

**МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ РФ**  
Федеральное государственное бюджетное образовательное учреждение  
высшего образования  
Томский государственный университет систем управления и радиоэлектроники



УТВЕРЖДАЮ

Документ подписан электронной подписью  
Сертификат: 1с6сfa0a-52a6-4f49-aef0-5584d3fd4820  
Владелец: Троян Павел Ефимович  
Действителен: с 19.01.2016 по 16.09.2019

« \_\_\_\_ » \_\_\_\_\_ 2017 г.

**РАБОЧАЯ ПРОГРАММА УЧЕБНОЙ ДИСЦИПЛИНЫ**  
**«ЯЗЫКИ И МЕТОДЫ ПРОГРАММИРОВАНИЯ»**

Уровень основной образовательной программы бакалавриат

Направление подготовки 01.03.02 Прикладная математика и информатика

Профиль Прикладная математика и информатика

Форма обучения очная

Факультет систем управления

Кафедра автоматизированных систем управления

Курс 4

Семестр 7

Учебный план набора 2013 года и последующих лет

**Распределение учебного времени**

Виды учебной работы	Семестр 7	Всего	Единицы
Лекции	36	36	часов
Лабораторные работы	36	36	часов
Практические занятия	нет	нет	часов
Курсовой проект/работа (аудиторная)	нет	нет	часов
Всего аудиторных занятий	72	72	часов
Из них в интерактивной форме	18	18	часов
Самостоятельная работа студентов (СРС)	72	72	часов
Всего (без экзаменов)	144	144	часов
Самост. работа на подготовку и сдачу экзамена	36	36	часов
Общая трудоемкость	180	180	часов
(в зачётных единицах)	5	5	ЗЕТ

Экзамен - 7 семестр

Томск 2017

Рабочая программа составлена с учетом требований Федерального Государственного образовательного стандарта высшего образования (ФГОС ВО) по направлению 01.02.03 Прикладная математика и информатика (квалификация (степень) бакалавр), утвержденного Приказом Министерства образования и науки Российской Федерации 12.03.2015 №228, рассмотрена и утверждена на заседании кафедры «12» января 2017 г., протокол № 1.

Разработчик д.т.н., профессор каф. АСУ \_\_\_\_\_ В.Т. Калайда

Зав. обеспечивающей кафедрой АСУ  
д.т.н., профессор \_\_\_\_\_ А.М. Кориков

Рабочая программа согласована с факультетом, профилирующей и выпускающей кафедрами специальности.

Декан, к.т.н., доцент \_\_\_\_\_ П.В. Сенченко

Заведующий профилирующей и выпускающей  
кафедрой АСУ, д.т.н., профессор \_\_\_\_\_ А.М. Кориков

**Эксперты:**

Доцент кафедры АСУ \_\_\_\_\_ А.И. Исакова

## **1. ЦЕЛИ И ЗАДАЧИ ДИСЦИПЛИНЫ**

Дисциплина «Языки и методы программирования» читается в 7 семестре предусматривает чтение лекций, выполнение лабораторных работ, выполнение контрольных работ, получение различного рода консультаций.

**Цель преподавания дисциплины** - систематическое рассмотрение основных понятий языков программирования; синтаксиса, семантики, формальных способов описания языков программирования; типов данных, способов и механизмов управления данными; методов и основных этапов трансляции; конструкции распределенного и параллельного программирования.

**Задачей дисциплины** является получение знаний в области теории формальных языков, формальных грамматик, теории автоматов и методов трансляции.

## **2. МЕСТО ДИСЦИПЛИНЫ В СТРУКТУРЕ ООП**

Успешное овладение дисциплиной «Языки и методы программирования» предполагает предварительные знания по теории множеств, теории графов, полученные в дисциплинах: «Математический анализ», «Основы информатики», «Дискретная математика», «Основы программирования».

Зная теорию по языкам программирования и методам трансляции, студенты смогут использовать эти знания при дальнейшем проектировании программных систем, при изучении дисциплины «Программное обеспечение ЭВМ и сетей».

## **3. ТРЕБОВАНИЯ К РЕЗУЛЬТАТАМ ОСВОЕНИЯ ДИСЦИПЛИНЫ**

Процесс изучения дисциплины «Языки и методы программирования» направлен на формирование следующих компетенций:

- Способностью к разработке алгоритмических и программных решений в области системного и прикладного программирования, математических, информационных и имитационных моделей, созданию информационных ресурсов глобальных сетей, образовательного контента, прикладных баз данных, тестов и средств тестирования систем и средств на соответствие стандартам и исходным требованиям (ОПК-3).
- Способностью к разработке и применению алгоритмических и программных решений в области системного и прикладного программного обеспечения (ПК-7).

В результате изучения дисциплины студент должен:

### **Знать:**

- теоретические основы методов проектирования и способы описания языков программирования,
- основные положения теории формальных грамматик и языков,
- методы синтаксического анализа и перевода для класса формальных языков, используемых для описания основных конструкций языков программирования,
- стандарты, используемые для языков программирования;

### **Уметь:**

- самостоятельно выполнять формальное описание синтаксиса и семантики, несложных процедурно - ориентированных и проблемно - ориентированных языков программирования,
- разрабатывать алгоритмы, реализующие методы синтаксического анализа и перевода для наиболее часто используемых классов формальных грамматик,
- пользоваться стандартными терминами и определениями,
- читать научные статьи и пользоваться литературой для самостоятельного решения научно - исследовательских задач, связанных с разработкой языков и реализацией систем программирования;

### **Владеть:**

- перспективными направлениями работ и методическими подходами в области формальных методов описания и введения стандартов, используемых для описания языков программирования.

#### 4. ОБЪЕМ ДИСЦИПЛИНЫ И ВИДЫ УЧЕБНОЙ РАБОТЫ

Общая трудоёмкость дисциплины составляет 5 зачётные единицы.

