

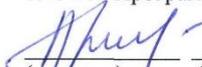


МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ РОССИЙСКОЙ ФЕДЕРАЦИИ  
Федеральное государственное автономное образовательное учреждение высшего образования  
«Дальневосточный федеральный университет»  
(ДФУ)

**ШКОЛА ЕСТЕСТВЕННЫХ НАУК**

«СОГЛАСОВАНО»

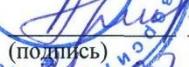
Руководитель ОП Разработка программно-информационных систем по направлению 09.04.04 Программная инженерия

  
(подпись) Артемяева И.Л.  
(Ф.И.О. рук. ОП) « 21 » 07 2018 г.



«УТВЕРЖДАЮ»

Заведующая кафедрой прикладной математики, механики, управления и программного обеспечения

  
(подпись) Артемяева И.Л.  
(Ф.И.О. зав. каф.) « 21 » 07 2018 г.

**РАБОЧАЯ ПРОГРАММА УЧЕБНОЙ ДИСЦИПЛИНЫ**

Разработка формальных языков и языковых процессоров

**Направление подготовки 09.04.04 Программная инженерия**

магистерская программа «Разработка программно-информационных систем»

**Форма подготовки очная**

курс 1 семестр 2

лекции 6 час.

практические занятия 0 час.

лабораторные работы 30 час.

в том числе с использованием МАО лек. 0 /пр. 0 /лаб. 18 час.

в том числе в электронной форме лек. 0 /пр. 0 /лаб. 0 час.

всего часов аудиторной нагрузки 36 час.

в том числе с использованием МАО 18 час.

в том числе контролируемая самостоятельная работа 0 час.

в том числе в электронной форме 0 час.

самостоятельная работа 72 час.

в том числе на подготовку к экзамену 0 час.

курсовая работа / курсовой проект    семестр

зачет 2 семестр

экзамен    семестр

Рабочая программа составлена в соответствии с требованиями образовательного стандарта, самостоятельно устанавливаемого ДВФУ, утвержденного приказом ректора от 07.07.2015 № 12-13-1282.

Рабочая программа обсуждена на заседании кафедры прикладной математики, механики, управления и программного обеспечения, протокол № 7.2 от 21.07.2018 г.

Заведующая кафедрой прикладной математики, механики, управления и программного обеспечения д.т.н., профессор Артемяева И.Л.

Составитель: научный сотрудник кафедры прикладной математики, механики, управления и программного обеспечения, к.т.н. Рябченко Н.В.

**Оборотная сторона титульного листа РПУД**

**I. Рабочая программа пересмотрена на заседании кафедры:**

Протокол от «\_\_\_\_\_» \_\_\_\_\_ 20\_\_ г. № \_\_\_\_\_

Заведующий кафедрой \_\_\_\_\_  
(подпись) (И.О. Фамилия)

**II. Рабочая программа пересмотрена на заседании кафедры:**

Протокол от «\_\_\_\_\_» \_\_\_\_\_ 20\_\_ г. № \_\_\_\_\_

Заведующий кафедрой \_\_\_\_\_  
(подпись) (И.О. Фамилия)

## ABSTRACT

**Master's degree in** 09.04.04 – Software engineering

**Master's Program** “Development of software and information systems”

**Course title:** The design of formal languages and language processors

**Variable part of Block, 3 credits**

**Instructor:** Ryabchenko N.

**At the beginning of the course a student should be able to:** study independently, be self-organized; know about main concepts, principles, theories and facts related to computer science; have knowledge of the structures of computers and systems; use the bases of programming and computer science for the design and the tests of a software product; apply main methods and tools of the design of software; work with various technologies of software design

**Learning outcomes:** an ability to design translators and interpreters of programming languages; an ability to design the main elements of operating systems

**Course description:** the modern methods of the design of formal languages, principles, methods and the tools of language processor development

**Main course literature:**

1. Serebryakov V.A., Galochkin M.P., Gonchar D.R. Teoriya i realizatsiya yazykov programmirovaniya [Theory and realization of programming languages]. – Moscow: MZ Press, 2012. – 352 p. (rus) – Access:

<http://lib.dvfu.ru:8080/lib/item?id=chamo:704386&theme=FEFU>

2. Kotlyarov V.P., Kolikova T.V. Osnovy testirovaniya programmnoho obespecheniya [Bases of software testing]. – Moscow: BINOM, 2016. – 285 p. (rus) – Access:

<http://lib.dvfu.ru:8080/lib/item?id=chamo:797490&theme=FEFU>

3. Artemeva I.L. Teoriya vychislitelnykh protsessov i structur. T. 1: Yazyki i sposoby ikh zadaniya. [Theory of computational processes and structures. P. 1. Languages and methods of their determination]. – Vladivostok: Far Eastern Federal University, 2011. – 60 p. (rus)

4. Artemeva I.L. Teoriya vychislitelnykh protsessov i structur. T. 2: Metody sintaksicheskogo analiza programm. [Theory of computational processes and structures. P. 2. Methods for the syntax analysis of programs]. – Vladivostok: Far Eastern Federal University, 2011. – 56 p. (rus)

5. Gagarina L.G., Kokoreva E.V. Vvedenie v teoriyu algoritmicheskikh yazykov i kompilyatorov [Introduction to the theory of algorithmic languages and compilers]. – Moscow: FORUM, 2011. – 176 p. (rus) – Access:

<http://znanium.com/catalog/product/265617>

6. Virt N. Postroenie kompilyatorov [Building of compilers]. – Moscow: DMK Press, 2010. – 192 p. (rus) – Access:

<http://www.studentlibrary.ru/book/ISBN9785940745853.html>

7. Aho A., Ullman J. Teoriya sintaksicheskogo analiza, perevoda i kompilyatsii. Sintaksicheskiy analiz. [The theory of syntactic analysis, translation and compilation. Syntax analysis]. – M.: Book on Demand, 2012. – 613 p. (rus)

