МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ
РОССИЙСКОЙ ФЕДЕРАЦИИ

Федеральное государственное бюджетное образовательное учреждение
высшего образования
«Магнитогорский государственный технический университет им. Г.И. Носова»

УТВЕРЖДАЮ:
Директор института
_____ С.И. Лукьянов
«26» сентября 2018 г.

РАБОЧАЯ ПРОГРАММА ДИСЦИПЛИНЫ (МОДУЛЯ)

ПРАКТИЧЕСКИЕ АСПЕКТЫ РАЗРАБОТКИ КОМПИЛЯТОРОВ

Направление подготовки (специальность)
09.03.01 Информатика и вычислительная техника

Направленность (профиль/ специализация) программы
Автоматизированные системы обработки информации и управления
Уровень высшего образования – бакалавриат

Программа подготовки – прикладной бакалавриат

Форма обучения
Очная

Институт	*энергетики и автоматизированных систем*
Кафедра	*вычислительной техники и программирования*
Курс	*4*
Семестр	*8*

Магнитогорск
2018 г.

Рабочая программа составлена на основе ФГОС ВО по направлению подготовки (специальности) 09.03.01 Информатика и вычислительная техника, утвержденного приказом МО и Н РФ от 12.01.2016 № 5.

Рабочая программа рассмотрена и одобрена на заседании кафедры вычислительной техники и программирования «05» сентября 2018 г., протокол № *1*.

Зав. кафедрой _____ О.С. Логунова

Рабочая программа одобрена методической комиссией института энергетики и автоматизированных систем «26» сентября 2018 г., протокол № 1.

Председатель _____ С.И. Лукьянов

Рабочая программа составлена: доцентом кафедры вычислительной техники и программирования, канд. техн. наук, доцентом

_____ А.Н. Калитаевым

Рецензент: начальник отдела инновационных разработок ЗАО «КонсОМ-СКС», канд. техн. наук

_____ А.Н. Панов

Лист регистрации изменений и дополнений

№ п/п	Раздел программы	Краткое содержание изменения/дополнения	Дата. № протокола заседания кафедры	Подпись зав. кафедрой
1	8	Корректировка списка рекомендуемой литературы	2,09,2019, протокол №1	
2	9	Обновление ссылки на перечень программного обеспечения	2,09,2019, протокол №1	

3

1 Цели освоения дисциплины (модуля)

Целями освоения дисциплины (модуля) «Практические аспекты разработки компиляторов» являются: ознакомление студентов с основными структурами, видами и основными задачами трансляторов; основами теории формальных языков и грамматики, типах распознавателей и преобразователей, а также принципами и технологиями построения компиляторов для цифровых вычислительных машин.

Для достижения поставленной цели в курсе «Практические аспекты разработки компиляторов» решаются задачи:

- изучение понятий о методах трансляции, принципах, технологиях и программных средствах построения компиляторов;
- получение знаний о теории формальных языков и грамматик; распознавателей и преобразователей;
- получение знаний о формальных методах описания перевода: СУ-схемы, транслирующие грамматики, атрибутные транслирующие грамматики;
- получение знаний об алгоритмах синтаксического анализа для LL(K)-грамматик, LR(K)-грамматик, грамматик предшествования;
- получение знаний о включении семантики в алгоритмы синтаксического анализа.

2 Место дисциплины (модуля) в структуре образовательной программы подготовки бакалавра (магистра, специалиста)

Дисциплина «Практические аспекты разработки компиляторов» входит в вариативную часть блока 1 образовательной программы.

Для изучения дисциплины необходимы знания (умения, владения), сформированные в результате изучения курсов: прикладное программирование, структуры и модели данных, алгоритмы и теория сложности, алгоритмы на сетях и графах, теория вычислительных процессов, объектно-ориентированное программирование и т.д.

Знания (умения, владения), полученные при изучении данной дисциплины будут необходимы при подготовке к государственной итоговой аттестации студентов.

3 Компетенции обучающегося, формируемые в результате освоения дисциплины (модуля) и планируемые результаты обучения

В результате освоения дисциплины (модуля) «Практические аспекты разработки компиляторов» обучающийся должен обладать следующими компетенциями:

Структурный элемент компетенции	Планируемые результаты обучения
ОПК-2 Способность осваивать методики использования программных средств для решения практических задач	
Знать	– основные принципы работы и устройства компиляторов; – особенности компиляции программ на различных системах, – средства разработки анализаторов
Уметь	– применять средства для разработки лексического анализатора языков программирования высокого уровня; – применять средства для разработки синтаксического анализатора языков программирования высокого уровня; – применять средства для разработки семантического анализатора языков программирования высокого уровня
Владеть	– навыками реализации лексического анализатора для языков программирования высокого уровня; – навыками реализации синтаксического анализатора для языков програм-