Вид учебной работы	Всего часов	Семестр 7
<b>Аудиторные занятия (всего)</b>	<b>108</b>	<b>108</b>
В том числе:	–	–
Лекции	36	36
Практические занятия (ПЗ)	–	–
Семинары (С)	–	–
Лабораторные работы (ЛР)	36	36
<b>Самостоятельная работа (всего)</b>	<b>72</b>	<b>72</b>
В том числе:	–	–
Курсовой проект (работа)	–	–
Расчётно-графические работы	–	–
Проработка лекционного материала	24	24
Подготовка к лабораторным занятиям	36	36
Самостоятельное изучение тем теоретической части	12	12
<b>Подготовка к экзамену</b>		
Вид промежуточной аттестации (экзамен)	36	36
<b>Общая трудоёмкость</b>	<b>180</b>	<b>180</b>
час		
зач. ед.	<b>5</b>	<b>5</b>

#### 5. СОДЕРЖАНИЕ ДИСЦИПЛИНЫ

##### 5.1. Разделы дисциплин и виды занятий

№ п/п	Тематика лекций	Лек.	Лаб.	СРС	Всего час.	Формируемые компетенции
1.	<b>ВВЕДЕНИЕ В ДИСЦИПЛИНУ</b>	2		2	4	ОПК-3, ПК-7
2.	<b>МАТЕМАТИЧЕСКИЕ МЕТОДЫ ФОРМАЛЬНОГО ОПИСАНИЯ ЯЗЫКА</b>	4	6	10	20	ОПК-3, ПК-7
3.	<b>ВВЕДЕНИЕ В ТЕОРИЮ КОМПИЛЯЦИИ.</b>	4	6	10	20	ОПК-3, ПК-7
4.	<b>ТЕОРИЯ ЯЗЫКОВ.</b>	4	6	10	20	ОПК-3, ПК-7
5.	<b>КОНТЕКСТНО - СВОБОДНЫЕ ГРАММАТИКИ</b>	4	6	10	20	ОПК-3, ПК-7
6.	<b>АВТОМАТЫ С МАГАЗИННОЙ ПАМЯТЬЮ</b>	4		4	8	ОПК-3, ПК-7
7.	<b>МЕТОДЫ СИНТАКСИЧЕСКОГО АНАЛИЗА</b>	4	6	8	16	ОПК-3, ПК-7
8.	<b>ВКЛЮЧЕНИЕ ДЕЙСТВИЙ В СИНТАКСИС</b>	2		2	4	ОПК-3, ПК-7
9.	<b>РАСПРЕДЕЛЕНИЕ ПАМЯТИ</b>	2		2	4	ОПК-3, ПК-7
10.	<b>ИСПРАВЛЕНИЕ И ДИАГНОСТИКА ОШИБОК ОПТИМИЗАЦИЯ КОДА</b>	4	6	10	20	ОПК-3, ПК-7
<b>ВСЕГО</b>		<b>36</b>	<b>36</b>	<b>72</b>	<b>144</b>	

##### 5.2. Содержание разделов дисциплины (по лекциям)

№ п/п	Тематика лекций	Содержание разделов	Трудоёмкость (час)	Формируемые компетенции
1.	<b>Введение</b>	Краткая характеристика дисциплины, ее цели и задачи, порядок изучения материала, связи с другими дисциплинами учебного плана и место в подготовке инженеров по специальности 230100. Перечень дисциплин, усвоение которых необходимо студентам для изучения данной дисциплины. Основы методики и форм контроля самостоятельной работы, краткая характеристика учебной литературы. Организация компилятора	2	ОПК-3, ПК-7
2.	<b>Математические методы формального описания языка</b>	Множество цепочек. Операции над цепочками. Языки. Операции над языками. Итерация языка. Гомоморфизм. Алгоритмы. Частичные алгоритмы. Полные алгоритмы. Рекурсивные алгоритмы. Задание алгоритмов. Проблемы. Пример формального	4	ОПК-3, ПК-7

		описания языка.		
3.	<b>Введение в теорию компиляции</b>	Задание языков программирования. Синтаксис и семантика. Процесс компиляции. Лексический анализ. Работа с таблицами. Синтаксический анализ. Генерация кода. Алгоритм генерации кода. Оптимизация кода. Исправление ошибок. Анализ структурной схемы компилятора.	4	ОПК-3, ПК-7
4	<b>Теория языков</b>	Способы определения языков. Грамматика. Грамматика с ограничениями на правила. Распознаватели. Регулярные множества, их распознавание и порождение. Алгоритм решения системы линейных уравнений с регулярными выражениями. Регулярные множества и конечные автоматы. Проблема разрешимости. Графическое представление конечных автоматов. Минимизация конечных автоматов. Алгоритм построения канонического конечного автомата.	4	ОПК-3, ПК-7
5.	<b>Контекстно - свободные грамматики</b>	Основные понятия и определения. Деревья выводов. Преобразование КС-грамматик. Алгоритм устранения недостижимых символов. Алгоритм преобразования в грамматику без $\epsilon$ -правил. Алгоритм устранения цепных правил. Грамматика без циклов. Нормальная форма Хомского. Нормальная форма Грейбах. Алгоритм устранения левой рекурсии	4	ОПК-3, ПК-7
6.	<b>Автоматы с магазинной памятью</b>	Автоматы с магазинной памятью (МП-автоматы). Расширенные МП-автоматы. Недетерминированные и детерминированные МП-автоматы. Языки, допускаемые МП-автоматами. Эквивалентность МП-автоматов и КС-грамматик. Конечный преобразователь. Регулярный перевод. Преобразователь с магазинной памятью	4	ОПК-3, ПК-7
7.	<b>Методы синтаксического анализа</b>	Прямые и синтаксически ориентированные методы анализа. Нисходящая методика синтаксического анализа. <b>LL(K)</b> -грамматики. Алгоритмы разбора для <b>LL(1)</b> -грамматики. Метод рекурсивного спуска. Алгоритм разбора для <b>LR(K)</b> -грамматики. <b>S</b> -грамматика. Алгоритмы построения управляющей таблицы <b>LL(1)</b> -грамматики. Формальное определение алгоритма разбора типа "перенос-свертка". Грамматика предшествования. Простое, слабое, оперативное предшествование	4	ОПК-3, ПК-7
8.	<b>Включение Действий в синтаксис</b>	Получение четверок. Работа с таблицей символов	2	ОПК-3, ПК-7
9.	<b>Распределение памяти</b>	Стек времени прогона. Методы вызова параметров. Вызов по значению. Вызов по имени. Вызов по результату. Вызов по значению и результату. Вызов по ссылке. Куча. Счетчик ссылок. Сборка мусора.	2	ОПК-3, ПК-7
10.	<b>Исправление и диагностика ошибок оптимизация кода</b>	Типы ошибок. Лексические ошибки. Ошибки в употреблении скобок. Синтаксические ошибки. Режим переполоха. Исключение символов. Включение символов. Правила для ошибок. Сообщения о синтаксических ошибках. Контекстно-зависимые ошибки. Ошибки, связанные с употреблением типов. Ошибки, допускаемые во время прогона. Ошибки, связанные с нарушением ограничений	4	ОПК-3, ПК-7
<b>ВСЕГО</b>			<b>36</b>	

