

МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ  
РОССИЙСКОЙ ФЕДЕРАЦИИ

ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ БЮДЖЕТНОЕ  
ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ  
«РЯЗАНСКИЙ ГОСУДАРСТВЕННЫЙ РАДИОТЕХНИЧЕСКИЙ  
УНИВЕРСИТЕТ ИМЕНИ В.Ф. УТКИНА»

Факультет вычислительной техники  
Кафедра «Вычислительная и прикладная математика»

СОГЛАСОВАНО


Декан факультета  
вычислительной техники

  
Д.А. Перепелкин  
« 26 » 06 2020 г.



УТВЕРЖДАЮ

Проректор РОПиМД

  
А.В. Корячко  
« 26 » 06 2020 г.

Заведующий кафедрой:

ВПиМ  Г.В. Овечкин  
« 26 » 06 2020 г.

**РАБОЧАЯ ПРОГРАММА**

дисциплины

**Б1.В.ДВ.06.01 «Разработка компиляторов»**

Направление подготовки — 09.03.04 «Программная инженерия»

Уровень подготовки — академический бакалавриат

Направленность (профиль) подготовки

Программная инженерия

Квалификация (степень) выпускника — бакалавр


Форма обучения — очная

Рязань 2020 г.

## ЛИСТ СОГЛАСОВАНИЙ


Рабочая программа составлена с учетом требований федерального государственного образовательного стандарта высшего образования по направлению подготовки 09.03.04 – «Программная инженерия»,  
утвержденного \_\_\_\_\_ №920 от 19.09.2017 г. \_\_\_\_\_  
(дата утверждения ФГОС ВО)

**Разработчик:**  
доцент кафедры ВПМ

 \_\_\_\_\_ О.В. Антипов

Программа рассмотрена и одобрена на заседании кафедры ВПМ «11» июня 2020 г.,  
протокол № 11.

**Заведующий кафедрой ВПМ**

 \_\_\_\_\_ Г.В. Овечкин

## 1. ЦЕЛЬ И ЗАДАЧИ ОСВОЕНИЯ ДИСЦИПЛИНЫ

Одной из сложнейших задач проектирования программного обеспечения является задача по проектированию и разработке компиляторов. Решение этой задачи требует от программиста не только знания языков и умения работать с инструментальными системами программирования, но глубокого знания теории формальных грамматик, языков программирования и методов трансляции.

**Целью изучения дисциплины** является ознакомить студентов с основными понятиями языков программирования: синтаксиса, семантики, формальных способов описания грамматики языков программирования; методов и основных этапов трансляции программ: лексический, синтаксический и семантический анализ.

### **Основные задачи освоения учебной дисциплины:**

1. Получение системы базовых знаний по теории формальных грамматик и языков программирования, методов трансляции.
2. Формирование представления о существующих языках программирования и методах трансляции и их использовании в решении различных задач.
3. Систематизация и закрепление практических навыков и умений по программированию отдельных подсистем компилятора.

## 2. МЕСТО ДИСЦИПЛИНЫ В СТРУКТУРЕ ОБРАЗОВАТЕЛЬНОЙ ПРОГРАММЫ

Дисциплина относится к вариативной части профессиональных дисциплин по выбору блока № 1 (Б1.В.ДВ.06.01), изучается по очной форме обучения на 4 курсе во 7 семестре, базируется на знаниях, полученных в ходе изучения дисциплин: «Информатика», «Алгоритмические языки и программирование», «Низкоуровневое программирование», «Объектно-ориентированное программирование», «Функциональное программирование», «Основы программной инженерии», «Теория автоматов и формальных языков».

*Пререквизиты дисциплины. Для освоения дисциплины обучающиеся должны*

- *знать:*
  - основы программирования на алгоритмических языках императивного стиля;
  - основные концепции и методы низкоуровневого, объектно-ориентированного и функционального программирования;
  - стандартные алгоритмы обработки структур данных;
- *иметь:*
  - навыки работы в современных средах программирования;
  - навыки работы с языками объектно-ориентированного программирования C++ или C#;
  - понятие о структурах и типах данных;
  - навыки отладки и тестирования программ;
  - навыки алгоритмизации и понимание архитектуры программ.

*Постреквизиты дисциплины.* Полученные знания используются при изучении дисциплин «Параллельное программирование», «Разработка многопоточных приложений», «Конструирование программного обеспечения», при выполнении НИР и при подготовке выпускной квалификационной работы, а также при дальнейшем обучении в магистратуре по данному направлению подготовки.

*В результате изучения дисциплины студенты должны*

- *иметь представление и понимание об:*
  - теоретических основах методов проектирования и способов описания языков программирования;
  - особенностях языков программирования (как самого языка, так и транслятора), нюансах выполнения, написанных на них программ, вытекающих из принятых традиционных способах организации трансляторов (компиляция и интерпретация);

- *знать*:
  - существующие языки программирования и их классификацию;
  - парадигмы языков программирования;
  - методы трансляции: компиляторы и интерпретаторы;
  - схемы трансляции;
  - структуру компилятора;
  - методы анализа и синтеза исходной программы;
  - построение грамматики;
- *иметь навыки*:
  - разработки сложных алгоритмов и структур данных;
  - использования элементов трансляции при разработке прикладных программ.

### 3. КОМПЕТЕНЦИИ ОБУЧАЮЩЕГОСЯ, ФОРМИРУЕМЫЕ В РЕЗУЛЬТАТЕ ОСВОЕНИЯ ДИСЦИПЛИНЫ

Процесс изучения дисциплины направлен на формирование следующих компетенций в соответствии с ФГОС ВО, ПООП (при наличии) по данному направлению подготовки, а также компетенций (при наличии), установленных университетом.