Структурный элемент компетенции	Планируемые результаты обучения
	мирования высокого уровня; – навыками реализации семантического анализатора для языков программирования высокого уровня
ПК-1 Способность разрабатывать модели компонентов информационных систем, включая модели баз данных и модели интерфейсов "человек - электронно-вычислительная машина"	
Знать	– виды, структуру и основные задачи компиляторов; – фазы процесса компиляции и их назначение; – основы теории формальных языков и грамматик; – основы использования метаинформации и гипертекста в исходном коде
Уметь	– разрабатывать модель внешнего интерфейса компилятора с учетом принципов, технологий построения компиляторов; – разрабатывать модель внутреннего интерфейса компилятора с учетом принципов, технологий построения компиляторов
Владеть	– навыками реализации модели внешнего интерфейса компилятора; – навыками реализации модели внутреннего интерфейса компилятора

4 Структура и содержание дисциплины (модуля)

Общая трудоемкость дисциплины составляет _3_ зачетных единиц _108_ акад. часов, в том числе:
- контактная работа – _37_ акад. часов:
  - аудиторная – _36_ акад. часов;
  - внеаудиторная – _1_ акад. часов;
- самостоятельная работа – _71_ акад. часов.

Раздел/ тема дисциплины	Семестр	Аудиторная контактная работа (в акад. часах)			Самостоятельная работа (в акад. часах)	Вид самостоятельной работы	Форма текущего контроля успеваемости и промежуточной аттестации	Код и структурный элемент компетенции
		лекции	лаборат. занятия	практич. занятия				
Раздел 1. Компиляторы. Основные задачи и методики создания.	8							
1.1 Основные задачи компиляторов. Отличия интерпретатора от компилятора. Объектная программа.	8	1	–	–	4	1. Самостоятельное изучение учебной и научной литературы. 2. Работа с электронными библиотеками. 3. Подготовка к семинарскому, практическому, лабораторно-практическому занятию.	1. Устный опрос (собеседование). 2. Коллоквиумы.	ОПК-2-зув
1.2 Методики создания компиляторов. Основные фазы процесса трансляции и их назначение. Внешний и внутренний интерфейсы. Просмотры.	8	1	–	–	4	1. Самостоятельное изучение учебной и научной литературы. 2. Работа с электронными библиотеками. 3. Подготовка к семинарскому, практическому, лабораторно-практическому занятию.	1. Устный опрос (собеседование). 2. Коллоквиумы.	ОПК-2-зув

Раздел/ тема дисциплины	Семестр	Аудиторная контактная работа (в акад. часах)			Самостоятельная работа (в акад. часах)	Вид самостоятельной работы	Форма текущего контроля успеваемости и промежуточной аттестации	Код и структурный элемент компетенции
		лекции	лаборат. занятия	практич. занятия				
1.3 Особенности компиляции программ в RadixWare и Flora.	8	2	4/2И	–	12	1. Самостоятельное изучение учебной и научной литературы. 2. Работа с электронными библиотеками. 3. Подготовка к семинарскому, практическому, лабораторно-практическому занятию. 4. Выполнение практических работ (решение задач, письменных работ и т.п.), предусмотренных рабочей программой дисциплины.	1. Устный опрос (собеседование). 2. Коллоквиумы. 3. Лабораторные работы.	ОПК-2-зув
Итого по разделу	**8**	**4**	**4/2И**	**-**	**20**		1. Устный опрос (собеседование). 2. Коллоквиумы. 3. Лабораторные работы.	
Раздел 2. Основы теории формальных языков и грамматик.	8							
2.1 Языки и их представление. Алфавиты, цепочки и языки. Представление языков на примере C++ и F++ (Flora).	8	2	-	-	4	1. Самостоятельное изучение учебной и научной литературы. 2. Работа с электронными библиотеками. 3. Подготовка к семинарскому, практическому, лабораторно-практическому занятию.	1. Устный опрос (собеседование). 2. Коллоквиумы.	ОПК-2-зув

Раздел/ тема дисциплины	Семестр	Аудиторная контактная работа (в акад. часах)			Самостоятельная работа (в акад. часах)	Вид самостоятельной работы	Форма текущего контроля успеваемости и промежуточной аттестации	Код и структурный элемент компетенции
		лекции	лаборат. занятия	практич. занятия				
2.2 Грамматики. Формальное определение грамматики. Типы грамматик и их свойства. Свойства контекстно-свободных грамматик.	8	2	-	-	4	1. Самостоятельное изучение учебной и научной литературы. 2. Работа с электронными библиотеками. 3. Подготовка к семинарскому, практическому, лабораторно-практическому занятию.	1. Устный опрос (собеседование). 2. Коллоквиумы.	ОПК-2-зув
2.3 Использование метаинформации и гипертекста в исходном коде.	8	2	4/2И	-	12	1. Самостоятельное изучение учебной и научной литературы. 2. Работа с электронными библиотеками. 3. Подготовка к семинарскому, практическому, лабораторно-практическому занятию. 4. Выполнение практических работ (решение задач, письменных работ и т.п.), предусмотренных рабочей программой дисциплины.	1. Устный опрос (собеседование). 2. Коллоквиумы. 3. Лабораторные работы.	ОПК-2-зув
Итого по разделу	**8**	**6**	**4/2И**	**-**	**20**		1. Устный опрос (собеседование). 2. Коллоквиумы. 3. Лабораторные работы.	