### 5.3. Разделы дисциплины и междисциплинарные связи с обеспечивающими (предыдущими) и последующими дисциплинами

№ п/п	Наименование обеспечивающих (предыдущих) дисциплин	№ № разделов данной дисциплины, для которых необходимо изучение обеспечивающих (предыдущих) дисциплин									
		1	2	3	4	5	6	7	8	9	10
1.	«Математический анализ»		+		+						+
2.	«Основы информатики»	+						+			+
3.	«Дискретная математика»		+	+	+	+	+	+			+
4.	«Основы программирования»			+	+			+	+	+	+

№ п/п	Наименование последующих дисциплин	№ № разделов данной дисциплины, которые необходимы при изучении последующих дисциплин									
		1	2	3	4	5	6	7	8	9	10
1.	«Программное обеспечение ЭВМ и сетей»	+	+	+	+	+	+	+	+	+	+

### 5.4. Соответствие компетенций, формируемых при изучении дисциплины, и видов занятий

Перечень компетенций	Л.	ЛЗ.	СРС	Формы контроля
				ОПК-3
ПК-7	+	+	+	Опрос на лекции, защита лаб. работ, проверка дом. задания

Л – лекция, ЛЗ – лабораторные занятия, СРС – самостоятельная работа студента

### 6. МЕТОДЫ И ФОРМЫ ОРГАНИЗАЦИИ ОБУЧЕНИЯ

Для успешного освоения дисциплины применяются различные образовательные технологии, которые обеспечивают достижение планируемых результатов обучения согласно основной образовательной программе, с учётом требований к объёму занятий в интерактивной форме.

#### Технологии интерактивного обучения при разных формах занятий

Методы	Формы	Лекции (час)	Практические занятия (час)	Всего (час)
Работа в команде			8	8
Игра		4		4
Поисковый метод			6	6
Итого интерактивных занятий		4	14	18

#### Примечание.

1. Работа в команде» происходит при коллективном выполнении заданий всех лабораторных работ.
2. «Поисковый метод» студенты используют при выполнении заданий (лаб. работа № 2, лаб. работа № 3).
3. Различные игровые моменты предлагаются студентам во время лекций.

### 7. ЛАБОРАТОРНЫЕ РАБОТЫ

№ п/п	№ раздела дисциплины из табл. 5.1	Темы лабораторной работы	Трудоёмкость (час)	Формируемые компетенции
1.	2, 3, 4	Генерации и оптимизации промежуточного кода для арифметического выражения	8	ОПК-3, ПК-7
2.	2, 3, 4, 10	Построить грамматику и написать программы разбора описания структуры для языков Pascal и C++ методом рекурсивного спуска	10	ОПК-3, ПК-7
3.	4, 5, 7, 10	Построить грамматику и построить таблицу разбора <b>LL(1)</b> и написать программу разбора описания структуры для языков Pascal и C++.	10	ОПК-3, ПК-7
4.	4, 5, 7, 10	Построить грамматику, таблицу разбора <b>LR(1)</b> и написать программу разбора описания структуры для языков Pascal и C++	8	ОПК-3, ПК-7
<b>ВСЕГО</b>			<b>36</b>	

### 8. ПРАКТИЧЕСКИЕ ЗАНЯТИЯ (СЕМИНАРЫ) – не предусмотрен.

## 9. САМОСТОЯТЕЛЬНАЯ РАБОТА

№ п/п	№ раздела дисциплины из табл. 5.1	Тематика самостоятельной работы (детализация)	Трудоёмкость (час)	Формируемые компетенции	Контроль выполнения работы (Опрос, тест, дом. задание, и т.д.)
1.	1-10	Проработка лекционного материала	24	ОПК-3, ПК-7	Опрос на занятиях (устно), экзамен
2.	2, 3, 4, 5, 7, 10	Подготовка к лабораторным работам	36	ОПК-3, ПК-7	Отчет, защита лабораторных работ
3.	3, 5	Самостоятельное изучение тем теоретической части	12	ОПК-3, ПК-7	Дом. задание, тест
<b>ИТОГО</b>			<b>72</b>		

### Темы для самостоятельного изучения (Всего 12 часов).

1. Оптимизация линейных блоков (4 час.).
2. Базовые топологические преобразования (4 час.).
3. Топологические преобразования для оптимизации циклов (4 час.).

### 10. КУРСОВОЙ ПРОЕКТ – не предусмотрен.

### 11. РЕЙТИНГОВАЯ СИСТЕМА ОЦЕНКИ КАЧЕСТВА

#### 11.1 Балльная раскладка отдельных элементов контроля по видам занятий

##### Курс 4, семестр 7

##### Контроль обучения – Экзамен.

Таблица 11.1 Балльные оценки для элементов контроля (лекции, лабораторные работы, экзамен).

Элементы учебной деятельности	Максимальный балл на 1-ую контрольную точку с начала семестра	Максимальный балл за период между 1КТ и 2КТ	Максимальный балл за период между 2КТ и на конец семестра	Всего за семестр
Посещение занятий	4	4	4	<b>12</b>
Лабораторные работы	10	10	10	<b>30</b>
Тестовый контроль	5	5	5	<b>15</b>
Компонент своевременности	4	4	5	<b>13</b>
<b>Итого максимум за период:</b>	<b>23</b>	<b>23</b>	<b>24</b>	<b>70</b>
<b>Нарастающим итогом</b>	<b>23</b>	<b>46</b>	<b>70</b>	
<b>Экзамен</b>			<b>30</b>	<b>30</b>
<b>ИТОГО</b>				<b>100</b>

Таблица 11.2 – Пересчет баллов в оценки за контрольные точки

Баллы на дату контрольной точки	Оценка
≥ 90 % от максимальной суммы баллов на дату КТ	5
От 70% до 89% от максимальной суммы баллов на дату КТ	4
От 60% до 69% от максимальной суммы баллов на дату КТ	3
< 60 % от максимальной суммы баллов на дату КТ	2

