

**МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ  
РОССИЙСКОЙ ФЕДЕРАЦИИ**

**ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ БЮДЖЕТНОЕ ОБРАЗОВАТЕЛЬНОЕ  
УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ  
«Рязанский государственный радиотехнический университет имени В.Ф. Уткина»**


**КАФЕДРА «ЭЛЕКТРОННЫЕ ВЫЧИСЛИТЕЛЬНЫЕ МАШИНЫ»**


**«СОГЛАСОВАНО»**

Декан факультета ВТ  
 Д.А. Перепелкин  
«\_\_» \_\_\_\_\_ 2020 г.



**«УТВЕРЖДАЮ»**

Проректор по РОПиМД  
 А.В. Корячко  
«\_\_» \_\_\_\_\_ 2020 г.

Заведующий кафедрой ЭВМ  
 Б.В. Костров  
«\_\_» \_\_\_\_\_ 2020 г.

**РАБОЧАЯ ПРОГРАММА ДИСЦИПЛИНЫ  
ФТД.03 «Функциональное программирование»**

Направление подготовки

02.03.03 Математическое обеспечение и администрирование информационных систем

ОПОП академического бакалавриата

«Математическое обеспечение и администрирование информационных систем»

Квалификация (степень) выпускника — бакалавр

Форма обучения — очная

Рязань, 2020 г.

## ЛИСТ СОГЛАСОВАНИЙ

Рабочая программа составлена с учетом требований федерального государственного образовательного стандарта высшего образования по направлению подготовки (специальности) 02.03.03 Математическое обеспечение и администрирование информационных систем (уровень бакалавриата), утвержденным приказом Минобрнауки России от 23.08.2017г. № 809.

Программу составил

д.т.н., доцент,

доцент кафедры

«Электронных вычислительных машин»



В.А. Саблина

Программа рассмотрена и одобрена на заседании кафедры ЭВМ

«11» 06 2020 г., протокол № 10

Заведующий кафедрой

«Электронные вычислительные машины»,

д.т.н., проф. кафедры ЭВМ



Б.В. Костров



## 1. Перечень планируемых результатов обучения по дисциплине, соотнесенных с планируемыми результатами освоения образовательной программы магистратуры

Рабочая программа по дисциплине «Функциональное программирование» является составной частью основной профессиональной образовательной программы (ОПОП) академического бакалавриата «Математическое обеспечение и администрирование информационных систем», разработанной в соответствии с ФГОС ВО по направлению подготовки 02.03.03 Математическое обеспечение и администрирование информационных систем (уровень бакалавриата), утвержденным приказом Минобрнауки России от 23.08.2017 г. № 809.

*Целью освоения дисциплины «Функциональное программирование» является изучение основных понятий языков функционального программирования и особенностей их применения.*

Основные задачи освоения учебной дисциплины:

1. получение теоретических знаний об основных понятиях языков функционального программирования;
2. Приобретение практических навыков функционального программирования на языке JavaScript;
3. Овладение навыками разработки и отладки программ на языке JavaScript с использованием возможностей функционального программирования.

### Перечень планируемых результатов обучения по дисциплине

Код компетенции	Содержание компетенций	Перечень планируемых результатов обучения по дисциплине
ОПК-2	Способен применять современный математический аппарат, связанный с проектированием, разработкой, реализацией и оценкой качества программных продуктов и программных комплексов в различных областях человеческой деятельности	ОПК-2.1. Знает: математические основы программирования и языков программирования, организации баз данных и компьютерного моделирования; математические методы оценки качества, надежности и эффективности программных продуктов; математические методы организации информационной безопасности при разработке и эксплуатации программных продуктов и программных комплексов. ОПК-2.2. Умеет использовать этот аппарат в профессиональной деятельности. ОПК-2.3. Имеет навыки применения данного математического аппарата при решении конкретных задач.

