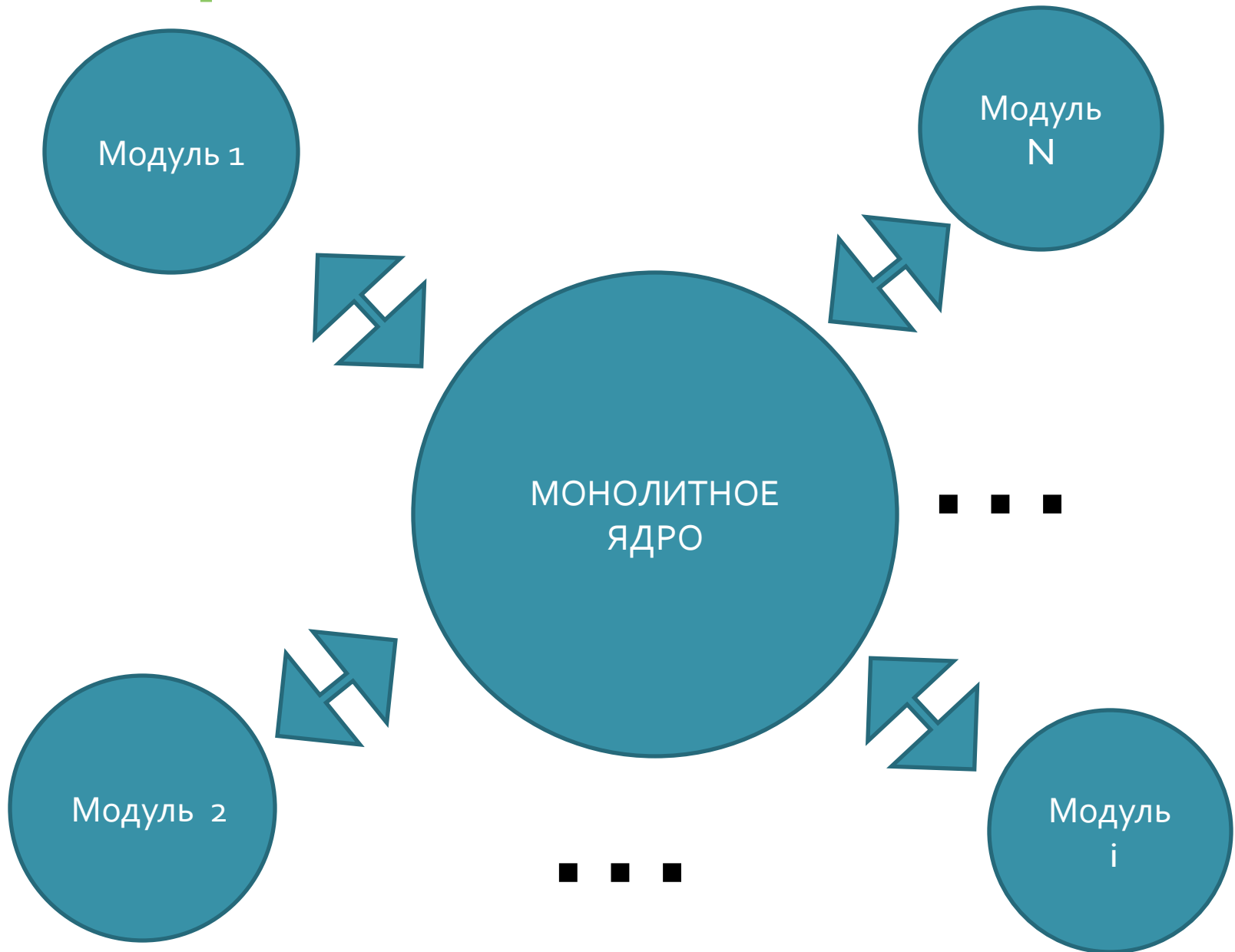
Иногда операционные системы делят на 32 и 64-разрядные по размеру одновременно обрабатываемого слова в микропроцессоре.

# Операционная система



ВЗАИМОДЕЙСТВИЕ  
ПРОГРАММНОГО И  
ТЕХНИЧЕСКОГО  
ОБЕСПЕЧЕНИЯ

# Операционная система





# Операционная система

Слоеная система Technische Hogeschool Eindhoven  
(THE)

5	Интерфейс пользователя
4	Управление вводом-выводом
3	Драйвер устройства связи оператора и консоли
2	Управление памятью
1	Планирование задач и процессов
0	Hardware

Дейкстрой (Dijkstra), 1968 г.

