

**МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ РОССИЙСКОЙ ФЕДЕРАЦИИ
ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ БЮДЖЕТНОЕ ОБРАЗОВАТЕЛЬНОЕ
УЧРЕЖДЕНИЕ ВЫСШЕГО ПРОФЕССИОНАЛЬНОГО ОБРАЗОВАНИЯ**

**ТОМСКИЙ ГОСУДАРСТВЕННЫЙ ПЕДАГОГИЧЕСКИЙ УНИВЕРСИТЕТ
(ТГПУ)**

«УТВЕРЖДАЮ»
Декан физико-математического факультета



А.Н. Макаренко

«10» августа 2011 года

ПРОГРАММА ДИСЦИПЛИНЫ

Трансляция с языка высокого уровня

СД.04

Направление подготовки

230200.62 - Информационные системы

Степень (квалификация) –

Бакалавр информационных систем

1. Цели и задачи дисциплины

1.1. Цель преподавания дисциплины

Цель преподавания дисциплины – ознакомление студентов с методами решения проблемы трансляции программ, созданных на языке высокого уровня в машинные коды.

1.2. Задачи изучения дисциплины

Задача изучения дисциплины – ознакомление с теорией формальных языков, конкретными методами анализа автоматных и контекстно-свободных языков, методами генерации и интерпретации обратной польской записи.

1.3. Перечень дисциплин, усвоение которых студентами необходимо для изучения данного курса

«Дискретная математика», «Математическая логика и теория алгоритмов», «Алгоритмы и структуры данных».

2. Требования к уровню освоения содержания дисциплины

В результате изучения дисциплины студент должен знать основы теории формальных языков и уметь построить простой транслятор.

3. Объем дисциплины и виды учебной работы

Вид учебной работы	Всего часов	Семестры			
Общая трудоемкость дисциплины	178	6			
Аудиторные занятия	114	114			
Лекции	57	57			
Практические занятия (ПЗ)					
Семинары (С)					
Лабораторные работы (ЛР)	57	57			
И (или) другие виды аудиторных занятий					
Самостоятельная работа	64	64			
Курсовой проект (работа)					
Расчетно-графические работы					
Реферат					
И (или) другие виды самостоятельной работы					
Вид итогового контроля (зачет, экзамен)		экзамен			

4. Содержание дисциплины

4.1. Разделы дисциплины и виды занятий

№ п/п	Разделы дисциплины	Лекции	Практические занятия или семинары	Лабораторные занятия
1.	Обзор задач, требующих решения в процессе трансляции с языка высокого уровня.	4		
2.	Лексический анализ.	12		16
3.	Синтаксический анализ.	20		25
4.	Обратная польская запись, как второй внутренний язык.	12		16
5.	Генерация команд.	9		

4.2. Содержание разделов дисциплины

1. Обзор задач, требующих решения в процессе трансляции с языка высокого уровня.

Понятие о языках высокого и низкого уровня. Трансляция. Компиляция и интерпретация. Этапы трансляции: лексический анализ, синтаксический анализ, генерация команд. Требования, предъявляемые к внутренним языкам.

2. Лексический анализ.

Понятие лексемы. Виды лексем – ключевые слова, константы, переменные, комментарии и т.д.

Понятие об автоматной грамматике и конечном автомате. Детерминированный и недетерминированный конечный автомат, приведение конечного автомата к детерминированному виду. Конечный автомат с лямбда-переходами. Применение регулярных выражений для описания лексики языка программирования.

Методика применения конечного автомата в процессе лексического анализа и формирования программы в первом внутреннем представлении.

3. Синтаксический анализ.

Контекстно-свободные грамматики (КС-грамматики) и их анализ. Представление синтаксиса в форме КС-грамматик и в форме Бэкуса-Науэра. Магазиновый автомат и его построение по контекстно-свободной грамматике для нисходящего и восходящего анализа.

Практические алгоритмы анализа контекстно-свободных языков. Левое и правое порождение. LL-анализаторы и LR-анализаторы. Рекурсивный спуск.

Восходящий анализ и дилемма «перенос-свертка». Использование отношений предшествования в процессе синтаксического анализа. Простое и операторное предшествование. Линеаризация матриц предшествования.

4. Обратная польская запись, как второй внутренний язык.

Обратная польская запись арифметических выражений. Алгоритм интерпретации обратной польской записи с использованием стека. Расширение обратной польской записи. Безусловные переходы и ветвления в обратной польской записи. Организация циклов. Использование индексов. Call и get в обратной польской записи. Генерация обратной польской записи в процессе синтаксического анализа на примере рекурсивного спуска. Генерация обратной польской записи для выражений с помощью алгоритма Дейкстры.