Раздел/ тема дисциплины	Семестр	Аудиторная контактная работа (в акад. часах)			Самостоятельная работа (в акад. часах)	Вид самостоятельной работы	Форма текущего контроля успеваемости и промежуточной аттестации	Код и структурный элемент компетенции
		лекции	лаборат. занятия	практич. занятия				
Раздел 3. Основные фазы компиляции.	8							
3.1 Лексический анализ. Связь между грамматиками и автоматами. Построение лексического анализатора по регулярному выражению. Способы записи регулярных выражений в Lex-программе. Практическая реализация на примере компилятора Flora.	8	2	2/1И	-	8	1. Самостоятельное изучение учебной и научной литературы. 2. Работа с электронными библиотеками. 3. Подготовка к семинарскому, практическому, лабораторно-практическому занятию. 4. Выполнение практических работ (решение задач, письменных работ и т.п.), предусмотренных рабочей программой дисциплины.	1. Устный опрос (собеседование). 2. Коллоквиумы. 3. Лабораторные работы.	ОПК-2-зув ПК-1-зув
3.2 Синтаксический анализ. Алгоритмы синтаксического анализа для LL(K), LR(K) - грамматик. Промежуточные представления программы: ориентированный граф, синтаксическое дерево разбора, трехадресный код, линеаризованные представления. Практическая реализация на примере компилятора Flora.	8	2	2/1И	-	8	1. Самостоятельное изучение учебной и научной литературы. 2. Работа с электронными библиотеками. 3. Подготовка к семинарскому, практическому, лабораторно-практическому занятию. 4. Выполнение практических работ (решение задач, письменных работ и т.п.), предусмотренных рабочей программой дисциплины.	1. Устный опрос (собеседование). 2. Коллоквиумы. 3. Лабораторные работы.	ОПК-2-зув ПК-1-зув

Раздел/ тема дисциплины	Семестр	Аудиторная контактная работа (в акад. часах)			Самостоятельная работа (в акад. часах)	Вид самостоятельной работы	Форма текущего контроля успеваемости и промежуточной аттестации	Код и структурный элемент компетенции
		лекции	лаборат. занятия	практич. занятия				
3.3 Семантический анализ. Обработка определяющего вхождения идентификатора. Конструирование типов. Представление типов. Контроль типов. Эквивалентность типов. Преобразование типов. Практическая реализация на примере компилятора Flora.	8	2	2/1И	-	8	1. Самостоятельное изучение учебной и научной литературы. 2. Работа с электронными библиотеками. 3. Подготовка к семинарскому, практическому, лабораторно-практическому занятию. 4. Выполнение практических работ (решение задач, письменных работ и т.п.), предусмотренных рабочей программой дисциплины.	1. Устный опрос (собеседование). 2. Коллоквиумы. 3. Лабораторные работы.	ОПК-2-зув ПК-1-зув
3.4 Оптимизация. Виды оптимизации. Примеры. Практическая реализация на примере компилятора Flora.	8	1	2/1И	-	4	1. Самостоятельное изучение учебной и научной литературы. 2. Работа с электронными библиотеками. 3. Подготовка к семинарскому, практическому, лабораторно-практическому занятию. 4. Выполнение практических работ (решение задач, письменных работ и т.п.), предусмотренных рабочей программой дисциплины.	1. Устный опрос (собеседование). 2. Коллоквиумы. 3. Лабораторные работы.	ОПК-2-зув ПК-1-зув
3.5 Генерация кода. Основные задачи, решаемые на этапе подготовки кода к гене-	8	1	2	-	3	1. Самостоятельное изучение учебной и научной литературы.	1. Устный опрос (собеседование).	ОПК-2-зув

Раздел/ тема дисциплины	Семестр	Аудиторная контактная работа (в акад. часах)			Самостоятельная работа (в акад. часах)	Вид самостоятельной работы	Форма текущего контроля успеваемости и промежуточной аттестации	Код и структурный элемент компетенции
		лекции	лаборат. занятия	практич. занятия				
рации. Практическая реализация на примере компилятора Flora.						2. Работа с электронными библиотеками. 3. Подготовка к семинарскому, практическому, лабораторно-практическому занятию. 4. Выполнение практических работ (решение задач, письменных работ и т.п.), предусмотренных рабочей программой дисциплины.	2. Коллоквиумы. 3. Лабораторные работы.	ПК-1-зув
Итого по разделу	**8**	**8**	**10/4И**	**-**	**31**		1. Устный опрос (собеседование). 2. Коллоквиумы. 3. Лабораторные работы.	
Итого по дисциплине		**18**	**18/8И**	**-**	**71**		**Зачет с оценкой**	

И – в том числе, часы, отведенные на работу в интерактивной форме.