Таблица 11.3 – Пересчет суммы баллов в традиционную и международную оценку

Оценка (ГОС)	Итоговая сумма баллов, учитывает успешно сданный экзамен	Оценка (ECTS)
5 (отлично) (зачтено)	<b>90 – 100</b>	A (отлично)
4 (хорошо) (зачтено)	<b>85 – 89</b>	B (очень хорошо)
	<b>75 – 84</b>	C (хорошо)
	<b>70 – 74</b>	D (удовлетворительно)
3 (удовлетворительно) (зачтено)	<b>65 – 69</b>	
	<b>60 – 64</b>	E (посредственно)
2 (неудовлетворительно), (не зачтено)	<b>Ниже 60 баллов</b>	F (неудовлетворительно)

## **12. УЧЕБНО-МЕТОДИЧЕСКИЕ МАТЕРИАЛЫ ПО ДИСЦИПЛИНЕ**

### **12.1. Основная литература**

1. Калайда, В. Т. Теория языков программирования методов трансляции: Методическое пособие [Электронный ресурс] / Калайда В. Т. — Томск: ТУСУР, 2012. — 219 с. — Режим доступа: <https://edu.tusur.ru/publications/2063>

### **12.2 Дополнительная литература**

1. Калайда В.Т. Теория языков программирования и методов трансляции. Учебное пособие. – Томск: Изд. – во, ТУСУР, 2007 – 244 с. (45 экз.).

2. Рейуорд-Смит, В. Дж. Теория формальных языков. Вводный курс : Пер. с англ. / В. Дж. Рейуорд-Смит ; пер. Б. А. Кузьмин, ред. пер. Б. А. Шестаков. - М.: Радио и связь, 1988. - 124 с. (10 экз.)

3. Льюис Ф., Розешкранц Д., Стирнз Р. Теоретические основы проектирования компиляторов. — М.: Мир, 1979. – 656 с. (2 экз.)

### **12.3 Перечень пособий, методических указаний и материалов, используемых в учебном процессе**

1. Калайда В.Т., Романенко В.В. Теория языков программирования и методов трансляции: методическое пособие по выполнению практических работ по дисциплине «Теория языков программирования и методов трансляции» для студентов специальности 230105 – «Программное обеспечение вычислительной техники и автоматизированных систем». - Томск: ТУСУР, 2012. - 37 с. – [электронный ресурс] – Режим доступа [http://asu.tusur.ru/learning/spec230105/d44/s230105\\_d44\\_pract.pdf](http://asu.tusur.ru/learning/spec230105/d44/s230105_d44_pract.pdf)

2. Надреев, И. И. Языки и технологии программирования: Методические рекомендации к практическим занятиям, лабораторным работам и самостоятельной работе [Электронный ресурс] / Надреев И. И. — Томск: ТУСУР, 2012. — 16 с. — Режим доступа: <https://edu.tusur.ru/publications/2540>

### **12.4 Базы данных, информационно-справочные и поисковые системы**

1. [www.compress.ru](http://www.compress.ru) – Журнал «КомпьютерПресс»
2. [www.osp.ru](http://www.osp.ru) – Издательство «Открытые системы»
3. [www.cnews.ru](http://www.cnews.ru) – Издание о высоких технологиях
4. [www.it-daily.ru](http://www.it-daily.ru) – Новости российского ИТ-рынка
5. [www.isn.ru](http://www.isn.ru) – Российская сеть информационного общества

## **13 МАТЕРИАЛЬНО-ТЕХНИЧЕСКОЕ ОБЕСПЕЧЕНИЕ ДИСЦИПЛИНЫ**

Для проведения практических занятий по дисциплине используются персональный ПК с процессором Pentium 4, операционная система MS WindowsXP, пакет Microsoft VisualStudio 2008.



Приложение к рабочей программе

**МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ РОССИЙСКОЙ ФЕДЕРАЦИИ**

Федеральное государственное бюджетное образовательное учреждение высшего образования

«ТОМСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ СИСТЕМ УПРАВЛЕНИЯ И  
РАДИОЭЛЕКТРОНИКИ» (ТУСУР)

**УТВЕРЖДАЮ**

**Проректор по учебной работе**

**П. Е. Троян**

«\_\_» \_\_\_\_\_ 2017 г.

**ФОНД ОЦЕНОЧНЫХ СРЕДСТВ ПО УЧЕБНОЙ ДИСЦИПЛИНЕ**

**«ЯЗЫКИ И МЕТОДЫ ПРОГРАММИРОВАНИЯ»**

Уровень основной образовательной программы \_\_\_\_\_ бакалавриат \_\_\_\_\_

Направление подготовки \_\_\_\_\_ 01.03.02 Прикладная математика и информатика \_\_\_\_\_

Профиль Прикладная математика и информатика \_\_\_\_\_

Форма обучения \_\_\_\_\_ очная \_\_\_\_\_

Факультет \_\_\_\_\_ систем управления \_\_\_\_\_

Кафедра \_\_\_\_\_ автоматизированных систем управления \_\_\_\_\_

Курс \_\_\_\_\_ 4 \_\_\_\_\_

Семестр \_\_\_\_\_ 7 \_\_\_\_\_

Учебный план набора 2013 года и последующих лет \_\_\_\_\_

**Экзамен – 7 семестр**

**Томск 2017**

## 1 ВВЕДЕНИЕ

Фонд оценочных средств (ФОС) является приложением к рабочей программе дисциплины «Языки и методы программирования» и представляет собой совокупность контрольно-измерительных материалов (типовые задачи (задания), контрольные работы, тесты и др.) и методов их использования, предназначенных для измерения уровня достижения студентом установленных результатов обучения.

ФОС по дисциплине (практике) используется при проведении текущего контроля успеваемости и промежуточной аттестации студентов. Перечень закрепленных за дисциплиной «Языки и методы программирования» компетенций приведен в таблице 1.