Контекстный анализ.

Алгоритм интерпретации обратной польской записи с помощью стека.

5. Генерация команд.

Особенности машинного языка. Структура исполняемого модуля. Триады и тетрады. Оптимизация на уровне триад. Генерация команд по тетрадам.

5. Лабораторный практикум

№ п/п	№ раздела дисциплины	Наименование лабораторных работ
1	2	Проектирование лексического анализатора простого языка программирования
2	3,4	Проектирование синтаксического анализатора простого языка программирования
3	4	Интерпретация обратной польской записи

6. Учебно-методическое обеспечение дисциплины

6.1. Рекомендуемая литература

а) основная литература:

1. Свердлов, С.З. Языки программирования и методы трансляции / С.З. Свердлов. – СПб.: Питер, 2007. – 637 с.

б) дополнительная литература

1. Ахо, А. Компиляторы: принципы, технологии, инструменты / А. Ахо, Р. Сети, Д. Ульман; пер. С англ. И. В. Красикова. – М.: Вильямс, 2003. – 767 с.
2. Ахо, А. Структуры данных и алгоритмы / А. В. Ахо, Д. Э. Хопкрофт, Д. Д. Ульман. – М.: Вильямс, 2003. – 382 с.
3. Вирт, Н. Алгоритмы и структуры данных / Н. Вирт. – СПб.: Невский диалект, 2007. – 351 с.
4. Кнут, Д. Искусство программирования: в 3 т. / Д. Э. Кнут. – 3-е изд., испр. и доп. – М.: Вильямс, 2003. – т. 1-3.
5. Новиков, Ф.А. Дискретная математика для программистов: учебное пособие для вузов / Ф. А. Новиков. – 2-е изд. – СПб.: Питер, 2009. – 363 с.
6. Хопкрафт, Д. Введение в теорию автоматов, языков и вычислений / Д. Хопкрафт, Р. Мотвани, Д. Ульман. – М.: Вильямс, 2002. – 254 с.

7. Материально-техническое обеспечение дисциплины

Лекции желательно проводить с использованием мультимедиа-проектора.

Лабораторный практикум проводится в среде Free Pascal, или на базе иного компилятора. По желанию студента возможно выполнение лабораторных работ на других языках программирования высокого уровня в других средах.

8 Методические рекомендации по организации изучения дисциплины

8.1. Методические рекомендации для преподавателей

Преподаватель должен последовательно излагать теоретический материал в рамках лекционных занятий. При этом предлагаемого материала должно быть достаточно для того, чтобы студент мог самостоятельно углублять полученные знания по мере необходимости. Важно помнить, что данная дисциплина, с одной стороны носит фундаментальный характер, так в ней достаточно подробно рассматривается теория формальных языков, с другой сторо-

ны дисциплина направлена на решение прикладных задач – построение трансляторов с языков высокого уровня различных типов. В лекционном курсе, материал целесообразно сопровождать примерами на основе упрощенного алгоритмического языка, реализация транслятора с которого предполагается лабораторным практикумом.

Простейший язык, транслятор с которого реализуется студентом, должен в обязательном порядке содержать оператор присваивание, арифметические операции, операции сравнения, условный оператор (в короткой и длинной форме), оператор цикла (либо оператор цикла с предусловием, либо оператор цикла с постусловием, либо оператор цикла с параметром), одномерные массивы, вызов подпрограмм с возможностью организации рекурсии.

В составе учебно-методического комплекса дисциплины имеются варианты реализации такого интерпретатора на языке Паскаль.

В рамках лабораторного практикума крайне важно добиться от студентов навыков самостоятельной реализации предлагаемых алгоритмов. На экзамене преподаватель должен убедиться не только в знании студентом вопросов конкретного билета, но и убедиться в общих знаниях по предмету. С этой целью могут непосредственно на экзамене задаваться дополнительные вопросы. При выставлении оценки, преподаватель должен ориентироваться не столько на объем информации, которую студент может запомнить, сколько на понимание материала и способность к его практическому применению.

8.2. Методические рекомендации для студентов

Студентам предлагается использовать предлагаемый курс лекций, а также основную и дополнительную литературу для изучения предмета. Стоит обратить внимание на то, что для того, чтобы иметь возможность освоить материала в рамках данной дисциплины, следует знать основы программирования на языках высокого уровня, основы теории вычислимости, теории алгоритмов и элементы теории формальных языков. Список предлагаемой литературы позволяет в случае необходимости ликвидировать пробелы в данной области.