# Операционная система

ОС- виртуальная машина

Программа пользователя	Программа пользователя	Программа пользователя
MS-DOS	Linux	Windows-NT
Виртуальное hardware	Виртуальное hardware	Виртуальное hardware
Реальная операционная система		
Реальное hardware		

# Операционная система

## МНОГОЯДЕРНАЯ СТРУКТУРА ОС



# Программирование

## *История программирования*

(история лингвистического и **программного обеспечения**) — это прежде всего история алгоритмических языков, операционных систем и технологий разработки программного обеспечения.

**Программное обеспечение (ПО)** - компьютерные программы, обеспечивающие функционирование вычислительных систем или решение задач определенного приложения.

# Программирование



**Ада Лавлейс**  
(1815-1852)

написала комментарии к статье о машине Чарльза. Бэббеджа, которые можно считать первым трудом по программированию. Описала ряд примеров применения машины, разработала программу вычисления на ней чисел Бернулли.

# Программирование

## ***История алгоритмических языков***

- Одна из первых попыток создания алгоритмического языка программирования, предпринятой К.Цузе в 1945 г. для электрорелейной ЦВМ "Z-4". Это язык PLANKALKHL, который мог применяться для формализации разнообразных сложных вычислительных процедур.
- Первые алгоритмические языки эпохи ЭВМ относятся к середине 50-х годов. Джону Бэкусу (J.Backus) принадлежит фундаментальный вклад в создание в 1954 г. языка Фортран (FORTRAN — FORmular TRANslation) и в разработку компилятора для IBM 705 (1958 г.). Коллектив под руководством Бэкуса разработал не только спецификацию языка, но и создал компилятор для него. Кроме того, Фортран был первым языком, для которого были созданы компиляторы для многих ЭВМ. Это обеспечило достаточно быструю и широкую распространенность языка.
- Язык Алгол с первоначальным названием IAL и с более поздним ALGOL-58 был разработан в 1958 г. международным комитетом под руководством Питера Наура (P.Naur).

# Программирование



Джон Бэкус и Питер Наур

# Программирование

- Язык Кобол (COBOL – Common Business Oriented Language), представленный в 1960 г., был предназначен для расчетов в сфере бизнеса.
- Язык ЛИСП для исследований в области искусственного интеллекта (LISP — сокращение от LISt Processing — обработка списков), предназначенный для работы со списками и лямбда – выражениями, создан Маккарти в МТИ в 1960 г.
- Разработка языка PL/1 относится к 1964 г. Его разработчики из английского отделения IBM стремились в этом языке совместить возможности Алгола и Кобола.
- Язык моделирования Simula-67 разработан в Норвегии Найгардом (Nygard) и Далом (Dahl). В нем уже используются многие черты объектно-ориентированного программирования.



# Программирование



Заметным вкладом в создание технологий разработки программного обеспечения стала концепция структурного программирования, основанная на положениях блочно-иерархического подхода к программированию. Ее автором является голландский ученый Эдстер Дейкстра (1930-2002), внесший заметный вклад в разработку информационных технологий. Ему принадлежат также определения стека, семафора, разработка ряда алгоритмов (например, поиск кратчайшего пути) и т.п.

# Программирование



Язык программирования Паскаль (Pascal) создан швейцарцем Н. Виртом (Niklaus Wirth) на кафедре информатики Стэнфордского университета на базе структурного программирования. Первая публикация описания языка относится к 1970 г. Особенность языка — его четкая структурированность, ясная логика определений, лаконичность описаний. Популярность язык приобрел после создания Андерсом Хейльсбергом (Anders Hejlsberg) в компании Borland версии Turbo Паскаль, в которой компилятор был объединен с редактором текста. Хейльсберг руководил разработкой среды Delphi, в которой Pascal стал объектно-ориентированным языком Object Pascal.

# Программирование



Язык Си (C) создан в 1972 г. Д. Ритчем (D.Ritchie) и К.Томпсоном (K.Thompson) из Bell Labs/Lucent Technologies при поддержке Б. Кернигана. Ими же в 1973 г. на языке Си представлена операционная система UNIX. В рекламных целях UNIX в университетах распространялся почти бесплатно, вместе с UNIX распространялся и Си. Благодаря эффективности исполнения программ, написанных на этом языке, он получил широкое распространение. До сих пор Си – один из лучших языков для системного программирования. Он стандартизован в 1989 г., стандартная версия языка обозначается ANSI C.