8. Fedoseeva L.I., Adilov R.M., Shmotkin M.N. Osnovy teorii konechnykh avtomatov i formalnykh yazykov [Fundamentals of the theory of finite-state automata and formal languages]. – Penza: Penza State Technical University, 2013. – 136 p. (rus) – Access:

<https://e.lanbook.com/book/62703>

**Form of final control:** pass-fail exam.

## **Аннотация к рабочей программе дисциплины «Разработка формальных языков и языковых процессоров»**

Рабочая программа дисциплины «Разработка формальных языков и языковых процессоров» разработана для студентов 1 курса, обучающихся по направлению 09.04.04 Программная инженерия, магистерская программа «Разработка программно-информационных систем». Дисциплина является обязательной дисциплиной вариативной части учебного плана Б1.В.02.02.

Трудоёмкость дисциплины составляет 3 зачётных единицы (108 часов). Дисциплина реализуется во 2 семестре. Во 2 семестре дисциплина содержит 6 часов лекций, 30 часов лабораторных работ, из них 18 часов лабораторных работ с использованием методов активного обучения; на самостоятельную работу студентов отводится 72 часа.

Дисциплина «Разработка формальных языков и языковых процессоров» базируется на дисциплине «Методология научных исследований в программной инженерии» и дисциплинах бакалавриата, связанных с изучением компиляторов. Знания, полученные при её изучении, будут использованы в дисциплине «Интеллектуальный анализ данных» учебного плана.

**Цель** дисциплины – изучение современных методов разработки формальных языков, принципов, технологий и инструментов разработки языковых процессоров.

### **Задачи:**

- Изучение понятия формальный язык, современных классов языков и их характеристик.
- Изучение методов разработки формальных языков различных классов и построения их формальных моделей.
- Изучение принципов, технологий и инструментов разработки языковых процессоров.
- Разработка формального языка некоторого класса для некоторой предметной области.
- Разработка языкового процессора на основе модели языка.

Для успешного изучения дисциплины «Разработка формальных языков и языковых процессоров» у обучающихся должны быть сформированы следующие предварительные компетенции:

- способность к самоорганизации и самообразованию;
- владение основными концепциями, принципами, теориями и фактами, связанными с информатикой;
- владение архитектурой электронных вычислительных машин и систем;

- готовность применять основы информатики и программирования к проектированию, конструированию и тестированию программных продуктов;
- готовность применять основные методы и инструменты разработки программного обеспечения;
- владение навыками использования различных технологий разработки программного обеспечения.

Планируемые результаты обучения по данной дисциплине (знания, умения, владения), соотнесенные с планируемыми результатами освоения образовательной программы, характеризуют этапы формирования следующих компетенций (общекультурные/ общепрофессиональные/ профессиональные компетенции (элементы компетенций)):

Код и формулировка компетенции	Этапы формирования компетенции	
ПК-10 – способность проектировать трансляторы и интерпретаторы языков программирования	Знает	методы разработки формальных языков, проектирования языковых процессоров языков
	Умеет	разрабатывать модели формальных языков, применять методы проектирования и разработки языковых процессоров.
	Владеет	формальными средствами разработки нового языка, инструментальными средствами разработки программных систем, в том числе языковых процессоров
ПК-12 – способность проектировать основные компоненты операционных систем	Знает	методы разработки формальных языков описания данных и управления заданиями
	Умеет	проектировать и реализовывать языковые процессоры, используемые в различных операционных системах
	Владеет	современными способами и системами разработки инструментальных средств, предназначенных для создания языковых процессоров
ПК-13 – способность проектировать вспомогательные и специализированные языки программирования и языки представления данных	Знает	методы разработки, обоснования и исследования моделей, методов, алгоритмов и программной инфраструктуры, требуемых для создания специализированных языков и языковых процессоров в различных предметных областях
	Умеет	разрабатывать специализированные языки для автоматизации профессиональной деятельности
	Владеет	методами обоснования моделей формальных языков, необходимых для решения задач профессиональной деятельности

Для формирования вышеуказанных компетенций в рамках дисциплины «Разработка формальных языков и языковых процессоров» применяются следующие методы активного/ интерактивного обучения: метод круглого стола и метод проектов.

## **I. СТРУКТУРА И СОДЕРЖАНИЕ ТЕОРЕТИЧЕСКОЙ ЧАСТИ КУРСА**

Содержание теоретической части курса разбивается на два раздела, каждый из которых содержит темы для детального рассмотрения.

### **Раздел I. Формальные языки и методы их разработки (4 час.)**

#### **Тема 1. Формальные языки и их классификация, модель языка**

Рассматривается понятие специализированного языка, структура языка, классификация языков, модель языка, способы представления модели языка, способы определения синтаксиса и семантики языков, порождающие и вычислительные модели.

#### **Тема 2. Язык описания данных**

Рассматривается понятие языка описания данных, методы представления языка описания данных, примеры существующих языков (SQL, ER-модель, схема данных, как вариант представления данных), достоинства и недостатки.

#### **Тема 3. Язык описания знаний**

Рассматривается понятие языка описания знаний, методы представления языка описания знаний, примеры существующих языков, использование формализма семантических сетей, фреймов при разработке моделей языков, анализируются достоинства и недостатки.

#### **Тема 4. Язык, основанный на правилах**

Рассматривается понятие языка, основанного на правилах, методы представления языка, основанного на правилах, примеры существующих языков, анализируются их достоинства и недостатки.

#### **Тема 5. Язык спецификаций**

Рассматривается понятие языка спецификаций, методы представления моделей языков спецификаций, примеры существующих языков, UML, язык предикатов первого порядка без правил вывода, язык множеств, многосортные языки предикатов, анализируются достоинства и недостатки рассмотренных языков.

