

Государственное образовательное учреждение  
«Приднестровский государственный университет им. Т.Г. Шевченко»

Инженерно-технический институт

Кафедра программного обеспечения вычислительной техники  
и автоматизированных систем

УТВЕРЖДАЮ

Директор института, доцент



О.Ю. Бурменко

«16» 09 2019 г.



# РАБОЧАЯ ПРОГРАММА

на 2019/2020 учебный год

УЧЕБНОЙ ДИСЦИПЛИНЫ

**Б1.В.06 «КОНСТРУИРОВАНИЕ ТРАНСЛЯТОРОВ,  
ИНТЕРПРЕТАТОРОВ»**

Направление подготовки:

**2.09.04.04 Программная инженерия**

Профиль

**Разработка программно-информационных систем**

Для набора

**2018 года**

Квалификация (степень) выпускника

**магистр**

Форма обучения:

**очная**

Тирасполь, 2019

Рабочая программа дисциплины «Конструирование трансляторов, интерпретаторов»  
/сост. С.В. Помян – Тирасполь: ГОУ ИГУ, 2019. – 15 с.

Рабочая программа предназначена для преподавания дисциплины вариативной части дисциплин (модулей) обучающимся очной формы обучения по направлению подготовки 2.09.04.04 «ПРОГРАММНАЯ ИНЖЕНЕРИЯ»

Рабочая программа составлена с учетом Федерального Государственного образовательного стандарта высшего образования по направлению подготовки 09.04.04 «ПРОГРАММНАЯ ИНЖЕНЕРИЯ», утвержденного приказом Министерства образования и науки Российской Федерации от 30 октября 2014 г. № 1406.

Составитель  /С.В. Помян, доцент

«30» августа 2019 г.

## 1. Цели и задачи освоения дисциплины

**Цели дисциплины:** освоение основ конструирования трансляторов, интерпретаторов, формирование основных концепций теории реализации трансляторов, интерпретаторов; освоение новых парадигм программирования, связанных с разработкой метаязыка, позволяющих программировать специализированные задачи для конкретной области применения, основанных на методах разработки трансляторов.

### **Задачи дисциплины:**

- формирование знаний об общих принципах организации процесса трансляции и структуры трансляторов;
- изучение основ построения трансляторов;
- практическое закрепление полученных теоретических знаний на примере разработки транслятора для простого языка;
- освоение новых парадигм программирования, связанных с разработкой метаязыка.

## 2. Место дисциплины в структуре ООП ВО

Б1.В.06. Дисциплины (модули). Вариативная часть.

Трудоемкость 6 зачетных единицы, 216 часов.

Для освоения дисциплины «Конструирование трансляторов, интерпретаторов» обучающиеся используют знания, умения, навыки, способы деятельности и установки, полученные и сформированные в ходе изучения дисциплин «Программирование», «Теория автоматов и формальных языков».

Изучение дисциплины «Конструирование трансляторов, компиляторов» является базой для дальнейшего освоения обучающимися дисциплин направления «Программная инженерия», курсов по выбору, для прохождения практики.

Знания, умения и навыки, приобретенные при изучении данной дисциплины, могут быть использованы при написании выпускной квалификационной работы и в профессиональной деятельности по профилю «Разработка программно-информационных систем».

## 3. Требования к результатам освоения дисциплины

Изучение дисциплины направлено на формирование следующих компетенций: ПК-9, ПК-12, ПК-16. Расшифровка компетенций дана в следующей таблице.

Код компетенции	Формулировка компетенций
ПК-9	способность проектировать трансляторы и интерпретаторы языков программирования;
ПК-12	способность проектировать вспомогательные и специализированные языки программирования и языки представления данных;
ПК-16	владение навыками создания трансляторов и интерпретаторов языков программирования.

В результате изучения курса обучающийся должен

### **3.1 Знать:**

- алгоритмы и методы построения лексических анализаторов;
- алгоритмы и методы построения возвратных и однопроходных синтаксических анализаторов;

- методы статического семантического анализа, основанные на атрибутивных грамматиках и атрибутивных трансляциях;
- алгоритмы и методы генерации промежуточного представления программы, основанные на атрибутивных грамматиках и атрибутивных трансляциях.

