

ФГБОУ ВПО  
"Сибирская государственная автомобильно-дорожная академия"  
(СибАДИ)  
Кафедра "ИТ"

**Контрольная работа № 1  
по информатике**

**Тема № 50: «Эволюция и классификация языков программирования.  
Основные понятия языков программирования.  
Трансляция. Компиляция и интерпретация»**

**Выполнил:  
Шифр зачетки –  
Проверила:**

**Омск - 2015**

# ОГЛАВЛЕНИЕ:

1. Введение

2. Описание языков программирования

3. Трансляция

4. Компиляция

5. Интерпритация

6. Линкование

7. Два способа линкования

8. Заключение

Список литературы

Завершить  
показ

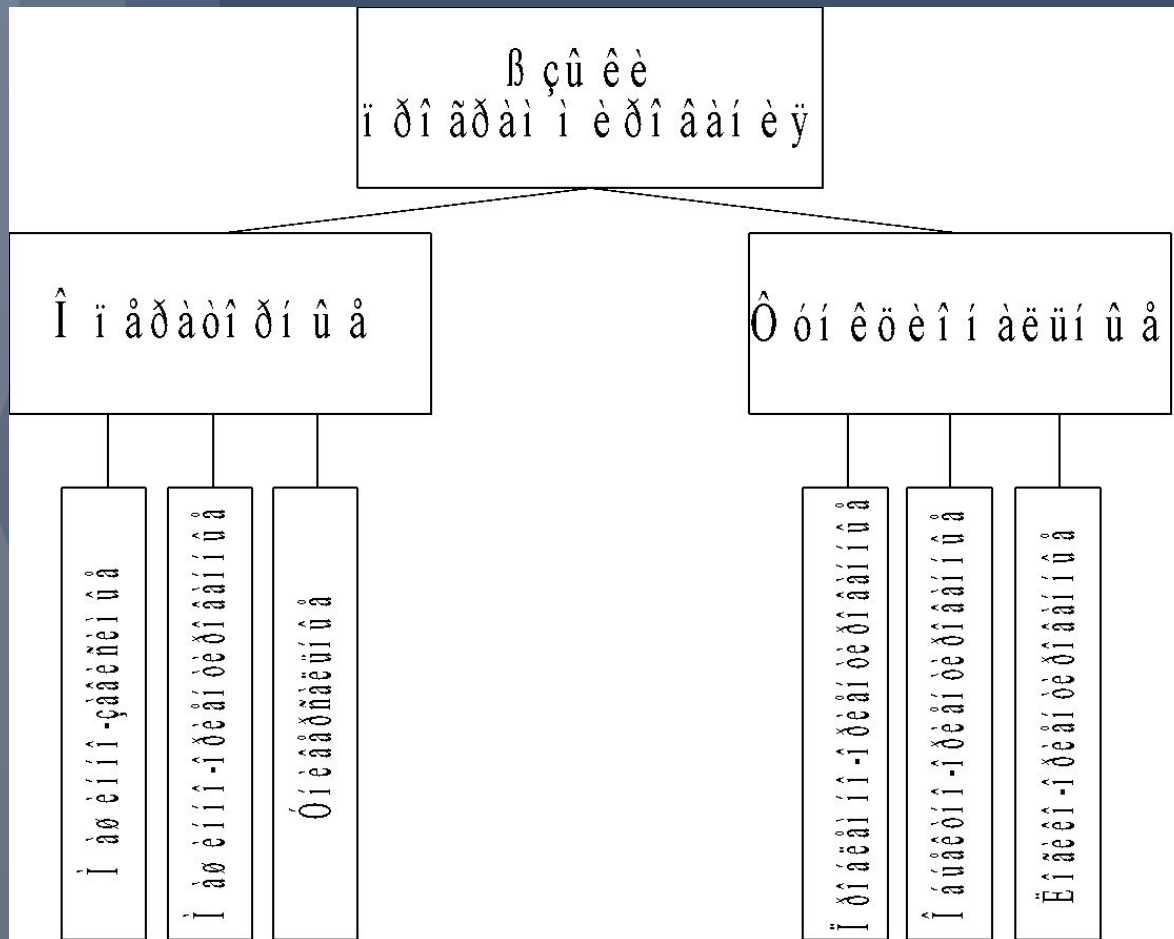
# 1. Введение

Язык программирования - это специальный язык, на котором пишут команды для управления компьютером. Языки программирования созданы для того, чтобы людям было проще читать и писать для компьютера, но они затем должны транслироваться (транслятором или интерпретатором) в машинный код, который только и может исполняться компьютером. Языки программирования можно разделить на языки высокого уровня и языки низкого уровня.

Язык низкого уровня - это язык программирования предназначенный для определенного типа компьютера и отражающий его внутренний машинный код; языки низкого уровня часто называют машинно-ориентированными языками. Их сложно конвертировать для использования на компьютерах с разными центральными процессорами, а также довольно сложно изучать, поскольку для этого требуется хорошо знать принципы внутренней работы компьютера.