## **Тема 6. Языки, основанные на объектно-ориентированном формализме**

Рассматривается понятие языка, основанного на объектно-ориентированном формализме, методы представления языка, основанного на объектно-ориентированном формализме, примеры существующих языков и редакторов, KIF, анализируются достоинства и недостатки.

## **Раздел II. Принципы, технологии и инструменты разработки языковых процессоров (2 час.)**

### **Тема 1. Языковые процессоры и их основные типы. Принципы компиляции**

Рассматривается понятие языкового процессора, типы языковых процессоров, интерпретаторы, трансляторы, основные понятия компиляции, грамматики языков, классификация грамматик, структура компилятора, методы реализации компиляторов.

### **Тема 2. Лексический, синтаксический анализ**

Рассматривается понятие лексем, особенности лексического анализа, понятие автомата, методы синтаксического анализа, синтаксическое дерево, особенности реализации методов синтаксического анализа, семантический анализ.

### **Тема 3. Генерация кода, оптимизация**

Рассматривается понятие промежуточного представления, методы генерации кода, существующие методы оптимизации кода.

## **II. СТРУКТУРА И СОДЕРЖАНИЕ ПРАКТИЧЕСКОЙ ЧАСТИ КУРСА**

Практическая часть курса включает в себя 30 часов лабораторных работ.

### **Практические занятия**

Не предусмотрено.

### **Лабораторные работы (30 час.)**

#### **Лабораторная работа №1. Классы формальных языков (2 час.)**

Студентам или бригаде студентов предлагается выбрать задачу некоторой предметной области, позволяющую использовать формальный язык для ее решения. Предложенные задачи выносятся на рассмотрение

круглого стола с вынесением предложений по ее усовершенствованию. Ставится задача по реализации компилятора для создаваемого языка.

### **Лабораторная работа №2. Описание формального языка (2 час.)**

Студенты ставят задачу, с внесением изменений, предложенном на круглом столе, описывают основные объекты формального языка, свойства этих объектов, операции и отношения между ними. На неформальном языке определяются условия выполнения операций.

### **Лабораторная работа №3. Модель формального языка (2 час.)**

Студенты описывают синтаксис выбранного языка, семантику этого языка и строят порождающую или вычислительную модель языка.

### **Лабораторная работа №4. Постановка задачи для компилятора (2 час.)**

Студенты ставят неформальную постановку задачи для реализации программного обеспечения. Описывают требования разрабатываемой системы.

### **Лабораторная работа №5. Эскизное проектирование компилятора (2 час.)**

Студенты разрабатывают эскизный проект будущего «классического» компилятора и принимают при этом основные проектные решения, определяя представление всех требуемых структур данных.

### **Лабораторная работа №6. Проектирование лексического анализатора (2 час.)**

Студенты определяют классы лексем на входном языке, структуру токенов, спецификацию, проект лексического анализатора, тесты.

### **Лабораторная работа №7. Реализация лексического анализатора (2 час.)**

Студенты реализуют разработанный лексический анализатор, отлаживают его и проходят все разработанные тесты.

### **Лабораторная работа №8. Проектирование синтаксического анализатора (2 час.)**

Студенты определяют метод реализации синтаксического анализатора, спецификацию, прорабатывают структуры для синтаксического дерева,

проект синтаксического и семантического анализатора, тесты для синтаксического и семантического анализа.

**Лабораторная работа №9. Реализация синтаксического анализатора (2 час.)**

Студенты реализуют разработанный синтаксический и семантический анализатор, отлаживают его и проходят все разработанные тесты.

**Лабораторная работа №10. Проектирование генератора кода (2 час.)**

Студенты определяют метод реализации генератора кода (в некоторых случаях интерпретатора), разрабатывают проект нижнего уровня для программной системы.

**Лабораторная работа №11. Реализация генератора кода (2 час.)**

Студенты реализуют разработанный генератор кода (в некоторых случаях интерпретатор), отлаживают его и проходят все разработанные тесты.

**Лабораторная работа №12. Оптимизация генератора кода (2 час.)**

Студенты реализуют выбранные методы оптимизации кода.

**Лабораторная работа №13. Сборка, отладка и тестирование компилятора (2 час.)**

Студенты осуществляют слияние разработанных программных средств, отлаживают передачу данных и управления, тестируют общую систему.

**Лабораторная работа №14. Подготовка документации по разработке компилятора и оформление отчета (4 час.)**

Студенты подготавливают и оформляют полный комплект документации по разработке формального языка и языкового процессора этого языка.

### **III. УЧЕБНО-МЕТОДИЧЕСКОЕ ОБЕСПЕЧЕНИЕ САМОСТОЯТЕЛЬНОЙ РАБОТЫ ОБУЧАЮЩИХСЯ**

Трудоёмкость самостоятельной работы студента – 72 часа.

Учебно-методическое обеспечение самостоятельной работы обучающихся по дисциплине «Разработка формальных языков и языковых процессоров» представлено в Приложении 1 и включает в себя:

план-график выполнения самостоятельной работы по дисциплине, в том числе примерные нормы времени на выполнение по каждому заданию;

характеристика заданий для самостоятельной работы обучающихся и методические рекомендации по их выполнению;

требования к представлению и оформлению результатов самостоятельной работы;

критерии оценки выполнения самостоятельной работы.

