

МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ
 ФЕДЕРАЦИИ
 федеральное государственное автономное образовательное учреждение высшего
 образования
 "САНКТ-ПЕТЕРБУРГСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ
 АЭРОКОСМИЧЕСКОГО ПРИБОРОСТРОЕНИЯ"

Кафедра № 44

УТВЕРЖДАЮ
 Руководитель направления

д.т.н., проф. _____
 (должность, уч. степень, звание)

М.Б. Сергеев _____
 (инициалы, фамилия)

_____ (подпись)

«05» марта 2020 г

РАБОЧАЯ ПРОГРАММА ДИСЦИПЛИНЫ

«Системное программное обеспечение»
 (Наименование дисциплины)

Код направления подготовки/ специальности	09.03.01
Наименование направления подготовки/ специальности	Информатика и вычислительная техника
Наименование направленности	Вычислительные машины, комплексы, системы и сети
Форма обучения	заочная

Лист согласования рабочей программы дисциплины

Программу составил (а)

доц., к.т.н., доц. _____ 05.03.2020
 (должность, уч. Степень, звание) (подпись, дата)

А.Ю. Молчанов _____
 (инициалы, фамилия)

Программа одобрена на заседании кафедры № 44

«05» марта 2020 г, протокол № 5-19/20

Заведующий кафедрой № 44

д.т.н., проф. _____ 05.03.2020
 (уч. Степень, звание) (подпись, дата)

М.Б. Сергеев _____
 (инициалы, фамилия)

Ответственный за ОП ВО 09.03.01(02)

доц., к.т.н., доц. _____ 05.03.2020
 (должность, уч. Степень, звание) (подпись, дата)

Н.В. Соловьев _____
 (инициалы, фамилия)

Заместитель директора института №4 по методической работе

доц., к.т.н., доц. _____ 05.03.2020
 (должность, уч. степень, звание) (подпись, дата)

А.А. Ключарев _____
 (инициалы, фамилия)

Аннотация

Дисциплина «Системное программное обеспечение» входит в образовательную программу высшего образования по направлению подготовки/ специальности 09.03.01 «Информатика и вычислительная техника» направленности «Вычислительные машины, комплексы, системы и сети». Дисциплина реализуется кафедрой «№44».

Дисциплина нацелена на формирование у выпускника следующих компетенций:

ПК-4 «Способен разрабатывать требования и проектировать программное обеспечение».

Содержание дисциплины охватывает круг вопросов, связанных с теоретическим базисом, необходимым для создания и использования языков программирования высокого уровня (грамматики и распознаватели для формальных языков), созданием программных продуктов с помощью современных систем программирования, организацией взаимодействия между различными программными продуктами, а также технологиями разработки программного обеспечения для сетевых распределенных вычислительных систем с использованием современных операционных систем (ОС) и систем управления базами данных (СУБД).

Преподавание дисциплины предусматривает следующие формы организации учебного процесса: лекции, лабораторные работы, самостоятельная работа студентов, консультации.

Программой дисциплины предусмотрены следующие виды контроля: текущий контроль успеваемости, промежуточная аттестация в форме экзамена.

Общая трудоемкость освоения дисциплины составляет 7 зачетных единиц, 252 часа.

Язык обучения по дисциплине «русский»

1. Перечень планируемых результатов обучения по дисциплине

1.1. Цели преподавания дисциплины

Цель преподавания дисциплины – получение студентами необходимых знаний в области теоретических основ и получение практических навыков описания формальных языков, методов реализации трансляторов, принципов построения и функционирования операционных систем, технологий разработки программного обеспечения для распределенных вычислительных систем, технологий обеспечения взаимодействия программ в современных вычислительных системах.

1.2. Дисциплина входит в состав части, формируемой участниками образовательных отношений, образовательной программы высшего образования (далее – ОП ВО).

1.3. Перечень планируемых результатов обучения по дисциплине, соотнесенных с планируемыми результатами освоения ОП ВО.

В результате изучения дисциплины обучающийся должен обладать следующими компетенциями или их частями. Компетенции и индикаторы их достижения приведены в таблице 1.

Таблица 1 – Перечень компетенций и индикаторов их достижения

Категория (группа) компетенции	Код и наименование компетенции	Код и наименование индикатора достижения компетенции
Профессиональные компетенции	ПК-4 Способен разрабатывать требования и проектировать программное обеспечение	ПК-4.3.1 знать принципы построения архитектуры программного обеспечения и виды архитектуры программного обеспечения ПК-4.У.1 уметь использовать существующие типовые решения и шаблоны проектирования программного обеспечения ПК-4.В.1 владеть навыками проектирования структур данных, баз данных, программных интерфейсов

2. Место дисциплины в структуре ОП

Дисциплина может базироваться на знаниях, ранее приобретенных обучающимися при изучении следующих дисциплин:

- Математика. Математический анализ.
- Информатика.
- Основы программирования.
- Организация ЭВМ и систем.
- Операционные системы.

Знания, полученные при изучении материала данной дисциплины, имеют как самостоятельное значение, так и могут использоваться при изучении других дисциплин:

- Интерфейсы периферийных устройств.
- Теория надежности ВС и ПО.

3. Объем и трудоемкость дисциплины

Данные об общем объеме дисциплины, трудоемкости отдельных видов учебной работы по дисциплине (и распределение этой трудоемкости по семестрам) представлены в таблице 2.

Таблица 2 – Объем и трудоемкость дисциплины

Вид учебной работы	Всего	Трудоемкость по семестрам	
		№7	№8
1	2	3	4

Общая трудоемкость дисциплины, ЗЕ/ (час)	7/ 252	3/ 108	4/ 144
Аудиторные занятия, всего час.	32	16	16
в том числе:			
лекции (Л), (час)	16	8	8
практические/семинарские занятия (ПЗ), (час)			
лабораторные работы (ЛР), (час)	16	8	8
курсовой проект (работа) (КП, КР), (час)			
экзамен, (час)	9		9
Самостоятельная работа, всего (час)	211	92	119
Вид промежуточной аттестации: зачет, дифф. зачет, экзамен (Зачет, Дифф. зач, Экз.**)	Дифф. Зач., Экз.	Дифф. Зач.	Экз.

Примечание: ** кандидатский экзамен

4. Содержание дисциплины

4.1. Распределение трудоемкости дисциплины по разделам и видам занятий.

Разделы, темы дисциплины и их трудоемкость приведены в таблице 3.

Таблица 3 – Разделы, темы дисциплины, их трудоемкость

Разделы, темы дисциплины	Лекции (час)	ПЗ (СЗ) (час)	ЛР (час)	КП (час)	СРС (час)
Семестр 7					
Раздел 1. Общие принципы построения трансляторов. Теория формальных языков и грамматик. Тема 1.1. Трансляторы и компиляторы. Основные понятия. Этапы компиляции. Тема 1.2. Языки и грамматики. Формальное определение грамматик. Классификация языков и грамматик. Тема 1.3. Распознаватели и преобразователи. Основные компоненты распознавателей. Классификация распознавателей. Тема 1.4. Таблицы идентификаторов. Основные методы организации таблиц идентификаторов.	1		2		25
Раздел 2. Лексический анализ и регулярные языки. Тема 2.1. Лексический анализ. Назначение и функции лексического анализа. Тема 2.2. Регулярные языки и грамматики. Автоматные грамматики. Тема 2.3. Конечные автоматы (КА). Использование КА в качестве лексического анализатора. Тема 2.4. Регулярные множества и регулярные выражения. Тема 2.5. Свойства регулярных языков.	2		3		15
Раздел 3. Синтаксический анализ и контекстно-свободные языки Тема 3.1. Синтаксический анализ. Назначение и функции синтаксического анализа. Тема 3.2. Контекстно-свободные языки и	4		3		20