### 3.2 Уметь:

- разрабатывать лексические анализаторы;
- разрабатывать как однопроходные, так и многопроходные синтаксические анализаторы;
- применять методы обработки и нейтрализации синтаксических ошибок;
- применять методы статического семантического анализа для проверки типов и контекстно-зависимых условий;
- разрабатывать генераторы промежуточного представления программы, основанные на атрибутивных грамматиках и атрибутивных трансляциях.

### 3.3 Владеть:

- методами построения распознавателей;
- навыками конструирования трансляторов.

Рабочая программа учебной дисциплины рассчитана на 46 часов аудиторных занятий, в том числе 16 часов отводится на лекционные занятия, 30 часов – на лабораторные занятия.

С целью систематизации и закрепления полученных теоретических знаний и практических умений в рабочей программе учебной дисциплины предусмотрена самостоятельная работа обучающихся. Внеаудиторная самостоятельная работа обучающихся – 134 часов. На экзамен – 36 часов.

Учебная дисциплина изучается один семестр и заканчивается промежуточной аттестацией обучающихся в форме экзамена.

## 4. Структура и содержание дисциплины (модуля)

### 4.1. Распределение трудоемкости в з.е./часах по видам аудиторной и самостоятельной работы студента по семестрам

Семестр	Трудоемкость з.е./часы	Количество часов				Самост. работа	Экзамен	Форма итогового контроля
		Всего	В том числе Аудиторных					
			Лекции	Лаб. раб.	Практич. занятия			
2	6/216	46	16	30	-	134	36	экзамен
<b>Итого</b>	6/216	46	16	30	-	134	36	экзамен

#### 4.2 Распределение видов учебной работы и их трудоемкости по разделам дисциплины:

дисциплины:

№ раздела	Наименование раздела дисциплины (модуля)	Всего	Аудиторная работа			Внеаудиторная работа СР
			Л	ЛЗ	ЛР	
1	Место трансляторов в системном программном обеспечении	16	2	-	-	14
2	Распознаватели и преобразователи	58	4	-	12	42
3	Распределение памяти как фаза трансляции, вызываемая генератором кода	58	6	-	10	42
4	Генерация промежуточного кода	48	4	-	8	36
		<b>180</b>	<b>16</b>	<b>-</b>	<b>30</b>	<b>134</b>
	Подготовка к экзамену	<b>36</b>	-	-	-	-
	<b>Итого:</b>	<b>216</b>	-	-	-	-

#### 4.3 Тематический план по видам учебной деятельности

Лекции				
№ п/п	Номер раздела дисциплины	Объем часов	Тема лекции	Учебно-наглядные пособия
1	1	2	Место трансляторов в системном программном обеспечении. Основы теории формальных языков и грамматик	Слайды презентации
2	2	2	Распознаватели и преобразователи	
3	2	2	Нисходящий синтаксический анализ. Восходящий синтаксический анализ	
4	3	2	Не-контекстно-свободные характеристики языков. Распределение памяти как фаза трансляции, вызываемая генератором кода	Слайды презентации
5	3	2	Память: глобальная и локальная, статическая и динамическая	Слайды презентации
6	3	2	Управление динамической памятью. Стек, куча	
7	4	2	Генерация промежуточного кода.	
8	4	2	Нормализация грамматик. Представление выводов. Построение выводов.	
	<b>Итого:</b>	<b>16</b>		

#### Практические (семинарские) занятия

Учебным планом не предусмотрены.