Язык высокого уровня - это язык программирования, предназначенный для удовлетворения требований программиста; он не зависит от внутренних машинных кодов компьютера любого типа. Языки высокого уровня используют для решения проблем и поэтому их часто называют проблемно-ориентированными языками. Каждая команда языка высокого уровня эквивалентна нескольким командам в машинных кодах, поэтому программы, написанные на языках высокого уровня, более компактны, чем аналогичные программы в машинных кодах.

## 2. Описание языков программирования



**Трансляция программы** — преобразование программы, представленной на одном из языков программирования, в программу на другом языке и, в определённом смысле, равносильную первой. При трансляции выполняется перевод программы, понятной человеку, на язык, понятный компьютеру. Выполняется специальными программными средствами (транслятором).

Трансляторы реализуются в виде компиляторов или интерпретаторов. С точки зрения выполнения работы компилятор и интерпретатор существенно различаются. Если цель трансляции – преобразование всего исходного текста на внутренний язык компьютера (т.е. получение некоторого нового кода) и только, то такая трансляция называется также компиляцией. Исходный текст называется также исходной программой или исходным модулем, а результат компиляции – объектным кодом или объектным модулем. Если же трансляции подвергаются отдельные операторы исходных текстов и при этом полученные коды сразу выполняются, такая трансляция называется интерпретацией. Поскольку трансляция выполняется специальными программными средствами (трансляторами), последние носят название компилятора или интерпретатора, соответственно.

**Цель трансляции** — преобразовать текст с одного языка на другой, который понятен адресату текста. В случае программ-трансляторов, адресатом является техническое устройство (процессор) или программа-интерпретатор.

Виды трансляторов

**Трансляторы подразделяют на:**

*Адресный.* Функциональное устройство, преобразующее виртуальный адрес в реальный адрес

*Диалоговый.* Обеспечивает использование языка программирования в режиме разделения времени.

*Многопроходной.* Формирует объектный модуль за несколько просмотров исходной программы.

*Обратный.* То же, что детранслятор (декомпилятор, дизассемблер).

*Однопроходной.* Формирует объектный модуль за один последовательный просмотр исходной программы.

*Оптимизирующий.* Выполняет оптимизацию кода в создаваемом объектном модуле.

*Синтаксически-ориентированный (синтаксически-управляемый).* Получает на вход описание синтаксического и семантического языка и текст на описанном языке, который и транслируется в соответствии с заданным описанием.

*Тестовый.* Набор макрокоманд языка ассемблера, позволяющих задавать различные отладочные процедуры в программах

## 4. Компиляция

Компиляция — преобразование программой-компилятором исходного текста программы, написанного на языке высокого уровня в машинный язык, в язык, близкий к машинному, или в объектный модуль. Результатом компиляции является объектный файл с необходимыми внешними ссылками для компоновщика.

Компилятор читает всю программу целиком, делает ее перевод и создает законченный вариант программы на машинном языке, который затем и выполняется.

Виды компиляции

*Пакетная.* Компиляция нескольких исходных модулей в одном пункте задания.

*Построчная.* То же, что и интерпретация.

*Условная.* Компиляция, при которой транслируемый текст зависит от условий, заданных в исходной программе. Так, в зависимости от значения некоторой константы, можно включать или выключать трансляцию части текста программы.

## 5. Интерпритация

**Интерпритация** — процесс непосредственного покомандного выполнения программы без предварительной компиляции, «на лету»; в большинстве случаев интерпритация намного медленнее работы уже скомпилированной программы, но не требует затрат на компиляцию, что в случае небольших программ может повышать общую производительность.

Типы интерпритаторов

Простой интерпритатор анализирует и тут же выполняет (собственно интерпритация) программу покомандно (или построчно), по мере поступления её исходного кода на вход интерпритатора. Его достоинство - мгновенная реакция. Недостаток — такой интерпритатор обнаруживает ошибки в тексте программы только при попытке выполнения команды (или строки) с ошибкой.

Интерпритатор компилирующего типа — это система из компилятора, переводящего исходный код программы в промежуточное представление, например, в байт-код или р-код, и собственно интерпритатора, который выполняет полученный промежуточный код (так называемая виртуальная машина). Его достоинство – большее быстродействие выполнения программ (за счёт выноса анализа исходного кода в отдельный, разовый проход, и минимизации этого анализа в интерпритаторе).

Недостатки — большее требование к ресурсам и требование на корректность исходного кода.

Алгоритм работы простого интерпритатора

прочитать инструкцию;

проанализировать инструкцию и определить соответствующие действия;

выполнить соответствующие действия;

если не достигнуто условие завершения программы, прочитать следующую инструкцию и перейти к пункту 2.

