

**Программа вступительного испытания  
по дисциплине «Языки программирования»  
для поступающих на направление подготовки магистратуры  
02.04.02 – Фундаментальная информатика и информационные технологии**

***История развития и основные классы языков программирования***

История возникновения и развития языков программирования. Императивное программирование. Языки функционального программирования. Языки логического программирования. Визуальное программирование. Языки моделирования. Языки параллельного и распределенного программирования.

***Основные свойства языков объектно-ориентированного программирования***

Наследование в объектно-ориентированном программировании. Инкапсуляция в объектно-ориентированном программировании. Полиморфизм в объектно-ориентированном программировании.

***Средства формального определения языков программирования***

Цели и принципы разработки языков программирования. Стандартизация языков. Математическое моделирование языков. Формальные языки. Формальные грамматики, примеры. Классификация Хомского формальных грамматик. Пример грамматики класса 3. Пример грамматики класса 2. Пример грамматики класса 1. БНФ как средство описания языков программирования. Синтаксические диаграммы как средство описания языков программирования. Контекстно-свободные языки как основа большинства языков программирования.

***Типы данных***

Понятие типа данных. Необходимость использования типов данных. Классификация типов данных в языках программирования. Системы типизации. Контроль типов. Примеры языков, поддерживающих различные системы типизации.

***Принципы виртуализации***

Управление памятью в реализации языка программирования. Понятие виртуальной машины. Техника реализации: эмуляция, виртуализация на уровне операционной системы. Виртуальная машина Java. Java Runtime Environment (JRE).

## ***Инфраструктура языков программирования***

Общая инфраструктура языков (CLI). Составные части спецификации CLI. Общая система типов в CLI. Виртуальная система выполнения. Промежуточные языки. Язык CIL.

## ***Основные задачи реализации языков программирования высшего уровня***

Проблемы безопасности выполнения программного кода на другой машине. Трансляции как отображения языков. Сравнение компиляторов и интерпретаторов. Методы получения объектного кода (абсолютный и переместимый код, преобразование в ассемблерный код, в байт-код, компиляция «на лету»). Схема компилятора. Основные фазы компиляции.

Просмотры в компиляторе. Сравнение однопросмотрового (однопроходного) и многопроходного компиляторов.

## ***Лексический анализ***

Задачи лексического анализа. Заглядывание вперед при лексическом анализе. Построение таблиц идентификаторов в компиляторе. Хеширование как метод построения таблиц идентификаторов.

## ***Синтаксический анализ***

Общие методы синтаксического анализа. Использование грамматик. Нисходящий разбор при синтаксическом анализе. Восходящий разбор при синтаксическом анализе. Однопроходный синтаксический анализ без возвратов. Алгоритм синтаксического LL(1)-языков. Построения множества FIRST при синтаксическом анализе LL(1)-языков. Построение функций (процедур) синтаксического анализа конструкции ⟨программа⟩. Пример на С или Паскале. Построение функций (процедур) синтаксического анализа конструкции ⟨блок⟩. Пример на С или Паскале. Построение функций (процедур) синтаксического анализа конструкции ⟨оператор while⟩. Пример на С или Паскале. Построение функций (процедур) синтаксического анализа для правил, содержащих { }. Синтаксический анализ конструкции ⟨составной оператор⟩. Пример на С или Паскале. Синтаксический анализ конструкции ⟨переменная⟩. Пример на С или Паскале. Синтаксический анализ конструкции ⟨выражение⟩. Пример на С или Паскале. Нейтрализация синтаксических ошибок в процессе трансляции. Назначение, методы.

## ***Семантический анализ***

Семантический анализ программы при трансляции. Виды семантических ошибок. Контроль типов. Идентификация идентификаторов. Таблицы трансляторов. Описатель процедуры. Дескрипторы типов различных типов данных. Атрибутное дерево разбора.

## ***Генерация кода***

Промежуточные формы представления программы. Обратная польская запись. Перевод в ОПЗ. Исполнение ОПЗ. Трехадресный код как промежуточная форма программы. Тетрады, триады. Синтаксически управляемый перевод. Схема. Процесс перевода. Пример. Перевод и генерация кода для арифметических выражений. Генерация кода операторов FOR для системы команд 80x86. Генерация кода для операторов FOR. Генерация кода операторов WHILE для системы команд 80x86. Генерация кода операторов REPEAT для системы команд 80x86.

## ***Оптимизация исполняемого кода***

Графы потока управления и графы потока данных. Оптимизирующие трансляторы. Задачи оптимизации. Виды оптимизации, зависимость между оптимизациями. Стадии оптимизации. Покадровая оптимизация.

Пример оптимизационных преобразований. Удаление мертвого кода. Упрощение выражений. Свертка констант. Экономия общих подвыражений. Объединение циклов. Раскрутка циклов. Пример оптимизационных преобразований. Чистка циклов. Понижение силы операций. Оптимизация. Зависимость качества оптимизации от размера участка экономии. Распределение регистров при компиляции.

## **Основная литература**

1. Ахо А., Лам М., Сети Р., Ульман Д. Компиляторы: принципы, технологии и инструментарий. – М.: ООО «И.Д. Вильямс», 2014.
2. Вирт Н. Построение компиляторов. – М.: ДМК Пресс, 2010.
3. Королев Л.Н., Миков А.И. Информатика. Введение в компьютерные науки. – М.: ООО «Абрис», 2012.

4. Миков А.И. Информационные процессы и нормативные системы в IT: Математические модели. Проблемы проектирования. Новые подходы. – М.: Книжный дом «Либроком», 2013.

5. Миков А.И. Распределенные алгоритмы в компьютерных сетях. – Ростов-на-Дону: изд. ЮФУ, 2015.

### **Дополнительная литература**

1. Ахо А., Ульман Дж. Теория синтаксического анализа, перевода и компиляции. Том 1. Синтаксический анализ. – М.: Мир, 1978.

2. Ахо А., Ульман Дж. Теория синтаксического анализа, перевода и компиляции. Том 2. Компиляция. – М.: Мир, 1978.

3. Грис Д. Конструирование компиляторов для цифровых вычислительных машин. – М.: Мир, 1975.

4. Залогова Л.А. Разработка Паскаль-компилятора. – М.: БИНОМ-Лаборатория знаний, 2007.

5. Карпов Ю. Теория и технология программирования. Основы построения трансляторов. – СПб.: ВHV-СПб, 2003.

6. Компаниец Р., Маньков Е., Филатов Н. Системное программирование. Основы построения трансляторов. – М.: КОРОНА принт, 2002.

7. Королев Л.Н., Миков А.И. Информатика. Введение в компьютерные науки (гл. 3, 7). – М.: Высшая школа, 2003.

8. Макаров А.В., Скоробогатов С.Ю., Чеповский А.М. Common Intermediate Language и системное программирование в Microsoft .NET. – М.: Интернет Университет.

9. Питерсон Дж. Теория сетей Петри и моделирование систем. – М.: Мир, 1984.

10. Рейуорд-Смит В.Дж. Теория формальных языков. Вводный курс. – М.: Радио и связь, 1988.

11. Саломеа А. Жемчужины теории формальных языков. – М.: Мир, 1986.

12. Хантер Р. Основные концепции компиляторов. – М.: Вильямс, 2005.

13. Хоар Ч. Взаимодействующие последовательные процессы. – М.: Мир, 1989.