**Таблица 1** – Перечень закрепленных за дисциплиной компетенций

Код	Формулировка компетенции	Этапы формирования компетенции
ОПК-3	Способностью к разработке алгоритмических и программных решений в области системного и прикладного программирования, математических, информационных и имитационных моделей, созданию информационных ресурсов глобальных сетей, образовательного контента, прикладных баз данных, тестов и средств тестирования систем и средств на соответствие стандартам и исходным требованиям	<p><b>Знать:</b></p> <ul style="list-style-type: none"> <li>– теоретические основы методов проектирования и способы описания языков программирования,</li> <li>– основные положения теории формальных грамматик и языков,</li> <li>– методы синтаксического анализа и перевода для класса формальных языков, используемых для описания основных конструкций языков программирования,</li> <li>– стандарты, используемые для языков программирования;</li> </ul> <p><b>Уметь:</b></p> <ul style="list-style-type: none"> <li>– самостоятельно выполнять формальное описание синтаксиса и семантики, несложных процедурно - ориентированных и проблемно - ориентированных языков программирования,</li> <li>– разрабатывать алгоритмы, реализующие методы синтаксического анализа и перевода для наиболее часто используемых классов формальных грамматик,</li> <li>– пользоваться стандартными терминами и определениями,</li> <li>– читать научные статьи и пользоваться литературой для самостоятельного решения научно - исследовательских задач, связанных с разработкой языков и реализацией систем программирования;</li> </ul> <p><b>Владеть:</b></p> <ul style="list-style-type: none"> <li>– перспективными направлениями работ и методическими подходами в области формальных методов описания и введения стандартов, используемых для описания языков программирования.</li> </ul>
ПК-7	Способностью к разработке и применению алгоритмических и программных решений в области системного и прикладного программного обеспечения.	

## 2 РЕАЛИЗАЦИЯ КОМПЕТЕНЦИЙ

### 2.1 Компетенции ОПК-3

**ОПК-3:** Способность к разработке алгоритмических и программных решений в области системного и прикладного программирования, математических, информационных и имитационных моделей, созданию информационных ресурсов глобальных сетей, образовательного контента, прикладных баз данных, тестов и средств тестирования систем и средств на соответствие стандартам и исходным требованиям.

Для формирования компетенций необходимо осуществить ряд этапов. Этапы формирования компетенций, применяемые для этого вида занятий и используемые средства оценивания, представлены в таблице 2.

**Таблица 2** – Этапы формирования компетенции и используемые средства оценивания

Состав	Знать	Уметь	Владеть
--------	-------	-------	---------

<b>Содержание этапов</b>	<p><b><u>Благодаря способности к разработке алгоритмических и программных решений в области системного и прикладного программирования</u></b></p> <p><b><i>Знать:</i></b></p> <ul style="list-style-type: none"> <li>– теоретические основы методов проектирования и способы описания языков программирования,</li> <li>– основные положения теории формальных грамматик и языков,</li> <li>– методы синтаксического анализа и перевода для класса формальных языков, используемых для описания основных конструкций языков программирования,</li> <li>– стандарты, используемые для языков программирования;</li> </ul>	<p><b><u>Благодаря способности к разработке алгоритмических и программных решений в области системного и прикладного программирования</u></b></p> <p><b><i>Уметь:</i></b></p> <ul style="list-style-type: none"> <li>– самостоятельно выполнять формальное описание синтаксиса и семантики, несложных процедурно - ориентированных и проблемно - ориентированных языков программирования,</li> <li>– разрабатывать алгоритмы, реализующие методы синтаксического анализа и перевода для наиболее часто используемых классов формальных грамматик,</li> <li>– пользоваться стандартными терминами и определениями,</li> <li>– читать научные статьи и пользоваться литературой для самостоятельного решения научно - исследовательских задач, связанных с разработкой языков и реализацией систем программирования;</li> </ul>	<p><b><u>Благодаря способности к разработке алгоритмических и программных решений в области системного и прикладного программирования</u></b></p> <p><b><i>Владеть:</i></b></p> <ul style="list-style-type: none"> <li>– перспективными направлениями работ и методическими подходами в области формальных методов описания и введения стандартов, используемых для описания языков программирования.</li> </ul>
<b>Виды занятий</b>	Лекции, лабораторные занятия, групповые консультации	Лабораторные занятия, выполнение домашнего задания, СРС	Лабораторные занятия, СРС
<b>Используемые средства оценивания</b>	<ul style="list-style-type: none"> <li>– Тест;</li> <li>– Контрольная работа;</li> <li>– Реферат;</li> <li>– Экзамен.</li> </ul>	<ul style="list-style-type: none"> <li>– Проверка правильности выполнения лабораторные работ;</li> <li>– Контрольная работа;</li> <li>– Конспект самостоятельной работы;</li> <li>– Экзамен.</li> </ul>	<ul style="list-style-type: none"> <li>– Проверка правильности выполнения лабораторные работ;</li> <li>– Контрольная работа;</li> <li>– Конспект самостоятельной работы.</li> </ul>

Общие характеристики показателей и критериев оценивания компетенции на всех этапах приведены в таблице 3.

**Таблица 3** – Общие характеристики показателей и критериев оценивания компетенции по этапам

<b>Показатели и критерии</b>	<b>Знать</b>	<b>Уметь</b>	<b>Владеть</b>
<b>ОТЛИЧНО (высокий уровень)</b>	Обладает фактическими и теоретическими знаниями в пределах изучаемой области с пониманием границ применимости	Обладает диапазоном практических умений, требуемых для развития творческих решений, абстрагирования проблем	Контролирует работу, проводит оценку, совершенствует действия работы

<b>ХОРОШО</b> (базовый уровень)	Знает факты, принципы, процессы, общие понятия в пределах изучаемой области	Обладает диапазоном практических умений, требуемых для решения определенных проблем в области исследования	Берет ответственность за завершение задач в исследовании, приспосабливает свое поведение к обстоятельствам в решении проблем
<b>УДОВЛЕТВОРИТЕЛЬНО</b> (низкий уровень)	Обладает низким уровнем общих знаний	Обладает умениями на низком уровне, которые не достаточны для выполнения даже простых задач	Работает только при прямом наблюдении

Формулировка показателей и критериев оценивания данной компетенции приведена в таблице 4.