грамматики. Тема 3.3. Автоматы с магазинной памятью (МП-автоматы). Расширенные МП-автоматы. Тема 3.4. Построение синтаксического распознавателя по заданной грамматике. Тема 3.5. Линейные синтаксические распознаватели. Тема 3.6. Свойства КС-языков. Тема 3.7. Внутренние представления программ.					
Раздел 4. Особенности построения современных компиляторов Тема 4.1. Проход. Понятие прохода. Однопроходные и многопроходные компиляторы. Тема 4.2. Язык ассемблера. Особенности компиляторов с языка ассемблера. Тема 4.3. Макрогенерация и макроподстановки. Условная компиляция. Тема 4.3. Мобильность и переносимость программного обеспечения. Особенности компиляторов с языков Java и C#.	1		0		32
Итого в семестре:	8		8		92
Семестр 8					
Раздел 5. Генерирование и оптимизация результирующего кода Тема 5.1. Семантический анализ. Основные функции семантического анализа. Тема 5.2. Распределение памяти. Организация компилятором таблиц RTTI. Тема 5.3. Исключительные ситуации. Обработка исключительных ситуаций. Тема 5.4. Синтаксически управляемый перевод. Генерация результирующего кода. Тема 5.5. Оптимизация. Основные принципы и методы оптимизации.	3		5		25
Раздел 6. Современные системы программирования Тема 6.1. Структура систем программирования. Тема 6.2. История развития систем программирования. Тема 6.3. Функции основных компонент в составе систем программирования.	1		0		30
Раздел 7. Технологии разработки программного обеспечения Тема 7.1. Принципы организации распределенных вычислений. Тема 7.2. Технология «файл-сервер». Тема 7.3. Терминальный сервер. Тема 7.4. Технология «клиент-сервер». Тема 7.5. Трехуровневые и многоуровневые технологии разработки программного обеспечения. Тема 7.6. Облачные вычисления и сервисы.	2		3		28

Раздел 8. Разработка программного обеспечения для глобальных сетей Тема 8.1. Особенности разработки программного обеспечения для глобальных сетей. Применение современных интерпретаторов. Тема 8.2. Построение статических Web-страниц. Язык HTML. Тема 8.3. Построение динамических Web-страниц на основе компилируемых и интерпретируемых языков. Тема 8.4. Построение интерактивных Web-страниц.	2		0		36
Итого в семестре:	8		8		119
Итого	16	0	16	0	211

4.2. Содержание разделов и тем лекционных занятий.

Содержание разделов и тем лекционных занятий приведено в таблице 4.

Таблица 4 – Содержание разделов и тем лекционного цикла

Номер раздела	Название и содержание разделов и тем лекционных занятий
I	<p>Раздел 1. Общие принципы построения трансляторов. Теория формальных языков и грамматик.</p> <p>Тема 1.1. Трансляторы и компиляторы. Основные понятия. Этапы компиляции.</p> <p>Определение транслятора, компилятора, интерпретатора. Различия между транслятором и компилятором, транслятором и интерпретатором. Понятие объектной программы и целевой вычислительной системы. Типовая структура компилятора. Этапы компиляции (этапы анализа и этапы синтеза), взаимосвязь между ними.</p> <p>Тема 1.2. Языки и грамматики. Формальное определение грамматик. Классификация языков и грамматик.</p> <p>Понятие цепочки символов (строки), основные операции над цепочками символов. Определение алфавита, цепочки символов над алфавитом, множества всех возможных цепочек символов над заданным алфавитом. Формальное определение языка, способы задания языков. Формальное определение грамматики. Грамматика в форме Бэкуса-Наура (БНФ), пример грамматики в БНФ. Другие способы определения грамматик, в том числе форма грамматики с метасимволами и графическая форма представления грамматик. Непосредственная выводимость цепочки символов из правил грамматики. Понятие вывода и цепочки вывода, сентенциальная форма грамматики. Дерево вывода, правосторонний и левосторонний выводы. Примеры цепочек вывода, дерева вывода.</p> <p>Определение языка, заданного грамматикой. Однозначные и неоднозначные грамматики, проблема однозначности. Примеры однозначных и неоднозначных грамматик, неоднозначность естественных языков. Классификация языков и грамматик по Хомскому, проблема преобразования. Назначение основных типов языков и их применение в вычислительной технике.</p> <p>Тема 1.3. Распознаватели и преобразователи. Основные компоненты распознавателей. Классификация распознавателей.</p> <p>Определение распознавателя. Основные компоненты распознавателей: считывающее устройство, устройство управления (УУ), внешняя память (ВП). Основные принципы работы распознавателя, начальная конфигурация и конечные</p>

	<p>конфигурации распознавателя. Классификация распознавателей по считывающему устройству, односторонние и двусторонние распознаватели. Классификация распознавателей по УУ, детерминированные и недетерминированные распознаватели. Классификация распознавателей по ВП, распознаватели с неограниченной внешней памятью (в том числе машина Тьюринга), распознаватели с ограниченной внешней памятью, распознаватели без внешней памяти. Соответствие между классификациями распознавателей и классификацией языков по Хомскому, типы распознавателей для основных типов языков. Определение преобразователя, отличие преобразователя от распознавателя. Назначение распознавателей и преобразователей, задача разбора и ее решение для основных типов языков.</p> <p>Тема 1.4. Таблицы идентификаторов. Основные методы организации таблиц идентификаторов.</p> <p>Назначение таблиц идентификаторов, информация, хранящаяся в таблицах идентификаторов. Основные действия, выполняемые компилятором над таблицами идентификаторов, требования к организации таблиц идентификаторов. Простейшие способы организации таблиц идентификаторов – логарифмический поиск в линейном списке и бинарное дерево – их основные преимущества и недостатки. Понятие о хеш-функции. Организация таблиц идентификаторов на основе хеш-функции с помощью хеш-адресации. Проблемы, связанные с организацией таблиц идентификаторов на основе хеш-адресации, понятие о коллизии. Методы борьбы с коллизиями. Организация таблиц идентификаторов на основе рехеширования, основные преимущества и недостатки метода. Организация таблиц идентификаторов с помощью метода цепочек, основные преимущества и недостатки метода. Смешанные методы организации таблиц идентификаторов.</p>
2	<p>Тема 2.1. Лексический анализ. Назначение и функции лексического анализа.</p> <p>Определение лексемы, основные типы лексем в языках программирования. Задачи лексического анализа, функции, выполняемые лексическим анализатором. Преимущества, предоставляемые лексическим анализатором, возможность организации компиляторов без лексического анализатора. Проблема определения границ лексем, взаимодействие лексического и синтаксического анализаторов – последовательный и параллельный методы взаимодействия. Принцип выбора лексемы максимальной длины в современных компиляторах. Таблица лексем и ее назначение, информация, хранящаяся в таблице лексем. Пример построения таблицы лексем.</p> <p>Тема 2.2. Регулярные языки и грамматики. Автоматные грамматики.</p> <p>Регулярные языки – простейший тип языков по классификации Хомского. Определение регулярной грамматики, левосторонние и правосторонние грамматики. Задача разбора для регулярных языков и метод ее решения. Определение автоматной грамматики. Эквивалентность регулярных и автоматных грамматик, алгоритм построения автоматной грамматики на основе регулярной грамматики. Пример регулярной грамматики и ее преобразование к автоматному виду.</p> <p>Тема 2.3. Конечные автоматы (КА). Использование КА в</p>