5 Образовательные и информационные технологии

1. **Традиционные образовательные технологии,** ориентированные на организацию образовательного процесса и предполагающие прямую трансляцию знаний от преподавателя к студенту.

Формы учебных занятий с использованием традиционных технологий:

Информационная лекция – последовательное изложение материала в дисциплинарной логике, осуществляемое преимущественно вербальными средствами (монолог преподавателя).

Лабораторная работа – организация учебной работы с реальными материальными и информационными объектами, экспериментальная работа с аналоговыми моделями реальных объектов.

2. **Технологии проблемного обучения** – организация образовательного процесса, которая предполагает постановку проблемных вопросов, создание учебных проблемных ситуаций для стимулирования активной познавательной деятельности студентов.

Формы учебных занятий с использованием технологий проблемного обучения:

Практическое занятие в форме практикума – организация учебной работы, направленная на решение комплексной учебно-познавательной задачи, требующей от студента применения как научно-теоретических знаний, так и практических навыков.

6 Учебно-методическое обеспечение самостоятельной работы обучающихся

По дисциплине «Практические аспекты разработки компиляторов» предусмотрена аудиторная и внеаудиторная самостоятельная работа обучающихся.

Аудиторная самостоятельная работа студентов предполагает выполнение работ на лабораторных занятиях. Внеаудиторная самостоятельная работа обучающихся осуществляется в виде изучения литературы по соответствующему разделу с проработкой материала при подготовке к написанию коллоквиума и сдаче зачета с оценкой по данной дисциплине.

Примерные задания к лабораторным занятиям:

Работа №1. Разработка лексического анализатора языка программирования высокого уровня.

Цель работы: создать программу, выполняющую лексический анализ исходного кода программы.

Лексический анализатор, работающий в две стадии: сканирование и оценка. На первой стадии, сканировании, лексический анализатор реализуется в виде конечного автомата, определяемого регулярными выражениями. В нем кодируется информация о возможных последовательностях символов, которые могут встречаться в токенах. Например, токен «целое число» может содержать любую последовательность десятичных цифр.

Пример программы:

```
int a=9, i=10, j=5, b=0xabc, c=01351;
a+=i*j;
float dsf = 45.6;
```

#	Type	Specification	Position	Length	Text
1	opReserved	tInt	1	3	int
2	udIdentificator		5	1	a
3	opBinary	mpSet	6	1	=
4	tInt	DecIntNum	7	1	9
5	lkSpecial	lspComma	8	1	,
6	udIdentificator		9	1	i
7	opBinary	mpSet	10	1	=
8	tInt	DecIntNum	11	2	10
9	lkSpecial	lspComma	13	1	,
10	udIdentificator		14	1	j
11	opBinary	mpSet	15	1	=
12	tInt	DecIntNum	16	1	5
13	lkSpecial	lspComma	17	1	,
14	udIdentificator		19	1	b
15	opBinary	mpSet	20	1	=
16	tInt	HexIntNum	21	5	0xabc
17	lkSpecial	lspComma	26	1	,
18	udIdentificator		27	1	c
19	opBinary	mpSet	28	1	=
20	tInt	OctIntNum	29	5	01351
21	lkSpecial	lspSemiColon	34	1	;
22	udIdentificator		37	1	a
23	opBinary	mpAddSet	38	2	+=
24	udIdentificator		40	1	i
25	opBinary	mpMult	41	1	*
26	udIdentificator		42	1	j
27	lkSpecial	lspSemiColon	43	1	;
28	opReserved	tFloat	46	5	float
29	udIdentificator		52	3	dsf
30	opBinary	mpSet	56	1	=
31	tDouble	floatNum	58	4	45.6
32	lkSpecial	lspSemiColon	62	1	;

Пример работы лексического анализатора

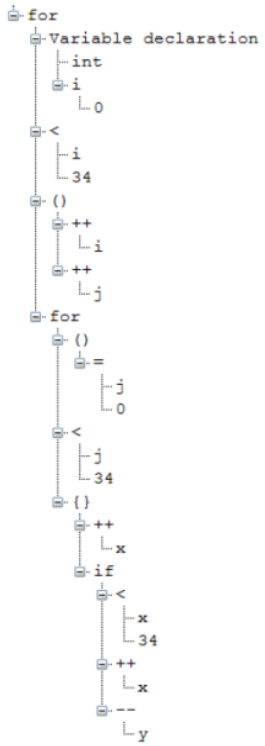
Работа №2. Разработка синтаксического анализатора.

Цель работы: создать программу, выполняющую процесс сопоставления линейной последовательности лексем (слов, токенов) исходного языка программирования с его формальной грамматикой.

Результатом выполнения программы является проверка исходного кода программы на наличие синтаксических ошибок и построение промежуточного представления программы (синтаксическое дерево разбора).

Пример фрагмента программы:

```
for (int i = 0; i < 34; i++, j++)
  for (j = 0; j < 34; )
  {
    x++;
   if (x < 34)
     x++;
   else y--;
  }
```



Пример работы синтаксического анализатора (синтаксическое дерево разбора)

Работа №3. Разработка семантического анализатора.