## 2. Место дисциплины в структуре ОПОП бакалавриата

Дисциплина «Функциональное программирование» является факультативной, относится к блоку «Факультативы» основной профессиональной образовательной программы академического бакалавриата «Математическое обеспечение и администрирование информационных систем» по направлению подготовки 02.03.03 Математическое обеспечение и администрирование информационных систем ФГБОУ ВО «РГРТУ».

Дисциплина изучается по очной форме обучения на 4 курсе в 7 семестре.

Программа курса ориентирована на возможность расширения и углубления знаний, умений и навыков бакалавра для успешной профессиональной деятельности.

### 3. Объем дисциплины и виды учебной работы

Общая трудоемкость (объем) дисциплины составляет 2 зачетные единицы (ЗЕ), 72 часов.

Вид учебной работы	Всего часов
Общая трудоемкость дисциплины	72
1. Контактная работа обучающихся с преподавателем (всего), в том числе:	32,25
лекции	32
практические занятия	-
лабораторные работы	-
консультации	-
иная контактная работа (промежуточная аттестация)	0,25
2. Самостоятельная работа обучающихся (всего), в том числе:	31
курсовой проект (работа)	-
иная самостоятельная работа	31
3. Контроль	8,75
Вид промежуточной аттестации обучающегося	зачет

### 4. Содержание дисциплины

#### 4.1 Содержание дисциплины, структурированное по темам

##### Тема 1. Основы функционального программирования.

Сущность функционального программирования. Декларативный характер функционального программирования. Чистые функции. Ссылочная прозрачность. Сохранение данных неизменяемыми. Преимущества функционального программирования.

##### Тема 2. Функциональное программирование в сравнении с объектно-ориентированным программированием.

Управление состоянием объектов в JavaScript. Обращение с объектами как со значениями. Глубокое замораживание подвижных частей. Перемещение по графам объектов и их модификация с помощью линз.

##### Тема 3. Функции, замыкания и области видимости.

Функции первого класса. Функции высшего порядка. Способы вызова функций. Методы функций. Трудности, связанные с соблюдением глобальной области видимости. Область видимости функций JavaScript. Область видимости псевдоблока. Практические примеры применения замыканий.

##### Тема 4. Концепции функционального программирования.

Общее представление о потоке управления прикладной программой. Связывание методов в цепочку. Связывание функций в цепочку. Анализ прикладного кода. Рекурсия. Повторно используемый, модульный код. Проектные шаблоны и сложность. Отказоустойчивость прикладного кода. Оптимизация функционального кода. Обработка асинхронных событий и данных.

##### Тема 5. JavaScript как язык для реализации модели клиент-сервер.

Язык JavaScript как язык для разработки программного обеспечения распределенных систем. Типы данных. Переменные и идентификаторы. Присваивание и операторы. Ветвление.

##### Тема 6. Функции JavaScript.

Определение функции. Параметры функции. Особенности использования областей видимости в JavaScript. Локальная область видимости. Глобальная область видимости. Рекурсия в JavaScript.

##### Тема 7. Объекты JavaScript.

Тип «объект». Ассоциативные массивы. Создание объекта. Операции с объектом. Наличие свойства. Доступ к свойству через переменную. Объявление со свойствами. Передача по ссылке. Встроенные объекты.

### Тема 8. NodeJS и работа с файловой системой.

Асинхронная событийная модель работы. Модульная структура основных библиотек. Работа с файловой системой. Основные действия: чтение файла, нахождение пути к файлу, асинхронное чтение файла, удаление файла, копирование файла, переименование файла, чтение директории, создание директории, удаление пустой директории, запись в файл, открытие файла.