#### IV. КОНТРОЛЬ ДОСТИЖЕНИЯ ЦЕЛЕЙ КУРСА

№ п/п	Контролируемые разделы/темы дисциплины	Коды и этапы формирования компетенций		Оценочные средства - наименование	
				текущий контроль	промежуточная аттестация
1	Раздел I. Формальные языки и методы их разработки. (Разработка формального языка для решения задач предметной области)	ПК-10 ПК-12	знает	Круглый стол УО-4	Вопросы к зачету № 1-8
			умеет владеет	Лабораторные работы № 1-7 ПР-6	
2	Раздел II. Принципы, технологии и инструменты разработки языковых процессоров	ПК-10 ПК-12	Знает	Круглый стол УО-4	Вопросы к зачету № 9-19
			умеет владеет	Лабораторные работы № 7-14 ПР-6	

Типовые контрольные задания, методические материалы, определяющие процедуры оценивания знаний, умений и навыков и (или) опыта деятельности, а также критерии и показатели, необходимые для оценки знаний, умений, навыков и характеризующие этапы формирования компетенций в процессе освоения образовательной программы, представлены в Приложении 2.

#### V. СПИСОК УЧЕБНОЙ ЛИТЕРАТУРЫ И ИНФОРМАЦИОННО-МЕТОДИЧЕСКОЕ ОБЕСПЕЧЕНИЕ ДИСЦИПЛИНЫ

## Основная литература

(электронные и печатные издания)

1. Серебряков, В. А. Теория и реализация языков программирования: учебное пособие / В. А. Серебряков, М. П. Галочкин, Д. Р. Гончар и др. – М. : МЗ Пресс, 2012. – 352 с.  
<http://lib.dvfu.ru:8080/lib/item?id=chamo:704386&theme=FEFU>
2. Котляров, В. П. Основы тестирования программного обеспечения: учебное пособие / В. П. Котляров, Т. В. Коликова. – М. : Интернет-Университет Информационных Технологий: БИНОМ. Лаборатория знаний, 2016. – 285 с. <http://lib.dvfu.ru:8080/lib/item?id=chamo:797490&theme=FEFU>
3. Артемьева И. Л. Теория вычислительных процессов и структур. Часть 1. Языки и способы их задания. Учебно-методическое пособие. – Владивосток : Издательство Дальневосточного федерального университета, 2011. – 60 с.
4. Артемьева И.Л. Теория вычислительных процессов и структур. Часть 2. Методы синтаксического анализа программ. Учебно-методическое пособие. – Владивосток : Издательство Дальневосточного федерального университета, 2011. – 56 с.
5. Гагарина, Л. Г. Введение в теорию алгоритмических языков и компиляторов: учеб. пособие / Л. Г. Кокорева, Е. В. Кокорева. – М. : ИД ФОРУМ, 2011. – 176 с. <http://znanium.com/catalog/product/265617>
6. Вирт, Н. Построение компиляторов / Н. Вирт // Пер. с англ. Е. В. Борисов, Л. Н. Чернышов. – М. : ДМК Пресс, 2010. – 192 с.  
<http://www.studentlibrary.ru/book/ISBN9785940745853.html>
7. Ахо, А. Теория синтаксического анализа, перевода и компиляции. Синтаксический анализ / А. Ахо, Дж. Ульман. – М. : Книга по требованию, 2012. – 613 с.
8. Федосеева Л.И., Адилов Р.М., Шмоткин М.Н. Основы теории автоматов и формальных языков: учебное пособие . – Пенза : Издательство ПензГТУ, 2013. – 136 с. <https://e.lanbook.com/book/62703>

## Дополнительная литература

(печатные и электронные издания)

1. Липаев, В. В. Программная инженерия: методологические основы: учебник для вузов / В. В. Липаев. – М. : ТЕИС, 2006. – 606 с.  
<http://lib.dvfu.ru:8080/lib/item?id=chamo:248067&theme=FEFU>

2. Ахо, А. Компиляторы. Принципы, технологии и инструментарий / А. Ахо, Р. Сети, Дж. Ульман. – М. : Издательский дом «Вильямс», 2001. – 767 с. <https://lib.dvfu.ru:8443/lib/item?id=chamo:334968&theme=FEFU>
3. Ахо, А. Теория синтаксического анализа, перевода и компиляции. Синтаксический анализ / А. Ахо, Дж. Ульман. – М. : Мир, 1978. – 612 с. <https://lib.dvfu.ru:8443/lib/item?id=chamo:295041&theme=FEFU>

### **Перечень ресурсов информационно-телекоммуникационной сети «Интернет»**

1. <http://linux-doc.ru/programming/assembler/book/compilers.pdf> Ахо А. В., Компиляторы. Принципы, технологии и инструментарий / Ахо А.В. Лам М. С., Сети Р., Ульман Дж. – М. : ООО «ИД «Вильямс», 2008. – 1184 с.
2. <https://e.lanbook.com/book/66125> Залогова, Л. А. Разработка Паскаль-компилятора / Л. А. Залогова. – М. : Лаборатория знаний, 2014. – 186 с.
3. <https://e.lanbook.com/book/39992> Вирт, Н. Разработка операционной системы и компилятора. Проект Оберон / Н. Вирт, Ю. Гуткнехт. – М. : ДМК Пресс, 2012. – 560 с.
4. <https://www.piter.com/collection/biblioteka-programmista> Библиотека программиста
5. <http://www.coders-library.ru/> Библиотека программиста: все, что нужно для комфортной работы

### **Перечень информационных технологий и программного обеспечения**

LibreOffice или Microsoft Office, интернет.

## **VI. МЕТОДИЧЕСКИЕ УКАЗАНИЯ ПО ОСВОЕНИЮ ДИСЦИПЛИНЫ**

Дисциплина «Разработка формальных языков и языковых процессоров» изучается в следующих организационных формах: лекционное занятие; лабораторное занятие; самостоятельное изучение теоретического материала; самостоятельное выполнение индивидуального проекта; индивидуальные и групповые консультации.

Основной формой самостоятельной работы студента является изучение конспекта лекций, их дополнение рекомендованной литературой, выполнение проекта, а также активная работа на лабораторных занятиях.

К прослушиванию лекции следует готовиться, для этого необходимо знать программу курса и рекомендованную литературу. Тогда в процессе лекции легче отделить главное от второстепенного, легче сориентироваться: что записать, что самостоятельно проработать, что является трудным для понимания, а что легко усвоить.