## 6. Линкование

**Линкование (компоновка)** - это процесс, при котором все "недокомпилированные" части программы доводятся до конца и связываются между собой в исполняемый файл (или файлы) формата, понятного данной операционной системе. В итоге, мы получаем исполняемую программу.

Линкование важно, так как при разработке больших проектов код обычно вырастает настолько, что приходится его для повышения надежности раскидывать по разным файлам.

Для связывания модулей компоновщик использует таблицы имён, созданные компилятором в каждом из объектных модулей.

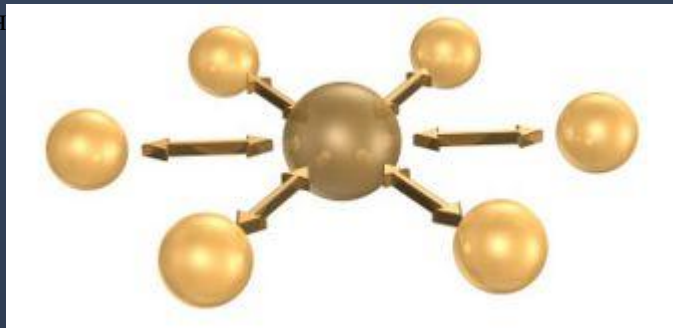
Такие имена могут быть двух типов:

*Определённые* или *экспортируемые* имена — функции и переменные, определённые в данном модуле и предоставляемые для использования другим модулям

*Неопределённые* или *импортируемые* имена — функции и переменные, на которые ссылается модуль, но не определяет их внутри себя

Работа компоновщика заключается в том, чтобы в каждом модуле разрешить ссылки на неопределённые имена. Для каждого импортируемого имени находится

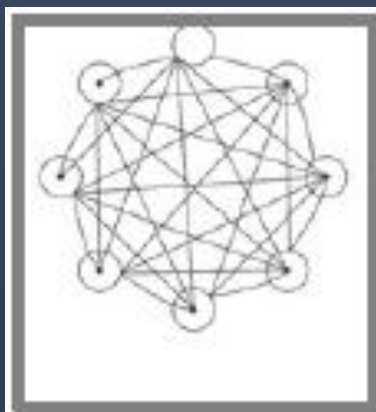
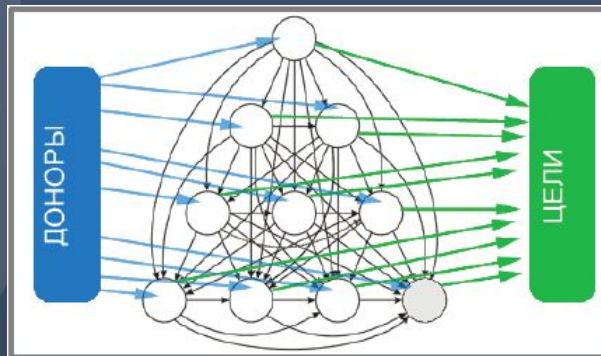
имени заменяется на его адрес.





## 7. Два способа линкования

Существует два способа создания линкования:



## 8. Заключение

есть из исходного текста на языке высокого уровня получается машинный код, пригодный для исполнения на данном процессоре. Если необходимо выполнить программу на другом типе процессора с несовместимой системой команд, то необходима перекомпиляция исходной программы на языке высокого уровня для данного типа процессора.

Программа же на интерпретируемом языке транслируется в машинные коды лишь при её выполнении. Делается это, по мере поступления команд, следующим образом. Каждой команде интерпретируемого языка в коде интерпретатора соответствует подпрограмма, созданная с использованием поддерживаемых аппаратной частью компьютера средств. Когда эта команда встречается в коде, связанная с ней подпрограмма запускается, и необходимые действия выполняются. Так как особенности реализации интерпретируемых языков скрыты, кажется, что имеющиеся в них команды проделываются напрямую компьютером. Поэтому соответствующие интерпретаторы называют виртуальными компьютерами (или, чаще, виртуальными машинами). Для переноса программы на другую архитектуру (или другую платформу) необходимо лишь изменить интерпретатор, а исходный код программы не трогать.

## Список литературы

1. Информатика. (Учебное пособие) Горяев Ю.А. (МИЭМП; 2005, 116с.)
2. Информатика. (Учебник) Макарова Н.В., Волков В.Б. (2011, 576с.)
3. Информатика. (Учебник) Могилев А.В., Пак Н.И., Хённер Е.К. (2004, 3-е изд., 848с.)
4. Информатика. (Учебник) Каймин В.А. (2001, 2-е изд., 272с.)
5. Информатика. (Учебник) Под общ. ред. А.Н. Данчула (2004, 528с.)
6. Информатика. (Учебник) Соболев Б.В. и др. (2007, 3-е изд, 446с.)
7. Информатика. (Учебник) Степанов А.Н. (2006, 4-е изд., 684с.)
8. Информатика. (Учебное пособие) Терехов А.В., Чернышев А.В., Чернышев В.Н. (ТГТУ; 2007, 128с.)