#### Лабораторные работы

№ п/п	Номер раздела дисциплины	Объем часов	Тема лабораторного занятия	Учебно-наглядные пособия
1	2	2	Распознаватели и преобразователи. Проектирование распознавателей	Индивидуальный вариант кейс-задачи

№ п/п	Номер раздела дисциплины	Объем часов	Тема лабораторного занятия	Учебно-наглядные пособия
2	2	2	Нисходящий распознаватель с возвратами.	Индивидуальный вариант кейс-задачи
3	2	2	Проектирование и реализация распознавателя	Индивидуальный вариант кейс-задачи
4	2	2	Нисходящий распознаватель без возвратов.	Индивидуальный вариант кейс-задачи
5	2	2	Проектирование и реализация распознавателя	Индивидуальный вариант кейс-задачи
6	2	2	Проектирование и реализация распознавателя	Индивидуальный вариант кейс-задачи
7	3	2	Распределение памяти как фаза трансляции, вызываемая генератором кода	Индивидуальный вариант кейс-задачи
8	3	2	Управление динамической памятью. Стек, куча	Индивидуальный вариант кейс-задачи
9	3	2	Управление динамической памятью. Стек, куча	Индивидуальный вариант кейс-задачи
10	3	2	Методы управления кучей: сборка мусора и использование счетчиков ссылок	Индивидуальный вариант кейс-задачи
11	3	2	Методы управления кучей: сборка мусора и использование счетчиков ссылок	Индивидуальный вариант кейс-задачи
12	4	2	Реализация компилятора метаязыка	Индивидуальный вариант кейс-задачи
13	4	2	Реализация компилятора метаязыка	Индивидуальный вариант кейс-задачи
14	4	2	Реализация компилятора метаязыка	Индивидуальный вариант кейс-задачи
15	4	2	Реализация компилятора метаязыка	Индивидуальный вариант кейс-задачи
<b>Итого:</b>		<b>30</b>		

#### Самостоятельная работа студента

Раздел дисциплины	№ п/п	Тема и вид СРС	Трудоёмкость (в часах)
Раздел 1	1	Тема: Место трансляторов в системном программном обеспечении СРС №1:- работа обучающихся с лекционным материалом и раздаточными материалами, - поиск и анализ литературы и электронных источников информации.	6
	2	Тема: Компиляторы и интерпретаторы. Основы теории формальных языков и грамматик СРС №2:- работа обучающихся с лекционным материалом и раздаточными материалами, - поиск и анализ литературы и электронных источников информации.	8
Раздел 2	3	Тема: Распознаватели и преобразователи СРС №3:- работа обучающихся с лекционным материалом и раздаточными материалами.	6

Раздел дисциплины	№ п/п	Тема и вид СРС	Трудоёмкость (в часах)
		<ul style="list-style-type: none"> <li>- поиск и анализ литературы и электронных источников информации.</li> <li>- подготовка презентации по результатам поиска и анализа литературных и электронных источников.</li> </ul>	
	4	<p>Тема: Восходящий синтаксический анализ</p> <p>СРС №4:- работа обучающихся с лекционным материалом и раздаточными материалами.</p> <ul style="list-style-type: none"> <li>- поиск и анализ литературы и электронных источников информации.</li> </ul>	6
	5	<p>Тема: Нисходящий синтаксический анализ</p> <p>СРС №5:- работа обучающихся с лекционным материалом и раздаточными материалами.</p> <ul style="list-style-type: none"> <li>- поиск и анализ литературы и электронных источников информации.</li> </ul>	6
	6	<p>Тема: Реализация распознавателей с возвратами</p> <p>СРС №6:- работа обучающихся с лекционным материалом и раздаточными материалами.</p> <ul style="list-style-type: none"> <li>- поиск и анализ литературы и электронных источников информации.</li> <li>- подготовка к выполнению индивидуальных заданий кейс-задачи.</li> <li>- подготовка ответов вопросы по обоснованию использованных, методов, алгоритмов, технологий.</li> </ul>	12
	7	<p>Тема: Реализация распознавателей без возвратов</p> <p>СРС №7:- работа обучающихся с лекционным материалом и раздаточными материалами.</p> <ul style="list-style-type: none"> <li>- поиск и анализ литературы и электронных источников информации.</li> <li>- подготовка к выполнению индивидуальных заданий кейс-задачи.</li> <li>- подготовка ответов вопросы по обоснованию использованных, методов, алгоритмов, технологий.</li> </ul>	12
Раздел 3	8	<p>Тема: Распределение памяти как фаза трансляции, вызываемая генератором кода</p> <p>СРС №8:- работа обучающихся с лекционным материалом и раздаточными материалами.</p> <ul style="list-style-type: none"> <li>- поиск и анализ литературы и электронных источников информации.</li> <li>- подготовка презентации по результатам поиска и анализа литературных и электронных источников.</li> </ul>	12
	9	<p>Тема: Распределение памяти как фаза трансляции, вызываемая генератором кода</p> <p>СРС №9:- работа обучающихся с лекционным материалом и раздаточными материалами.</p> <ul style="list-style-type: none"> <li>- поиск и анализ литературы и электронных источников информации.</li> <li>- подготовка к выполнению индивидуальных заданий кейс-задачи.</li> <li>- подготовка ответов вопросы по обоснованию использованных, методов, алгоритмов, технологий.</li> </ul>	12