	<p>качестве лексического анализатора.</p> <p>Определение КА. Способы задания КА: с помощью множеств и функции переходов, с помощью графа переходов. КА как распознаватель для регулярных языков, эквивалентность КА и регулярных языков. Детерминированные и недетерминированные КА, преобразование КА к детерминированному виду. Минимизация КА, множества эквивалентных состояний КА, алгоритм минимизации КА. Полное решение задачи разбора для регулярных языков, эквивалентность регулярных грамматик, автоматных грамматик и КА. Алгоритм построения КА по заданной автоматной грамматике. Пример автоматной грамматики, построение по ней КА, приведение КА к детерминированному виду и минимизация построенного КА.</p> <p>Тема 2.4. Регулярные множества и регулярные выражения.</p> <p>Определение регулярного множества и регулярного выражения. Свойства регулярных выражений. Уравнения с регулярными коэффициентами и системы уравнений с регулярными коэффициентами, методы решения систем уравнений с регулярными коэффициентами. Построение системы уравнений с регулярными коэффициентами на основе регулярной грамматики и ее решение. Автоматизированные методы построения лексических распознавателей, программа LEX и ее аналоги.</p> <p>Тема 2.5. Свойства регулярных языков.</p> <p>Способы определения регулярных языков, эквивалентность регулярных множеств, регулярных грамматик и КА. Свойства регулярных языков, решение проблем однозначности и проблем преобразования для регулярных грамматик. Лемма о разрастании для регулярных языков.</p>
3	<p>Тема 3.1. Синтаксический анализ. Назначение и функции синтаксического анализа.</p> <p>Задачи и функции синтаксического анализатора, синтаксический анализатор – основная и самая сложная по вычислительным затратам часть компилятора. Этапы построения синтаксического анализатора, исходные данные, необходимые для выполнения синтаксического анализа. Восходящие и нисходящие синтаксические распознаватели.</p> <p>Тема 3.2. Контекстно-свободные языки и грамматики.</p> <p>Контекстно-свободные языки (КС-языки) – второй по сложности тип языков в классификации Хомского. Определение контекстно-свободной (КС) грамматики. Проблемы, связанные с КС-языками. Понятие о том, что языки программирования в строгом смысле слова не являются КС-языками, поэтому компилятор разбивает их анализ на две части: синтаксический и семантический анализ.</p> <p>Тема 3.3. Автоматы с магазинной памятью (МП-автоматы). Расширенные МП-автоматы.</p> <p>Определение МП-автомата, способы задания МП-автоматов, принципы функционирования МП-автоматов. Детерминированные (ДМП) и недетерминированные МП-автоматы, невозможность построения ДМП-автомата, эквивалентного исходному МП-автомату в общем случае (невозможность приведения МП-автомата к детерминированному виду в общем случае). Определение детерминированных КС-языков (ДКС-языков), взаимосвязь ДКС-языков и ДМП-автоматов, взаимосвязь ДКС-языков с однозначными КС-грамматиками. Расширенные МП-автоматы –</p>

	<p>определение, отличия от обычных МП-автоматов. Эквивалентность МП-автоматов, расширенных МП-автоматов и КС-языков.</p> <p>Тема 3.4. Построение синтаксического распознавателя по заданной грамматике.</p> <p>Задача разбора и ее решение для КС-языков. Построение синтаксического распознавателя с подбором альтернатив на основе МП-автомата по заданной КС-грамматике, проблемы распознавателя с подбором альтернатив. Построение синтаксического распознавателя «сдвиг-свертка» на основе расширенного МП-автомата по заданной КС-грамматике, проблемы распознавателя «сдвиг-свертка». Преобразования КС-грамматик: условия преобразований, цели преобразований. Основные алгоритмы преобразования КС-грамматик: удаление бесплодных (бесполезных) символов, удаление недостижимых символов, устранение пустых правил, устранение цепных правил, устранение левой рекурсии. Синтаксические распознаватели с возвратом – распознаватель с подбором альтернатив и распознаватель типа «сдвиг-свертка»: построение, функционирование, основные преимущества и недостатки. Универсальные синтаксические распознаватели (табличные распознаватели) – алгоритм Кока-Янгера-Касами и алгоритм Эрли – их построение, функционирование, основные преимущества и недостатки.</p> <p>Тема 3.5. Линейные синтаксические распознаватели.</p> <p>Понятие о линейном синтаксическом распознавателе, основные преимущества линейных синтаксических распознавателей, отсутствие универсальности при применении линейных синтаксических распознавателей. Простейший линейный синтаксический распознаватель – распознаватель по методу рекурсивного спуска – его построение, свойства, ограничения на применение. Расознаватели на основе LL(k)-грамматик – основные свойства, принципы функционирования, ограничения. Построение простейшего распознавателя на основе LL(1)-грамматики, множества FIRST₁ и FOLLOW₁. Расознаватели на основе LR(k)-грамматик – основные свойства, принципы функционирования, ограничения. Часто используемые распознаватели на основе SLR(1)-грамматик и LALR(1)-грамматик. Расознаватели на основе грамматик предшествования – основные свойства, принципы функционирования, ограничения. Алгоритм распознавателя «сдвиг-свертка» для грамматик предшествования. Построение простейшего распознавателя на основе грамматики операторного предшествования, множества L(U) и R(U), L'(U) и R'(U), матрица предшествования и ее заполнение. Автоматизированные методы построения синтаксических распознавателей, программа YACC и ее аналоги.</p> <p>Тема 3.6. Свойства КС-языков.</p> <p>Способы определения КС-языков, эквивалентность КС-грамматик и МП-автоматов. Свойства КС-языков, неразрешимость проблем однозначности и проблем преобразования для КС-грамматик в общем случае. Классы КС-языков и КС-грамматик, решение проблем однозначности и проблем преобразования для определенных классов КС-грамматик. Лемма о разрастании для КС-языков.</p> <p>Тема 3.7. Внутренние представления программ.</p> <p>Понятие о внутреннем представлении программы, информация, хранящаяся во внутреннем представлении программы, использование</p>
--	--

	<p>компилятором нескольких вариантов внутреннего представления программы по мере необходимости. Машинно-зависимые и машинно-независимые способы внутреннего представления программ. Основные машинно-независимые способы внутреннего представления программ: дерево операций (и его взаимосвязь с синтаксическим деревом), трехадресный код (тетрады), двухадресный код (триады), постфиксная (или обратная польская) запись операций. Основные преимущества и недостатки способов внутреннего представления программ. Вычисления выражений в форме постфиксной записи операций.</p>
4	<p>Тема 4.1. Проход. Понятие прохода. Однопроходные и многопроходные компиляторы.</p> <p>Понятие прохода. Последовательность выполнения проходов компилятором, взаимосвязь между последовательно выполняемыми проходами. Количество проходов компилятора, однопроходные и многопроходные компиляторы. Влияние количества проходов на вычислительную сложность компилятора, зависимость эффективности результирующей программы от количества проходов.</p> <p>Тема 4.2. Язык ассемблера. Особенности компиляторов с языка ассемблера.</p> <p>Определение языка ассемблера (мнемокода). Области применения языков ассемблера в современных вычислительных системах. Компилятор с языка ассемблера как простейший случай компилятора. Типовая грамматика языка ассемблера. Особенности компиляторов с языка ассемблера.</p> <p>Тема 4.3. Макрогенерация и макроподстановки. Условная компиляция.</p> <p>Понятие о макрогенерации, основные цели выполнения макрогенерации. Макроопределения («макросы») и их структура, выполнение макроподстановок на основе макроопределений. Понятие о макропроцессоре (препроцессоре) как о текстовом трансляторе с макроязыка. Примеры выполнения макроподстановок без параметров и с параметрами. Возможные проблемы при выполнении макроподстановок, сравнение выполнения макроподстановок с вызовом функций. Организация условной компиляции на основе макрогенерации, задачи, решаемые с помощью условной компиляции.</p> <p>Тема 4.3. Мобильность и переносимость программного обеспечения. Особенности компиляторов с языков Java и C#.</p> <p>Понятие мобильности и переносимости программного обеспечения. Основные сложности обеспечения мобильности кода. Основные способы обеспечения переносимости программного кода: стандартизация (соответствие стандартам переносимости), условная компиляция (с помощью макроподстановок), использование интерпретируемых языков. Переносимость кода с помощью сочетания компиляции и интерпретации: компилятор – промежуточный код – интерпретатор – выполнение кода. Особенности компиляторов с языков Java и C#, ориентированных на сочетание компиляции и интерпретации кода.</p>
5	<p>Тема 5.1. Семантический анализ. Основные функции семантического анализа.</p> <p>Семантика и синтаксис языка, зависимость семантики от синтаксиса. Семантический анализ, основные задачи семантического анализа, исходные данные, необходимые для</p>

