МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ
РОССИЙСКОЙ ФЕДЕРАЦИИ

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

УТВЕРЖДАЮ
Директор ИЭиАС
_____ В.Р. Храмшин

13.02.2024 г.

РАБОЧАЯ ПРОГРАММА ДИСЦИПЛИНЫ (МОДУЛЯ)

ТЕОРИЯ ЯЗЫКОВ ПРОГРАММИРОВАНИЯ

Направление подготовки (специальность)
09.03.01 Информатика и вычислительная техника

Направленность (профиль/специализация) программы
Программное обеспечение средств вычислительной техники и автоматизированных систем

Уровень высшего образования - бакалавриат

Форма обучения
очная

Институт/ факультет	Институт энергетики и автоматизированных систем
Кафедра	Вычислительной техники и программирования
Курс	4
Семестр	8

Магнитогорск
2024 год

Рабочая программа составлена на основе ФГОС ВО - бакалавриат по направлению подготовки 09.03.01 Информатика и вычислительная техника (приказ Минобрнауки России от 19.09.2017 г. № 929)

Рабочая программа рассмотрена и одобрена на заседании кафедры Вычислительной техники и программирования
25.01.2024, протокол № 5

Зав. кафедрой _____ О.С. Логунова

Рабочая программа одобрена методической комиссией ИЭиАС
13.02.2024 г. протокол № 4

Председатель _____ В.Р. Храмшин

Рабочая программа составлена:
доцент кафедры ВТиП, канд. техн. наук _____ А.Н. Калитаев

Рецензент:
директор НИИ «Промбезопасность» , д-р техн. наук _____ М.Ю. Наркевич

Лист актуализации рабочей программы

Рабочая программа пересмотрена, обсуждена и одобрена для реализации в 2025 - 2026 учебном году на заседании кафедры Вычислительной техники и программирования

Протокол от __ _____ 20__ г. № __
Зав. кафедрой _____ О.С. Логунова

Рабочая программа пересмотрена, обсуждена и одобрена для реализации в 2026 - 2027 учебном году на заседании кафедры Вычислительной техники и программирования

Протокол от __ _____ 20__ г. № __
Зав. кафедрой _____ О.С. Логунова

Рабочая программа пересмотрена, обсуждена и одобрена для реализации в 2027 - 2028 учебном году на заседании кафедры Вычислительной техники и программирования

Протокол от __ _____ 20__ г. № __
Зав. кафедрой _____ О.С. Логунова

Рабочая программа пересмотрена, обсуждена и одобрена для реализации в 2028 - 2029 учебном году на заседании кафедры Вычислительной техники и программирования

Протокол от __ _____ 20__ г. № __
Зав. кафедрой _____ О.С. Логунова

1 Цели освоения дисциплины (модуля)

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

Для достижения поставленной цели в курсе «Теория языков программирования» 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

Дисциплина Теория языков программирования входит в часть учебного плана формируемую участниками образовательных отношений образовательной программы.

Для изучения дисциплины необходимы знания (умения, владения), сформированные в результате изучения дисциплин/ практик:

Программирование

Структуры и модели данных

Алгоритмы и теория сложности

Объектно-ориентированное программирование

Знания (умения, владения), полученные при изучении данной дисциплины будут необходимы для изучения дисциплин/практик:

Выполнение и защита выпускной квалификационной работы

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

В результате освоения дисциплины (модуля) «Теория языков программирования» обучающийся должен обладать следующими компетенциями:

Код индикатора	Индикатор достижения компетенции
ПК-5 Обладает способностью к разработке компиляторов, загрузчиков, сборщиков	
ПК-5.1	Определяет целесообразность разработки компиляторов, загрузчиков, сборщиков
ПК-5.2	Оценивает работоспособность компиляторов, загрузчиков, сборщиков

4. Структура, объём и содержание дисциплины (модуля)

Общая трудоемкость дисциплины составляет 3 зачетных единиц 108 акад. часов, в том числе:

– контактная работа – 51,5 акад. часов:

– аудиторная – 48 акад. часов;

– внеаудиторная – 3,5 акад. часов;

– самостоятельная работа – 20,8 акад. часов;

– в форме практической подготовки – 0 акад. час;