#### Рекомендуемые профессиональные компетенции выпускников и индикаторы их достижения (при наличии)

Задача ПД	Объект или область знания	Код и наименование профессиональной компетенции	Код и наименование индикатора достижения профессиональной компетенции	Обоснование (ПС, анализ опыта)
Направленность (профиль), специализация: Программная инженерия				
Тип задач профессиональной деятельности: проектный				
Формирование требований к информатизации и автоматизации прикладных процессов, формализация предметной области проекта; технико-экономическое обоснование проектных решений и составление технического задания на разработку ПП; проектирование программно-аппаратных средств в соответствии с техническим заданием; применение современных инструментальных средств при разработке ПО; документирование компонентов ИС на всех стадиях ЖЦ	Прикладные и информационные процессы  Информационные технологии  Программное обеспечение	ПК-4. Владение навыками моделирования, анализа и использования формальных методов конструирования ПО	ИД – 1ПК-4 <i>Знать</i> : основы моделирования и формальные методы конструирования ПО.	06.001 Программист
			ИД – 2ПК-4 <i>Уметь</i> : использовать формальные методы моделирования и конструирования ПО.	06.004 Специалист по тестированию в области информационных технологий
			ИД – 3ПК-4 <i>Владеть</i> : методами формализации, моделирования и конструирования ПО.	06.022 Системный аналитик
				06.028 Системный программист

		<p>ПК-7. Разработка тестовых случаев, проведение тестирования программного продукта, анализ результатов и разработка тестовых документов</p>	<p>ИД – 1<sub>ПК-7</sub> <i>Знать:</i> виды и типы тестирования, существующие техники и инструменты проектирования и комбинаторики тестов, системы автоматиз. тестирования.</p> <p>ИД – 2<sub>ПК-7</sub> <i>Уметь:</i> документировать тесты, выбирать техники тестирования, разрабатывать скрипты для автоматизации тестирования, пользоваться специальным ПО для автоматизации тестирования, проводить анализ полученных результатов.</p> <p>ИД – 3<sub>ПК-7</sub> <i>Владеть:</i> навыками разработки тестовых случаев, проведения тестирования ПП, анализа результатов и разработки тестовых документов.</p>	
		<p>ПК-8. Разработка компонентов системных программных продуктов: разработка компиляторов и системных утилит</p>	<p>ИД – 1<sub>ПК-8</sub> <i>Знать:</i> синтаксис и стандартные библиотеки выбранного языка программирования, технологии разработки и отладки системных продуктов.</p> <p>ИД – 2<sub>ПК-8</sub> <i>Уметь:</i> применять выбранные технологии и языки программирования для разработки компонентов системных ПП и осуществлять их отладку.</p> <p>ИД – 3<sub>ПК-8</sub> <i>Владеть:</i> навыками написания исходного кода и отладки разработанных компонентов системных ПП, их сопровождения и реинжиниринга, разработки эксплуатационной документации.</p>	

#### 4. СТРУКТУРА И СОДЕРЖАНИЕ ДИСЦИПЛИНЫ

##### 4.1. Объем дисциплины по семестрам (курсам) и видам занятий в зачетных единицах

с указанием количества академических часов, выделенных на контактную работу обучающихся с преподавателем (по видам занятий) и на самостоятельную работу обучающихся

Общая трудоемкость изучения дисциплины для очной формы обучения составляет 5 зачетных единиц (ЗЕ), 180 академических часа.

Объем дисциплины	Всего часов	Семестр 7
Общая трудоемкость дисциплины, в том числе:	180	180
1.Контактная работа обучающихся с преподавателем (всего), в том числе:	50,35	50,35
Лекции	24	24
лабораторные работы	16	16
практические занятия	8	8
иная контактная работа (ИКР)	0,35	0,35
консультация	2	2
2.Самостоятельная работа	85	85
3.Курсовой проект	-	-
4.Контроль	44,65	44,65
Вид промежуточной аттестации		Экзамен

##### 4.2. Содержание дисциплины, структурированное по разделам и темам

В структурном отношении программа дисциплины представлена следующими разделами и темами:

###### Раздел 1. Языки программирования и задачи трансляции.

**Тема 1. Языки программирования.** Задачи обработки текстов. Языки программирования. Алфавит. Синтаксис. Семантика.

**Тема 2. Классификация языков программирования.** Архитектура фон Неймана. Язык программирования. Языки низкого уровня. Языки высокого уровня. Поколения языков программирования.

**Тема 3. Парадигмы языков программирования.** Парадигмы языков программирования. Императивные языки. Функциональные языки. Декларативные языки. Объектно-ориентированные языки.

**Тема 4. Трансляторы.** Проблема трансляции языка программирования. Виртуальная трехадресная машина. Переход к мнемокоду. Требования, предъявляемые к транслятору.

**Тема 5. Компиляторы и интерпретаторы.** Компиляция и интерпретация. Компилятор. Целевая программа. Интерпретатор. Система обработки языка. Препроцессор. Компилятор. Система обработки языка. Ассемблер. компоновщик. Загрузчик.

###### Раздел 2. Введение в компиляцию.

**Тема 1. Структура компилятора.** Анализ. Синтез. Фазы компиляции.

**Тема 2. Анализ исходной программы.** Лексический анализ. Tokeny и атрибуты. Синтаксический анализ. Грамматическая структура. Семантический анализ. Приведения (*coercion*).

**Тема 3. Синтез целевой программы.** Генерация промежуточного кода. Оптимизация кода. Генерация кода. Таблица символов. Проходы компилятора.

### **Раздел 3. Определение синтаксиса.**

**Тема 1. Простой синтаксически управляемый транслятор.** Синтаксически управляемая трансляция. Трёхадресный код. Контекстно-свободная грамматика. Инфиксная и постфиксная форма выражения. Назначение. Примеры.

**Тема 2. Модель начальной стадии компилятора.** Лексический анализатор. Токены. Фаза генерации промежуточного кода. Виды промежуточного кода. Абстрактное синтаксическое дерево.

**Тема 3. Грамматика.** Контекстно-свободная грамматика. Определение грамматик. Компоненты грамматик. Терминалы и нетерминалы. Лексемы. Токены. Терминалы. Пример составления грамматики для выражений. Выведение (Порождение). Пример для грамматики для выражений. Пример составления грамматики для функций и их параметров.