	<p>выполнения семантического анализа. Функции семантического анализа: проверка семантических ограничений входного языка, дополнение внутреннего представления программы неявными вызовами процедур и функций, проверка общих семантических ограничений языков программирования. Ошибки и предупреждения, порождаемые семантическим анализатором. Примеры ошибочных ситуаций и ситуаций, вызывающих предупреждения. Ограничения на возможности семантического анализа в компиляторе.</p> <p>Тема 5.2. Распределение памяти. Организация компилятором таблиц RTTI.</p> <p>Основные принципы распределения памяти компилятором, статическая и динамическая, локальная и глобальная области памяти. Вычисления объема памяти для простейших типов данных, зависимость объема памяти от целевой вычислительной системы. Формулы вычисления объема памяти для сложных структур данных: массивов, структур (записей), союзов (записей с вариантами), классов и объектов. Кратность адресации в вычислительных системах, выравнивание адресов ячеек памяти в зависимости от кратности адресации, влияние кратности на производительность результирующей программы и на объем требуемой оперативной памяти. Дисплей памяти процедуры (функции), стековая организация дисплея памяти процедуры (функции), преимущества и недостатки стековой организации дисплея памяти процедуры (функции). Особенности стековой организации дисплея памяти процедуры (функции) в различных вычислительных системах, соглашения о вызовах, два типа основных соглашений о вызовах: соглашение о вызовах языка Си и соглашение о вызовах языка Паскаль. Организация компилятором таблиц информации о типах данных (Run-Time Type Information – RTTI) в объектно-ориентированных языках программирования, информация, хранящаяся в таблицах RTTI и ее взаимосвязь с классами (объектами) в объектно-ориентированных языках. Проблемы использования таблиц RTTI при работе с библиотеками классов (объектов).</p> <p>Тема 5.3. Исключительные ситуации. Обработка исключительных ситуаций.</p> <p>Понятие об исключительных ситуациях, роль исключительных ситуаций в объектно-ориентированных языках программирования. Необходимость поддержки обработки исключительных ситуаций со стороны операционной системы (ОС) на целевой вычислительной системе. Порождение компилятором кода для обработки исключительных ситуаций, явный и неявный код обработки исключительных ситуаций. Возможные уровни обработки исключительных ситуаций – на уровне кода приложения, на уровне библиотеки системы программирования, на уровне ОС. Основные типы исключительных ситуаций – аппаратные, порождаемые ОС, порождаемые системой программирования, пользовательские. Новые возможности и ограничения при обработке исключительных ситуаций, вычислительные затраты на обработку исключительных ситуаций.</p> <p>Тема 5.4. Синтаксически управляемый перевод. Генерация результирующего кода.</p> <p>Проблемы, возникающие при порождении результирующего кода компилятором. Понятие синтаксически управляемого перевода (СУ-перевода), зависимость семантики кода от его синтаксиса.</p>
--	--

	<p>Использование метода СУ-перевода для генерации результирующего кода компилятором. Ограничения на применение метода СУ-перевода. Проблемы эффективности результирующего кода, возникающие из-за применения метода СУ-перевода.</p> <p>Тема 5.5. Оптимизация. Основные принципы и методы оптимизации.</p> <p>Причины необходимости применения оптимизации при порождении результирующего кода компилятором. Определение оптимизации, основные принципы выполнения оптимизации, условие отсутствия влияния оптимизации на семантику результирующей программы. Критерии эффективности результирующей программы: скорость выполнения и потребность в вычислительных ресурсах (в основном – в оперативной памяти), зависимость оптимизации от критериев эффективности. Машинно-зависимая и машинно-независимая оптимизация, сочетание оптимизации со способами внутреннего представления программы и с процессом генерации результирующего кода. Основные машинно-независимые методы оптимизации (для линейных участков кода): свертка объектного кода, исключение лишних операций, оптимизация арифметических выражений, оптимизация логических выражений, оптимизация вызовов процедур и функций.</p>
6	<p>Тема 6.1. Структура систем программирования.</p> <p>Определение системы программирования. Основные составные части системы программирования: текстовый редактор, компилятор, компоновщик, библиотеки, загрузчик, отладчик, компилятор ресурсов, редактор ресурсов. Структурная схема системы программирования, последовательность обработки данных в системе программирования.</p> <p>Тема 6.2. История развития систем программирования.</p> <p>Начальный этап развития систем программирования: выполнение всей последовательности обработки данных в ручном режиме, высокая трудоемкость разработки, высокая вероятность возникновения ошибок в процессе обработки данных. Первый этап развития систем программирования: появление языка описания процесса обработки данных (Makefile), выполнение процесса обработки данных на основе интерпретатора Makefile, передача функций загрузчика в операционную систему (ОС). Второй этап развития систем программирования: появление интегрированных сред разработки, выполнение процесса обработки данных в одной среде (в окне текстового редактора). Первоначальное внедрение интегрированных сред в качестве обучающих систем программирования, их дальнейшее развитие и выход на рынок профессиональных средств разработки. Противостояние систем на основе Makefile и интегрированных сред на рынке профессиональных средств разработки, поглощение систем на основе Makefile системами на основе интегрированных сред. Третий этап развития систем программирования: появление графического интерфейса пользователя (GUI – Graphical User Interface), включение в состав систем программирования средств создания и обработки ресурсов пользовательского интерфейса. Взаимодействие систем программирования и средств автоматизированного проектирования программного обеспечения (CASE – Computer Aided Software Engineering).</p> <p>Тема 6.3. Функции основных компонент в составе систем программирования.</p>

	<p>Функции текстовых редакторов в составе системы программирования. Функции компилятора с входного языка. Назначение компоновщика (редактора связей), основные функции компоновщика. Библиотеки: состав библиотек по назначению, типы библиотек – статические и динамические библиотеки. Преимущества и недостатки статических и динамических библиотек, технологии дальнейшего развития динамических библиотек. Функции загрузчика, настраивающий и динамический загрузчик, взаимодействие систем программирования с ОС целевой вычислительной системы. Функции отладчика, взаимодействие отладчика с другими компонентами системы программирования, формирование и использование отладочной информации в результирующей программе. Ресурсы пользовательского интерфейса: определение ресурсов пользовательского интерфейса, состав, формирование и использование ресурсов пользовательского интерфейса. Функции компилятора ресурсов пользовательского интерфейса. Функции редактора ресурсов пользовательского интерфейса, взаимодействие редактора ресурсов пользовательского интерфейса с внешними источниками данных.</p>
7	<p>Тема 7.1. Принципы организации распределенных вычислений.</p> <p>Результирующая программа как выполняемое приложение. Возможные схемы организации распределенных вычислений, многообразие способов организации распределенных вычислений. Организация распределенных вычислений с разбиением приложений по уровням. Четыре основных уровня при построении приложений: уровень файловых операций, уровень операций со структурами данных, уровень бизнес-логики, уровень интерфейса с пользователем.</p> <p>Тема 7.2. Технология «файл-сервер».</p> <p>Выделение файловых операций на отдельный компьютер, понятие файлового сервера. Общее описание технологии «файл-сервер». Поддержка требований к файловым серверам на уровне современных ОС. Главное преимущество технологии «файл-сервер»: отсутствие каких-либо дополнительных требований к разработке приложений. Недостатки технологии «файл-сервер» и методы борьбы с ними.</p> <p>Тема 7.3. Терминальный сервер.</p> <p>Выделение средств по организации взаимодействия с пользователем на отдельный компьютер, понятие терминального сервера. Общее описание терминального сервера. Протоколы, используемые для организации терминальных серверов: протокол XWindow и протокол RDP (Remote Desktop Protocol). Особенности работы приложений на терминальном сервере. Требования к ресурсам терминального сервера.</p> <p>Тема 7.4. Технология «клиент-сервер».</p> <p>Выделение операций со структурами данных на отдельный компьютер, понятие сервера данных. Общее описание технологии «клиент-сервер», технология «клиент-сервер» как следующий шаг в развитии технологии «файл-сервер». Общие требования к серверу данных, определяющее требование по структуризации данных. Организация сервера данных с помощью СУБД (Система Управления Базами Данных). Требования к средствам разработки клиентских приложений по технологии «клиент-сервер».</p>