Контроль за выполнением самостоятельной работы студента производится в виде контроля каждого этапа работы, отраженного в документации и защиты проекта.

Студент должен планировать график самостоятельной работы по дисциплине и придерживаться его.

## **VII. МАТЕРИАЛЬНО-ТЕХНИЧЕСКОЕ ОБЕСПЕЧЕНИЕ ДИСЦИПЛИНЫ**

Лекции проводятся с использованием проектора и внутренней системы портала ДВФУ. Лабораторные занятия проходят в аудиториях, оборудованных компьютерами типа Lenovo C360G-i34164G500UDK с лицензионными программами Microsoft Office 2013 и аудиовизуальными средствами проектор Panasonic DLPPjectorPT-D2110XE, плазма LG FLATRON M4716CCBAM4716CJ. Для выполнения самостоятельной работы студенты в жилых корпусах ДВФУ обеспечены Wi-Fi.



МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ РОССИЙСКОЙ ФЕДЕРАЦИИ  
Федеральное государственное автономное образовательное учреждение высшего образования  
**«Дальневосточный федеральный университет»**  
(ДВФУ)

---

**ШКОЛА ЕСТЕСТВЕННЫХ НАУК**

**УЧЕБНО-МЕТОДИЧЕСКОЕ ОБЕСПЕЧЕНИЕ САМОСТОЯТЕЛЬНОЙ  
РАБОТЫ ОБУЧАЮЩИХСЯ**

**по дисциплине «Разработка формальных языков и языковых  
процессоров»**

**Направление подготовки 09.04.04 Программная инженерия  
магистерская программа «Разработка программно-информационных систем»  
Форма подготовки очная**

**Владивосток  
2018**

## План-график выполнения самостоятельной работы по дисциплине

№ п/п	Формулировка задачи	Дата/сроки выполнения	Примерные нормы времени на выполнение	Форма контроля
1.	Формулировка задачи и определение формального языка	7 неделя обучения	4 час	проект
2.	Определение синтаксиса формального языка	8 неделя обучения	4 час	проект
3.	Определение семантики формального языка	9 неделя обучения	4 час	проект
4.	Построение модели формального языка	10 неделя обучения	4 час	проект
5.	Постановка задачи для компилятора	11 неделя обучения	4 час	проект
6.	Разработка требований	12 неделя обучения	4 час	проект
7.	Разработка проекта верхнего уровня	13 неделя обучения	4 час	проект
8.	Разработка детального проекта лексического анализатора	14 неделя обучения	4 час	проект
9.	Разработка тестов для лексического анализатора	15 неделя обучения	4 час	проект
10.	Реализация, отладка, тестирование лексического анализатора	17 неделя обучения	4 час	проект
11.	Разработка детального проекта синтаксического анализатора	2 неделя обучения	4 часов	проект
12.	Разработка тестов для синтаксического анализатора	3 неделя обучения	4 часа	проект
13.	Реализация, отладка, тестирование синтаксического анализатора	7 неделя обучения	4 часов	проект
14.	Разработка генератора кода	9 неделя обучения	4 часа	проект
15.	Реализация генератора кода	12 неделя обучения	4 часа	проект
16.	Сбор, отладка, тестирование компилятора	16 неделя обучения	8 часов	проект
17.	Предоставление документации по разработке компилятора	18 неделя обучения	4 часов	проект

## **Рекомендации по самостоятельной работе студентов**

Трудоёмкость самостоятельной работы студента – 72 часа.

Самостоятельная работа обучающихся подразумевает обязательную подготовку к лабораторным занятиям (оформление отчетов), изучение основной и дополнительно литературы по дисциплине, подготовку к текущему контролю и промежуточной аттестации в конце семестра, консультации преподавателей.

### **Рекомендации по работе с литературой**

Для более эффективного освоения и усвоения материала рекомендуется ознакомиться с теоретическим материалом по той или иной теме до проведения лабораторного занятия. Всю учебную литературу желательно изучать «под конспект».

Цель написания конспекта по дисциплине – сформировать навыки по поиску, отбору, анализу и формулированию учебного материала.

Работу с теоретическим материалом по теме можно проводить по следующей схеме:

- название темы;
- цели и задачи изучения темы;
- основные вопросы темы;
- характеристика основных понятий и определений, необходимых для усвоения данной темы;
- краткие выводы, ориентирующие на определенную совокупность сведений, основных идей, ключевых положений, систему доказательств, которые необходимо усвоить.

При работе над конспектом обязательно выявляются и отмечаются трудные для самостоятельного изучения вопросы, с которыми уместно обратиться к преподавателю при посещении консультаций, либо в индивидуальном порядке.

### **Подготовка к лабораторным работам**

Подготовку к лабораторной работе студент должен начать с изучения теоретического материала и ознакомления с планом, который отражает содержание предложенной темы. Все новые понятия по изучаемой теме необходимо выучить наизусть и внести в глоссарий, который целесообразно вести с самого начала изучения курса. Результат такой работы должен проявиться в способности студента свободно ответить на теоретические вопросы по теме задания, и правильном его выполнении.

В процессе выполнения лабораторной работы или практического задания студент должен создать требуемый документ с помощью предлагаемого программного средства и выполнить требуемые в задании операции. Задание по лабораторной или практической работе содержит методические указания по подготовке документа, который должен быть получен в результате выполнения работы. При подготовке следует их внимательно прочесть.

### **Критерии оценки лабораторных работ**

100-86 баллов – выполнены все задания практической (лабораторной) работы, студент четко и без ошибок ответил на все контрольные вопросы.

85-76 баллов – выполнены все задания практической (лабораторной) работы; студент ответил на все контрольные вопросы с замечаниями.