Раздел дисциплины	№ п/п	Тема и вид СРС	Трудоёмкость (в часах)
	10	Тема: Управление динамической памятью. Стек, куча СРС №10:- работа обучающихся с лекционным материалом и раздаточными материалами. - поиск и анализ литературы и электронных источников информации. - подготовка к выполнению индивидуальных заданий кейс-задачи. - подготовка ответов вопросы по обоснованию использованных, методов, алгоритмов, технологий.	18
Раздел 4	11	Тема: Генерация промежуточного кода СРС №11:- работа обучающихся с лекционным материалом и раздаточными материалами. - поиск и анализ литературы и электронных источников информации.	12
	12	Тема: Реализация компилятора метаязыка СРС №12:- поиск и анализ литературы и электронных источников информации. - подготовка к выполнению индивидуальных заданий кейс-задачи. - подготовка ответов вопросы по обоснованию использованных, методов, алгоритмов, технологий.	24
<b>Итого:</b>			<b>134</b>

### 5. Примерная тематика курсовых проектов (работ)

Курсовые работы учебным планом не предусмотрены

### 6. Образовательные технологии

Семестр	Вид занятия (Л, ПР, ЛР)	Используемые интерактивные образовательные технологии	Количество часов
3	Л	Компьютерные технологии обучения, проблемная лекция, лекция-диспут, лекция-визуализация: информационно-развивающие технологии, <i>case-study</i> (ситуационный анализ)	16
	ЛР	Компьютерные технологии обучения, Технологии учебного проектирования и моделирования; Ситуационный анализ	30
<b>Итого:</b>			<b>46</b>

### 7. Оценочные средства для текущего контроля успеваемости, промежуточной аттестации по итогам освоения дисциплины и учебно-методическое обеспечение самостоятельной работы студентов

Итоговой формой контроля знаний, умений и навыков по дисциплине является экзамен.

Оценка знаний студентов производится по следующим критериям:

... оценка «отлично» выставляется обучающемуся, если он глубоко и прочно усвоил программный материал курса, исчерпывающе, последовательно, четко и логически стройно его излагает, умеет тесно увязывать теорию с практикой, свободно справляется с задачами и

вопросами, причем не затрудняется с ответами при видоизменении заданий, правильно обосновывает принятые решения, владеет разносторонними навыками и приемами выполнения практических задач;

– оценка «хорошо» выставляется обучающемуся, если он твердо знает материал курса, грамотно и по существу излагает его, не допуская существенных неточностей в ответе на вопрос, правильно применяет теоретические положения при решении практических вопросов и задач, владеет необходимыми навыками и приемами их выполнения;

– оценка «удовлетворительно» выставляется обучающемуся, если он имеет знания только основного материала, но не усвоил его деталей, допускает неточности, недостаточно правильные формулировки, нарушения логической последовательности в изложении программного материала, испытывает затруднения при выполнении практических задач;

– оценка «неудовлетворительно» выставляется обучающемуся, который не знает значительной части программного материала, допускает существенные ошибки, неуверенно, с большими затруднениями решает практические задачи или не справляется с ними самостоятельно.