– подготовка к экзамену – 35,7 акад. час

Форма аттестации - экзамен

Раздел/ тема дисциплины	Семестр	Аудиторная контактная работа (в акад. часах)			Самостоятельная работа студента	Вид самостоятельной работы	Форма текущего контроля успеваемости и промежуточной аттестации	Код компетенции
		Лек.	лаб. зан.	практ. зан.				
1. Основы теории формальных языков и грамматик								
1.1 Основные задачи компиляторов. Отличия интерпретатора от компилятора. Объектная программа. Т-диаграммы.	8	1			1	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. Коллоквиум.	ПК-5.1, ПК-5.2
1.2 Методики создания компиляторов. Основные фазы процесса трансляции и их назначение. Внешний и внутренний ин-терфейсы. Просмотры.		1			1	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. Коллоквиум.	ПК-5.1, ПК-5.2

1.3 Языки и их представление. Алфавиты, цепочки и языки. Представление языков.		1			1	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. Коллоквиум.	ПК-5.1, ПК-5.2
1.4 Грамматики. Формальное определение грамматики. Типы грамматик и их свойства. Свойства контекстно-свободных грамматик.		1			1	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. Коллоквиум.	ПК-5.1, ПК-5.2
Итого по разделу		4			4			
2. Распознаватели и преобразователи: конечные автоматы и преобразователи, автоматы и преобразователи с магазинной памятью								
2.1 Распознаватели для различных классов грамматик. Конечные автоматы (детерминированные и недетерминированные).	8	1	2		1	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. Коллоквиум. 3. Лабораторные работы.	ПК-5.1, ПК-5.2

2.2 Эквивалентность конечных автоматов. Эквивалентность праволинейных грамматик и конечных автоматов. Магазинные автоматы. Алгоритмы построения конечных автоматов.		2	2		1	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. Коллоквиум. 3. Лабораторные работы.	ПК-5.1, ПК-5.2
Итого по разделу		3	4		2			
3. Лексический анализ. Связь между грамматиками и автоматами								
3.1 Машины Тьюринга. Связь машин Тьюринга и грамматик типа 0.	8	1	1		1	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. Коллоквиум. 3. Лабораторные работы.	ПК-5.1, ПК-5.2

3.2 Линейно-ограниченные автоматы и их связь с контекстно-зависимыми грамма-тиками. Связь регулярных множеств, конечных автоматов и регулярных грамматик.		2	2		1	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. Коллоквиум. 3. Лабораторные работы.	ПК-5.1, ПК-5.2
3.3 Лексический анализ различных языков программирования. Таблица представлений. Использование грамматик для лексического анализа.		1	1		1	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	ПК-5.1, ПК-5.2

3.4 Регулярные выражения. Построение лексического анализатора по регулярному выражению. Способы записи регулярных выражений в Lex-программе.		1	1		1	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. Коллоквиум. 3. Лабораторные работы.	ПК-5.1, ПК-5.2
Итого по разделу		5	5		4			
4. Синтаксический анализ. Алгоритмы синтаксического анализа для LL(K)-грамматик, LR(K)-грамматик, грамматик предшествования								
4.1 Контекстно-свободные грамматики и автоматы с магазинной памятью. Преобразования КС-грамматик.	8	2	4		1	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. Коллоквиум. 3. Лабораторные работы.	ПК-5.1, ПК-5.2

4.2 Алгоритм Кока-Янгера-Касами. Алгоритм разбора сверху-вниз. LL(1)-грамматики. LL(K)-грамматики. Разбор снизу-вверх. LR(1)-анализаторы. LR(K)-грамматики		2	4		1	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. Коллоквиум. 3. Лабораторные работы.	ПК-5.1, ПК-5.2
Итого по разделу		4	8		2			
5. Формальные методы описания перевода: СУ-схема, транслирующие грамматики, атрибутные транслирующие грамматики								
5.1 Преобразователи с магазинной памятью. Синтаксически управляемый перевод. Атрибутные грамматики. Классы атрибутных грамматик и их реализация. Язык описания атрибутных грамматик.	8	1	1		1	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. Коллоквиум. 3. Лабораторные работы	ПК-5.1, ПК-5.2

5.2 Промежуточные представления про-граммы. Представление в виде ориентированного графа. Синтаксическое дерево разбора.		1	1		1	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. Коллоквиум. 3. Лабораторные работы.	ПК-5.1, ПК-5.2
5.3 Трехадресный код (тройки, четверки, косвенные тройки). Линеаризованные представления.		1			1	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. Коллоквиум. 3. Лабораторные работы.	ПК-5.1, ПК-5.2

5.4 Организация таблиц символов. Таблица идентификаторов. Таблица внешних представлений.		1	1		1,8	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. Коллоквиум. 3. Лабораторные работы.	ПК-5.1, ПК-5.2
Итого по разделу		4	3		4,8			
6. Включение семантики в алгоритмы синтаксического анализа								
6.1 Семантический анализ. Обработка определяющего вхождения идентификатора	8	2	2		2	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. Коллоквиум. 3. Лабораторные работы.	ПК-5.1, ПК-5.2

6.2 Конструирование типов. Представление типов. Контроль типов. Эквивалентность типов. Преобразование типов		2	2		2	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. Коллоквиум. 3. Лабораторные работы.	ПК-5.1, ПК-5.2
Итого по разделу		4	4		4			
Итого за семестр		24	24		20,8		экзамен	
Итого по дисциплине		24	24		20,8		экзамен	

5 Образовательные 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

Представлено в приложении 1.