# Программирование

- Первый полностью объектно-ориентированный язык программирования SmallTalk создан в 1972 г. Алана Кей, компания Херох.
- В 1973 г. в Марсельском университете разработан язык логического программирования Пролог, в котором описываются факты и отношения между ними.
- Универсальный язык программирования Ада был разработан по заказу министерства обороны США в 1979 г. Список требований к языку прошел через несколько этапов утверждения. Руководителем группы разработчиков француз Жан Ихбиа.
- Вернувшись из Стенфорда в Швейцарию, Н.Вирт развивает методологию программирования. В 1980 г. в языке Модула-2 он реализует концепцию модульного программирования, затем в Оберон-2 – объектно-ориентированное и в Component Pascal – компонентно-ориентированное программирование.
- Создание объектно-ориентированного языка C++ относят к 1982 г. Его автором является датчанин Бьерн Страуструп (Bjarne Stroustrup), сотрудник лаборатории AT&T Bell Labs, в которой уже были разработаны операционная система UNIX и язык программирования Си, взятый за основу (вместе с идеями объектной ориентированности из языка Симулаб7) и для Си++.

# Программирование



А.Хейльсберг и Б.Страуструп

Язык программирования Java разработан в 1995 г. Джеймсом Гослингом из компании Sun Microsystems. В последние годы Андерс Хейльсберг, перешедший из Borland в Microsoft, разработал язык C# (Си-шарп) и занимается развитием среды Microsoft.Net Framework.

# Трансляция и интерпретация

В автоматизированных системах на разных этапах обработки информации, как правило, используется несколько разных **языков**. При подготовке данных для ввода в ЭВМ применяют входные языки, программы в исходном состоянии представлены на алгоритмических языках, исполнение программ происходит на машинном языке. Возникает задача перевода информации с одного языка на другой.

Язык, с которого осуществляется перевод, называют **исходным**, а язык, на который происходит перевод, — **объектным**.

# Трансляция и интерпретация

Для перевода программы с входного языка на объектный применяют специальные программы, называемые *языковыми процессорами*.

Перевод может выполняться методами интерпретации или трансляции.

# Трансляция и интерпретация

Перевод, заканчивающийся получением объектной программы, называют *трансляцией*, а программу, используемую для трансляции, — *транслятором*. Если объектный язык — машинный или близкий к машинному, то трансляцию и транслятор называют *компиляцией* и *компилятором* соответственно.

При *интерпретации*, выполняемой с помощью *интерпретаторов*, предложения исходной программы поочередно преобразуются в машинный код и тут же исполняются, т.е. объектная программа, как таковая, не формируется и не требуется отводить для нее место в памяти ЭВМ. Для интерпретации характерны меньшие затраты памяти, чем для трансляции. Однако в циклических вычислительных процессах интерпретация каждого предложения исходной программы будет повторяться многократно, поэтому, интерпретация характеризуется повышенными затратами времени.



# Трансляция и интерпретация

**Различают одно- и многопроходные трансляторы.**

В **однопроходных** трансляторах трансляция происходит в несколько этапов, называемых фазами.

На фазе *лексического анализа* исходное описание разделяется на структурные единицы — **лексемы** (идентификаторы, числа, метки и т.п.).

На этапе *синтаксического анализа* проверяется соблюдение **синтаксиса** исходного языка и при наличии ошибок выдаются соответствующие диагностические сообщения.

На этапе *генерации кода* формируется объектная программа.

# Трансляция и интерпретация

**Семантика.** — система правил определения поведения отдельных языковых конструкций — система правил определения поведения отдельных языковых конструкций. Семантика определяет смысловое значение предложений алгоритмического языка.