### 4.2. Разделы дисциплины и трудоемкость по видам учебных занятий (в академических часах).

Название раздела	Общая трудоемкость, всего часов	Контактная работа обучающихся с преподавателем					Самостоятельная работа обучающихся	Контроль
		Всего	Лекции	Практические занятия	Консультации	Иные виды контактной работы		
Тема 1. Основы функционального программирования	8	4	4	-	-	-	3	1
Тема 2. Функциональное программирование в сравнении с объектно-ориентированным программированием	9	4	4	-	-	-	4	1
Тема 3. Функции, замыкания и области видимости	9	4	4	-	-	-	4	1
Тема 4. Концепции функционального программирования	9	4	4	-	-	-	4	1
Тема 5. JavaScript как язык для реализации модели клиент-сервер	9	4	4	-	-	-	4	1
Тема 6. Функции JavaScript	9	4	4	-	-	-	4	1
Тема 7. Объекты JavaScript	9	4	4	-	-	-	4	1
Тема 8. NodeJS и работа с файловой системой	9,75	4	4	-	-	-	4	1,75
Промежуточная аттестация	0,25	0,25	-	-	-	0,25	-	-
Итого	72	32,25	32	-	-	0,25	31	8,75

### Виды самостоятельных работ

Тема	Вид работы	Наименование и содержание работы	Трудоемкость, часов
Тема 1. Основы функционального программирования.	Самостоятельная работа	Изучение конспекта лекций	2
		Подготовка к зачету и консультации	1
Тема 2. Функциональное программирование в сравнении с объектно-ориентированным программированием.	Самостоятельная работа	Изучение конспекта лекций	2
		Подготовка к зачету и консультации	2

Тема	Вид работы	Наименование и содержание работы	Трудоемкость, часов
Тема 3. Функции, замыкания и области видимости.	Самостоятельная работа	Изучение конспекта лекций	2
		Подготовка к зачету и консультации	2
Тема 4. Концепции функционального программирования.	Самостоятельная работа	Изучение конспекта лекций	2
		Подготовка к зачету и консультации	2
Тема 5. JavaScript как язык для реализации модели клиент-сервер.	Самостоятельная работа	Изучение конспекта лекций	2
		Подготовка к зачету и консультации	2
Тема 6. Функции JavaScript.	Самостоятельная работа	Изучение конспекта лекций	2
		Подготовка к зачету и консультации	2
Тема 7. Объекты JavaScript.	Самостоятельная работа	Изучение конспекта лекций	2
		Подготовка к зачету и консультации	2
Тема 8. NodeJS и работа с файловой системой.	Самостоятельная работа	Изучение конспекта лекций	2
		Подготовка к зачету и консультации	2

## 5 ПЕРЕЧЕНЬ УЧЕБНО-МЕТОДИЧЕСКОГО ОБЕСПЕЧЕНИЯ ДЛЯ САМОСТОЯТЕЛЬНОЙ РАБОТЫ ОБУЧАЮЩИХСЯ ПО ДИСЦИПЛИНЕ

- 1) Засорин С.В. Функциональное программирование: метод. указ. к лаб. работам. - РГРТУ. - Рязань, 2011. - 56 с.

## 6 ФОНД ОЦЕНОЧНЫХ СРЕДСТВ ДЛЯ ПРОВЕДЕНИЯ ПРОМЕЖУТОЧНОЙ АТТЕСТАЦИИ ОБУЧАЮЩИХСЯ ПО ДИСЦИПЛИНЕ

Фонд оценочных средств приведен в Приложении к рабочей программе дисциплины (см. документ «Оценочные материалы по дисциплине «Функциональное программирование»).