	<p>Взаимодействие клиентских приложений с сервером данных; организация взаимодействия с сервером данных, основные технологии взаимодействия, язык структурированных запросов SQL (Structured Query Language). Основные преимущества технологии «клиент-сервер» по сравнению с технологией «файл-сервер». Основные недостатки технологии «клиент-сервер» и методы борьбы с ними: перенос части бизнес-логики на сервер данных, использование терминальных серверов для функционирования клиентских приложений. Ограничения возможностей технологии «клиент-сервер».</p> <p>Тема 7.5. Трехуровневые и многоуровневые технологии разработки программного обеспечения.</p> <p>Вынесение операций бизнес-логики на отдельный компьютер, понятие сервера приложений. Общее описание трехуровневой технологии, трехуровневая технология как следующий шаг в развитии технологии «клиент-сервер». Общие требования к серверу приложений, взаимодействия сервера приложений и сервера данных. Взаимодействие сервера приложений и клиентских приложений, сложности в организации взаимодействия, технологии взаимодействия CORBA и COM/DCOM. Требования к средствам разработки клиентских приложений по трехуровневой технологии, понятие об интерфейсах взаимодействия (интерфейсах сервера приложений), организация интерфейсов взаимодействия, сравнение интерфейсов с таблицей RTTI и динамическими библиотеками. Шаблоны для построения серверов приложений, предлагаемые системами программирования. Возможность разбиения сервера приложений на несколько серверов, переход от трехуровневой к многоуровневой (multi-tier) технологии разработки. Основные преимущества трехуровневой и многоуровневой технологий по сравнению с технологиями «клиент-сервер» и «файл-сервер». Основные сложности разработки приложений по трехуровневой и многоуровневой технологиям. Современное состояние рынка средств разработки.</p> <p>Тема 7.6. Облачные вычисления и сервисы.</p> <p>Понятие об облачных вычислениях и сервисах, требования, предъявляемые к облачным вычислениям и сервисам. Облачные вычисления и сервисы, построенные по принципам IaaS (Infrastructure as a Service) и SaaS (Software as a Service) – основные понятия и возможности. Организация облачных сервисов на основе основных технологий разработки приложений: «файл-сервер», «клиент-сервер» и многоуровневая технология.</p>
8	<p>Тема 8.1. Особенности разработки программного обеспечения для глобальных сетей. Применение современных интерпретаторов.</p> <p>Главная особенность глобальных сетей – многообразие целевых вычислительных систем, отсутствие ограничений на структуру целевой вычислительной системы. Следствия по разработке программ для глобальных сетей: невозможность использования компиляторов, необходимость использования интерпретаторов. Глобальные сети – основная область применения современных интерпретаторов и интерпретируемых языков. Типовая структура программы навигации по сети Интернет (browser): сетевая часть, поддерживающая обмен данными по сети, и интерпретатор поступивших данных.</p> <p>Тема 8.2. Построение статических Web-страниц. Язык</p>

	<p>HTML.</p> <p>Статическая Web-страница – хранилище неизменяемых данных, организация статических Web-страниц, как сетевого хранилища структурированных файлов. Организация доступа к Web-страницам: протокол доступа HTTP (Hyper-Text Transfer Protocol), язык описания страниц HTML (Hyper-Text Markup Language). Любая программа навигации содержит интерпретатор HTML. Основной недостаток статических Web-страниц – предоставляемая информация не зависит от запросов и условий, задаваемых пользователями.</p> <p>Тема 8.3. Построение динамических Web-страниц на основе компилируемых и интерпретируемых языков.</p> <p>Принципы организации динамических Web-страниц: формирование текста HTML страницы с учетом данных и условий, полученных в запросе пользователя. Использование различных технологий разработки приложений (технология «клиент-сервер» и многоуровневая технология) для организации динамических Web-страниц. Сочетание динамических Web-страниц и технологии «клиент-сервер» как простейшее средство построения сервера приложений. Возможности взаимодействия сервера Интернет с приложениями, технологии CGI (Common-Gateway Interface) и ISAPI (Internet Server Application Program Interface), преимущества и недостатки каждой технологии. Построение взаимодействующих с сервером Интернет приложений с помощью компилятора или с помощью интерпретатора, преимущества и недостатки каждого способа. Ограничения по интерактивности динамических Web-страниц.</p> <p>Тема 8.4. Построение интерактивных Web-страниц.</p> <p>Требования к интерактивности Web-страниц и ограничения по организации интерактивности на стороне сервера. Принципы построения интерактивных Web-страниц: организация выполнения кода на стороне клиента. Способы организации выполнения кода на стороне клиента: скрипты (scripts) и скриптовые языки, интерпретаторы и интерпретируемые языки, встраиваемые компоненты (технология ActiveX и подобные ей). Ограничения по выполнению кода на стороне клиента. Вопросы безопасности выполнения кода на стороне клиента. Общая схема организации Web-страниц с учетом динамичности и интерактивности.</p>
--	--

4.3. Практические (семинарские) занятия

Темы практических занятий и их трудоемкость приведены в таблице 5.

Таблица 5 – Практические занятия и их трудоемкость

№ п/п	Темы практических занятий	Формы практических занятий	Трудоемкость, (час)	№ раздела дисциплины
Учебным планом не предусмотрено				
Всего				

4.4. Лабораторные занятия

Темы лабораторных занятий и их трудоемкость приведены в таблице 6.

Таблица 6 – Лабораторные занятия и их трудоемкость

№ п/п	Наименование лабораторных работ	Трудоемкость, (час)	№ раздела дисциплины
Семестр 7			
1	Вводное занятие	1	1
2	Работа с таблицами идентификаторов	1	1
3	Построение распознавателя для регулярной грамматики	1	2
4	Построение лексического анализатора	2	2
5	Построение синтаксического анализатора	2	3
6	Построение простейшего дерева вывода	1	3
Семестр 8			
7	Вводное занятие	1	5
8	Генерация объектного кода	1	5
9	Оптимизация объектного кода	1	5
10	Построение распознавателя для анализа описаний типов данных и переменных	1	5
11	Определение объема памяти для заданных структур данных	1	5
12	Разработка простейшего приложения на основе технологии «клиент-сервер»	3	7
Всего		16	

4.5. Курсовое проектирование/ выполнение курсовой работы
Учебным планом не предусмотрено

4.6. Самостоятельная работа обучающихся
Виды самостоятельной работы и ее трудоемкость приведены в таблице 7.

Таблица 7 – Виды самостоятельной работы и ее трудоемкость

Вид самостоятельной работы	Всего, час	Семестр 7, час	Семестр 8, час
1	2	3	4
Изучение теоретического материала дисциплины (ТО)		52	62
Курсовое проектирование (КП, КР)			
Расчетно-графические задания (РГЗ)			
Выполнение реферата (Р)			
Подготовка к текущему контролю успеваемости (ТКУ)		24	24
Домашнее задание (ДЗ)			
Контрольные работы заочников (КРЗ)			
Подготовка к промежуточной аттестации (ПА)		16	33
Всего:	211	92	119

5. Перечень учебно-методического обеспечения

для самостоятельной работы обучающихся по дисциплине (модулю)

Учебно-методические материалы для самостоятельной работы обучающихся указаны в п.п. 7-11.

6. Перечень печатных и электронных учебных изданий

Перечень печатных и электронных учебных изданий приведен в таблице 8.