### **Вопросы к экзамену по курсу «Конструирование трансляторов, интерпретаторов»**

1. Определения компилятора и интерпретатора.
2. Входной язык, целевой язык, язык реализации.
3. Т-диаграммы.
4. Прямой компилятор.
5. Кросс-трансляторы.
6. Виртуальные машины.
7. Компиляция "на лету".
8. Различные способы задания языков в компиляции: грамматики, конечные и магазинные автоматы.
9. Соотношения между различными способами задания языков.
10. Приложения в компиляции.
11. Основные задачи лексического анализа.
12. Регулярные выражения.
13. Использование конечных автоматов для лексического анализа.
14. Утилита *Lex*.
15. Синтаксический анализ.
16. Контекстно-свободные грамматики.
17. Нисходящие анализаторы.
18. Метод рекурсивного спуска.
19. Восходящие анализаторы.
20. *LR (k)*-анализаторы.
21. Построение *LR (0)*-анализатора.
22. *LR (1)*-анализатор.
23. *LALR*-анализаторы.
24. Неоднозначные грамматики.
25. Различные типы конфликтов. Разрешение конфликтов.
26. Генератор анализаторов *YACC*.
27. Фаза контроля типов.
28. Идентификация. Работа с таблицами.
29. Работа с типами.
30. Причины использования промежуточных языков в компиляторах.
31. Атрибутные деревья разбора.
32. Прямая и обратная польские записи.
33. Триады и тетрады.
34. Управление памятью с точки зрения разработчика компилятора.

35. Проблемы управления памятью.
36. Статическое и динамическое размещение памяти.
37. Стековый механизм управления памятью.
38. Управление кучей.
39. Подсчет ссылок и разметка памяти.
40. Задачи оптимизации кода программы.
41. Виды оптимизирующих преобразований.
42. Представления программы, используемые в оптимизирующих преобразованиях.
43. Примеры оптимизирующих преобразований.
44. Задачи анализа потока управления.
45. Граф потока управления. Доминирование.
46. Глубинное остовное дерево.
47. Основные виды фрагментов графа потока управления и их свойства.
48. Определение анализа потоков данных.
49. Достижимые определения и живые переменные.
50. Формализация задач анализа потоков данных.
51. Итеративный алгоритм для решения задач анализа потоков данных.
52. Постановка задачи выбора оптимальных инструкций.
53. Деревянные языки.
54. Деревянные грамматики.

### Образцы тестов для проведения текущего контроля

#### Типовой вариант теста

1. Строки, образованные из последовательности 0 и последующей последовательности 1, включая пустую строку, начинающиеся с 0, описываются регулярным выражением:

- а)  $0+(0|1)^*$ ;                      б)  $0(0|1)^*$ ;                      в)  $0(01)^*$ ;  
 г)  $0(0^*1^*)$ ;                      д)  $0^*(0|1)$ .

2. Правила вида  $A \rightarrow B$ , где  $A, B \in V_N$  называются:

- а) леворекурсивными;                      б) ценными;  
 в) факторизованными;                      г) бесполезными;  
 д) праворекурсивными.

3. Процедура *procedure B: begin if CH = 'b' then begin gc; B end else Err end* реализует рекурсивный спуск для:

- а) правила  $B \rightarrow bB | b$ ;                      б) правила  $B \rightarrow bB$ ;  
 в) правила  $B \rightarrow b$ ;                      г) правила  $B \rightarrow Bb$ ;  
 д) правила  $B \rightarrow Bb | b$ .

4. Достаточные условия применимости метода рекурсивного спуска выполняются в правиле:

- а)  $S \rightarrow aS | aB$ ;                      б)  $S \rightarrow Sa | aB$ ;  
 в)  $S \rightarrow Sa | a$ ;                      г)  $S \rightarrow aS | bB$ ;  
 д)  $S \rightarrow aS | a$ .

5. В грамматике с правилами  $P = \{S \rightarrow S+T | T; T \rightarrow a | S[S]\}$  справедливо:

- а)  $FIRST(T) = \{a\}$ ;                      б)  $FIRST(T) = \{S, T\}$ ;  
 в)  $FIRST(T) = \{a, +\}$ ;                      г)  $FIRST(T) = \{S, T, a\}$ ;  
 д)  $FIRST(T) = \{a, [, ], +\}$ .