**Тема 4. Деревья разбора.** Свойства. Узлы. Метки. Корень. Родитель. Дочерний узел. Родственный узел. Лист. Внутренний узел. Потомок. Предок. Пример построения дерева для выражения.

**Тема 5. Построение грамматики.** Неоднозначности грамматики. Ассоциативность операторов. Пример грамматики и дерева разбора. Приоритет операторов. Пример построения грамматики.

### **Раздел 4. Синтаксически управляемая трансляция.**

**Тема 1. Синтаксически управляемая трансляция.** Атрибуты. Схемы трансляции. Постфиксная запись. Правило трансляции. Пример трансляции. Арность. Правило вычисления. Пример вычисления. Синтезированные атрибуты. Синтаксически управляемое определение. Вычисление атрибутов.

**Тема 2. Схемы трансляции.** Семантические действия. Пример продукции и дерева разбора с семантическим действием. Пример действий трансляции выражения в постфиксную запись, дерево разбора.

### **Раздел 5. Разбор.**

**Тема 1. Аннотированное дерево разбора.** Синтаксически управляемые определения для трансляции инфиксных выражений в постфиксные. Обходы дерева. Обход в ширину и глубину. Прямой и обратный порядок.

**Тема 2. Классы методов разбора.** Нисходящий анализ. Предиктивный анализ. Пример грамматики и дерева разбора для неё.

**4.3. Разделы дисциплины и трудоемкость по видам учебных занятий  
(в академических часах)**

№ п/п	Тема	Общая трудоемкость, всего часов	Контактная работа обучающихся с преподавателем				Самостоятельная работа обучающихся
			всего	лекции	лабораторные работы	практич. занятия	
<b>1.</b>	<b>Языки программирования и задачи трансляции</b>	<b>28</b>	<b>8</b>	<b>4</b>	<b>2</b>	<b>2</b>	<b>20</b>
1.1.	<i>Языки программирования</i>	5,5	1,5	0,5		1	4
1.2.	<i>Классификация языков программирования</i>	5	1	0,5		0,5	4
1.3.	<i>Парадигмы языков программирования</i>	5,5	1,5	1		0,5	4
1.4.	<i>Трансляторы</i>	6	2	1	1		4
1.5.	<i>Компиляторы и интерпретаторы</i>	6	2	1	1		4
<b>2.</b>	<b>Введение в компиляцию</b>	<b>22</b>	<b>10</b>	<b>6</b>		<b>4</b>	<b>12</b>
2.1.	<i>Структура компилятора</i>	6	2	2			4
2.2.	<i>Анализ исходной программы</i>	8	4	2		2	4
2.3.	<i>Синтез целевой программы</i>	8	4	2		2	4
<b>3.</b>	<b>Определение синтаксиса</b>	<b>48</b>	<b>16</b>	<b>6</b>	<b>8</b>	<b>2</b>	<b>32</b>
3.1.	<i>Простой синтаксически управляемый транслятор</i>	6	2	1		1	4
3.2.	<i>Модель начальной стадии компилятора</i>	10	4	1	2	1	6
3.3.	<i>Грамматика</i>	9	3	1	2		6
3.4.	<i>Деревья разбора</i>	11	3	1	2		8
3.5.	<i>Построение грамматики</i>	12	4	2	2		8
<b>4.</b>	<b>Синтаксически управляемая трансляция</b>	<b>16</b>	<b>6</b>	<b>4</b>	<b>2</b>		<b>10</b>
4.1.	<i>Синтаксически управляемая трансляция</i>	7	3	2	1		4
4.2.	<i>Схемы трансляции</i>	9	3	2	1		6
<b>5.</b>	<b>Разбор</b>	<b>21</b>	<b>8</b>	<b>4</b>	<b>4</b>		<b>13</b>
5.1.	<i>Аннотированное дерево разбора</i>	10	4	2	2		6
5.2.	<i>Классы методов разбора</i>	11	4	2	2		7
	Экзамены и консультации	<b>45</b>					<b>45</b>
	<b>Всего:</b>	<b>180</b>	<b>48</b>	<b>24</b>	<b>16</b>	<b>8</b>	<b>132</b>



#### 4.3.1. Лекционные занятия

№ п/п	Темы лекционных занятий	Трудоемкость (час.)	Формируемые компетенции	Форма контроля
1.	Языки программирования. Классификация языков программирования. Парадигмы языков программирования	2	ПК-4, ПК-8	экзамен
2.	Трансляторы. Компиляторы и интерпретаторы	2	ПК-4, ПК-8	экзамен
3.	Структура компилятора	2	ПК-4, ПК-8	экзамен
4.	Анализ исходной программы	2	ПК-4, ПК-8	экзамен
5.	Синтез целевой программы	2	ПК-4, ПК-8	экзамен
6.	Простой синтаксически управляемый транслятор. Модель начальной стадии компилятора	2	ПК-4, ПК-8	экзамен
7.	Грамматика. Деревья разбора	2	ПК-4, ПК-7, ПК-8	экзамен
8.	Построение грамматики	2	ПК-4, ПК-7, ПК-8	экзамен
9.	Синтаксически управляемая трансляция	2	ПК-4, ПК-8	экзамен
10.	Схемы трансляции	2	ПК-4, ПК-7, ПК-8	экзамен
11.	Аннотированное дерево разбора	2	ПК-4, ПК-7, ПК-8	экзамен
12.	Классы методов разбора	2	ПК-4, ПК-8	экзамен

#### 4.3.2. Практические занятия (семинары)

№ п/п	Тематика практических занятий (семинаров)	Трудоемкость (час.)	Формируемые компетенции	Форма контроля
1.	Языки программирования и задачи трансляции	2	ПК-4, ПК-8	экзамен
2.	Анализ исходной программы	2	ПК-4, ПК-8	экзамен
3.	Синтез целевой программы	2	ПК-4, ПК-8	экзамен
4.	Синтаксически управляемая трансляция	2	ПК-4, ПК-8	экзамен

## План практических занятий

### Тема 1. Языки программирования.

Язык. Языки программирования. Алфавит. Синтаксис. Семантика. (2 часа)

*Цель занятия.* Закрепить теоретические знания о способах описания языков программирования; об их синтаксисе и семантике, особенностях описания (как самого языка, так и транслятора), нюансах выполнения, написанных на них программ, вытекающих из принятых традиционных способах организации трансляторов (компиляция и интерпретация)

*Задача закрепления теоретических знаний и практических навыков:* студент должен знать существующие языки программирования и их классификацию; существующие парадигмы организации языков программирования.