Таблица 8– Перечень печатных и электронных учебных изданий

Шифр/ URL адрес	Библиографическая ссылка	Количество экземпляров в библиотеке (кроме электронных экземпляров)
004.7 K95	Интерфейсы взаимодействия приложений и распределенные вычисления /Н.В. Кучин, А.Ю. Молчанов – СПб.: Изд-во ГУАП, 2020 – 125 с.	40
004.7 K95	Многоуровневые системы и облачные вычисления /Н.В. Кучин, А.Ю. Молчанов – СПб.: Изд-во ГУАП, 2018 – 113 с.	40
004.4 M76	Системное программное обеспечение: лабораторный практикум /А. Ю. Молчанов; С.-Петербург. гос. ун-т аэрокосм. приборостроения. - СПб.: Изд-во ГУАП, 2014. – 131 с.	35
004.4(075) M75	Системное программное обеспечение: учебник / А. Ю. Молчанов. - СПб.: ПИТЕР, 2010. - 395 с	49
004.4(075) M75	Системное программное обеспечение: лабораторный практикум / А. Ю. Молчанов. - СПб.: ПИТЕР, 2005. - 283 с.	38
004.4 Г67	Системное программное обеспечение: учебник / А. В. Гордеев, А. Ю. Молчанов. - СПб.: ПИТЕР, 2002. - 734 с.	61

7. Перечень электронных образовательных ресурсов информационно-телекоммуникационной сети «Интернет»

Перечень электронных образовательных ресурсов информационно-телекоммуникационной сети «Интернет», необходимых для освоения дисциплины приведен в таблице 9.

Таблица 9 – Перечень электронных образовательных ресурсов информационно-телекоммуникационной сети «Интернет»

URL адрес	Наименование
http://www.fort-inform.ru/~mill/student.htm	Методические указания, перечень основных вопросов и вариантов по выполнению лабораторных работ
http://www.embarcadero.com/	Описание технологий и средств разработки, предлагаемых компанией Embarcadero
http://www.corba.com/	Описание технологий разработки, предлагаемых консорциумом OMG

https://www.visualstudio.com/ru-ru	Описание средств разработки, предлагаемых компанией Microsoft
---	---

8. Перечень информационных технологий

8.1. Перечень программного обеспечения, используемого при осуществлении образовательного процесса по дисциплине.

Перечень используемого программного обеспечения представлен в таблице 10.

Таблица 10– Перечень программного обеспечения

№ п/п	Наименование
1.	Свободно распространяемые средства разработки проекта GNU
2.	Свободно распространяемые СУБД типа Microsoft SQL Server Express

8.2. Перечень информационно-справочных систем, используемых при осуществлении образовательного процесса по дисциплине

Перечень используемых информационно-справочных систем представлен в таблице 11.

Таблица 11– Перечень информационно-справочных систем

№ п/п	Наименование
	Не предусмотрено

9. Материально-техническая база

Состав материально-технической базы, необходимой для осуществления образовательного процесса по дисциплине, представлен в таблице 12.

Таблица 12 – Состав материально-технической базы

№ п/п	Наименование составной части материально-технической базы	Номер аудитории (при необходимости)
1	Лекционная аудитория	32-04
2	Вычислительная лаборатория	52-09

10. Оценочные средства для проведения промежуточной аттестации

10.1. Состав оценочных средств для проведения промежуточной аттестации обучающихся по дисциплине приведен в таблице 13.

Таблица 13 – Состав оценочных средств для проведения промежуточной аттестации

Вид промежуточной аттестации	Перечень оценочных средств
Экзамен	Список вопросов к экзамену; Экзаменационные билеты; Задачи; Тесты.
Дифференцированный зачёт	Список вопросов; Тесты; Задачи.

10.2. В качестве критериев оценки уровня сформированности (освоения) компетенций обучающимися применяется 5-балльная шкала оценки сформированности компетенций, которая приведена в таблице 14. В течение семестра может использоваться

100-балльная шкала модульно-рейтинговой системы Университета, правила использования которой, установлены соответствующим локальным нормативным актом ГУАП.

Таблица 14 –Критерии оценки уровня сформированности компетенций

Оценка компетенции	Характеристика сформированных компетенций
5-балльная шкала	
«отлично» «зачтено»	– обучающийся глубоко и всесторонне усвоил программный материал; – уверенно, логично, последовательно и грамотно его излагает; – опираясь на знания основной и дополнительной литературы, тесно привязывает усвоенные научные положения с практической деятельностью направления; – умело обосновывает и аргументирует выдвигаемые им идеи; – делает выводы и обобщения; – свободно владеет системой специализированных понятий.
«хорошо» «зачтено»	– обучающийся твердо усвоил программный материал, грамотно и по существу излагает его, опираясь на знания основной литературы; – не допускает существенных неточностей; – увязывает усвоенные знания с практической деятельностью направления; – аргументирует научные положения; – делает выводы и обобщения; – владеет системой специализированных понятий.
«удовлетворительно» «зачтено»	– обучающийся усвоил только основной программный материал, по существу излагает его, опираясь на знания только основной литературы; – допускает несущественные ошибки и неточности; – испытывает затруднения в практическом применении знаний направления; – слабо аргументирует научные положения; – затрудняется в формулировании выводов и обобщений; – частично владеет системой специализированных понятий.
«неудовлетворительно» «не зачтено»	– обучающийся не усвоил значительной части программного материала; – допускает существенные ошибки и неточности при рассмотрении проблем в конкретном направлении; – испытывает трудности в практическом применении знаний; – не может аргументировать научные положения; – не формулирует выводов и обобщений.

10.3. Типовые контрольные задания или иные материалы.

Вопросы (задачи) для экзамена представлены в таблице 15.

Таблица 15 – Вопросы (задачи) для экзамена

№ п/п	Перечень вопросов (задач) для экзамена
1.	Трансляторы, компиляторы и интерпретаторы. Структура компилятора, основные этапы компиляции.
2.	Определение алфавита, цепочки символов, языка. Операции над цепочками символов. Способы задания языков.
3.	Грамматики. Определение грамматики. Форма Бэкуса-Наура. Другие формы представления грамматик.
4.	Классификация языков и грамматик по Хомскому. Проблема преобразования грамматик.
5.	Цепочки вывода. Сентенциальная форма. Правосторонний и левосторонний выводы. Дерево вывода. Однозначные и неоднозначные грамматики.

6.	Распознаватели. Общая схема распознавателя. Классификация распознавателей.
7.	Конечные автоматы (КА). Определение КА. Детерминированные и недетерминированные КА.
8.	Регулярные множества и регулярные выражения. Свойства регулярных выражений. Уравнения с регулярными коэффициентами.
9.	Связь регулярных множеств, регулярных грамматик и КА. Построение автоматной грамматики по заданной регулярной грамматике.
10.	Связь регулярных множеств, регулярных грамматик и КА. Построение КА по заданной автоматной грамматике.
11.	Автоматы с магазинной памятью (МП-автоматы). Расширенные МП-автоматы. Детерминированные МП-автоматы и ДКС-языки.
12.	Построение МП-автомата на основе КС-грамматики. МП-автомат с подбором альтернатив.
13.	Построение расширенного МП-автомата на основе КС-грамматики. Расширенный МП-автомат на основе алгоритма «сдвиг-свертка».
14.	Преобразование КС-грамматик. Алгоритмы разбора с возвратами и их характеристики. Табличные распознаватели и их характеристики.
15.	Свойства регулярных языков и грамматик. Свойства КС-языков и КС-грамматик.
16.	Распознавание цепочек КС-языков методом рекурсивного спуска.
17.	LL(k)-грамматики. Определение, свойства, принципы построения распознавателей.
18.	LR(k)-грамматики. Определение, свойства, принципы построения распознавателей. SLR(1) и LALR(1)-грамматики.
19.	Грамматика предшествования. Алгоритм «сдвиг-свертка» для грамматик предшествования.
20.	Грамматика предшествования. Построение отношений предшествования для грамматик операторного предшествования.
21.	Автоматизированные методы построения распознавателей. Программы LEX и YACC.
22.	Сканеры (лексические анализаторы). Назначение и функции лексического анализатора. Способы взаимосвязи синтаксического и лексического анализаторов.
23.	Таблицы символов. Организация таблиц символов в виде линейных списков и по методу бинарного дерева.
24.	Таблицы символов. Организация таблиц символов с помощью хэш-функций на основе рехэширования и на основе метода цепочек.
25.	Семантический анализ. Основные действия, выполняемые на этапе семантического анализа.
26.	Внутреннее представление программы. Методы внутреннего представления программ.
27.	Обратная польская запись. Вычисление выражений в форме обратной польской записи.
28.	Распределение памяти. Динамическая и статическая память, глобальная и локальная память.
29.	Распределение памяти. Распределение памяти под простые и сложные структуры данных.
30.	Распределение памяти. Стековая организация дисплея памяти, соглашения о вызовах.
31.	Таблицы RTTI – структура и назначение. Организация компилятором таблиц RTTI.