## 7 ПЕРЕЧЕНЬ ОСНОВНОЙ И ДОПОЛНИТЕЛЬНОЙ УЧЕБНОЙ ЛИТЕРАТУРЫ, НЕОБХОДИМОЙ ДЛЯ ОСВОЕНИЯ ДИСЦИПЛИНЫ

### Основная учебная литература:

1. Васильев В.И., Ильясов Б.Г. Интеллектуальные системы управления. Теория и практика : учеб. пособие. - М. : Радиотехника, 2009. - 392 с.
2. Кузьмин Ю.М., Пржегорлинский В.Н. Методы программирования : учеб. пособие. Ч.1. Основы объектно-ориентированного программирования. - РГРТУ. - Рязань, 2015. - 64 с.
3. Рогозин О.В. Функциональное и рекурсивно-логическое программирование [Электронный ресурс] : учебное пособие / О.В. Рогозин. — Электрон. текстовые данные. — М. : Евразийский открытый институт, 2009. — 139 с. — 978-5-374-00182-2. — Режим доступа: <http://www.iprbookshop.ru/11119.html> (дата обращения: 01.02.2018).
4. Городняя Л.В. Основы функционального программирования [Электронный ресурс] / Л.В. Городняя. — 2-е изд. — Электрон. текстовые данные. — М. : Интернет-Университет Информационных Технологий (ИНТУИТ), 2016. — 246 с. — 2227-8397. — Режим доступа: <http://www.iprbookshop.ru/73703.html> (дата обращения: 01.02.2018).

### **Дополнительная учебная литература:**

1. Ручкин В.Н., Фулин В.А. Универсальный искусственный интеллект и экспертные системы . - СПб. : БХВ-Петербург, 2009. - 240 с.
2. Афонин В.Л., Макушкин В.А. Интеллектуальные робототехнические системы: курс лекций : учеб. пособие для вузов. - М. : Интернет-Ун-т Информ.Технологий, 2009. - 207 с.
3. Салмина Н.Ю. Функциональное программирование и интеллектуальные системы [Электронный ресурс] : учебное пособие / Н.Ю. Салмина. — Электрон. текстовые данные. — Томск: Томский государственный университет систем управления и радиоэлектроники, 2016. — 100 с. — 2227-8397. — Режим доступа: <http://www.iprbookshop.ru/72216.html> (дата обращения: 01.02.2018).
4. Сошников Д.В. Функциональное программирование на F# [Электронный ресурс] / Д.В. Сошников. — Электрон. текстовые данные. — Саратов: Профобразование, 2017. — 191 с. — 978-5-4488-0131-0. — Режим доступа: <http://www.iprbookshop.ru/63961.html> (дата обращения: 01.02.2018).
5. Душкин Р.В. Функциональное программирование на языке Haskell [Электронный ресурс] / Р.В. Душкин. — Электрон. текстовые данные. — Саратов: Профобразование, 2017. — 608 с. — 978-5-4488-0044-3. — Режим доступа: <http://www.iprbookshop.ru/64053.html> (дата обращения: 01.02.2018).
6. Практикум по дисциплине Логическое и функциональное программирование [Электронный ресурс] / . — Электрон. текстовые данные. — М. : Московский технический университет связи и информатики, 2016. — 16 с. — 2227-8397. — Режим доступа: <http://www.iprbookshop.ru/61489.html> (дата обращения: 01.02.2018).

### **Законодательные и нормативные акты:**

ГОСТ Р 7.0.8-2013. Национальный стандарт Российской Федерации. Система стандартов по информации, библиотечному и издательскому делу. Делопроизводство и архивное дело. Термины и определения" (утв. Приказом Росстандарта от 17.10.2013 N 1185-ст) // Официальный сайт справочной правовой системы КонсультантПлюс <http://www.consultant.ru>

## **8 ПЕРЕЧЕНЬ РЕСУРСОВ ИНФОРМАЦИОННО-ТЕЛЕКОММУНИКАЦИОННОЙ СЕТИ «ИНТЕР-НЕТ», НЕОБХОДИМЫХ ДЛЯ ОСВОЕНИЯ ДИСЦИПЛИНЫ**

Обучающимся предоставлена возможность индивидуального доступа к следующим электронно-библиотечным системам.

Электронно-библиотечная система «IPRbooks», режим доступа – с любого компьютера РГРТУ без пароля, из сети интернет по паролю. – URL: <https://iprbookshop.ru/>.

## **9 МЕТОДИЧЕСКИЕ УКАЗАНИЯ ДЛЯ ОБУЧАЮЩИХСЯ ПО ОСВОЕНИЮ ДИСЦИПЛИНЫ**

### **Указания в рамках лекций**

Во время лекции студент должен вести краткий конспект.