7 Оценочные средства для проведения промежуточной аттестации

Представлены в приложении 2.

8 Учебно-методическое и информационное обеспечение дисциплины (модуля)

а) Основная литература:

1. Гагарина, Л. Г. Введение в теорию алгоритмических языков и компиляторов: учеб. пособие / Л.Г. Гагарина, Е.В. Кокорева. - Москва : ИД ФОРУМ, 2011. - 176 с.: ил.; . - (Высшее образование). ISBN 978-5-8199-0404-6. - Текст : электронный. - URL: https://znanium.com/catalog/product/265617 (дата обращения: 29.10.2020). – Режим доступа: по подписке.

2. Малявко, А. А. Формальные языки и компиляторы / А. А. Малявко. - Новосибирск : НГТУ, 2014. - 431 с. - SBN 978-5-7782-2318-9. - ISBN 978-5-7782-2318-9. - Текст : электронный. - URL: https://znanium.com/catalog/product/548152 (дата обращения: 29.10.2020). – Режим доступа: по подписке.

б) Дополнительная литература:

1. Гуськова, О.И. Объектно ориентированное программирование в Java : учебное пособие / О. И. Гуськова. - Москва : МПГУ, 2018. - 240 с. - ISBN 978-5-4263-0648-6. - Текст : электронный. - URL: https://znanium.com/catalog/product/1020593 (дата обращения: 29.10.2020). – Режим доступа: по подписке.

2. Практикум по программированию на языке С++ : учебное пособие / В. Е. Торчинский, А. Н. Калитаев, В. Д. Тутарова, Ю. В. Федосеева ; МГТУ. - Магнитогорск : МГТУ, 2017. - 1 электрон. опт. диск (CD-ROM). - Загл. с титул. экрана. - URL: https://magtu.informsystema.ru/uploader/fileUpload?name=3004.pdf&show=dcatalogues/1/1134950/3004.pdf&view=true(дата обращения: 23.10.2020). - Макрообъект. - Текст : электронный. - Сведения доступны также на CD-ROM.

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

Наименование ПО	№ договора	Срок действия лицензии
Borland Turbo C++	№112301 от 23.11.2005	бессрочно
Eclipse	свободно распространяемое ПО	бессрочно
JetBrains IDEA Community Edition	свободно распространяемое ПО	бессрочно
Borland Turbo Delphi	№112301 от 23.11.2005	бессрочно
MS Visual Studio 2017 Community Edition	свободно распространяемое ПО	бессрочно
NetBeans	свободно распространяемое ПО	бессрочно

Профессиональные базы данных и информационные справочные системы

Название курса	Ссылка
Федеральное государственное бюджетное учреждение «Федеральный институт промышленной собственности»	URL: http://www1.fips.ru/

9 Материально-техническое обеспечение дисциплины (модуля)

Материально-техническое обеспечение дисциплины включает:

1. Лекционная аудитория ауд. 282. Мультимедийные средства хранения, передачи и представления информации.

2. 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.

3. 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.

4. Аудиторий для групповых и индивидуальных консультаций, текущего контроля и промежуточной аттестации. Ауд. 282 и классы УИТ и АСУ.

5. 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.

6. Помещения для хранения и профилактического обслуживания учебного оборудования. Центр информационных технологий – ауд. 372.

Приложение 1

Учебно-методическое обеспечение самостоятельной работы обучающихся

По дисциплине «Теория языков программирования» 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 экзамена 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

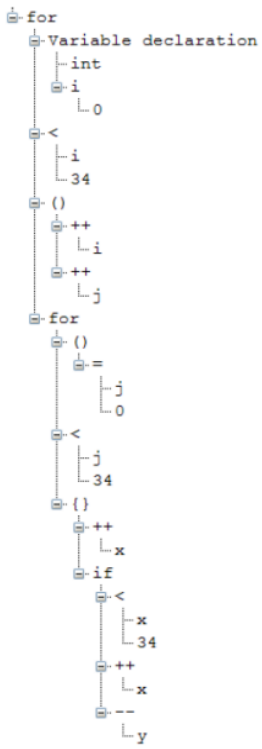
Работа №2. Разработка синтаксического анализатора.