# ЯЗЫК ПРОГРАММИРОВАНИЯ

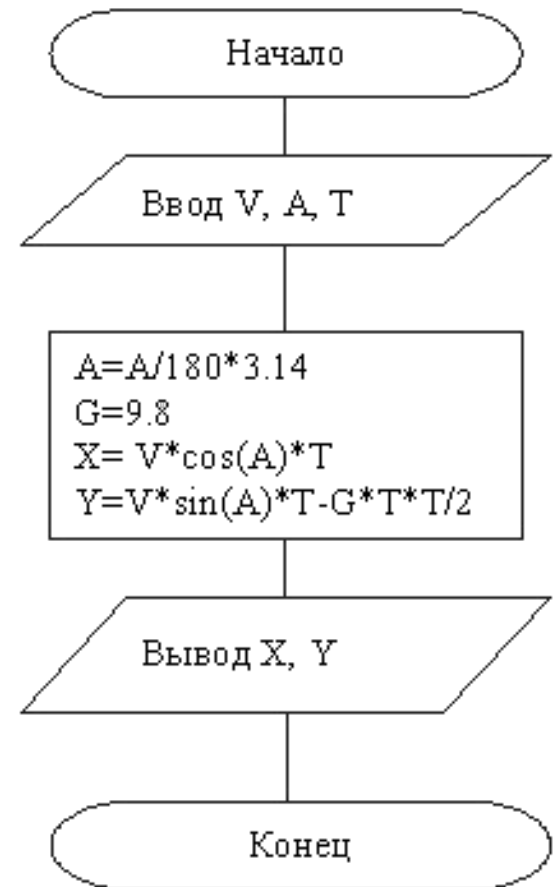
- **Язык программирования** — формальная знаковая система, предназначенная для записи программ.
- **Программа** - некоторый алгоритм - некоторый алгоритм в форме, понятной для исполнителя (например, компьютера).

# ЯЗЫК

# ПРОГРАММИРОВАНИЯ

- **Алгоритм** — строго детерминированная последовательность действий, описывающая процесс преобразования объекта из начального состояния в конечное, записанная с помощью понятных исполнителю команд.

(Николай Дмитриевич Угринович, учебник «Информатика и информ. технологии»)



# ЯЗЫК

# ПРОГРАММИРОВАНИЯ

- **Язык программирования** определяет набор лексических определяет набор лексических, синтаксических определяет набор лексических, синтаксических и семантических определяет набор лексических, синтаксических и семантических правил, используемых при составлении компьютерной программы определяет набор лексических, синтаксических и семантических правил, используемых при составлении компьютерной программы. Он позволяет программисту определяет набор лексических, синтаксических и семантических правил, используемых при составлении компьютерной программы. Он позволяет программисту точно

# ЯЗЫК

## ПРОГРАММИРОВАНИЯ **ЛЕКСИКА**

- *Лéксика* (от греч. (от греч. λεξικός — «относящийся к слову», от греч. (от греч. λεξικός — «относящийся к слову», от греч. λέξις — «слово», «оборот речи») — раздел науки о языке (от греч. λεξικός — «относящийся к слову», от греч. λέξις — «слово», «оборот речи») — раздел науки о языке, изучающий значения СЛОВ.

# ЯЗЫК ПРОГРАММИРОВАНИЯ

- **Синтаксис** - раздел описания формального математического языка или языка программирования, исследующий вид, форму и структуру конструкций (без учета их значения или практической применимости).

# ЯЗЫК ПРОГРАММИРОВАНИЯ

## СЕМАНТИКА

**Семантика.** — система правил определения поведения отдельных языковых конструкций.

Семантика определяет смысловое значение предложений алгоритмического языка.

Существует несколько подходов к определению семантики языков программирования. Наиболее широко распространены разновидности следующих подходов к семантике языков программирования:

- операционный (математический),
- деривационный (аксиоматический),
- денотационный



# ЯЗЫК

# ПРОГРАММИРОВАНИЯ

## Семантика.

- **Операционная** семантика интерпретирует исполнение конструкций языка программирования с помощью некоторой воображаемой (абстрактной) ЭВМ.
- **Деривационная** семантика описывает последствия выполнения конструкций языка с помощью языка логики и задания пред- и постусловий.
- **Денотационная** семантика оперирует понятиями, типичными для математики — множества, соответствия, а также суждения, утверждения и др.

# ЯЗЫК

# ПРОГРАММИРОВАНИЯ

## Семантика.

- **Операционная** семантика интерпретирует исполнение конструкций языка программирования с помощью некоторой воображаемой (абстрактной) ЭВМ.
- **Деривационная** семантика описывает последствия выполнения конструкций языка с помощью языка логики и задания пред- и постусловий.
- **Денотационная** семантика оперирует понятиями, типичными для математики — множества, соответствия, а также суждения, утверждения и др.

# ЯЗЫК ПРОГРАММИРОВАНИЯ

**Компьютерная программа —**

последовательность инструкций, предназначенная для исполнения устройством управления вычислительной машины. В зависимости от контекста, рассматриваемый термин может относиться также и к исходным текстам программы.