75-61 баллов – выполнены все задания практической (лабораторной) работы с замечаниями; студент ответил на все контрольные вопросы с замечаниями.

60-50 баллов – студент не выполнил или выполнил неправильно задания практической (лабораторной) работы; студент ответил на контрольные вопросы с ошибками или не ответил на контрольные вопросы.

### **Подготовка презентации и доклада**

Для подготовки презентации рекомендуется использовать: PowerPoint, MS Word, Acrobat Reader, LaTeX-овский пакет bphzleamer. Последовательность подготовки презентации:

1. Четко сформулировать цель презентации: вы хотите свою аудиторию мотивировать, убедить, заразить какой-то идеей или просто формально отчитаться.

2. Определить каков будет формат презентации: живое выступление (тогда, сколько будет его продолжительность) или электронная рассылка (каков будет контекст презентации).

3. Отобрать всю содержательную часть для презентации и выстроить логическую цепочку представления.

4. Определить ключевые моменты в содержании текста и выделить их.

5. Определить виды визуализации (иллюстрации, образы, диаграммы, таблицы) для отображения их на слайдах в соответствии с логикой, целью и спецификой материала.

6. Подобрать дизайн и форматировать слайды (количество картинок и текста, их расположение, цвет и размер).

7. Проверить визуальное восприятие презентации.

***Практические советы по подготовке презентации*** - готовьте отдельно:

- печатный текст + слайды + раздаточный материал;
- *слайды* – визуальная подача информации, которая должна содержать минимум текста, максимум изображений, несущих смысловую нагрузку, выглядеть наглядно и просто;
- *текстовое содержание презентации* – устная речь или чтение, которая должна включать аргументы, факты, доказательства и эмоции;
- *рекомендуемое число слайдов* 17-22;
- *обязательная информация для презентации*: тема, фамилия и инициалы выступающего; план сообщения; краткие выводы из всего сказанного; список использованных источников;
- *раздаточный материал* – должен обеспечивать ту же глубину и охват, что и живое выступление: люди больше доверяют тому, что они могут унести с собой, чем исчезающим изображениям, слова и слайды забываются, а раздаточный материал остается постоянным осязаемым напоминанием; раздаточный материал важно раздавать в конце презентации; раздаточный материалы должны отличаться от слайдов, должны быть более информативными.

#### **Критерии оценки презентации доклада:**

<b>Оценка</b>	<b>50-60 баллов (неудовлетворительно)</b>	<b>61-75 баллов (удовлетворительно)</b>	<b>76-85 баллов (хорошо)</b>	<b>86-100 баллов (отлично)</b>
<b>критерии</b>	<b>Содержание критериев</b>			
<b>Раскрытие проблемы</b>	Проблема не раскрыта. Отсутствуют выводы	Проблема раскрыта не полностью. Выводы не сделаны и/или выводы не обоснованы	Проблема раскрыта. Проведен анализ проблемы без привлечения дополнительной литературы. Не все выводы сделаны и/или обоснованы	Проблема раскрыта полностью. Проведен анализ проблемы с привлечением дополнительной литературы. Выводы обоснованы
<b>Представление</b>	Представляемая информация логически не связана. Не использованы профессиональные термины	Представляемая информация не систематизирована и/или не последовательна	Представляемая информация не систематизирована и последовательна. Использовано более 2 профессиональных терминов	Представляемая информация систематизирована, последовательна и логически связана. Использовано более 5

		1-2 профессиональных термина		профессиональных терминов
<b>Оформление</b>	Не использованы технологии Power Point. Больше 4 ошибок в представляемой информации	Использованы технологии Power Point частично. 3-4 ошибки в представляемой информации	Использованы технологии Power Point. Не более 2 ошибок в представляемой информации	Широко использованы технологии (Power Point и др.). Отсутствуют ошибки в представляемой информации
<b>Ответы на вопросы</b>	Нет ответов на вопросы	Только ответы на элементарные вопросы	Ответы на вопросы полные и/или частично полные	Ответы на вопросы полные, с приведением примеров и/или пояснений



МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ РОССИЙСКОЙ ФЕДЕРАЦИИ  
Федеральное государственное автономное образовательное учреждение высшего образования  
**«Дальневосточный федеральный университет»**  
(ДВФУ)

---

**ШКОЛА ЕСТЕСТВЕННЫХ НАУК**

**ФОНД ОЦЕНОЧНЫХ СРЕДСТВ**  
**по дисциплине «Разработка формальных языков и языковых**  
**процессоров»**  
**Направление подготовки 09.04.04 Программная инженерия**  
**магистерская программа «Разработка программно-информационных систем»**  
**Форма подготовки очная**

**Владивосток**  
**2018**

## Паспорт ФОС

Код и формулировка компетенции	Этапы формирования компетенции	
	ПК-10 – способность проектировать трансляторы и интерпретаторы языков программирования	Знает
Умеет		разрабатывать модели формальных языков, применять методы проектирования и разработки языковых процессоров.
Владеет		формальными средствами разработки нового языка, инструментальными средствами разработки программных систем, в том числе языковых процессоров
ПК-12 – способность проектировать основные компоненты операционных систем	Знает	методы разработки формальных языков описания данных и управления заданиями
	Умеет	проектировать и реализовывать языковые процессоры, используемые в различных операционных системах
	Владеет	современными способами и системами разработки инструментальных средств, предназначенных для создания языковых процессоров
ПК-13 – способность проектировать вспомогательные и специализированные языки программирования и языки представления данных	Знает	методы разработки, обоснования и исследования моделей, методов, алгоритмов и программной инфраструктуры, требуемых для создания специализированных языков и языковых процессоров в различных предметных областях
	Умеет	разрабатывать специализированные языки для автоматизации профессиональной деятельности
	Владеет	методами обоснования моделей формальных языков, необходимых для решения задач профессиональной деятельности