6. Распознаватели LR(k)-грамматик основаны на построении:

- а) дерева разбора снизу вверх и левостороннем выводе цепочек;
- б) дерева разбора сверху вниз и левостороннем выводе цепочек;
- в) дерева разбора снизу вверх и правостороннем выводе цепочек;
- г) дерева разбора сверху вниз и правостороннем выводе цепочек;
- д) дерева разбора сверху вниз и комбинированном выводе цепочек.

7 Если символ в верхушке стека имеет меньшее или равное простое предшествование, чем очередной входной символ, то:

- а) МП-автомат выполняет свертку;
- б) МП-автомат выполняет выброс;
- в) МП-автомат выполняет сдвиг;
- г) МП-автомат выдает ошибку;
- д) МП-автомат успешно завершает работу.

8 На этапе синтеза компилятор выполняет:

- а) группировку символов в лексемы;
- б) синтаксический разбор программы;
- в) проверку семантической корректности программы;
- г) заполнение таблиц идентификаторов;
- д) генерацию объектной программы.

9 Многоадресный код с явно именуемым результатом называется:

- а) тетрадами;
- б) триадами;
- в) ПОЛИЗом;
- г) синтаксическим деревом;
- д) машинной командой.

10 Этап компиляции, на котором символы исходной программы, группируются в отдельные минимальные единицы текста, несущие смысловую нагрузку:

- а) лексический анализ;
- б) синтаксический анализ;
- в) семантический анализ;
- г) генерация кода;
- д) оптимизация кода.

#### *Примерный перечень тематик презентаций*

- 1) LEX – генератор лексических анализаторов
- 2) Системы автоматизации построения трансляторов. Система СУИЕР.
- 3) Системы автоматизации построения трансляторов. Система УАСС.
- 4) Динамическая компиляция
- 5) Обзор компилируемых языков программирования
- 6) Обзор интерпретируемых языков программирования
- 7) Кросс-трансляторы
- 8) Виртуальная машина
- 9) Сборщики мусора в трансляторах, интерпретаторах
- 10) Основные алгоритмы построения распознавателей с возвратами
- 11) Основные алгоритмы построения распознавателей без возвратов
- 12) Табличные распознаватели

#### *Типовое задание для реферата*

Привести алгоритмы построения распознавателя без возвратов на конкретном примере своего формального языка, алгоритм выбрать самостоятельно, показать графически работу соответствующего построенного МП-автомата. Привести примеры цепочек формального языка, которые будут приняты и отвергнуты, продемонстрировать графически по графу МП-автомата.

Аналогично привести алгоритм построения распознавателя с возвратами.

### Типовой вариант кейс-задачи №1

Даны:

1) формальный язык: язык содержит арифметические выражения, разделенные символом ;(точка с запятой). Арифметические выражения состоят из идентификаторов, десятичных чисел с плавающей точкой (в обычной и логарифмической форме), знака присваивания ( $:=$ ), знаков операций  $+$ ,  $-$ ,  $*$ ,  $/$  и круглых скобок.

2) алгоритм распознавателя: алгоритм с подбором альтернативы.

3) грамматика КС-языка:

$S \rightarrow a := F;$

$F \rightarrow F+T \mid T$

$T \rightarrow T*E \mid T/E \mid E$

$E \rightarrow (F) \mid -(F) \mid a$

Задание:

1) для заданного формального языка и заданной грамматики реализовать на любом выбранном самостоятельно языке программирования распознаватель с возвратами для отдельной строки по заданному алгоритму;

2) представить дерево вывода и вывод цепочек, используя правила приведенной грамматики, цепочки, принятые распознавателем и отвергнутые;

3) привести классификацию ошибок.

Типовой вариант кейс-задачи №2

Даны:

1) формальный язык: язык содержит арифметические выражения, разделенные символом ;(точка с запятой). Арифметические выражения состоят из идентификаторов, десятичных чисел с плавающей точкой (в обычной и логарифмической форме), знака присваивания ( $:=$ ), знаков операций  $+$ ,  $-$ ,  $*$ ,  $/$  и круглых скобок.