# ЯЗЫК ПРОГРАММИРОВАНИЯ

## КЛАССИФИКАЦИЯ



# ЯЗЫК ПРОГРАММИРОВАНИЯ

## КЛАССИФИКАЦИЯ

**Машинно-зависимые языки** – (Ассемблер, макроассемблер-языки низкого уровня) применяются для написания программ, явно использующих специфику конкретной аппаратуры.

**Машинно-ориентированные языки** – (СИ ). объединяют достоинства низкоуровневых возможностей ассемблеров и мощных выразительных средств языков программирования высокого уровня.

**Универсальные языки** (языки высокого уровня) - (Фортран, PL/I, Паскаль Turbo Pascal и др.) наиболее эффективны при численных расчетах, просты по структуре и удобны при выполнении программ, получили большое распространение при разработке прикладных программ для решения инженерных, экономических и научных задач. Самыми распространенными на ЭВМ являются различные версии языка Бейсик, простота которого делает его превосходным средством для начинающих программистов.

# ЯЗЫК ПРОГРАММИРОВАНИЯ

## КЛАССИФИКАЦИЯ

**Проблемно-ориентированные языки** – (Лого, CPSS, Форт и Смолток.) языки программирования, управляющие структуры и структуры данных которых отражают особенности класса решаемых задач. Например, **Лого** — диалоговый процедурный язык, реализованный на принципе интерпретации и работающий со списками, текстами, графическими средствами и т. д. Язык очень перспективен для обучения, создания электронных игрушек и т. д.

Развитием проблемно-ориентированных языков является **объектно-ориентированный подход** (языки Смолток, Форт, Модула и Ада). Отличительные особенности таких языков: модульность построения процедур, абстракция данных, динамическая связка программ и использование механизма наследования иерархического типа. Например, **Смолток** предназначен для решения нечисловых задач при построении систем искусственного интеллекта. В языке Форт применены структурное программирование и очень компактный машинный код.

# ЯЗЫК ПРОГРАММИРОВАНИЯ

## КЛАССИФИКАЦИЯ

**Функциональные языки (Языки функционального программирования)** (Лисп, AutoLISP, Пролог, СНОБОЛ)- языки, основным конструктивным элементом которых является математическое понятие функции. Тексты программ на функциональных языках программирования *описывают* «как решить задачу», но не *предписывают* последовательность действий для решения. Например, [AutoLISP](#) последовательность действий для решения. Например, AutoLISP широко используется в системе автоматизированного проектирования [AutoCAD](#).

В качестве основных свойств функциональных языков программирования обычно рассматриваются следующие:

- краткость и простота;
- строгая типизация;
- модульность;
- функции — объекты вычисления.

Используются также для разработки систем искусственного интеллекта (языки Лисп, Пролог и СНОБОЛ – эти языки относятся к так называемым языкам представления знаний).

# ЯЗЫК

# ПРОГРАММИРОВАНИЯ

## Используемые символы

Современные языки программирования рассчитаны на использование ASCII.

**ASCII** (англ. *American Standard Code for Information Interchange* — Американский —  
Американский стандартный —  
Американский стандартный код —  
Американский стандартный код для  
обмена информацией).

ASCII представляет собой 8-битную кодировку для представления десятичных цифр, латинского и национального алфавитов, знаков препинания и управляющих символов. *Управляющие* символы ASCII используются ограниченно



# ЯЗЫК ПРОГРАММИРОВАНИЯ

## Кодировка

	.0	.1	.2	.3	.4	.5	.6	.7	.8	.9	.A	.B	.C	.D	.E	.F
0.	NUL	SOH	STX	ETX	EOT	ENQ	ACK	BEL	BS	TAB	LF	VT	FF	CR	SO	SI
1.	DLE	DC1	DC2	DC3	DC4	NAK	SYN	ETB	CAN	EM	SUB	ESC	FS	GS	RS	US
2.		!	"	#	\$	%	&	'	(	)	*	+	,	-	.	/
3.	0	1	2	3	4	5	6	7	8	9	:	;	<	=	>	?
4.	@	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O
5.	P	Q	R	S	T	U	V	W	X	Y	Z	[	\	]	^	_
6.	`	a	b	c	d	e	f	g	h	i	j	k	l	m	n	o
7.	p	q	r	s	t	u	v	w	x	y	z	{		}	~	DEL

# ЯЗЫК

# ПРОГРАММИРОВАНИЯ

Кодировка Windows-1251 (синоним CP1251)