Работа с конспектом лекций предполагает просмотр конспекта в тот же день после занятий. При этом необходимо пометить материалы конспекта, которые вызывают затруднения для понимания. При этом обучающийся должен стараться найти ответы на затруднительные вопросы, используя рекомендуемую литературу. Если ему самостоятельно не удалось разобраться в материале, необходимо сформулировать вопросы и обратиться за помощью к преподавателю на консультации или ближайшей лекции.

Обучающимся необходимо регулярно отводить время для повторения пройденного материала, проверяя свои знания, умения и навыки по контрольным вопросам.

### **Указания в рамках подготовки к промежуточной аттестации**

При подготовке к экзамену в дополнение к изучению конспектов лекций, учебных пособий, слайдов и другого раздаточного материала предусмотренного рабочей программой дисциплины, необходимо пользоваться учебной литературой, рекомендованной к настоящей рабочей программе.

При подготовке к экзамену нужно изучить теорию: определения всех понятий и подходы к оцениванию до состояния понимания материала и самостоятельно решить по нескольким типовым задачам из каждой темы (в том случае если тема предусматривает решение задач). При решении задач всегда необходимо уметь качественно интерпретировать итог решения.

### **Указания в рамках самостоятельной работы студентов**

Самостоятельная работа студентов по дисциплине играет важную роль в ходе всего учебного процесса. Методические материалы и рекомендации для обеспечения самостоятельной работы студентов готовятся преподавателем и выдаются студентам в виде раздаточных материалов или оформляются в виде электронного ресурса используемого в рамках системы дистанционного обучения ФГБОУ ВО «РГРТУ».

Самостоятельное изучение тем учебной дисциплины способствует:

- закреплению знаний, умений и навыков, полученных в ходе аудиторных занятий;
- углублению и расширению знаний по отдельным вопросам и темам дисциплины;
- освоению умений прикладного и практического использования полученных знаний.

Самостоятельная работа как вид учебной работы может использоваться на лекциях и практических, а также иметь самостоятельное значение – внеаудиторная самостоятельная работа обучающихся – при подготовке к лекциям, практическим занятиям, а также к экзамену.

Основными видами самостоятельной работы по дисциплине являются:

- самостоятельное изучение отдельных вопросов и тем дисциплины;
- выполнение практического задания;
- выполнение домашнего задания;
- подготовка к защите практического задания, оформление отчета.

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

Теоретический материал курса становится более понятным, когда дополнительно к прослушиванию лекции и изучению конспекта, изучается дополнительная рекомендованная литература. Литературу по курсу рекомендуется изучать в библиотеке, с использованием доступной электронной библиотечной системы или с помощью сети Интернет (источники, которые могут быть использованы без нарушения авторских прав).

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

При проведении занятий по дисциплине используются следующие информационные технологии:

- удаленные информационные коммуникации между студентами и преподавателем, ведущим лекционные и практические занятия, посредством информационной образовательной среды ФГБОУ ВО «РГРТУ», позволяющие осуществлять оперативный контроль графика выполнения и содержания образовательного процесса, решение организационных вопросов, консультирование;
- доступ к современным профессиональным базам данных (в том числе международным реферативным базам данных научных изданий) и информационным справочным системам;
- проведение аудиторных занятий с использованием презентаций и раздаточных материалов в электронном виде;
- выполнение студентами различных видов учебных работ с использованием лицензион-



ного программного обеспечения, установленного на рабочих местах студента в компьютерных классах и в помещениях для самостоятельной работы, а также для выполнения самостоятельной работы в домашних условиях.

Обучающимся по данной дисциплине предоставляется доступ к дистанционным курсам, расположенным в системе дистанционного обучения ФГБОУ ВО «РГРТУ»:

- 1) Аналитические этапы проектирования информационных систем [Электронный ресурс]. URL: <http://cdo.rsreu.ru/user/view.php?id=4764&course=1453> (дата обращения 21.12.2017).
- 2) Технология разработки программного обеспечения [Электронный ресурс]. URL: <http://cdo.rsreu.ru/course/view.php?id=1058> (дата обращения 21.12.2017).
- 3) Современные технологии разработки интегрированных ИС [Электронный ресурс]. URL: <http://cdo.rsreu.ru/course/view.php?id=1175> (дата обращения 21.12.2017).
- 4) Базы данных. Проектирование информационных систем [Электронный ресурс]. URL: <http://cdo.rsreu.ru/course/view.php?id=1036> (дата обращения 21.12.2017).
- 5) Современные технологии БД [Электронный ресурс]. URL: <http://cdo.rsreu.ru/course/view.php?id=1174> (дата обращения 21.12.2017).
- 6) Базы данных. Язык SQL [Электронный ресурс]. URL: <http://cdo.rsreu.ru/course/view.php?id=1001> (дата обращения 21.12.2017).

Система дистанционного обучения ФГБОУ ВО «РГРТУ» доступна как из внутренней информационной системы организации, так и из глобальной сети Интернет.

#### **Перечень лицензионного программного обеспечения:**

- 1) Операционная система Windows XP Professional (лицензия Microsoft DreamSpark Membership ID 700102019);
- 2) LibreOffice 4.4 (лицензия LGPLv3);
- 3) Microsoft Visual Studio (лицензия Microsoft DreamSpark Membership ID 700102019).

#### **Перечень профессиональных баз данных (в том числе международным реферативным базам данных научных изданий) и информационных справочных систем:**

- 1) Информационно-правовой портал ГАРАНТ.РУ [Электронный ресурс]. – URL: <http://www.garant.ru>. – Режим доступа: свободный доступ (дата обращения 02.02.2018).
- 2) Справочная правовая система КонсультантПлюс [Электронный ресурс]. – URL: <http://www.consultant.ru/online/>. – Режим доступа: свободный доступ (будние дни – 20.00 - 24.00, выходные и праздничные дни – круглосуточно) (дата обращения 02.02.2018).

### **11 ОПИСАНИЕ МАТЕРИАЛЬНО-ТЕХНИЧЕСКОЙ БАЗЫ, НЕОБХОДИМОЙ ДЛЯ ОСУЩЕСТВЛЕНИЯ ОБРАЗОВАТЕЛЬНОГО ПРОЦЕССА ПО ДИСЦИПЛИНЕ**

Для освоения дисциплины необходимы:

- 1) для проведения лекционных занятий необходима аудитория с достаточным количеством посадочных мест, соответствующая необходимым противопожарным нормам и санитарно-гигиеническим требованиям;
- 2) для проведения практических занятий необходим класс персональных компьютеров с установленными операционными системами Microsoft Windows XP (или выше) и установленным открытым программным обеспечением MS Visual Studio;
- 3) для проведения лекций аудитория должна быть оснащена проекционным оборудованием.

**МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ  
РОССИЙСКОЙ ФЕДЕРАЦИИ**

**ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ БЮДЖЕТНОЕ ОБРАЗОВАТЕЛЬНОЕ  
УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ  
«Рязанский государственный радиотехнический университет имени В.Ф. Уткина»**

**КАФЕДРА «ЭЛЕКТРОННЫЕ ВЫЧИСЛИТЕЛЬНЫЕ МАШИНЫ»**

**ОЦЕНОЧНЫЕ МАТЕРИАЛЫ  
ФТД.03 «Функциональное программирование»**

Направление подготовки

02.03.03 Математическое обеспечение и администрирование информационных систем

ОПОП академического бакалавриата

«Математическое обеспечение и администрирование информационных систем»

Квалификация (степень) выпускника — бакалавр

Форма обучения — очная

Рязань, 2020 г.

## 1 ОБЩИЕ ПОЛОЖЕНИЯ

Оценочные материалы – это совокупность учебно-методических материалов (практических заданий, описаний форм и процедур проверки), предназначенных для оценки качества освоения обучающимися данной дисциплины как части ОПОП.

Цель – оценить соответствие знаний, умений и владений, приобретенных обучающимся в процессе изучения дисциплины, целям и требованиям ОПОП в ходе проведения промежуточной аттестации.

Основная задача – обеспечить оценку уровня сформированности компетенций, закрепленных за дисциплиной.