2) грамматика КС-языка:

$S \rightarrow a := F;$

$F \rightarrow F+T \mid T$

$T \rightarrow T*E \mid T/E \mid E$

$E \rightarrow (F) \mid -(F) \mid a$

3) алгоритм распознавателя: выбрать любой из предложенных по своему усмотрению.

Задание: построить и реализовать на любом языке программирования синтаксический анализатор, для чего:

1. Выполнить простейшие преобразования над заданной КС-грамматикой.

2. Проверить принадлежность КС-грамматики, получившейся в результате преобразований, к одному из известных классов КС-грамматик, для которых существуют линейные распознаватели.

3. Если соответствующий класс найден, взять за основу для построения распознавателя алгоритм разбора входных цепочек, известный для этого класса, если найдено несколько классов линейных распознавателей -- выбрать из них один по своему усмотрению.

4. Иначе, если соответствующий класс по п. 2 не был найден или же найденный класс КС-грамматик не устраивает разработчиков компилятора -- попытаться выполнить над грамматикой неформальные преобразования с целью подвести ее под интересующий класс КС-грамматик для линейных распознавателей и вернуться к п. 2.

5. Если же ни в п. 3, ни в п. 4 соответствующий распознаватель найти не удалось (что для современных языков программирования практически невозможно), необходимо использовать один из универсальных распознавателей.

6. Определить, в какой форме синтаксический распознаватель будет передавать результаты своей работы другим фазам компилятора (*внутренним представлением программы* в компиляторе).

Реализовать выбранный в п. 3 или 5 алгоритм с учетом структур данных, соответствующих п. 6.

В качестве основного пути выполнения задания предлагается распознаватель на основе грамматик операторного предшествования.

#### *Типовой вариант кейс-задачи №3*

*Задание:* Дополнить программу, выполняющую синтаксический контроль программы (результат выполнения предыдущей КЗ 2), написанной на заданном языке, фрагментами семантического анализа. Составить тестовые наборы данных. Дополнить программу, выполняющую синтаксический контроль программы, написанной на заданном языке, фрагментами, обеспечивающими генерацию выходного текста программы на языке ассемблера. Составить тестовые наборы данных. Получить результатов обработки тестовых примеров.

### **8. Учебно-методическое и информационное обеспечение дисциплины (модуля)**

#### **8.1. Основная литература**

1. Алдан А. Введение в генерацию программного кода [Электронный ресурс] / М.: Национальный Открытый Университет «ИНТУИТ», 2016. – 189с. – Режим доступа: [http://biblioclub.ru/index.php?page=book\\_red&id=428825](http://biblioclub.ru/index.php?page=book_red&id=428825)

2. Свердлов С. Конструирование компиляторов: учебное пособие. – Lambert Academic Publishing, 2015. – 575с.

#### **8.2 Дополнительная литература**

3. Ахо А. Компиляторы: принципы, технологии и инструменты. – М.: Изд. Дом» Вильямс», 2003. – 768с.

4. Опалева Э.А., Языки программирования и методы трансляции: учебное пособие для студентов вузов. – СПб.: БХВ – Петербург, 2005. – 480с.

5. Хонкрофт Дж., Мотвани Р., Ульман Дж. Введение в теорию автоматов, языков и вычислений: Пер. с англ. – М.: Издательский дом "Вильямс", 2002. – 528 с.: ил

#### **8.3 Периодические издания**

«Мир ПК»:

«Компьютер-Пресс»;

«PC-Magazine»;

«Byte (Россия)»;

«Программная инженерия»;

«Программирование»;

«Программные продукты и системы»;

«Теория и системы управления».

#### **8.4. Программное обеспечение и Интернет-ресурсы**

Интернет-ресурсы:

1) [msdn.microsoft.com/ru-ru/library/719exd8s.aspx](http://msdn.microsoft.com/ru-ru/library/719exd8s.aspx)

2) <http://www.secr.ru/> - Software Engineering Conference (Russia) 2005, 2006, 2007

Программное обеспечение:

1) ОС Windows,

2) Visual Studio.NET,

#### **8.5. Методические указания и материалы по видам занятий**

Варианты заданий к кейс-задам по дисциплине «Конструирование трансляторов, компиляторов» в электронном варианте.