Таблица 4 – Показатели и критерии оценивания компетенции на этапах

Показатели и критерии	Знать	Уметь	Владеть
<b>ОТЛИЧНО</b> (высокий уровень)	<u>Благодаря способности к разработке алгоритмических и программных решений в области системного и прикладного программирования</u> – <i>На высоком уровне</i> <i>Знать:</i> теоретические основы методов проектирования и способы описания языков программирования, – основные положения теории формальных грамматик и языков, – методы синтаксического анализа и перевода для класса формальных языков, используемых для описания основных конструкций языков программирования, стандарты, используемые для языков программирования;	<u>Благодаря способности к разработке алгоритмических и программных решений в области системного и прикладного программирования</u> – <i>На высоком уровне</i> <i>Уметь:</i> самостоятельно выполнять формальное описание синтаксиса и семантики, несложных процедурно ориентированных и проблемно ориентированных языков программирования, – разрабатывать алгоритмы, реализующие методы синтаксического анализа и перевода для наиболее часто используемых классов формальных грамматик, – пользоваться стандартными терминами и определениями, читать научные статьи и пользоваться литературой для самостоятельного решения научно-исследовательских задач, связанных с разработкой языков и реализацией систем программирования;	<u>Благодаря способности к разработке алгоритмических и программных решений в области системного и прикладного программирования</u> <i>На высоком уровне</i> <i>Владеть:</i> перспективными направлениями работ и методическими подходами в области формальных методов описания и введения стандартов, используемых для описания языков программирования.
<b>ХОРОШО</b> (базовый уровень)	<u>Благодаря способности к разработке</u>	<u>Благодаря способности к разработке алгоритмических и</u>	<u>Благодаря способности к разработке</u>

	<p><b><u>алгоритмических и программных решений в области системного и прикладного программирования</u></b>  – <i>хорошо знать</i>:  теоретические основы методов проектирования и способы описания языков программирования,  – основные положения теории формальных грамматик и языков,  – методы синтаксического анализа и перевода для класса формальных языков, используемых для описания основных конструкций языков программирования,</p>	<p><b><u>программных решений в области системного и прикладного программирования</u></b>  – <i>хорошо уметь</i>:  самостоятельно выполнять формальное описание синтаксиса и семантики, несложных процедурно - ориентированных и проблемно - ориентированных языков программирования,  – пользоваться стандартными терминами и определениями,  – читать научные статьи и пользоваться литературой для самостоятельного решения научно - исследовательских задач;</p>	<p><b><u>алгоритмических и программных решений в области системного и прикладного программирования</u></b>  <i>хорошо владеть</i>:  перспективными направлениями работ и методическими подходами в области формальных методов описания и введения стандартов, используемых для описания языков программирования.</p>
<p><b>УДОВЛЕТВОРИТЕЛЬНО (низкий уровень)</b></p>	<p><b><u>Благодаря способности к разработке алгоритмических и программных решений в области системного и прикладного программирования</u></b>  <i>Знать</i>:  – теоретические основы методов проектирования и способы описания языков программирования,  – основные положения теории формальных грамматик и языков, методы синтаксического анализа и перевода для класса формальных языков, используемых для описания основных конструкций языков программирования,</p>	<p><b><u>Благодаря способности к разработке алгоритмических и программных решений в области системного и прикладного программирования</u></b>  – <i>Уметь</i>: на <i>низком уровне</i> самостоятельно выполнять формальное описание синтаксиса и семантики, несложных процедурно - ориентированных и проблемно - ориентированных языков программирования,  – пользоваться стандартными терминами и определениями, читать научные статьи и пользоваться литературой для самостоятельного решения научно - исследовательских задач;</p>	<p><b><u>Благодаря способности к разработке алгоритмических и программных решений в области системного и прикладного программирования</u></b>  <i>Владеть</i>: на <i>низком уровне</i>  перспективными направлениями работ и методическими подходами в области формальных методов описания и введения стандартов, используемых для описания языков программирования имитационного моделирования. разрабатывать требования к ППП</p>

## 2.2 Компетенции ПК-7

**ПК-7:** способность к разработке и применению алгоритмических и программных решений в области системного и прикладного программного обеспечения.

Для формирования компетенций необходимо осуществить ряд этапов. Этапы формирования компетенций, применяемые для этого виды занятий и используемые средства оценивания, представлены в таблице 5.

**Таблица 5** – Этапы формирования компетенции и используемые средства оценивания

Состав	Знать	Уметь	Владеть
<b>Содержание этапов</b>	<p><b><u>Благодаря способности к разработке и применению алгоритмических и программных решений</u></b></p> <p>– <b><i>Знать:</i></b> теоретические основы методов проектирования и способы описания языков программирования, – основные положения теории формальных грамматик и языков, – методы синтаксического анализа и перевода для класса формальных языков, используемых для описания основных конструкций языков программирования, стандарты, используемые для языков программирования;</p>	<p><b><u>Благодаря способности к разработке и применению алгоритмических и программных решений</u></b></p> <p>– <b><i>Уметь:</i></b> самостоятельно выполнять формальное описание синтаксиса и семантики, несложных процедурно - ориентированных и проблемно - ориентированных языков программирования, – разрабатывать алгоритмы, реализующие методы синтаксического анализа и перевода для наиболее часто используемых классов формальных грамматик, – пользоваться стандартными терминами и определениями, читать научные статьи и пользоваться литературой для самостоятельного решения научно - исследовательских задач, связанных с разработкой языков и реализацией систем программирования.</p>	<p><b><u>Благодаря способности к разработке и применению алгоритмических и программных решений</u></b></p> <p><b><i>Владеть:</i></b> перспективными направлениями работ и методическими подходами в области формальных методов описания и введения стандартов, используемых для описания языков программирования.</p>
<b>Виды занятий</b>	Лекции, лабораторные занятия, групповые консультации	Лабораторные занятия, выполнение домашнего задания, СРС	Лабораторные занятия, СРС
<b>Используемые средства оценивания</b>	<ul style="list-style-type: none"> <li>– Тест;</li> <li>– Контрольная работа;</li> <li>– Реферат;</li> <li>– Экзамен</li> </ul>	<ul style="list-style-type: none"> <li>– Проверка правильности выполнения лабораторные работ;</li> <li>– Контрольная работа;</li> <li>– Конспект самостоятельной работы;</li> <li>– Экзамен</li> </ul>	<ul style="list-style-type: none"> <li>– Проверка правильности выполнения лабораторные работ;</li> <li>– Контрольная работа;</li> <li>– Конспект самостоятельной работы</li> </ul>

Общие характеристики показателей и критериев оценивания компетенции на всех этапах приведены в таблице 3.

