

Тема 2.1 Языки и системы программирования

1. Поколения языков программирования

Язык программирования (ЯП) – это система обозначений, служащая для точного описания алгоритмов решения задач на ЭВМ.

Программа – упорядоченный список команд для решения некоторой задачи.

В развитии программного обеспечения выделяют 5 поколений языков программирования, которые от поколения к поколению улучшают свои характеристики, становясь все более доступными в освоении.

Таблица – **Поколения ЯП**

Поколения	Языки программирования	Характеристика
I	Машинные	Набор машинных команд в двоичном или восьмеричном формате, который определяется архитектурой конкретной ЭВМ
II	Процедурные низкого уровня (Ассемблер)	Машинно-зависимые языки, использующие символьные обозначения
III	Процедурные высокого уровня (Fortran, Pascal, Basic, C)	Машинно-независимые мобильные языки, ориентированные на структурное программирование
IV	Непроцедурные языки (Prolog)	Минимальное число синтаксических правил, используются для задач искусственного интеллекта, ориентированы на непрофессионального пользователя
	Объектно-ориентированные языки (C++, Delphi, Visual Basic)	Основаны на понятии программного объекта
	Языки запросов SQL	Получение информации из баз данных
	Языки параллельного программирования (FP)	Ориентированы на создание программного обеспечения ЭВМ с параллельной архитектурой
V	Языки искусственного интеллекта, экспертных систем, баз знаний (LISP, HAL)	Ориентированы на повышение интеллектуального уровня ЭВМ и интерфейса с языками

2. Классификация ЯП

Языки программирования **классифицируют по следующим факторам:**

1) **Алгоритмичность:**

- *процедурные (алгоритмические) языки* – представляют собой последовательность команд для решения конкретной задачи, используя понятия операторов и данных – Ассемблер, Pascal, Basic;
- *непроцедурные языки* – языки обработки символьной информации, описывающие только *что*, а не *как* надо сделать – Prolog.

2) **Уровень ЯП:**

- *языки низкого уровня (машинно-зависимые или машинно-ориентированные)* – это языки, реализующие набор машинных команд, заложенных в центральный процессор конкретного компьютера - Ассемблер;

- *языки высокого уровня (машинно-независимые)* – представляют набор заданных команд на языке близком к английскому - Pascal, Basic, C;
 - *языки сверхвысокого уровня* – Сетл.
- 3) *Специализация ЯП:*
- *языки общего назначения (универсальные)* - Pascal, Basic;
 - *специализированные языки:*
 - ✓ инженерные – Fortran;
 - ✓ коммерческие – Cobol;
 - ✓ символьная обработка – LISP;
 - ✓ программирование в реальном времени – Ada.

Однако компьютер не может обрабатывать команды и данные до тех пор, пока они не будут представлены машинными кодами, т.е. последовательностью 0 и 1. Перевод программ, записанных на каком-либо языке программирования в машинные коды, осуществляется специальной служебной программой, называемой *транслятором*.

Известны два вида трансляторов – компилятор и интерпретатор, которые отличаются способом выполнения перевода.

Компилятор читает всю программу целиком, делает ее перевод и задает законченный вариант на машинном языке, который затем загружается и выполняется компьютером.

Интерпретатор переводит и выполняет программу строка за строкой.

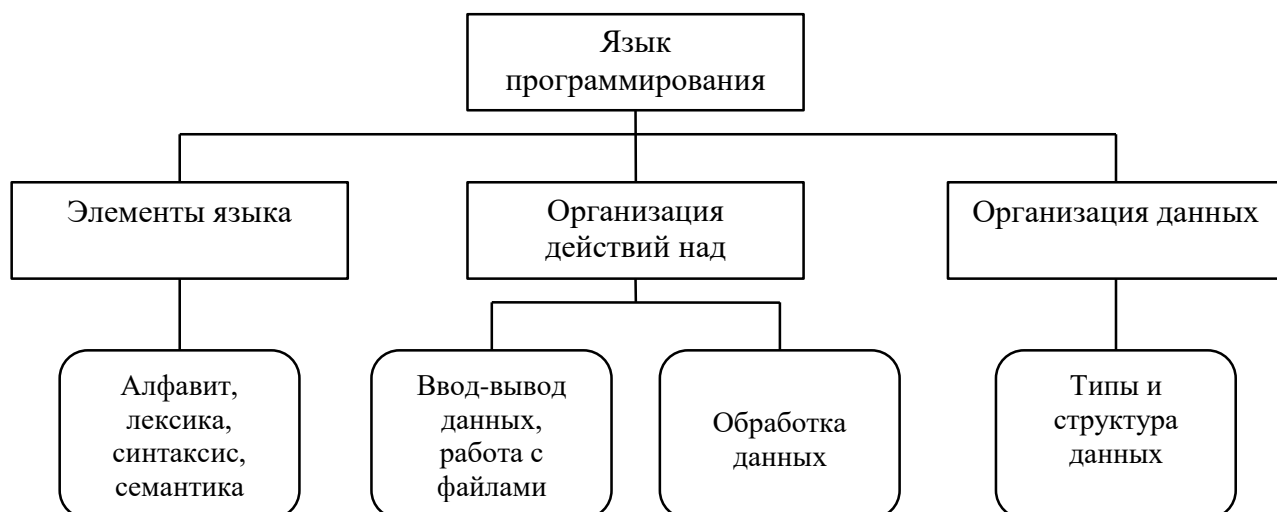
Для решения большинства задач программирования можно использовать любой язык. Однако на практике оказывается, что написать программу для решения конкретной задачи гораздо легче на одних языках, чем на других.