*Форма проведения:* подготовка рефератов для выступления на практическом занятии и коллективной дискуссии.

*Учебно-методическая литература* [1 – 10].

### Тема 2. Анализ исходной программы. Синтез целевой программы.

Лексический анализ. Токены и атрибуты. Синтаксический анализ. Грамматическая структура. Семантический анализ. Приведения (*coercion*). Генерация промежуточного кода. Оптимизация кода. Генерация кода. Таблица символов. Проходы компилятора. (4 часа)

*Цель занятия.* Освоить основные механизмы анализа и синтеза программ.

*Задачи закрепления теоретических знаний и практических навыков:* студент должен знать основы лексического, синтаксического и семантического анализа программ, генерации и оптимизации кода программы.

*Форма проведения:* подготовка рефератов для выступления на практическом занятии и коллективной дискуссии..

*Учебно-методическая литература* [1 – 10].

**Тема 3. Синтаксически управляемая трансляция.** Атрибуты. Схемы трансляции. Постфиксная запись. Правило и пример трансляции. Арность. Правило и примеры вычисления. Синтезированные атрибуты. Синтаксически управляемое определение. Вычисление атрибутов. Семантические действия. Пример продукции и дерева разбора с семантическим действием. Пример действий трансляции выражения в постфиксную запись, дерево разбора. (2 часа)

*Цель занятия.* Освоить основные элементы и правила схем трансляции на конкретных примерах.

*Задачи закрепления теоретических знаний и практических умений и навыков:* студент должен знать схемы трансляции и формы записи выражений, и их трансляцию.

*Форма проведения:* решение и анализ типовых задач на аудиторных занятиях, самостоятельное решение студентами аналогичных задач в аудитории, выполнение домашнего задания по изучаемой теме.

*Учебно-методическая литература* [1 – 10].

### 4.3.3. Лабораторные занятия

Лабораторные работы по дисциплине «Разработка компиляторов» для направления 09.03.04 – «Программная инженерия» имеют своей целью освоить на практике основные методы трансляции и компиляции и создания компилятора преобразования инфиксных арифметических выражений в постфиксные. Основной упор сделан на начальную стадию компиляции, в частности на лексический анализ, синтаксический анализ и генерацию промежуточного кода.

Изученные модели, теория и алгоритмы, связанные с компиляторами, в дальнейшем могут применяться студентами для решения широкого диапазона задач проектирования и разработки программного обеспечения, в том числе системного программного обеспечения.

### Перечень лабораторных работ

№ п/п	Наименование лабораторных работ	Трудоемкость (час.)	Формируемые компетенции	Форма контроля
1.	Создание консольной утилиты для генерации арифметических выражений и их трансляции в словесное представление	2	ПК-4, ПК-7, ПК-8	экзамен, защита л.р.
2.	Создание консольной утилиты для лексического анализа арифметических выражений, создание последовательности токенов и таблицы символов	4	ПК-4, ПК-7, ПК-8	экзамен, защита л.р.
3.	Модификация консольной утилиты, добавление синтаксического анализа, проверка синтаксических ошибок, генерация дерева разбора	4	ПК-4, ПК-7, ПК-8	экзамен, защита л.р.
4.	Модификация консольной утилиты, добавление семантического анализа, проверка семантических ошибок, построение постфиксной записи арифметического выражения	4	ПК-4, ПК-7, ПК-8	экзамен, защита л.р.
5.	Модификация консольной утилиты, генерация промежуточного кода, оптимизация кода, выполнение вычисления арифметического выражения	2	ПК-4, ПК-7, ПК-8	экзамен, защита л.р.

#### 4.3.4. Самостоятельная работа

Самостоятельное изучение тем учебной дисциплины способствует закреплению знаний, умений и навыков, полученных в ходе аудиторных занятий, углублению и расширению знаний по отдельным вопросам и темам дисциплины; освоению умений практического использования полученных знаний по разработке программ системного характера.

*Самостоятельная работа обучающихся по данному курсу заключается:*

- при подготовке к лекциям и практическим занятиям в изучении и доработке конспекта лекции с применением учебно-методической литературы, подборе дополнительных примеров к теоретическим положениям курса по данной теме;
- при подготовке к лабораторным работам в разработке, отладке и выполнении программы своего варианта задания по данной теме;
- при самостоятельном изучении отдельных вопросов и тем курса с применением рекомендуемой учебно-методической литературы;
- при подготовке к экзамену в изучении, осмыслении и повторении пройденного теоретического материала и выполненных практических заданий с применением конспекта лекций и учебно-методической литературы.