Цель работы: создать программу, выполняющую процесс соблюдения контекстных условий для исходного языка программирования, предполагающий три типа проверок: обработка описаний, анализ выражений и проверка правильности использования операторов.

Входные данные: таблицы лексем, идентификаторов, внешних представлений и промежуточное представление программы (синтаксическое дерево разбора).

Выходные данные: заключение о семантической правильности программы или о типе обнаруженной семантической ошибке. Видозависимый анализ (type checking), иногда также называемый семантическим анализом (semantic analysis), обычно заключается в проверке правильности типов данных, используемых в программе. Кроме того, на этом этапе компилятор должен также проверить, соблюдаются ли определенные контекстные условия входного языка. В современных языках программирования одним из примеров контекстных условий может служить обязательность описания переменных: для каждого использующего вхождения идентификатора должно существовать единственное определяющее вхождение. Другой пример контекстного условия: число и атрибуты фактических параметров вызова функции должны быть согласованы с определением этой функции.

Примерные индивидуальные задания к лабораторным занятиям:

1. Реализация лексического анализатора языка программирования C++ на основе конечных автоматов.
2. Реализация лексического анализатора языка программирования C# на основе конечных автоматов.
3. Реализация лексического анализатора языка программирования Java на основе конечных автоматов.
4. Реализация лексического анализатора языка программирования Object Pascal на основе конечных автоматов.

5. Реализация лексического анализатора языка программирования Visual Basic на основе конечных автоматов.
6. Реализация лексического и синтаксического анализаторов языка программирования C++ с применением генераторов Flex/Bison.
7. Реализация лексического и синтаксического анализаторов языка программирования Object Pascal с применением генератора Flex/Bison.
8. Реализация лексического и синтаксического анализаторов языка программирования Visual Basic с применением генератора Flex/Bison.
9. Реализация лексического и синтаксического анализаторов языка программирования C++ с применением генератора Coco/R.
10. Реализация лексического и синтаксического анализаторов языка программирования C# с применением генератора Coco/R.
11. Реализация лексического и синтаксического анализаторов языка программирования Java с применением генератора Coco/R.
12. Реализация лексического и синтаксического анализаторов языка программирования Object Pascal с применением генератора Coco/R.
13. Реализация лексического и синтаксического анализаторов языка программирования Visual Basic с применением генератора Coco/R.
14. Реализация лексического и синтаксического анализаторов языка программирования C++ с применением генератора ANTLR.
15. Реализация лексического и синтаксического анализаторов языка программирования C# с применением генератора ANTLR.
16. Реализация лексического и синтаксического анализаторов языка программирования Java с применением генератора ANTLR.
17. Реализация лексического и синтаксического анализаторов языка программирования Object Pascal с применением генератора ANTLR.
18. Реализация лексического и синтаксического анализаторов языка программирования Visual Basic с применением генератора ANTLR.

Тестовые задания

Определите правильные ответы на вопросы, приведенные в таблице.

№	Вопрос	Ответы
1	Какой из перечисленных языков программирования относится к интерпретируемым языкам программирования?	а) Visual Basic; б) C++; в) PHP; г) Pascal; д) C
2	Какой из перечисленных языков программирования транслируется в специальный байт-код, выполняемый виртуальной машиной?	а) Visual Basic; б) Visual Basic for Application; в) Pascal; г) Java; д) C++
3	Перечислите этапы компиляции, которые в большей степени зависят от исходного языка программирования, чем от целевого (*несколько вариантов*)?	а) лексический анализ; б) синтаксический анализ; в) семантический анализ; г) генерация кода; д) оптимизация кода
4	4. Какой тип лексем должен быть исключен из дальнейшей обработки при выполнении лексического анализа?	а) идентификаторы; б) ключевые слова; в) комментарии; г) операторы; д) литералы (константы)

№	Вопрос	Ответы
5	К какому лексическому классу (язык программирования C++) относится лексема “ >>= “?	а) идентификаторы; б) ключевые слова; в) комментарии; г) операторы и пунктуаторы; д) литералы (константы)
6	Этап компиляции, на котором проводится проверка правильности конструкций программы (выражений, описаний, операторов и др.), образованных из лексем?	а) лексический анализ; б) синтаксический анализ; в) семантический анализ; г) генерация кода; д) оптимизация кода
7	При записи выражения (*Y**2+*X*/5)**Z* в виде тетрад/четверок (форма промежуточного представления программы виде трехадресного кода), последней будет тетрада вида?	а) /(*T*3, 5, *T*4); б) +(*T*3, *T*4, *T*5); в) +(*T*2, *T*3, *T*4); г) *(*T*2, *Z*, *T*3); д) *(*T*3, *Z*, *T*4)
8	Последовательности триад/троек (форма промежуточного представления программы виде трехадресного кода) *(10, X); 2) *(X, Y); 3) -((1), (2)); 4) /((3), 2) соответствует выражение?	а) (*X***Y* - 10**X*)/2; б) 2/(10**X*-*X***Y*); в) (10**Y*-*X***X*)/2; г) (10**X*-*X***Y*)/2; д) 2/(10**Y*-*X***X*)
9	Значение *false* предопределяет логическую операцию?	а) конъюнкция; б) дизъюнкция; в) импликация; г) отрицание; д) эквивалентность
10	Этап компиляции, на котором проводится проверка эквивалентности типов данных?	а) лексический анализ; б) синтаксический анализ; в) семантический анализ; г) генерация кода; д) оптимизация кода