Формулировка показателей и критериев оценивания данной компетенции приведена в таблице 6.

Таблица 6 – Показатели и критерии оценивания компетенции на этапах

Показатели и критерии	Знать	Уметь	Владеть
<p><b>ОТЛИЧНО (высокий уровень)</b></p>	<p><u>Благодаря способности к разработке и применению алгоритмических и программных решений</u> – <i>На высоком уровне</i> <i>Знать:</i> теоретические основы методов проектирования и способы описания языков программирования, – основные положения теории формальных грамматик и языков, – методы синтаксического анализа и перевода для класса формальных языков, используемых для описания основных конструкций языков программирования, стандарты, используемые для языков программирования;</p>	<p><u>Благодаря способности к разработке и применению алгоритмических и программных решений</u> – <i>На высоком уровне</i> <i>Уметь:</i> самостоятельно выполнять формальное описание синтаксиса и семантики, несложных процедурно ориентированных и проблемно ориентированных языков программирования, – разрабатывать алгоритмы, реализующие методы синтаксического анализа и перевода для наиболее часто используемых классов формальных грамматик, – пользоваться стандартными терминами и определениями, читать научные статьи и пользоваться литературой для самостоятельного решения научно-исследовательских задач, связанных с разработкой языков и реализацией систем программирования.</p>	<p><u>Благодаря способности к разработке и применению алгоритмических и программных решений</u> <i>На высоком уровне</i> <i>Владеть:</i> перспективными направлениями работ и методическими подходами в области формальных методов описания и введения стандартов, используемых для описания языков программирования.</p>
<p><b>ХОРОШО (базовый уровень)</b></p>	<p><u>Благодаря способности к разработке и применению алгоритмических и программных решений</u> – <i>Хорошо</i> <i>знать:</i> теоретические основы методов проектирования и способы описания языков программирования, – основные положения теории формальных грамматик и языков, – методы синтаксического анализа и перевода для класса формальных языков.</p>	<p><u>Благодаря способности к разработке и применению алгоритмических и программных решений</u> – <i>Хорошо</i> <i>уметь:</i> самостоятельно выполнять формальное описание синтаксиса и семантики, несложных процедурно ориентированных и проблемно ориентированных языков программирования, – разрабатывать алгоритмы, реализующие методы синтаксического</p>	<p><u>Благодаря способности к разработке и применению алгоритмических и программных решений</u> <i>Хорошо</i> <i>владеть:</i> перспективными направлениями работ и методическими подходами в области формальных методов описания и введения стандартов, используемых для описания языков программирования.</p>

		анализа и перевода для наиболее часто используемых классов формальных грамматик.	
<b>УДОВЛЕТВОРИТЕЛЬНО</b> (низкий уровень)	<p><b><u>Благодаря способности к разработке и применению алгоритмических и программных решений</u></b></p> <p>– <i>Знать:</i></p> <p>теоретические основы методов проектирования и способы описания языков программирования,</p> <p>– основные положения теории формальных грамматик и языков.</p>	<p><b><u>Благодаря способности к разработке и применению алгоритмических и программных решений</u></b></p> <p>– <i>Уметь:</i></p> <p>самостоятельно выполнять формальное описание синтаксиса и семантики, несложных процедурно ориентированных и проблемно ориентированных языков программирования.</p>	<p><b><u>Благодаря способности к разработке и применению алгоритмических и программных решений</u></b></p> <p><i>Владеть:</i></p> <p>перспективными направлениями работ и методическими подходами в области формальных методов описания и введения стандартов, используемых для описания языков программирования.</p>

### 3 ТИПОВЫЕ КОНТРОЛЬНЫЕ ЗАДАНИЯ

Для реализации вышеперечисленных задач обучения используются следующие материалы: типовые контрольные задания или иные материалы, необходимые для оценки знаний, умений, навыков и (или) опыта деятельности, характеризующих этапы формирования компетенций в процессе освоения образовательной программы, в составе, приведенном ниже.

#### 3.1 Темы лабораторных занятий

1. Генерации и оптимизации промежуточного кода для арифметического выражения
2. Построить грамматику и написать программы разбора описания структуры для языков Pascal и C++ методом рекурсивного спуска
3. Построить грамматику и построить таблицу разбора **LL(1)** и написать программу разбора описания структуры для языков Pascal и C++.
4. Построить грамматику, таблицу разбора **LR(1)** и написать программу разбора описания структуры для языков Pascal и C++

#### 3.2 Пример вариантов вопросов для подготовки к контрольным точкам

- 1 Организация компилятора.
- 4 Языки.
- 5 Алгоритмы.
- 6 Частичные алгоритмы. Полные алгоритмы. Рекурсивные алгоритмы. Задание алгоритмов.
- 7 Задание языков программирования.
- 8 Процесс компиляции. Лексический анализ.
- 9 Работа с таблицами.
- 10 Синтаксический анализ.
- 11 Генерация кода. Алгоритм генерации кода. Оптимизация кода. Исправление ошибок.
- 12 Способы определения языков.
- 13 Грамматики. Грамматики с ограничениями на правила.
- 14 Распознаватели.
- 15 Регулярные множества, их распознавание и порождения.
- 16 Алгоритм решения системы линейных выражений с регулярными выражениями.
- 17 Регулярные множества и конечные автоматы.