Выбор языка программирования определяется тремя факторами:

- язык должен быть удобен для программиста;
- язык должен быть пригоден для данного компьютера;
- язык должен быть пригоден для решения данной задачи.

3. Элементы языка программирования

В любом языке программирования определены элементы ЯП, способы организации данных и способы организации действий над ними.



Языки программирования имеют основные составляющие (элементы):

- *алфавит* – совокупность символов, отображаемых на устройствах печати и экране ЭВМ;

- *лексика* – совокупность правил образования цепочек символов (лексем), образующих идентификаторы, операторы, операции и другие компоненты языка;
- *синтаксис* – совокупность правил образования слов и предложений ЯП;
- *семантика* – смысловое содержание слов и предложений ЯП.

4. Системы программирования

Для разработки программ на ЯП необходима соответствующая система программирования.

Система программирования – это совокупность средств разработки программ, обеспечивающих автоматизацию составления и отладки программ пользователя.

Система программирования включает в себя, кроме ЯП:

- 1) *текстовый редактор*, осуществляющий функции записи и редактирования текстов программы;
- 2) *транслятор*, осуществляемый перевод исходного текста программы в машинные коды с диагностикой синтаксических и логических ошибок;
- 3) *загрузчик программ*, позволяющий выбрать из директория нужный текстовый файл программы;
- 4) *запускатель программ*, осуществляющий процесс выполнения программ;
- 5) *диспетчер файлов*, предоставляющий возможность выполнять операции с файлами;
- 6) *отладчик*, выполняющий сервисные функции по отладке и тестированию программы;
- 7) *компоновщик (редактор связей)*, осуществляющий объединение объектных и загрузочных модулей в единый модуль с последующей записью в файл;
- 8) *библиотека подпрограмм*, т.е. организованный соответствующим образом набор подпрограмм, используемый при решении разных задач.

Системы программирования классифицируют по 3 признакам:

1. *по набору исходных языков:*

- одноязыковые;
- многоязыковые, в которых отдельные части программ могут быть подготовлены на различных языках и объединены в единый модуль во время или перед выполнением программ.

2. *по возможности расширения:*

- замкнутые;
- открытые, в которые можно ввести новый входной язык с транслятором, не требуя изменений в системе.

3. *по способу трансляции:*

- компиляция;
- интерпретация.

Структура многоязыковой, открытой, компилирующей системы программирования:

(Схема разработки прикладных программ в среде СП):

Программа на исходном языке (*исходный модуль*) готовится с помощью текстовых редакторов и в виде текстового файла поступает на вход транслятора.

Трансляция исходной программы преобразует исходный модуль в промежуточную - объектную форму.

Объектный модуль представляет собой текст программы на машинном языке, включающий машинные инструкции, словари, служебную информацию.

С помощью специальных средств (компоновщика, редактора связей) объединяется объектный и загрузочный модуль в единый *загрузочный модуль*, с последующей записью в библиотеку или файл.

Загрузочный модуль после сборки помещается в пользовательскую библиотеку программ или непосредственно направляется на исполнение.

Выполнение модуля состоит в загрузке его в оперативную память, настройке по месту в памяти и передаче ему управления. Образ загрузочного модуля в памяти называется *абсолютным модулем*, поскольку все команды ЭВМ здесь приобретают окончательную форму и получают абсолютные адреса в памяти.

Современные системы программирования являются *интегрированной средой программирования*, которые содержат текстовый редактор, компилятор, компоновщик, встроенный отладчик, и предоставляют дополнительные удобства для написания и отладки программ.