## **9. Материально-техническое обеспечение дисциплины**

Учебный кабинет, лаборатория ИТО ИТИ.

## **10. Методические рекомендации по организации изучения дисциплины:**

Обучающийся, изучающий дисциплину, должен, с одной стороны, овладеть общим понятийным аппаратом, а с другой стороны, должен научиться применять теоретические знания на практике.

В результате изучения дисциплины обучающийся должен знать основные определения, понятия, аксиомы, методы доказательств.

Успешное освоение курса требует напряженной самостоятельной работы обучающегося. В программе курса отведено минимально необходимое время для работы обучающегося над темой. Самостоятельная работа включает в себя:

- чтение и конспектирование рекомендованной литературы;
- проработку учебного материала (по конспектам занятий, учебной и научной литературе), подготовку ответов на вопросы, предназначенные для самостоятельного изучения, доказательство отдельных утверждений, свойств, решение задач;
- подготовка к экзамену.

Руководство и контроль за самостоятельной работой обучающегося осуществляется в форме индивидуальных консультаций.

Важно добиться понимания изучаемого материала, а не механического его запоминания. При затруднении изучения отдельных тем, вопросов следует обращаться за консультациями к лектору.

Рабочая учебная программа по дисциплине «Конструирование трансляторов, интерпретаторов» составлена в соответствии с требованиями Федерального Государственного образовательного стандарта ВО по направлению 09.04.04 «ПРОГРАММНАЯ ИНЖЕНЕРИЯ» и учебного плана по профилю «Разработка программно-информационных систем».

## 11. Технологическая карта дисциплины

Курс 1

Семестр 3

Группа ИТ18ДР68НИ1

Преподаватель лектор **Помян С.В.**

Преподаватели, ведущие лабораторные, практические занятия - **Помян С.В.**

Кафедра программного обеспечения вычислительной техники и автоматизированных систем

Наименование дисциплины/курса	Уровень образования (бакалавриат, специалитет, магистратура)	Статус дисциплины в учебном плане (А, Б, В)	Количество зачетных единиц	
Конструирование трансляторов, интерпретаторов	магистратура	Б	6	
<b>СМЕЖНЫЕ ДИСЦИПЛИНЫ ПО УЧЕБНОМУ ПЛАНУ:</b>				
Научно-исследовательская работа, выполнение магистерской диссертации				
<b>БАЗОВЫЙ МОДУЛЬ</b> (проверка знаний и умений по дисциплине)				
Тема, задание или мероприятие текущего контроля	Виды текущей аттестации	Аудиторная или внеаудиторная	Минимальное количество баллов	Максимальное количество баллов
Презентация №1	П1	Аудиторная	10	20
Кейс-задача №1	К31	Аудиторная	7	15
Кейс-задача №2	К32	Аудиторная	8	15
<b>РУБЕЖНЫЙ КОНТРОЛЬ</b>	<b>РК</b>		<b>25</b>	<b>50</b>
Реферат №1	Р1	Внеаудиторная	7	15
Тест №1	Т1	Аудиторная	5	10
Кейс-задача №3	К33	Аудиторная	13	25
<b>РУБЕЖНАЯ АТТЕСТАЦИЯ</b>	<b>РА</b>		<b>25</b>	<b>50</b>
		<b>Итого</b>	<b>50</b>	<b>100</b>

Составитель, доцент



С.В. Помян

Рабочая учебная программа рассмотрена методической комиссией инженерно-технического института протокол № 1 от «12» 09 2019 г. и признана соответствующей требованиям Федерального Государственного образовательного стандарта и учебного плана по направлению 09.04.04 «ПРОГРАММНАЯ ИНЖЕНЕРИЯ».

Председатель МК ИТИ



Е.И. Андрианова

Зав. кафедрой ПОВТ и АС, доцент



С.Г. Федорченко