32.	Генерация кода. Назначение генерации кода. Принцип СУ-перевода.
33.	Оптимизация программ. Назначение оптимизации, основные принципы и методы оптимизации.
34.	Оптимизация программ. Оптимизация линейных участков: свертка объектного кода, исключение лишних операций.
35.	Оптимизация программ. Особенности оптимизации логических и арифметических выражений, вызовов процедур и функций.
36.	Понятие прохода. Однопроходные и многопроходные компиляторы.
37.	Язык ассемблера. Особенности построения компиляторов с языка ассемблера.
38.	Макроопределения и макрогенерация. Условная компиляция.
39.	Особенности построения интерпретаторов. Области применения современных интерпретаторов.
40.	Современные системы программирования. Структура и составные части систем программирования.
41.	Современные системы программирования. Функции текстовых редакторов в системах программирования.
42.	Современные системы программирования. Функции компоновщиков и загрузчиков в системах программирования.
43.	Современные системы программирования. Организация библиотек процедур и функций, статические и динамические библиотеки.
44.	Современные системы программирования. Ресурсы пользовательского интерфейса: создание, обработка и использование ресурсов.
45.	Мобильность и переносимость кода. Принципы разработки переносимого кода. Особенности разработки на языках Java и C#.
46.	Концепция организации распределенных вычислений. Технология «файл-сервер». Недостатки и преимущества технологии «файл-сервер».
47.	Концепция организации распределенных вычислений. Технология «клиент-сервер». Недостатки и преимущества технологии «клиент-сервер».
48.	Построение приложений по трехуровневой и многоуровневой технологии. Недостатки и преимущества многоуровневых технологий.
49.	Особенности разработки программного обеспечения для сети Интернет. Разработка статических Web-страниц.
50.	Разработка динамических Web-страниц на основе CGI, ISAPI и интерпретаторов. Разработка интерактивных Web-страниц.

Вопросы (задачи) для зачета / дифф. зачета представлены в таблице 16.

Таблица 16 – Вопросы (задачи) для зачета / дифф. зачета

№ п/п	Перечень вопросов (задач) для зачета / дифф. зачета
1.	Трансляторы, компиляторы и интерпретаторы. Структура компилятора, основные этапы компиляции.
2.	Определение алфавита, цепочки символов, языка. Операции над цепочками символов. Способы задания языков.
3.	Грамматика. Определение грамматики. Форма Бэкуса-Наура. Другие формы представления грамматик.
4.	Классификация языков и грамматик по Хомскому. Проблема преобразования грамматик.
5.	Цепочки вывода. Сентенциальная форма. Правосторонний и левосторонний вывод. Дерево вывода. Однозначные и неоднозначные грамматики.
6.	Распознаватели. Общая схема распознавателя. Классификация распознавателей.
7.	Конечные автоматы (КА). Определение КА. Детерминированные и

8.	недетерминированные КА. Регулярные множества и регулярные выражения. Свойства регулярных выражений. Уравнения с регулярными коэффициентами.
9.	Связь регулярных множеств, регулярных грамматик и КА. Построение автоматной грамматики по заданной регулярной грамматике.
10.	Связь регулярных множеств, регулярных грамматик и КА. Построение КА по заданной автоматной грамматике.
11.	Автоматы с магазинной памятью (МП-автоматы). Расширенные МП-автоматы. Детерминированные МП-автоматы и ДКС-языки.
12.	Построение МП-автомата на основе КС-грамматики. МП-автомат с подбором альтернатив.
13.	Построение расширенного МП-автомата на основе КС-грамматики. Расширенный МП-автомат на основе алгоритма «сдвиг-свертка».
14.	Преобразование КС-грамматик. Алгоритмы разбора с возвратами и их характеристики. Табличные распознаватели и их характеристики.
15.	Свойства регулярных языков и грамматик. Свойства КС-языков и КС-грамматик.
16.	Распознавание цепочек КС-языков методом рекурсивного спуска.
17.	LL(k)-грамматики. Определение, свойства, принципы построения распознавателей.
18.	LR(k)-грамматики. Определение, свойства, принципы построения распознавателей. SLR(1) и LALR(1)-грамматики.
19.	Грамматика предшествования. Алгоритм «сдвиг-свертка» для грамматик предшествования.
20.	Грамматика предшествования. Построение отношений предшествования для грамматик операторного предшествования.
21.	Автоматизированные методы построения распознавателей. Программы LEX и YACC.
22.	Сканеры (лексические анализаторы). Назначение и функции лексического анализатора. Способы взаимосвязи синтаксического и лексического анализаторов.
23.	Таблицы символов. Организация таблиц символов в виде линейных списков и по методу бинарного дерева.
24.	Таблицы символов. Организация таблиц символов с помощью хэш-функций на основе рехэширования и на основе метода цепочек.

Перечень тем для курсового проектирования/выполнения курсовой работы представлены в таблице 17.

Таблица 17 – Перечень тем для курсового проектирования/выполнения курсовой работы

№ п/п	Примерный перечень тем для курсового проектирования/выполнения курсовой работы
	Учебным планом не предусмотрено

Вопросы для проведения промежуточной аттестации в виде тестирования представлены в таблице 18.

Таблица 18 – Примерный перечень вопросов для тестов

№ п/п	Примерный перечень вопросов для тестов
1.	Определите принадлежность заданной грамматики к определенному классу по структуре ее правил.
2.	Постройте правосторонний и левосторонний вывод, а также дерево вывода для заданных цепочек символов на основе данной грамматики.

3.	Построить конечный автомат на основе заданной регулярной грамматики.
4.	Найти решение системы уравнений с регулярными коэффициентами.
5.	Построить систему уравнений с регулярными коэффициентами на основе заданной регулярной грамматики.
6.	Привести заданный конечный автомат к детерминированному виду.
7.	Удалить недостижимые символы из правил заданной грамматики.
8.	Удалить бесплодные (бесполезные) символы из правил заданной грамматики.
9.	Удалить λ -правила из заданной грамматики.
10.	Устранить цепные правила из заданной грамматики.
11.	Устранить левую рекурсию из заданной грамматики.
12.	Построить таблицу предшествования для заданной грамматики.
13.	Построить бинарное дерево для заданной последовательности идентификаторов.
14.	Определить необходимый объем памяти для заданного набора структур данных на основе заданного перечня скалярных типов данных и кратности размещения данных.
15.	Построить обратную постфиксную запись для заданной последовательности операций.
16.	Вычислить значение выражения, заданного в форме обратной постфиксной записи.
17.	Построить последовательность триад для заданной последовательности операций.
18.	Вычислить значение выражения на основе заданной последовательности триад.
19.	Выполнить оптимизацию методом свертки объектного кода для заданной последовательности триад.
20.	Выполнить оптимизацию методом исключения лишних операций для заданной последовательности триад.

Перечень тем контрольных работ по дисциплине обучающихся заочной формы обучения, представлены в таблице 19.

Таблица 19 – Перечень контрольных работ

№ п/п	Перечень контрольных работ
	Не предусмотрено

10.4. Методические материалы, определяющие процедуры оценивания индикаторов, характеризующих этапы формирования компетенций, содержатся в локальных нормативных актах ГУАП, регламентирующих порядок и процедуру проведения текущего контроля успеваемости и промежуточной аттестации обучающихся ГУАП.

11. Методические указания для обучающихся по освоению дисциплины

11.1. Методические указания для обучающихся по освоению лекционного материала.