№ п/п	Тематика самостоятельной работы	Трудоемкость (час.)	Формируемые компетенции	Форма контроля
1.	Языки программирования. Классификация языков программирования. Парадигмы языков программирования	12	ПК-4, ПК-8	экзамен
2.	Трансляторы. Компиляторы и интерпретаторы	8	ПК-4, ПК-8	экзамен
3.	Структура компилятора	4	ПК-4, ПК-8	экзамен
4.	Анализ исходной программы	4	ПК-4, ПК-8	экзамен
5.	Синтез целевой программы	4	ПК-4, ПК-8	экзамен
6.	Простой синтаксически управляемый транслятор. Модель начальной стадии компилятора	10	ПК-4, ПК-8	экзамен
7.	Грамматика. Деревья разбора	14	ПК-4, ПК-7, ПК-8	экзамен
8.	Построение грамматики	8	ПК-4, ПК-7, ПК-8	экзамен
9.	Синтаксически управляемая трансляция	4	ПК-4, ПК-8	экзамен
10.	Схемы трансляции	6	ПК-4, ПК-7, ПК-8	экзамен
11.	Аннотированное дерево разбора	6	ПК-4, ПК-7, ПК-8	экзамен
12.	Классы методов разбора	7	ПК-4, ПК-8	экзамен

## 5. ФОНД ОЦЕНОЧНЫХ СРЕДСТВ ДЛЯ ПРОВЕДЕНИЯ ПРОМЕЖУТОЧНОЙ АТТЕСТАЦИИ ОБУЧАЮЩИХСЯ ПО ДИСЦИПЛИНЕ

Фонд оценочных средств для проведения промежуточной аттестации обучающихся по дисциплине представлен в виде оценочных материалов и приведен в **Приложении** к рабочей программе дисциплины (см. документ «Оценочные материалы по дисциплине «Разработка компиляторов»).

## 6. УЧЕБНО-МЕТОДИЧЕСКОЕ ОБЕСПЕЧЕНИЕ ДИСЦИПЛИНЫ

### 6.1. Основная литература

1. Карпов Ю.Г. Теория и технология программирования. Основы построения трансляторов: учеб. пособие / СПб.: БХВ-Петербург, 2005. — 272 с. (21 экз. в БФ РГРТУ)
2. Богданов В.С. Теория языков программирования и методы трансляции: метод. ук. / В.С. Богданов, РГРТА — Рязань, 2003. — 48 с. (78 экз. в БФ РГРТУ)
3. Богданов В.С. Формальные грамматики и языки: метод. ук. к реш. зад. / В.С. Богданов, РГРТУ — Рязань, 2007. — 36 с. № 3957 (79 экз. в БФ РГРТУ)
4. Маркин А.В. Основы построения трансляторов: метод. ук. к л. раб. / А.В. Маркин, РГРТА — Рязань, 1999. — 44 с. № 2875 (31 экз. в БФ РГРТУ)

5. Пентус А.Е. Математическая теория формальных языков [Электронный ресурс] / А.Е. Пентус, М.Р. Пентус. — Электрон. текстовые данные. — М.: Интернет-Университет Информационных Технологий (ИНТУИТ), 2016. — 218 с. — 5-9556-0062-0. — Режим доступа: <http://www.iprbookshop.ru/52201.html>

6. Малявко А.А. Формальные языки и компиляторы [Электронный ресурс]: учебник / А.А. Малявко. — Электрон. текстовые данные. — Новосибирск: Новосибирский государственный технический университет, 2014. — 431 с. — 978-5-7782-2318-9. — Режим доступа: <http://www.iprbookshop.ru/47725.html>

7. Малявко А.А. Системное программное обеспечение. Формальные языки и методы трансляции. Часть 1 [Электронный ресурс]: учебное пособие / А.А. Малявко. — Электрон. текстовые данные. — Новосибирск: Новосибирский государственный технический университет, 2010. — 104 с. — 978-5-7782-1429-3. — Режим доступа: <http://www.iprbookshop.ru/45017.html>

8. Малявко А.А. Системное программное обеспечение. Формальные языки и методы трансляции. Часть 2 [Электронный ресурс]: учебное пособие / А.А. Малявко. — Электрон. текстовые данные. — Новосибирск: Новосибирский государственный технический университет, 2011. — 160 с. — 978-5-7782-1668-6. — Режим доступа: <http://www.iprbookshop.ru/45018.html>

9. Малявко А.А. Системное программное обеспечение. Формальные языки и методы трансляции. Часть 3 [Электронный ресурс]: учебное пособие / А.А. Малявко. — Электрон. текстовые данные. — Новосибирск: Новосибирский государственный технический университет, 2012. — 120 с. — 978-5-7782-1960-1. — Режим доступа: <http://www.iprbookshop.ru/45019.html>

10. Молдованова О.В. Языки программирования и методы трансляции [Электронный ресурс]: учебное пособие / О.В. Молдованова. — Электрон. текстовые данные. — Новосибирск: Сибирский государственный университет телекоммуникаций и информатики, 2012. — 134 с. — 2227-8397. — Режим доступа: <http://www.iprbookshop.ru/54809.html>

## **6.2. Дополнительная литература**

11. Молчанов А.Ю. Системное программное обеспечение: учебник / А.Ю. Молчанов—3-е изд. — СПб.: БХВ-Петербург, 2010. — 397 с. (21 экз. в БФ РГРТУ)

12. Ахо А., Сети Р., Ульман Дж. Компиляторы: принципы, технологии и инструменты: пер. с англ. — М.: Вильямс, 2001. — 767 с. (2 экз. в БФ РГРТУ)

13. Хопкрофт Дж., Мотвани Р., Ульман Дж. Введение в теорию автоматов, языков и вычислений: 2-е издание. — М.: Вильямс, 2002. — 528 с. (1 экз. в БФ РГРТУ)

14. Теория и реализация языков программирования [Электронный ресурс] / В.А. Серебряков [и др.]. — 2-е изд. — Электрон. текстовые данные. — М.: Интернет-Университет Информационных Технологий (ИНТУИТ), 2016. — 372 с. — 2227-8397. — Режим доступа: <http://www.iprbookshop.ru/73731.html>

15. Разработка компиляторов [Электронный ресурс] / Н.Н. Вояковская [и др.]. — 2-е изд. — Электрон. текстовые данные. — М.: Интернет-Университет Информационных Технологий (ИНТУИТ), 2016. — 374 с. — 2227-8397. — Режим доступа: <http://www.iprbookshop.ru/73654.html>

## **6.3. Методические указания для обучающихся по освоению дисциплины**

### **6.3.1. Методические указания к освоению лекционного материала**

*Для освоения лекционного материала следует:* изучить конспект лекции в тот же день, после лекции: 10 – 15 минут, повторно прочитать конспект лекции за день перед следующей лекцией: 10 – 15 минут. Также следует изучить теоретический лекционный материал по рекомендуемому учебнику/учебному пособию: 1 час в неделю. Следует максимально использовать лекционное время для изучения дисциплины, понимания лекционного материала и написания конспекта лекций. В процессе лекционного занятия студент должен уметь выделять важные моменты и основные положения.