7 Оценочные средства для проведения промежуточной аттестации

а) Планируемые результаты обучения и оценочные средства для проведения промежуточной аттестации:

Структурный элемент компетенции	Планируемые результаты обучения	Оценочные средства
ОПК-2 Способность осваивать методики использования программных средств для решения практических задач		
Знать	– основные принципы работы и устройства компиляторов; – особенности компиляции программ на различных системах, – средства разработки анализаторов	*Перечень теоретических вопросов* 1. Основные задачи компиляторов. Отличия интерпретатора от компилятора. Объектная программа. 2. Методики создания компиляторов. Основные фазы процесса трансляции и их назначение. Внешний и внутренний интерфейсы. Просмотры. 3. Особенности компиляции программ в RadixWare и Flora. 4. Языки и их представление. Алфавиты, цепочки и языки. Представление языков на примере C++ и F++ (Flora). 5. Грамматики. Формальное определение грамматики. Типы грамматик и их свойства. Свойства контекстно-свободных грамматик. 6. Использование метаинформации и гипертекста в исходном коде 7. Лексический анализ. Связь между грамматиками и автоматами. Построение лексического анализатора по регулярному выражению. Способы записи регулярных выражений в Lex-программе. Практическая реализация на примере компилятора Flora. 8. Синтаксический анализ. Алгоритмы синтаксического анализа для LL(K), LR(K) - грамматик. Промежуточные представления программы: ориентированный граф, синтаксическое дерево разбора, трехадресный код, линеаризованные представления. Практическая реализация на примере компилятора Flora. 9. Семантический анализ. Обработка определяющего вхождения идентификатора. Конструирование типов. Представление типов. Контроль типов. Эквивалентность типов. Преобразование типов. Практическая реализация на примере компилятора Flora.

Структурный элемент компетенции	Планируемые результаты обучения	Оценочные средства
		10. Оптимизация. Виды оптимизации. Примеры. Практическая реализация на примере компилятора Flora. 11. Генерация кода. Основные задачи, решаемые на этапе подготовки кода к генерации. Практическая реализация на примере компилятора Flora.
Уметь	– применять средства для разработки лексического анализатора языков программирования высокого уровня; – применять средства для разработки синтаксического анализатора языков программирования высокого уровня; – применять средства для разработки семантического анализатора языков программирования высокого уровня	*Практические задания* 1. Реализация диаграммы Вирта и регулярного выражения для проверки принадлежности символов к лексическому классу «целые восьмеричные числа». 2. Реализация диаграммы Вирта и регулярного выражения для проверки принадлежности символов к лексическому классу «целые шестнадцатеричные числа» 3. Реализация диаграммы Вирта и регулярного выражения для проверки принадлежности символов к лексическому классу «целые десятичные числа». 4. Реализация диаграммы Вирта и регулярного выражения для проверки принадлежности символов к лексическому классу «действительные числа». 5. Реализация диаграммы Вирта и регулярного выражения для проверки принадлежности символов к лексическому классу «идентификаторы и ключевые слова». 6. Реализация диаграммы Вирта и регулярного выражения для проверки принадлежности символов к лексическому классу «строковый литерал». 7. Реализация диаграммы Вирта и регулярного выражения для проверки принадлежности символов к лексическому классу «символьный литерал». 8. Реализация диаграммы Вирта и регулярного выражения для проверки принадлежности символов к классу «блочный комментарий».
Владеть	– навыками реализации лексического анализатора для языков программирования высокого уровня; – навыками реализации синтаксического анализатора для языков программирования высокого уровня; – навыками реализации семантического	*Задания на решение задач из профессиональной области, комплексные задания* 1. Программная реализация конечного автомата для проверки принадлежности символов к лексическому классу «целые восьмеричные числа». 2. Программная реализация конечного автомата для проверки принадлежности символов к лексическому классу «целые шестнадцатеричные числа» 3. Программная реализация конечного автомата для проверки принадлежности символов к лексическому классу «целые десятичные числа». 4. Программная реализация конечного автомата проверки принадлежности символов к