Контроль знаний обучающихся проводится в форме промежуточной аттестации. Промежуточная аттестация проводится в форме зачета.

Форма проведения зачета – тестирование, письменный опрос по теоретическим вопросам.

## 2 ОПИСАНИЕ ПОКАЗАТЕЛЕЙ И КРИТЕРИЕВ ОЦЕНИВАНИЯ КОМПЕТЕНЦИЙ

Сформированность каждой компетенции (или ее части) в рамках освоения данной дисциплины оценивается по трехуровневой шкале:

- 1) пороговый уровень является обязательным для всех обучающихся по завершении освоения дисциплины;
- 2) продвинутый уровень характеризуется превышением минимальных характеристик сформированности компетенций по завершении освоения дисциплины;
- 3) эталонный уровень характеризуется максимально возможной выраженностью компетенций и является важным качественным ориентиром для самосовершенствования.

**Уровень освоения компетенций, формируемых дисциплиной:**

**Описание критериев и шкалы оценивания теоретического вопроса (каждого):**

Шкала оценивания	Критерий
3 балла (эталонный уровень)	выставляется студенту, который дал полный ответ на вопрос, показал глубокие систематизированные знания, смог привести примеры, ответил на дополнительные вопросы преподавателя
2 балла (продвинутый уровень)	выставляется студенту, который дал полный ответ на вопрос, но на некоторые дополнительные вопросы преподавателя ответил только с помощью наводящих вопросов
1 балл (пороговый уровень)	выставляется студенту, который дал неполный ответ на вопрос в билете и смог ответить на дополнительные вопросы только с помощью преподавателя
0 баллов	выставляется студенту, который не смог ответить на вопрос

На промежуточную аттестацию (зачет) выносятся два теоретических вопроса. Максимально студент может набрать 6 баллов. Итоговый суммарный балл студента, полученный при прохождении промежуточной аттестации, переводится в традиционную форму по системе «зачтено», «не зачтено».

**Оценка «зачтено»** выставляется студенту, который набрал в сумме не менее 4 баллов (выполнил одно задание на эталонном уровне, другое – не ниже

порогового, либо оба задания выполнит на продвинутом уровне). Обязательным условием является выполнение всех предусмотренных в течение семестра практических и лабораторных работ заданий.

**Оценка «не зачтено»** выставляется студенту, который набрал в сумме менее 4 баллов, либо имеет к моменту проведения промежуточной аттестации несданные практические, либо лабораторные работы.

### 3 ПАСПОРТ ОЦЕНОЧНЫХ МАТЕРИАЛОВ ПО ДИСЦИПЛИНЕ

<b>Контролируемые разделы (темы) дисциплины</b>	<b>Код контролируемой компетенции (или её части)</b>	<b>Вид, метод, форма оценочного мероприятия</b>
Тема 1. Основы функционального программирования	<b>ОПК-2</b>	<b>Зачет</b>
Тема 2. Функциональное программирование в сравнении с объектно-ориентированным программированием	<b>ОПК-2</b>	<b>Зачет</b>
Тема 3. Функции, замыкания и области видимости	<b>ОПК-2</b>	<b>Зачет</b>
Тема 4. Концепции функционального программирования	<b>ОПК-2</b>	<b>Зачет</b>
Тема 5. JavaScript как язык для реализации модели клиент-сервер	<b>ОПК-2</b>	<b>Зачет</b>
Тема 6. Функции JavaScript	<b>ОПК-2</b>	<b>Зачет</b>
Тема 7. Объекты JavaScript	<b>ОПК-2</b>	<b>Зачет</b>
Тема 8. NodeJS и работа с файловой системой	<b>ОПК-2</b>	<b>Зачет</b>

## 4 ТИПОВЫЕ КОНТРОЛЬНЫЕ ЗАДАНИЯ ИЛИ ИНЫЕ МАТЕРИАЛЫ

### 4.1. Промежуточная аттестация в форме зачета

Код компетенции	Результаты освоения ОПОП Содержание компетенций
ОПК-1	Способен применять естественнонаучные и общетехнические знания, методы математического анализа и моделирования, теоретического и экспериментального исследования в профессиональной деятельности