При написании *конспекта лекций* следует придерживаться следующих правил и рекомендаций.

1. При ведении конспекта рекомендуется структурировать материал по разделам, главам, темам. Вести нумерацию формул, схем, рисунков. Выделять по каждой теме постановку задачи, основные положения, выводы. Кратко записывать те пояснения лектора, которые оказались особенно важными. Это позволит при подготовке к сдаче зачёта и экзамена не запутаться в структуре лекционного материала.

2. Лекционный материал следует записывать в конспект лишь после того, как излагаемый лектором тезис будет дослушан до конца и понят.

3. При конспектировании следует отмечать непонятные, на данном этапе, положения.

### **6.3.2. Методические указания к практическим/лабораторным занятиям**

*Практические занятия* по дисциплине существенно дополняют лекции. В процессе анализа теоретических положений и решения практических задач студенты расширяют и углубляют свои знания, полученные из лекционного курса и учебников, приобретают умение применять общие закономерности к конкретным случаям. В процессе решения задач развивается логическое мышление, и вырабатываются навыки вычислений, работы со справочной литературой. Практические занятия способствуют закреплению знаний и практических навыков, формированию конструктивного стиля мышления, расширению кругозора.

При подготовке к практическому занятию необходимо внимательно ознакомиться с соответствующим теоретическим материалом по конспекту лекций и рекомендуемому учебнику, затем изучить конспект или материалы предыдущего практического занятия и выполнить заданное расчетное задание: 1 – 2 часа в неделю. Следует максимально использовать аудиторное время практических занятий. В процессе занятия студент должен активно участвовать в дискуссиях, обсуждениях и решениях практических задач и вести *конспект практических занятий* отдельно от конспекта лекций. Дополнительно в часы самостоятельной работы студенты могут повторно решить задачи, с которыми они плохо освоились во время аудиторных занятий, и обязательно те задачи, которые не получились дома при предыдущей подготовке к практическим занятиям.

*Подготовка к лабораторным работам.* Перед началом лабораторной работы необходимо ознакомиться с методическими указаниями к данной лабораторной работе, внимательно ознакомиться с заданием и желательно заранее выполнить подготовку программного проекта в используемой инструментальной среде, чтобы время лабораторного занятия использовать для исправления ошибок, модификации проекта и защиты данной работы.

Выполнение каждой из запланированных работ заканчивается предоставлением отчета. Требования к форме и содержанию отчета приведены в методических указаниях к лабораторным работам или определяются преподавателем на первом занятии.

*Отчет по лабораторной работе* студент должен начать оформлять еще на этапе подготовки к ее выполнению. Для допуска к лабораторной работе, студент должен представить преподавателю «заготовку» отчета, содержащую: оформленный титульный лист или название и номер работы при ведении общего конспекта, цель работы, задание, проект решения, и при наличии полученные результаты, выводы.

Изучение методических указаний к лабораторной работе – 2 часа перед выполнением лабораторной работы и в ходе разработки проекта и 2 часа для оформления отчета, отладки проекта и подготовки к сдаче работы. После выполнения лабораторной работы необходимо согласовать полученные результаты с преподавателем.

Важным этапом является *защита лабораторной работы*. В процессе защиты студент отвечает на вопросы преподавателя, касающиеся теоретического материала, относящегося к данной работе, и проекта, реализующего его задание, комментирует полученные в ходе работы результаты. При подготовке к защите лабораторной работы рекомендуется ознакомиться со списком вопросов по изучаемой теме и попытаться самостоятельно на них ответить, используя конспект лекций и рекомендуемую литературу, информационные ресурсы сети Интернет.

### 6.3.3. Методические указания к подготовке и сдаче экзамена

*Экзамен* – форма промежуточной проверки знаний, умений, навыков, степени освоения дисциплины. Главная задача экзамена состоит в том, чтобы у студента по окончании изучения данной дисциплины сформировались определенное представление об общем содержании дисциплины, определенные теоретические знания и практические навыки, определенный кругозор. Готовясь к экзамену, студент приводит в систему знания, полученные на лекциях, на практических и лабораторных занятиях, разбирается в том, что осталось непонятным, и тогда изучаемая им дисциплина может быть воспринята в полном объеме с присущей ей строгостью и логичностью, ее практической направленностью.

Экзамены дают возможность преподавателю определить теоретические знания студента и его практические навыки при решении определенных прикладных задач. Оцениваются: понимание и степень усвоения теоретического материала; степень знакомства с основной и дополнительно литературой, а также с современными публикациями; умение применить теорию к практике, решать определенные практические задачи данной предметной области, правильно проводить расчеты и т. д.; знакомство с историей данной науки; логика, структура и стиль ответа, умение защищать выдвигаемые положения.

Значение экзаменов не ограничивается проверкой знаний, являясь естественным завершением обучения студента по данной дисциплине, они способствуют обобщению и закреплению знаний и умений, приведению их в стройную систему, а также устранению возникших в процессе обучения пробелов.

*Подготовка к экзамену* – это тщательное изучение и систематизация учебного материала, осмысление и запоминание теоретических положений, формулировок, формул, установление и осмысление внутри предметных связей между различными темами и разделами дисциплины, закрепление теоретических знаний путем решения определенных задач.

Перед экзаменом назначается *консультация*, ее цель – дать ответы на вопросы, возникшие в ходе самостоятельной подготовки студента, студент имеет возможность получить ответ на все неясные ему вопросы, кроме того, преподаватель будет отвечать на вопросы других студентов, что будет способствовать повторению и закреплению знаний всех присутствующих. Преподаватель на консультации, как правило, обращает внимание на те разделы, по которым на предыдущих экзаменах ответы были неудовлетворительными, а также фиксирует внимание на наиболее трудных разделах или темах курса.