Структурный элемент компетенции	Планируемые результаты обучения	Оценочные средства
	анализатора для языков программирования высокого уровня	лексическому классу «действительные числа». 5. Программная реализация конечного автомата для проверки принадлежности символов к лексическому классу «идентификаторы и ключевые слова». 6. Программная реализация конечного автомата для проверки принадлежности символов к лексическому классу «строковый литерал». 7. Программная реализация конечного автомата для проверки принадлежности символов к лексическому классу «символьный литерал». 8. Программная реализация конечного автомата для проверки принадлежности символов к классу «блочный комментарий».
ПК-1 Способность разрабатывать модели компонентов информационных систем, включая модели баз данных и модели интерфейсов "человек - электронно-вычислительная машина"		
Знать	– виды, структуру и основные задачи компиляторов; – фазы процесса компиляции и их назначение; – основы теории формальных языков и грамматик; – основы использования метаинформации и гипертекста в исходном коде	*Перечень теоретических вопросов* 9. Компиляторы и интерпретаторы. Основные задачи компиляторов. Отличия интерпретатора от компилятора. Объектная программа. 10. Т-диаграммы. Методики создания компиляторов. 11. Основные фазы процесса трансляции и их назначение. Примеры. 12. Внешний и внутренний интерфейсы. Просмотры. 13. Внутреннее представление программы на разных этапах трансляции. Структура данных транслятора. Массив лексем, таблица идентификаторов. 14. Формы промежуточного представления программы (синтаксическое дерево, ориентированный ациклический граф и т.д.). 15. Промежуточное представление программы в виде синтаксического дерева. Порядок обхода дерева.
Уметь	– разрабатывать модель внешнего интерфейса компилятора с учетом принципов, технологий построения компиляторов; – разрабатывать модель внутреннего интерфейса компилятора с учетом принципов, технологий построения компиляторов	*Практические задания* 1. Реализация модели внешнего интерфейса компилятора языка C++ с применением генераторов Flex-Bison/CocoR/ANTLR. 2. Реализация модели внешнего интерфейса компилятора языка C# с применением генераторов Flex-Bison/CocoR/ANTLR. 3. Реализация модели внешнего интерфейса компилятора языка Java с применением

Структурный элемент компетенции	Планируемые результаты обучения	Оценочные средства
		генераторов Flex-Bison/CocoR/ANTLR. 4. Реализация модели внешнего интерфейса компилятора языка Object Pascal с применением генераторов Flex-Bison/CocoR/ANTLR. 5. Реализация модели внешнего интерфейса компилятора языка Visual Basic с применением генераторов Flex-Bison/CocoR/ANTLR.
Владеть	– навыками реализации модели внешнего интерфейса компилятора; – навыками реализации модели внутреннего интерфейса компилятора	*Задания на решение задач из профессиональной области, комплексные задания* 1. Реализация спецификации модели внешнего интерфейса компилятора языка C++ с применением генераторов Flex-Bison/CocoR/ANTLR. 2. Реализация спецификации модели внешнего интерфейса компилятора языка C# с применением генераторов Flex-Bison/CocoR/ANTLR. 3. Реализация спецификации модели внешнего интерфейса компилятора языка Java с применением генераторов Flex-Bison/CocoR/ANTLR. 4. Реализация спецификации модели внешнего интерфейса компилятора языка Object Pascal с применением генераторов Flex-Bison/CocoR/ANTLR. 5. Реализация спецификации модели внешнего интерфейса компилятора языка Visual Basic с применением генераторов Flex-Bison/CocoR/ANTLR.

б) Порядок проведения промежуточной аттестации, показатели и критерии оценивания:

Промежуточная аттестация по дисциплине «Практические аспекты разработки компиляторов» включает теоретические вопросы, позволяющие оценить уровень усвоения обучающимися знаний, и практические задания, выявляющие степень сформированности умений и владений, проводится в форме зачета с оценкой.

Зачет с оценкой по дисциплине проводится по результатам отчетности на лабораторных занятиях с опросом в устной форме по этапам выполнения и результатам коллоквиума, проводимого по материалам лекционных занятий.

Показатели и критерии оценивания зачета с оценкой:

– на оценку **«отлично»** (5 баллов) – обучающийся демонстрирует высокий уровень сформированности компетенций, всестороннее, систематическое и глубокое знание учебного материала, свободно выполняет практические задания, свободно оперирует знаниями, умениями, применяет их в ситуациях повышенной сложности.

– на оценку **«хорошо»** (4 балла) – обучающийся демонстрирует средний уровень сформированности компетенций: основные знания, умения освоены, но допускаются незначительные ошибки, неточности, затруднения при аналитических операциях, переносе знаний и умений на новые, нестандартные ситуации.

– на оценку **«удовлетворительно»** (3 балла) – обучающийся демонстрирует пороговый уровень сформированности компетенций: в ходе контрольных мероприятий допускаются ошибки, проявляется отсутствие отдельных знаний, умений, навыков, обучающийся испытывает значительные затруднения при оперировании знаниями и умениями при их переносе на новые ситуации.

– на оценку **«неудовлетворительно»** (2 балла) – обучающийся демонстрирует знания не более 20% теоретического материала, допускает существенные ошибки, не может показать интеллектуальные навыки решения простых задач.

– на оценку **«неудовлетворительно»** (1 балл) – обучающийся не может показать знания на уровне воспроизведения и объяснения информации, не может показать интеллектуальные навыки решения простых задач.