№ п/п	Контролируемые разделы/темы дисциплины	Коды и этапы формирования компетенций	Оценочные средства - наименование		
			текущий контроль	промежуточная аттестация	
1	Раздел I. Формальные языки и методы их разработки. (Разработка формального языка для решения задач		знает	Круглый стол УО-4	Вопросы к зачету № 1-8
			умеет владеет	Лабораторная работа № 1-7 ПР-6	

	предметной области)					
2	Раздел Принципы, технологии инструменты разработки языковых процессоров	II. и	ПК-10 ПК-12	Знает  умеет владеет	Круглый стол УО-4  Лабораторная работа № 7-14 ПР-6	Вопросы к зачету № 9-19

### Шкала оценивания уровня сформированности компетенций

Код и формулировка компетенции	Этапы формирования компетенции		критерии	показатели
ПК-10 – способность проектировать трансляторы и интерпретаторы языков программирования	знает (пороговый уровень)	методы разработки формальных языков, проектирования языковых процессоров языков	Знание формализмов для описания языков, типов языковых процессоров и их компонентов	Способность дать ответы на вопросы
	умеет (продвинутый)	разрабатывать модели формальных языков, применять методы проектирования и разработки языковых процессоров.	Умение использовать формализмы при описании языка, разработать проект языкового процессора	Способность спроектировать компоненты языкового процессора
	владеет (высокий)	формальными средствами разработки нового языка, инструментальным и средствами разработки программных систем, в том числе языковых процессоров	Владение методами разработки программной системы по проекту	Способность выполнить реализацию языкового процессора

ПК-12 – способность проектировать основные компоненты операционных систем	знает (порогов ый уровень)	методы разработки формальных языков описания данных и управления заданиями	Знание особенностей языков данного класса	Способность дать ответы на вопросы
	умеет (продви нутый)	проектировать и реализовывать языковые процессоры, используемые в различных операционных системах	Умение использовать формализмы при описании языка описания данных и управления заданиями, разработать проект языкового процессора	Способность спроектировать компоненты языкового процессора для заданной операционной системы
	владеет (высоки й)	современными способами и системами разработки инструментальных средств, предназначенных для создания языковых процессоров	Владение навыками работы в современных средах программиров ания	Способность выполнить реализацию языкового процессора для заданной операционной системы
ПК-13 – способность проектировать вспомогательные и специализированн ые языки программировани я и языки представления данных	знает (порогов ый уровень)	методы разработки, обоснования и исследования моделей, методов, алгоритмов и программной инфраструктуры, требуемых для создания специализированн ых языков и языковых процессоров в различных предметных областях	Знание методов разработки моделей, методов, алгоритмов и программной инфраструкту ры, требуемых для создания специализиро ванных языков и языковых процессоров	Способность дать ответы на вопросы
	умеет (продви нутый)	разрабатывать специализированн ые языки для автоматизации профессиональной деятельности	Умение определить тип языка, разработать его модель и проект	Способность спроектировать компоненты языкового процессора для языка

			языкового процессора	требуемого типа
	владеет (высокий)	методами обоснования моделей формальных языков, необходимых для решения задач профессиональной деятельности	Владение методами обоснования моделей формальных языков, необходимых для решения задач профессиональной деятельности	Предоставление проекта модели формального языка

## Методические рекомендации, определяющие процедуры оценивания результатов освоения дисциплины

### Промежуточный контроль

Промежуточный контроль осуществляется в конце семестра и завершает изучение дисциплины. Помогает оценить более крупные совокупности знаний и умений, сформированность определенных профессиональных компетенций по дисциплине. Промежуточный контроль проводится в форме зачета, допуск к экзамену возможен для обучающихся, получивших оценку «зачтено» в результате выполнения самостоятельной работы и успешно выполнившие все лабораторные работы.

### Критерии выставления оценки магистранту на зачете/экзамене

Баллы (рейтинговой оценки)	Оценка зачета/ экзамена (стандартная)	Требования к сформированным компетенциям
86-100	«зачтено»/ «отлично»	Оценка «отлично» выставляется студенту, если он глубоко и прочно усвоил программный материал, исчерпывающе, последовательно, четко и логически стройно его излагает, умеет тесно увязывать теорию с практикой, свободно справляется с задачами, вопросами и другими видами применения знаний, причем не затрудняется с ответом при видоизменении заданий, правильно обосновывает принятое решение, владеет разносторонними навыками и приемами выполнения практических задач.

76-85	«зачтено»/ «хорошо»	Оценка «хорошо» выставляется студенту, если он твердо знает материал, 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.
61-75	«зачтено»/ «удовлетворительно»	Оценка «удовлетворительно» выставляется студенту, 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 работ.
0-60	«не зачтено»/ «неудовлетворительно»	Оценка «неудовлетворительно» выставляется студенту, который не знает значительной части программного материала, допускает существенные ошибки, неуверенно, с большими затруднениями выполняет практические работы. Как правило, оценка «неудовлетворительно» ставится студентам, которые не могут продолжить обучение без дополнительных занятий по соответствующей дисциплине.

## Оценочные средства для промежуточной аттестации

### Вопросы к зачету

1. Формальный язык, классификация формальных языков.
2. Язык описания данных.
3. Язык описания знаний.
4. Язык, основанный на правилах.
5. Язык спецификаций.
6. Языки, основанные на объектно-ориентированном формализме.
7. Модель формального языка.
8. Формальный язык и языковой процессор на его основе. Теоретический язык и входной язык языкового процессора.
9. Типы языковых процессоров (компиляторы, интерпретаторы, смешанные формы). Структура языкового процессора.
10. Реализация лексического анализа в языковом процессоре. Лексические ошибки.