На непосредственную подготовку к экзамену обычно дается 3 – 5 дней. Этого времени достаточно для углубления, расширения и систематизации знаний, полученных в ходе обучения, на устранение пробелов в знании отдельных вопросов, для определения объема ответов на каждый из вопросов рабочей программы дисциплины.

Планируя подготовку, обучаемый должен учитывать: неоднородность в сложности учебного материала и степени его проработки в ходе обучения, свои индивидуальные способности. Рекомендуется делать перерывы в занятиях через каждые 50-60 минут на 10 минут. После 3-4 часов занятий следует сделать часовой перерыв. Целесообразно разделять весь рабочий день на три рабочих периода – с утра до обеда, с обеда до ужина и с ужина до сна. Каждый рабочий период дня должен заканчиваться отдыхом не менее 1 часа. Работая в сессионном режиме, студент имеет возможность увеличить время занятий с 8 (как требовалось в семестре) до 10 часов в сутки. Подготовку к экзаменам следует начинать с общего планирования своей деятельности, с определения объема материала, подлежащего проработке, необходимо внимательно сверить свои конспекты с программой дисциплины, чтобы убедиться, все ли разделы отражены в лекциях, отсутствующие темы изучить по учебнику. Второй этап предусматривает системное изучение материала по данному предмету с обязательной записью всех выкладок, выводов, формул. На третьем этапе – этапе закрепления – полезно чередовать углубленное повторение особенно сложных вопросов с беглым повторением всего материала.

### 6.3.4. Рекомендации по работе с литературой

Теоретический материал курса становится более понятным, когда дополнительно к прослушиванию лекции и изучению конспекта изучаются и книги по данному предмету. Литературу по дисциплине рекомендуется читать как в бумажном, так и в электронном виде (если отсутствует бумажный аналог). Полезно использовать несколько учебников и пособий по дисциплине. Рекомендуется после изучения очередного параграфа ответить на несколько вопросов по данной теме. Полезно мысленно задать себе следующие вопросы (и попробовать ответить на них): «о чем этот параграф?», «какие новые понятия введены, каков их смысл?», «зачем мне это нужно по специальности?».

Рекомендуется самостоятельно изучать материал, который еще не прочитан на лекции и не применялся на лабораторном или практическом занятии, тогда занятия будут гораздо понятнее. В течение недели рекомендуется выбрать время (1 час) для работы с литературой.

### 6.3.5. Образовательные технологии, используемые в учебном процессе

Изучение дисциплины построено на методике *когнитивного или проблемного обучения*, как наиболее эффективной технологии при подготовке специалиста технического профиля. В рамках такого обучения студенты с самого начала учатся мыслить как программные инженеры, концентрировать внимание на решаемой проблеме, изучать требования и особенности проектирования перед созданием кода программы, работать по итеративной схеме, т.е. вырабатывают в себе привычку думать обо всех деталях, необходимых для разработки больших программ. При подготовке программных инженеров программирование остается основополагающим навыком, чем раньше и дольше студенты будут программировать, тем лучшими специалистами они смогут стать в дальнейшем, большинство возможностей их будущей работы требуют наличия профессиональных навыков программирования, поэтому необходимо делать основной упор на получение таких навыков. Применение данного подхода способствует тому, что студенты с самого начала чувствуют, что обучаются выбранной специальности. Для обеспечения результативности профессиональной практики рекомендуется поддерживать рейтинговую систему оценки знаний текущей успеваемости, выполнения лабораторных заданий. Активно поощряется профессионально выполненная работа с обязательным профессиональным оформлением отчетов.

Принятая технология обучения предусматривает применение активных форм обучения, таких как интерактивные методы чтения лекций и проведения практических и лабораторных занятий, организация устных презентаций в группе для конструктивного обсуждения и оценивания выполненных студенческих работ с целью демонстрации прогресса в обучении и для выработки у студентов лучшего понимания изучаемой проблематики, использование стандартизированных тестов для отслеживания и оценивания общего прогресса в группе и у каждого студента в отдельности. Чтение лекций и проведение практических и лабораторных занятий осуществляется с использованием мультимедийных средств и раздаточных материалов, с привлечением обучающихся и контролирующих учебных курсов.

## 7. ПЕРЕЧЕНЬ РЕСУРСОВ ИНФОРМАЦИОННО-ТЕЛЕКОММУНИКАЦИОННОЙ СЕТИ ИНТЕРНЕТ, НЕОБХОДИМЫХ ДЛЯ ОСВОЕНИЯ ДИСЦИПЛИНЫ

1. Электронно-библиотечная система «Лань». – Режим доступа: с любого компьютера РГРТУ без пароля. – URL: <https://e.lanbook.com/>
2. Электронно-библиотечная система «IPRbooks». – Режим доступа: с любого компьютера РГРТУ без пароля, из сети Интернет по паролю. – URL: <https://iprbookshop.ru/>.
3. Электронная библиотека РГРТУ. – URL: <http://weblib.rtu/ebs>.
4. Научная электронная библиотека eLibrary. – URL: <http://e.lib/vlsu.ru/www.uisrussia.msu.ru/elibrary.ru>
5. Библиотека и форум по программированию. – URL: <http://www.cyberforum.ru>
6. Национальный открытый университет ИНТУИТ. – URL: <http://www.intuit.ru/>
7. Информационно-справочная система. – URL: <http://window.edu.ru>



## 8. ПЕРЕЧЕНЬ ИНФОРМАЦИОННЫХ ТЕХНОЛОГИЙ, ИСПОЛЬЗУЕМЫХ ПРИ ОСУЩЕСТВЛЕНИИ ОБРАЗОВАТЕЛЬНОГО ПРОЦЕССА ПО ДИСЦИПЛИНЕ, ВКЛЮЧАЯ ПЕРЕЧЕНЬ ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ И ИНФОРМАЦИОННЫХ СПРАВОЧНЫХ СИСТЕМ

1. Операционная система Windows XP (MicrosoftImagine, номерподписки 700102019, бессрочно).
2. Операционная система Windows XP (MicrosoftImagine, номерподписки ID 700565239, бессрочно).
3. OpenOffice (OpenOffice.org, OO.org, OO.o, OOo) – свободно распространяемое ПО.
4. MicrosoftOfficeVisio (MicrosoftImagine:Номер подписки 700102019, бессрочно).
5. Свободно распространяемое ПО: Microsoft Visual Studio Express (<https://www.visualstudio.com/ru/vs/express>), Qt Framework 5.10 Open Source LGPL license (<https://www.qt.io/>).