- 18 Проблема разрешимости.
- 19 Графическое представление конечных автоматов. Минимизация конечных автоматов.
- 20 Алгоритм построения канонического конечного автомата. Контекстно-свободные грамматики.
- 21 Деревья выводов. Преобразование КС-грамматик.
- 22 Алгоритм устранения недостижимых символов. Алгоритм устранения бесполезных символов.
- 23 Алгоритм преобразования грамматику без  $\epsilon$ -правил. Алгоритм устранения цепных правил.
- 24 Грамматики без циклов. Нормальная форма Хомского. Алгоритм преобразования к нормальной форме Хомского. Нормальная форма Грейбах.
- 25 Алгоритм устранения левой рекурсии.
- 26 Автоматы с магазинной памятью.
- 27 Эквивалентность МП-автоматов КС-грамматик. LR(k)-грамматики. S-грамматика.
- 28 LL(1)-грамматика.
- 29 Алгоритм определения принадлежности грамматики к LL(1)-грамматики.
- 30 Таблица разбора. Технология разбора снизу-вверх.
- 31 Построение LR – таблицы разбора.
- 32 Метод определения количества состояний грамматики.
- 33 Разрешение конфликта сдвиг/приведение.
- 34 Сравнение LL - и LR – методов разбора
- 35 Структуры данных для генерации промежуточного кода.
- 36 Алгоритм генерации промежуточного кода для арифметических выражений.
- 37 Оптимизация линейного участка. Модель линейного участка.
- 38 Преобразование блоков. Топологические преобразования.
- 39 Графическое представление блоков. Критерий эквивалентности блоков. Оптимизация блоков.
- 40 Алгебраические преобразования.
- 41 Арифметические выражения. Модель машины.
- 42 Разметка деревьев. Алгоритм разметки синтаксического дерева.
- 43 Алгоритм построения на языке ассемблера для выражений.
- 44 Программы с командами STORE.
- 45 Влияние алгебраических законов. Алгоритм построения дерева с минимальной оценкой.
- 46 Алгоритм построения синтаксического дерева с минимальной оценкой в предположении, что одни операции коммутативны и ассоциативны.
- 47 Программы с циклами. Модель программы.
- 48 Анализ потоков управления.
- 49 Алгоритм вычисления прямого доминирования.
- 50 Преобразование программ.
- 51 Оптимизация циклов. (Перемещение кода. Индуктивные переменные. Замена сложных операций. Развёртывание цикла)
- 52 Анализ потоков данных. Интервалы.
- 53 Алгоритм разбиения графа управления на непересекающиеся интервалы.
- 54 Анализ потоков данных с помощью интервалов (Алгоритм вычисления функции IN)
- 55 Несводимые графы. Алгоритм общего вычисления функции IN.

### 3.3 Темы для самостоятельной работы (темы рефератов)

1. Оптимизация линейных блоков.
2. Базовые топологические преобразования.
3. Топологические преобразования для оптимизации циклов.

### 3.4 Вопросы для подготовки к экзамену

1. Множество цепочек. Операции над цепочками.
2. Языки. Операции над языками. Итерация языка.
3. Гомоморфизм. Алгоритмы. Частичные алгоритмы. Полные алгоритмы. Рекурсивные алгоритмы. Задание алгоритмов.
4. Задание языков программирования. Синтаксис и семантика. Процесс компиляции. Лексический анализ.

5. Работа с таблицами. Синтаксический анализ. Генерация кода. Алгоритм генерации кода. Оптимизация кода.
  6. Исправление ошибок. Анализ структурной схемы компилятора.
  7. Способы определения языков. Грамматики. Грамматики с ограничениями на правила.
  8. Распознаватели. Регулярные множества, их распознавание и порождение.
  9. Алгоритм решения системы линейных уравнений с регулярными выражениями.
  10. Регулярные множества и конечные автоматы. Проблема разрешимости.
  11. Графическое представление конечных автоматов. Минимизация конечных автоматов.
- Алгоритм построения канонического конечного автомата.
12. Деревья выводов. Преобразование КС-грамматик. Алгоритм устранения недостижимых символов. Алгоритм преобразования в грамматику без ε-правил. Алгоритм устранения цепных правил.
  13. Грамматика без циклов. Нормальная форма Хомского.
  14. Нормальная форма Грейбах. Алгоритм устранения левой рекурсии.
  15. Автоматы с магазинной памятью (МП-автоматы). Расширенные МП-автоматы.
  16. Недетерминированные и детерминированные МП-автоматы. Языки, допускаемые МП-автоматами.
  17. Эквивалентность МП-автоматов и КС-грамматик. Конечный преобразователь. Регулярный перевод. Преобразователь с магазинной памятью.
  18. Прямые и синтаксически ориентированные методы анализа. Нисходящая методика синтаксического анализа. LL(K)-грамматики. Алгоритмы разбора для LL(1)-грамматики.
  19. Метод рекурсивного спуска. Алгоритм разбора для LR(K)-грамматики. S-грамматика. Алгоритмы построения управляющей таблицы LL(1)-грамматики
  20. Формальное определение алгоритма разбора типа "перенос-свертка". Грамматика предшествования. Простое, слабое, оперативное предшествование.
  21. Получение четверок. Работа с таблицей символов
  22. Стек времени прогона. Методы вызова параметров. Вызов по значению. Вызов по имени. Вызов по результату. Вызов по значению и результату. Вызов по ссылке. Куча. Счетчик ссылок. Сборка мусора.
  23. Типы ошибок. Лексические ошибки. Ошибки в употреблении скобок. Синтаксические ошибки.
  24. Режим переполоха. Исключение символов. Включение символов. Правила для ошибок. Сообщения о синтаксических ошибках. Контекстно-зависимые ошибки.
  25. Ошибки, связанные с употреблением типов. Ошибки, допускаемые во время прогона. Ошибки, связанные с нарушением ограничений.

#### 4 МЕТОДИЧЕСКИЕ МАТЕРИАЛЫ

Для обеспечения процесса обучения и решения задач обучения используются следующие материалы, определяющие процедуры оценивания знаний, умений, навыков и (или) опыта деятельности, характеризующих этапы формирования компетенций, в составе:

- **Калайда, В. Т. Теория языков программирования методов трансляции: Методическое пособие [Электронный ресурс] / Калайда В. Т. — Томск: ТУСУР, 2012. — 219 с. — Режим доступа: <https://edu.tusur.ru/publications/2063>**

Методические указания и материалы, используемые в учебном процессе

3. Калайда В.Т., Романенко В.В. Теория языков программирования и методов трансляции: методическое пособие по выполнению практических работ по дисциплине «Теория языков программирования и методов трансляции» для студентов специальности 230105 – «Программное обеспечение вычислительной техники и автоматизированных систем». - Томск: ТУСУР, 2012. - 37 с. – [электронный ресурс] – Режим доступа [http://asu.tusur.ru/learning/spec230105/d44/s230105\\_d44\\_pract.pdf](http://asu.tusur.ru/learning/spec230105/d44/s230105_d44_pract.pdf)
4. Надреев, И. И. Языки и технологии программирования: Методические рекомендации к практическим занятиям, лабораторным работам и самостоятельной работе [Электронный ресурс] / Надреев И. И. — Томск: ТУСУР, 2012. — 16 с. — Режим доступа: <https://edu.tusur.ru/publications/2540>