Основное назначение лекционного материала – логически стройное, системное, глубокое и ясное изложение учебного материала. Назначение современной лекции в рамках дисциплины не в том, чтобы получить всю информацию по теме, а в освоении фундаментальных проблем дисциплины, методов научного познания, новейших достижений научной мысли. В учебном процессе лекция выполняет методологическую, организационную и информационную функции. Лекция раскрывает понятийный аппарат конкретной области знания, её проблемы, дает цельное представление о дисциплине, показывает взаимосвязь с другими дисциплинами.

Планируемые результаты при освоении обучающимися лекционного материала:

- получение современных, целостных, взаимосвязанных знаний, уровень которых определяется целевой установкой к каждой конкретной теме;
- получение опыта творческой работы совместно с преподавателем;
- развитие профессионально-деловых качеств, любви к предмету и самостоятельного творческого мышления.
- появление необходимого интереса, необходимого для самостоятельной работы;
- получение знаний о современном уровне развития науки и техники и о прогнозе их развития на ближайшие годы;
- научиться методически обрабатывать материал (выделять главные мысли и положения, приходить к конкретным выводам, повторять их в различных формулировках);
- получение точного понимания всех необходимых терминов и понятий.

Лекционный материал может сопровождаться демонстрацией слайдов и использованием раздаточного материала при проведении коротких дискуссий об особенностях применения отдельных тематик по дисциплине.

Структура предоставления лекционного материала:

- Изучение теоретических вопросов, связанных с рассматриваемой темой
- Описание программных средств, методов и алгоритмов, применяемых при разработке системного программного обеспечения
- Демонстрация примеров реализации описанных ранее алгоритмов и использования программных средств
- Обобщение изложенного материала
- Ответы на возникающие вопросы по теме лекции.

Конспекты лекций и указания по их использованию размещаются в личном кабинете преподавателя.

11.2. Методические указания для обучающихся по участию в семинарах

Не предусмотрено.

11.3. Методические указания для обучающихся по прохождению практических занятий

Не предусмотрено

11.4. Методические указания для обучающихся по выполнению лабораторных работ

В ходе выполнения лабораторных работ обучающийся должен углубить и закрепить знания, практические навыки, овладеть современной методикой и техникой эксперимента в соответствии с квалификационной характеристикой обучающегося. Выполнение лабораторных работ состоит из экспериментально-практической, расчетно-аналитической частей и контрольных мероприятий.

Выполнение лабораторных работ обучающимся является неотъемлемой частью изучения дисциплины, определяемой учебным планом, и относится к средствам, обеспечивающим решение следующих основных задач обучающегося:

- приобретение навыков исследования процессов, явлений и объектов, изучаемых в рамках данной дисциплины;
- закрепление, развитие и детализация теоретических знаний, полученных на лекциях;

- получение новой информации по изучаемой дисциплине;
- приобретение навыков самостоятельной работы с лабораторным оборудованием и приборами.

Задание и требования к проведению лабораторных работ

Вариант задания по каждой лабораторной работе обучающийся получает в соответствии с номером в списке группы. Перед началом выполнения лабораторной работы обучающемуся следует внимательно ознакомиться с методическими указаниями по ее выполнению. В соответствии с заданием обучающийся должен подготовить необходимые данные, получить от преподавателя допуск к выполнению лабораторной работы, разработать и отладить программный продукт, в результате выполнения разработанного программного продукта получить требуемые результаты, продемонстрировать их преподавателю, оформить и защитить отчет по лабораторной работе.

Структура и форма отчета о лабораторной работе

Отчет о лабораторной работе должен включать в себя:

- титульный лист;
- полную формулировку задания в соответствии с полученным вариантом;
- теоретические сведения, используемые при выполнении лабораторной работы;
- результаты выполненных предварительных расчетов;
- краткое описание использованных средств разработки;
- полученные результаты и выводы.

Требования к оформлению отчета о лабораторной работе

По каждой лабораторной работе выполняется отдельный отчет. В соответствии с методическими указаниями, допускается составлять единый отчет для 2 лабораторных работ, объединенных одной темой.

Титульный лист оформляется в соответствии с шаблоном (образцом) приведенным на сайте ГУАП (www.guap.ru) в разделе «Сектор нормативной документации». Текстовые и графические материалы оформляются в соответствии с действующими ГОСТами и требованиями, приведенными на сайте ГУАП (www.guap.ru) в разделе «Сектор нормативной документации».

Методические указания по прохождению лабораторных работ:

1. Системное программное обеспечение: лабораторный практикум / А. Ю. Молчанов; С.-Петербург. гос. ун-т аэрокосм. приборостроения. - СПб.: Изд-во ГУАП, 2014. - 131 с.: рис., табл.
2. Системное программное обеспечение: методические указания к выполнению лабораторных работ / С.-Петербург. гос. ун-т аэрокосм. приборостроения; сост. Т. Л. Прокофьева. - СПб. : Изд-во ГУАП, 2012. - 67 с.: рис.

11.5. Методические указания для обучающихся по прохождению курсового проектирования/выполнения курсовой работы

Не предусмотрено

11.6. Методические указания для обучающихся по прохождению самостоятельной работы

В ходе выполнения самостоятельной работы, обучающийся выполняет работу по заданию и при методическом руководстве преподавателя, но без его непосредственного участия.

Для обучающихся по заочной форме обучения, самостоятельная работа может включать в себя контрольную работу.

В процессе выполнения самостоятельной работы, у обучающегося формируется целесообразное планирование рабочего времени, которое позволяет им развивать умения и навыки в усвоении и систематизации приобретаемых знаний, обеспечивает высокий уровень успеваемости в период обучения, помогает получить навыки повышения профессионального уровня.

Методическими материалами, направляющими самостоятельную работу обучающихся являются:

- учебно-методический материал по дисциплине;
- методические указания по выполнению контрольных работ (для обучающихся по заочной форме обучения).

11.7. Методические указания для обучающихся по прохождению текущего контроля успеваемости.

Текущий контроль успеваемости предусматривает контроль качества знаний обучающихся, осуществляемого в течение семестра с целью оценивания хода освоения дисциплины.

Текущий контроль успеваемости проводится в форме устной беседы и выполнения тестов после предоставления обучающимся отчета по каждой лабораторной работе. Результаты промежуточного контроля успеваемости фиксируются в электронном журнале в личном кабинете преподавателя вместе с отчетами о выполнении лабораторных работ в виде баллов, полученных за выполнение каждой лабораторной работы. Сумма баллов, набранная обучающимся за семестр, принимается во внимание при проведении промежуточной аттестации.

11.8. Методические указания для обучающихся по прохождению промежуточной аттестации.

Промежуточная аттестация обучающихся предусматривает оценивание промежуточных и окончательных результатов обучения по дисциплине. Она включает в себя:

- экзамен – форма оценки знаний, полученных обучающимся в процессе изучения всей дисциплины или ее части, навыков самостоятельной работы, способности применять их для решения практических задач. Экзамен, как правило, проводится в период экзаменационной сессии и завершается аттестационной оценкой «отлично», «хорошо», «удовлетворительно», «неудовлетворительно».
- зачет – это форма оценки знаний, полученных обучающимся в ходе изучения учебной дисциплины в целом или промежуточная (по окончании семестра) оценка знаний обучающимся по отдельным разделам дисциплины с аттестационной оценкой «зачтено» или «не зачтено».
- дифференцированный зачет – это форма оценки знаний, полученных обучающимся при изучении дисциплины, при выполнении курсовых проектов, курсовых работ, научно-исследовательских работ и прохождении практик с аттестационной оценкой «отлично», «хорошо», «удовлетворительно», «неудовлетворительно».

Система оценок при проведении промежуточной аттестации осуществляется в соответствии с требованиями Положений «О текущем контроле успеваемости и промежуточной аттестации студентов ГУАП, обучающихся по программы высшего образования» и «О модульно-рейтинговой системе оценки качества учебной работы студентов в ГУАП».

Лист внесения изменений в рабочую программу дисциплины

Дата внесения изменений и дополнений. Подпись внесшего изменения	Содержание изменений и дополнений	Дата и № протокола заседания кафедры	Подпись зав. кафедрой