11. Работа с таблицей имен. Способы хранения длинных имен и строк.
12. Токены, шаблоны, лексемы. Атрибуты токенов. Их вычисление при разборе и хранение.
13. Язык спецификации лексических анализаторов.
14. Понятие о стратегии синтаксического анализа. Нисходящий и восходящий разбор. Важнейшие классы грамматик и соответствующие им анализаторы.
15. Формализмы для задания грамматик.
16. Обработка синтаксических ошибок. Методы продолжения разбора при ошибках.
17. Способы представления результатов анализа: дерево разбора, синтаксическое дерево, польская запись.
18. Работа с таблицей имен при синтаксическом анализе.
19. Методы диагностики синтаксических ошибок.

### **Текущий контроль**

Текущий контроль предполагает систематическую проверку усвоения учебного материала, сформированности компетенций или их элементов, регулярно осуществляемую на протяжении изучения дисциплины, в соответствии с ее рабочей программой.

Состоит в проверке правильности выполнения заданий по самостоятельной работе. Задание зачтено, если нет ошибок. По текущим ошибкам даются пояснения.

#### **Критерии оценки проектов и участия в тематической дискуссии**

- 100-86 баллов выставляется, если магистрант/группа точно определили содержание и составляющие части задания, умеют аргументировано отвечать на вопросы, связанные с заданием. Продемонстрировано знание и владение навыками самостоятельной исследовательской работы по теме. Фактических ошибок, связанных с пониманием проблемы, нет.

- 85-76 баллов – работа магистранта/группы характеризуется смысловой цельностью, связностью и последовательностью изложения; допущено не более 1 ошибки при объяснении смысла или содержания проблемы. Продемонстрированы исследовательские умения и навыки. Фактических ошибок, связанных с пониманием проблемы, нет.

- 75-61 балл – проведен достаточно самостоятельный анализ основных этапов и смысловых составляющих проблемы; понимание базовых основ и

теоретического обоснования выбранной темы. Привлечены основные источники по рассматриваемой теме. Допущено не более 2 ошибок в смысле или содержании проблемы

- 60-50 баллов – если работа представляет собой пересказанный или полностью переписанный исходный текст без каких бы то ни было комментариев, анализа. Не раскрыта структура и теоретическая составляющая темы. Допущено три или более трех ошибок смыслового содержания раскрываемой проблемы.

### Шкала оценивания проектов

Менее 60 баллов	Не зачтено
От 61 до 75 баллов	зачтено
От 76 до 85 баллов	зачтено
От 86 до 100 баллов	зачтено

### Текущий контроль

Текущий контроль предполагает систематическую проверку усвоения учебного материала, сформированности компетенций или их элементов, регулярно осуществляемую на протяжении изучения дисциплины, в соответствии с ее рабочей программой.

Состоит в проверке правильности выполнения заданий по самостоятельной работе. Задание зачтено, если нет ошибок. По текущим ошибкам даются пояснения.

Тесты предназначены для проверки знаний по компетенциям. Проверка достижения умений и навыков по компетенциям проверяется выполнением практических работ и курсовой работы.

### Примерные тесты для проверки сформированности компетенций

<b>ПК-10 – способность проектировать трансляторы и интерпретаторы языков программирования</b>	Знает методы разработки формальных языков, проектирования языковых процессоров языков
1. Работа нисходящего и восходящего анализаторов моделируется соответственно	ответы <b>а. автоматом с магазинной памятью и расширенным автоматом с магазинной памятью</b> б. детерминированным конечным автоматом и правоавтоматной грамматикой в. недетерминированным конечным автоматом и левоавтоматной грамматикой

2. Моделью синтаксического анализатора является	<p>ответы</p> <p>а. детерминированный конечный автомат</p> <p>б. недетерминированный конечный автомат</p> <p><b>в. автомат с магазинной памятью</b></p>
---	---

<b>ПК-12 – способность проектировать основные компоненты операционных систем</b>	Знает методы разработки формальных языков описания данных и управления заданиями
1. Продукцией или правилом подстановки называется упорядоченная пара, которая обычно записывается так $U ::= X$ , где X	<p>ответы</p> <p>1) символ</p> <p><b>2) цепочка символов</b></p> <p>3) не пустая конечная цепочка символов</p>
2. проверку соответствия типов выполняет	<p>ответы</p> <p>а. <b>семантический анализатор</b></p> <p>б синтаксический анализатор</p> <p>в. лексический анализатор</p>
3. при проверке выполнения контекстных условий языка используются	<p>ответы</p> <p><b>а. значения атрибутов – различных свойств конструкций языка</b></p> <p>б. таблица, задающая грамматику</p> <p>в. автомат с магазинкой памятью</p>
4. Пока в цепочке есть хотя бы один нетерминальный символ из нее можно вывести	<p>ответы</p> <p>1) терминальную цепочку</p> <p><b>2) новую цепочку</b></p>

<b>ПК-13 – способность проектировать вспомогательные и специализированные языки программирования и языки представления данных</b>	Знает методы разработки, обоснования и исследования моделей, методов, алгоритмов и программной инфраструктуры, требуемых для создания специализированных языков и языковых процессоров в различных предметных областях
1. модель алгоритмического языка описывается	<p>ответы</p> <p>а. <b>вычислительной моделью</b></p> <p>б. порождающей моделью</p> <p>в. конечным автоматом</p> <p>г. автоматом с магазинной памятью</p>
2. Все символы, которые встречаются в левых и правых частях правил грамматики, образуют	<p>ответы</p> <p>1) множество терминальных символов</p> <p>2) множество нетерминальных символов</p> <p>3) алфавит символов</p>
3. В заданной грамматике G символы, которые не входят во множество N нетерминальных символов называются	<p>ответы</p> <p>1) входными</p> <p><b>2) терминальными</b></p> <p>3) конечными</p>
4. чтобы задать вычислительную модель надо определить	<p>ответы</p> <p>а. <b>структуру состояний рабочей среды</b></p> <p>б. конечный автомат</p> <p>в. автомат с магазинной памятью</p>