8 Учебно-методическое и информационное обеспечение дисциплины (модуля)

а) Основная литература:

1. Ахо, А.В. Компиляторы: принципы, технологии, инструменты – (Dragon Book-2) [Текст] / Альфред В. Ахо, Моника С. Лам, Рави Сети, Джеффри Д. Ульман. –2-е изд.,– «ВИЛЬЯМС», 2011. – 1184 с.– ISBN 978-5-8459-1349-4.
2. Орлов, С. Теория и практика языков программирования. Учебник для вузов. Стандарт 3-го поколения. [Текст] / С.Орлов. – СПб.: Питер, 2014 г. – 688 с. – Электронное издание. – РАН. – ISBN 978-5-496-00032-1 http://ibooks.ru/reading.php?productid=26402 .

б) Дополнительная литература:

1. Бруно, К.Л. LLVM: инфраструктура для разработки компиляторов. [Электронный ресурс] / К.Л. Бруно, А. Рафаэль. — Электрон. дан. — М.: ДМК Пресс, 2015. – 342 с. – Электронное издание. – ISBN 978-5-97060-305-5. http://e.lanbook.com/book/90119 .
2. Вирт Н. Построение компиляторов [Текст] / Вирт Н. – М. : ДМК Пресс, 2010 г. – 192 с. – Электронное издание. – ISBN 978-5-94074-585-3. http://ibooks.ru/product.php?productid=22485 .
3. Гагарина, Л. Г. Введение в теорию алгоритмических языков и компиляторов [Текст] : учеб. пособие / Гагарина Л.Г., Кокорева Е.В. – М.: ИД ФОРУМ, 2011. – 176 с.: ил. ISBN 978-5-8199-0404-6. http://znanium.com/bookread.php?book=265617 .

в) Методические указания:

1. Кирпичев А.А. Создание и использование контейнерных классов в современных языках программирования [Текст]: учебное пособие / А.А.Кирпичев, Н.Т.Кирпичева, В.Е. Торчинский. – Магнитогорск : МГТУ, 2000.– 65 с. ISBN 5-89514-143-9

г) Программное обеспечение и Интернет-ресурсы:

Программное обеспечение: лицензионное программное обеспечение: операционная система MS Windows 2007; MS Office 2010; PacketTracerre, установленные на каждом персональном компьютере вычислительного центра ФГБОУ ВПО «МГТУ».

Перечень лицензионного программного обеспечения по ссылке:

http://sps.vuz.magtu.ru/Shared%20Documents/Forms/AllItems.aspx?RootFolder=%2FShared%20Documents%2F%D0%9F%D0%BE%D0%B4%D0%B3%D0%BE%D1%82%D0%BE%D0%B2%D0%BA%D0%B0%20%D0%BA%20%D0%B0%D0%BA%D0%BA%D1%80%D0%B5%D0%B4%D0%B8%D1%82%D0%B0%D1%86%D0%B8%D0%B8%202020%2F%D0%A1%D0%B0%D0%BC%D0%BE%D0%BE%D0%B1%D1%81%D0%BB%D0%B5%D0%B4%D0%BE%D0%B2%D0%B0%D0%BD%D0%B8%D0%B5%202019%D0%B3%2F%D0%9B%D0%B8%D1%86%D0%B5%D0%BD%D0%B7%D0%B8%D0%BE%D0%BD%D0%BD%D0%BE%D0%B5%20%D0%9F%D0%9E&InitialTabId=Ribbon.Document&VisibilityContext=WSSTabPersistence

Официальные сайты промышленных предприятий и организаций: http://www.mmk.ru , http://www.magtu.ru , и т.п.; разработчиков программных продуктов: http://www.statsoft.ru , http://www.microsoft.com , http://www.netacad.com и т.п.

9 Материально-техническое обеспечение дисциплины (модуля)

Материально-техническое обеспечение дисциплины включает:

Тип и название аудитории	Оснащение аудитории
Лекционная аудитория ауд. 282	Мультимедийные средства хранения, передачи и представления информации
Компьютерные классы Центра информационных технологий ФГБОУ ВО «МГТУ»	Персональные компьютеры, объединенные в локальные сети с выходом в Internet, оснащенные современными программно-методическими комплексами для решения задач в области информатики и вычислительной техники
Аудитории для самостоятельной работы: компьютерные классы; читальные залы библиотеки	Все классы УИТ и АСУ с персональными компьютерами, выходом в Интернет и с доступом в электронную информационно-образовательную среду университета
Аудиторий для групповых и индивидуальных консультаций, текущего контроля и промежуточной аттестации	Ауд. 282 и классы УИТ и АСУ
Помещения для самостоятельной работы обучающихся, оснащенных компьютерной техникой с возможностью подключения к сети «Интернет» и наличием доступа в электронную информационно-образовательную среду организации	Классы УИТ и АСУ
Помещения для хранения и профилактического обслуживания учебного оборудования	Центр информационных технологий – ауд. 379