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)
5	К какому лексическому классу (язык	а) идентификаторы;

№	Вопрос	Ответы
	программирования C++) относится лексема “ >>= “?	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

Приложение 2

Оценочные средства для проведения промежуточной аттестации

а) Оценочные средства для проведения промежуточной аттестации:

Код индикатора	Индикатор достижения компетенции	Оценочные средства
ПК-5: Обладает способностью к разработке компиляторов, загрузчиков, сборщиков		
ПК-5.1	Определяет целесообразность разработки компиляторов, загрузчиков, сборщиков	*Перечень теоретических вопросов* 1. Компиляторы и интерпретаторы. Основные задачи компиляторов. Отличия интерпретатора от компилятора. Объектная программа. 2. Т-диаграммы. Методики создания компиляторов. 3. Основные фазы процесса трансляции и их назначение. Примеры. 4. Внешний и внутренний интерфейсы. Просмотры. 5. Лексический анализ. Основные задачи. Пример конечного автомата, описывающий множество ключевых слов (if-int-“идентификатор”). 6. Для чего нужен лексический анализатор? Что порождает лексический анализатор? Структура лексем. 7. Транслитератор DPL. Общая организация транслитератора (виды лексических классов) и программная реализация (пример проверки принадлежности символа к лексическому классу). 8. Непрямой лексический анализатор DPL. Общая структура (объединение диаграмм Вирта). Пример диаграммы Вирта проверки принадлежности идентификатора к лексическому классу «ключевые слова». 9. Прямой лексический анализатор DPL. Общая структура (объединение диаграмм Вирта). Пример диаграммы Вирта проверки принадлежности символов к лексическому классу «вещественные числа». 10. Перечислите конструкции конкретного языка программирования, которые целесообразно распознать на фазе лексического анализа. 11. Синтаксический анализ. Основные задачи. Что является результатом синтаксического разбора? Классы синтаксических анализаторов. 12. Какие существуют методы разбора? Особенности нисходящего разбора. Метод рекурсивного спуска.

Код индикатора	Индикатор достижения компетенции	Оценочные средства
		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. 16. Семантический анализ. Основные задачи. Назначение таблицы идентификаторов и таблицы внешних представлений. Обработка определяющего вхождения идентификатора. 17. Семантический анализ. Конструирование типов. Представление типов. 18. Семантический анализ. Контроль типов. Эквивалентность типов. Преобразование типов. *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

Код индикатора	Индикатор достижения компетенции	Оценочные средства
		принадлежности символов к лексическому классу «символьный литерал». 8. Реализация диаграммы Вирта и регулярного выражения для проверки принадлежности символов к классу «блочный комментарий».
ПК-5.2	Оценивает работоспособность компиляторов, загрузчиков, сборщиков	*Перечень теоретических вопросов* 1. Оптимизация. Основные задачи оптимизации. Виды оптимизации. Зависимость между оптимизациями. Стадии оптимизации. 2. Оптимизация. Примеры: удаление пустого оператора, удаление мертвого кода, чистка циклов вверх, объединение и раскрутка циклов, понижение силы операций, упрощение выражений, экономия общих подвыражений. 3. Оптимизация. Зависимость качества оптимизации от размера участка экономии. Понятия: локальная, квазилокальная и глобальная оптимизации. 4. Генерация кода. Основные задачи генерации. *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 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».

Код индикатора	Индикатор достижения компетенции	Оценочные средства
		8. Программная реализация конечного автомата для проверки принадлежности символов к классу «блочный комментарий». *Тестовые задания:* 1. Какой из перечисленных языков программирования транслируется в специальный байт-код, выполняемый виртуальной машиной? а) Visual Basic б) Pascal в) C++ **г) Java** 2. Какой тип лексем должен быть исключен из дальнейшей обработки при выполнении лексического анализа? а) идентификаторы б) ключевые слова в) операторы г) литералы (константы) **д) комментарии** 3. Этап компиляции, на котором выполняется проверка эквивалентности типов данных? а) лексический анализ б) синтаксический анализ **в) семантический анализ** г) генерация кода д) оптимизация кода

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

Промежуточная аттестация по дисциплине «Теория языков программирования» включает теоретические вопросы, позволяющие оценить уровень усвоения обучающимися знаний, и практические задания, выявляющие степень сформированности умений и владений, проводится в форме экзамена.

Экзамен по данной дисциплине проводится в устной форме по экзаменационным билетам, каждый из которых включает 2 теоретических вопроса и одно практическое задание.

Показатели и критерии оценивания экзамена:

– 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.