## 9. МАТЕРИАЛЬНО-ТЕХНИЧЕСКОЕ ОБЕСПЕЧЕНИЕ ДИСЦИПЛИНЫ

Для освоения дисциплины необходимы следующие материально-технические ресурсы:

- 1) аудитория для проведения лекционных и практических занятий, групповых и индивидуальных консультаций, текущего контроля, промежуточной аттестации, оборудованная проектором и маркерной (меловой) доской;
- 2) классы, оснащенные персональными компьютерами для проведения лабораторных занятий;
- 3) аудитория для самостоятельной работы, оснащенная компьютерной техникой с подключением к локальной вычислительной сети и сети Интернет:

Наименование специальных помещений и помещений для самостоятельной работы	Оснащенность специальных помещений и помещений для самостоятельной работы	Перечень лицензированного ПО. Реквизиты подтверждающего документа
Учебная аудитория для проведения практический занятий, лабораторных работ и самостоятельной работы №206-3	Проектор: InFocus LP640 18 ПК с возможностью подключения к сети «Интернет» и обеспечением доступа в электронную информационно-образовательную среду: ЦП: Intel Pentium 4 class 2800 ОЗУ: 1 Гб ПЗУ: 50 Гб (11 шт.) ЦП: Intel Pentium 4 class 3200 ОЗУ: 1 Гб ПЗУ: 50 Гб (5 шт.) ЦП: Intel Pentium 4 class 2800 ОЗУ: 500 Мб ПЗУ: 50 Гб (1 шт.) ЦП: Intel Pentium 4 class 2800 ОЗУ: 2 Гб ПЗУ: 50 Гб (1 шт.)	1. Операционная система Windows XP/Vista/7/8/10 (Microsoft Imagine: Номер подписки 700102019, бессрочно). 2. Microsoft Visual Studio (Microsoft Imagine: Номер подписки 700102019, бессрочно). 3. Microsoft Office Access (Microsoft Imagine: Номер подписки 700102019, бессрочно). 4. Microsoft Office Visio (Microsoft Imagine: Номер подписки 700102019, бессрочно). 5. Microsoft SQL Server (Microsoft Imagine: Номер подписки 700102019, бессрочно). 6. Microsoft Project (Microsoft Imagine: Номер подписки 700102019, бессрочно). 7. Свободно распространяемое программное обеспечение под лицензиями GNU, Apache, Oracle, Mozilla, CeCILL.

<p>Учебная аудитория для проведения практической работы и самостоятельной работы <b>№206-4</b></p>	<p>18 ПК с возможностью подключения к сети «Интернет» и обеспечением доступа в электронную информационно-образовательную среду: ЦП: Pentium 4 class 2800 ОЗУ: 1 Гб ПЗУ: 50 Гб (8 шт.) ЦП: Intel Pentium II/III class 2327 ОЗУ: 2 Гб ПЗУ: 50 Гб (10 шт.)</p>	<ol style="list-style-type: none"> <li>1. Операционная система Windows XP/Vista/7/8/10 (Microsoft Imagine: Номер подписки 700102019, бессрочно).</li> <li>2. Microsoft Visual Studio (Microsoft Imagine: Номер подписки 700102019, бессрочно).</li> <li>3. Microsoft Office Access (Microsoft Imagine: Номер подписки 700102019, бессрочно).</li> <li>4. Microsoft Office Visio (Microsoft Imagine: Номер подписки 700102019, бессрочно).</li> <li>5. Microsoft SQL Server (Microsoft Imagine: Номер подписки 700102019, бессрочно).</li> <li>6. Microsoft Project (Microsoft Imagine: Номер подписки 700102019, бессрочно).</li> <li>7. Свободно распространяемое программное обеспечение под лицензиями GNU, Apache, Oracle, Mozilla, CeCILL.</li> </ol>
<p>Учебная аудитория для проведения практической работы и самостоятельной работы <b>№206-5</b></p>	<p>24 ПК с возможностью подключения к сети «Интернет» и обеспечением доступа в электронную информационно-образовательную среду: ЦП: Intel Pentium II/III class 2394 ОЗУ: 2 Гб ПЗУ: 70 Гб (17 шт.) ЦП: Intel Pentium II/III class 2327 ОЗУ: 2 Гб ПЗУ: 100 Гб (1 шт.) ЦП: Intel Pentium III Xeon 3093 ОЗУ: 4 Гб ПЗУ: 300 Гб (6 шт.)</p>	<ol style="list-style-type: none"> <li>1. Операционная система Windows XP/Vista/7/8/10 (Microsoft Imagine: Номер подписки 700102019, бессрочно).</li> <li>2. Microsoft Visual Studio (Microsoft Imagine: Номер подписки 700102019, бессрочно).</li> <li>3. Microsoft Office Access (Microsoft Imagine: Номер подписки 700102019, бессрочно).</li> <li>4. Microsoft Office Visio (Microsoft Imagine: Номер подписки 700102019, бессрочно).</li> <li>5. Microsoft SQL Server (Microsoft Imagine: Номер подписки 700102019, бессрочно).</li> <li>6. Microsoft Project (Microsoft Imagine: Номер подписки 700102019, бессрочно).</li> <li>7. Свободно распространяемое программное обеспечение под лицензиями GNU, Apache, Oracle, Mozilla, CeCILL.</li> </ol>