Дисциплина построена путем последовательного изучения различных этапов трансляции и их реализации: лексический анализ, синтаксический анализ, интерпретация / генерация команд. Для изучения внутренних представлений используются наиболее общие подходы, на которых, как правило, основаны все конкретные решения.

Важнейшую роль играет выполнение лабораторных работ, комплекс которых позволяет реализовать интерпретатор с простейшего языка, более сложные задачи могут решаться студентами в рамках самостоятельной работы, а также в качестве курсовых и дипломных работ.

Примерный перечень вопросов к экзамену

1. Основные парадигмы языков программирования.
2. Задача трансляции. Типы трансляции.
3. Основные этапы трансляции.
4. Понятие лексем и их типы. Задачи лексического анализа.
5. Понятие формального языка и формальной грамматики.
6. Классификация грамматик и языков по Хомскому.
7. Автоматные языки и конечные автоматы. ДКА и НКА (обзор).
8. Регулярные выражения, как форма записи автоматной грамматики.
9. Использование ДКА на этапе лексического анализа.
10. Особенности программирования лексических анализаторов.
11. Первые внутренние представления и их особенности.
12. Основные функции и методы синтаксического анализа.
13. Описание синтаксиса языка программирования в форме Бэкуса-Науэра.
14. Описание синтаксиса языка программирования в виде контекстно-свободной грамматике.

15. Простейшие приемы оптимизации контекстно-свободных грамматик.
16. Понятие о магазинном автомате.
17. Построение магазинного автомата по контекстно-свободной грамматике и нисходящий анализ.
18. Построение магазинного автомата по контекстно-свободной грамматике и восходящий анализ.
19. Левый и правый проходы, левое и правое порождения.
20. Прямая и косвенная левая рекурсии, ее устранение. Нормальная форма Грейбаха.
21. Построение таблицы LL-анализатора по контекстно-свободной грамматике.
22. Рекурсивный спуск, как метод реализации LL-анализа.
23. Понятие об обратной польской записи. Алгоритм Дейкстры преобразования инфиксной записи в обратной польской записи.
24. Различные методики контроля корректности арифметических выражений.
25. Восходящий анализ и проблема построения LR-анализаторов.
26. Простое предшествование и построение LR(1) — анализатора.
27. Операторное предшествование и построение LR(1) — анализатора.
28. Линеаризация матриц предшествования.
29. 2-е внутренние представления и их особенности.
30. Представление простых команд, а также условных операторов и циклов в обратной польской записи.
31. Представление вызовов подпрограмм и возврата из подпрограмм в обратной польской записи.
32. Основные принципы работы виртуальной машины Java.
33. Особенности синтаксического анализа непроедурных языков.
34. Контекстный анализ.
35. Интерпретация арифметических выражений в обратной польской записи.
36. Особенности интерпретации расширенной обратной польской записи. Работа с адресами. Простые команды.
37. Переходы и вызовы в процессе интерпретации расширенной обратной польской записи.
38. Оптимизация вторых внутренних представлений на примере обратной польской записи.
39. 3-е внутренние представление. Триады и тетрады.
40. Оптимизация последовательности триад. Генерация команд по тетрадам.

Задание для самостоятельной работы

Проектирование интерпретатора языка программирования должен в обязательном порядке содержать оператор присваивание, арифметические операции, операции сравнения, условный оператор (в короткой и длинной форме), оператор цикла (либо оператор цикла с предусловием, либо оператор цикла с постусловием, либо оператор цикла с параметром), одномерные массивы, вызов подпрограмм с возможностью организации рекурсии.

Синтаксис языка может быть определен индивидуально для каждого студента, в том числе и самим студентом.

Реализация транслятора студентом возможна в любой среде на любом языке программирования по его выбору.

Программа составлена в соответствии с Государственным образовательным стандартом высшего профессионального образования по направлению **230200.62 – Информационные системы**, степень (квалификация) - **бакалавр информационных систем**.

Программу составил:

к. т. н., доцент кафедры информатики  Стась А.Н.

Программа дисциплины утверждена на заседании кафедры информатики протокол № 1 от «30» апреля 2011 г.

Зав. кафедрой  А.Н. Макаренко

Программа дисциплины одобрена методической комиссией физико-математического факультета ТГПУ

Председатель методической комиссии физико-математического факультета  Г.К. Разина