	.0	.1	.2	.3	.4	.5	.6	.7	.8	.9	.A	.B	.C	.D	.E	.F
8.	Ъ 402	Ѓ 403	҃ 201A	҄ 453	҅ 201E	҆ 2026	҇ 2020	҈ 2021	€ 20AC	‰ 2030	Љ 409	< 2039	Њ 40A	Ѓ 40C	Ѕ 40B	Ц 40F
9.	Ђ 452	“ 2018	” 2019	“ 201C	” 201D	• 2022	— 2013	— 2014		™ 2122	Љ 459	> 203A	Њ 45A	Ѓ 45C	Ѕ 45B	Ц 45F
A.		Ў 40E	ў 45E	Ј 408	Љ A4	Ѓ 490	Ѕ A6	§ A7	Ё 401	© A9	Є 404	« AB	¬ AC		® AE	Ї 407
B.	° E0	± E1	І 406	і 456	҃ 491	μ E5	¶ E6	· E7	ё 451	№ 2116	е 454	» EB	ј 458	Ѕ 405	ѕ 455	ї 457
C.	А 410	Б 411	В 412	Г 413	Д 414	Е 415	Ж 416	З 417	И 418	Й 419	К 41A	Л 41B	М 41C	Н 41D	О 41E	П 41F
D.	Р 420	С 421	Т 422	У 423	Ф 424	Х 425	Ц 426	Ч 427	Ш 428	Щ 429	Ъ 42A	Ы 42B	Ь 42C	Э 42D	Ю 42E	Я 42F
E.	а 430	б 431	в 432	г 433	д 434	е 435	ж 436	з 437	и 438	й 439	к 43A	л 43B	м 43C	н 43D	о 43E	п 43F
F.	р 440	с 441	т 442	у 443	ф 444	х 445	ц 446	ч 447	ш 448	щ 449	ъ 44A	ы 44B	ь 44C	э 44D	ю 44E	я 44F

ЯЗЫК

ПРОГРАММИРОВАНИЯ

## Используемые символы

КОИ-8 (код обмена информацией, 8 битов), KOI8 — восьмибитовая ASCII — восьмибитовая ASCII-совместимая кодированная страница — восьмибитовая ASCII-совместимая кодированная страница, разработанная для кодирования букв кириллических алфавитов.

# ЯЗЫК ПРОГРАММИРОВАНИЯ

Кодировка KOI8-R (русская)

	. 0	. 1	. 2	. 3	. 4	. 5	. 6	. 7	. 8	. 9	. A	. B	. C	. D	. E	. F
8.	— 2500	 2502	┌ 250C	└ 2510	┐ 2514	┑ 2518	┒ 251C	┓ 2524	└ 252C	┘ 2534	┙ 253C	■ 2580	▀ 2584	▄ 2588	▨ 258C	▩ 2590
9.	▯ 2591	▮ 2592	▭ 2593	∫ 2320	■ 25A0	· 2219	√ 221A	≈ 2248	≤ 2264	≥ 2265	∞ A0	∫ 2321	◦ B0	2 B2	· B7	÷ F7
A.	= 2550	 2551	F 2552	ё 451	п 2553	р 2554	з 2555	т 2556	к 2557	л 2558	ц 2559	ч 255A	ш 255B	щ 255C	ъ 255D	ы 255E
B.	 255F	 2560	└ 2561	Ё 401	 2562	 2563	т 2564	т 2565	т 2566	┘ 2567	ц 2568	ч 2569	ч 256A	щ 256B	щ 256C	© A9
C.	Ю 44E	а 430	б 431	ц 446	д 434	е 435	ф 444	г 433	х 445	и 438	й 439	к 43A	л 43B	м 43C	н 43D	о 43E
D.	П 43F	Я 44F	р 440	с 441	т 442	у 443	ж 436	в 432	ь 44C	ы 44B	з 437	ш 448	э 44D	щ 449	ч 447	ъ 44A
E.	Ю 42E	А 410	Б 411	Ц 426	Д 414	Е 415	Ф 424	Г 413	Х 425	И 418	Й 419	К 41A	Л 41B	М 41C	Н 41D	О 41E
F.	П 41F	Я 42F	Р 420	С 421	Т 422	У 423	Ж 416	В 412	Ь 42C	Ы 42B	З 417	Ш 428	Э 42D	Щ 429	Ч 427	Ъ 42A