#### Типовые тестовые вопросы:

1. Какой метод решения задач особенно характерен для функционального программирования?  
+символьная обработка информации;  
преобразование состояний памяти;  
обработка сигналов;  
симплекс-метод.
2. Какой из перечисленных механизмов реализации языка Лисп способствовал практическому успеху функционального программирования?  
+автоматизация повторного использования памяти — «сбор мусора»;  
математическая основа исходных понятий;  
отсутствие «синтаксического сахара» в представлении программ;  
бэктрекинг.
3. Какая особенность функционального подхода дает путь к достижению надежности программ?  
+доказательность основных построений при разработке универсальных функций;  
разнообразие встроенных функций и библиотек;  
высокий уровень языковых средств;  
использование рекурсии.
4. Кто впервые сформулировал идеи языка программирования, послужившие основой для функционального программирования?  
Николас Вирт;  
Тони Хоар;  
+Джон Мак-Карти.  
Норберт Винер.
5. Какой из перечисленных механизмов реализации языка Лисп способствовал практичности функционального программирования?  
+традиционное включение в реализацию и интерпретатора, и компилятора одновременно;  
полнота средств управления вычислениями;  
отсутствие «синтаксического сахара» в представлении программ;  
бэктрекинг.
6. В каком языке программирования впервые поддержаны идеи функционального программирования?  
+Lisp;  
Python;  
Prolog;  
JavaScript.



7. Какова основная структура данных языка Lisp?

+список;  
атом;  
таблица;  
массив.

8. Где содержатся пары «переменная-значение» в элементарном языке Lisp?

в базе данных переменных;  
+в ассоциативном списке;  
в хэш-таблице;  
в стеке.

9. Какой подход в построении программ преимущественно используется в Lisp-системах программирования?

объектно-ориентированный;  
+функциональный;  
императивный;  
рекурсивно-логический.

10. Что понимается под термином «форма» в языке Lisp?

список из идентификатора функции, ее аргументов и ее всевозможных значений;  
+список из представления функции и перечня ее аргументов;  
программа на языке Lisp;  
отдельный модуль программы.

11. Что называют программой на языке Lisp?

последовательность вычисляемых атомов;  
последовательность хорновских дизъюнктов;  
последовательность вызовов функций;  
+последовательность вычисляемых форм и выражений.

12. Какой подход в построении программ наиболее подходит для использования в Lisp-системах программирования?

императивный;  
интерфейсный;  
+функциональный;  
декларативный.

13. Какой принцип является основным для функционального программирования?

представление функции строится заранее;  
+представление функции строится и вычисляется также как и представление данных;  
при символьном представлении данных представление функции нельзя построить теми же средствами, что и обработку значений;  
представление функции строится на основе рекурсии.

14. Что называют элементарными функциями в Lisp-системе?

любые неделимые элементы — атомы;  
встраиваемые S-выражения, имеющие определения на уровне исполнимого кода;  
+атомы, определенные как подпрограммы на уровне исполняемого кода;  
функции без параметров или с одним параметром.

15. Данные какого вида и типа могут выступать в качестве аргументов функции в Lisp?  
списковые домены;  
+любая форма произвольной длины и сложности;  
числа и строки;  
только атомы.

16. Каким языком программирования является JavaScript?  
+мультипарадигменный язык программирования;  
рекурсивно-логический язык программирования;  
компилируемый язык программирования;  
сильно типизированный язык.

17. Что получится в результате выполнения следующего кода на JavaScript?  
(function(){  
 return typeof arguments;  
})();  
+"object";  
"array";  
"arguments";  
"undefined".

18. Что получится в результате выполнения следующего кода на JavaScript?  
var f = function g(){ return 23; };  
 typeof g();  
"number";  
"undefined";  
"function";  
+Error.

19. Что получится в результате выполнения следующего кода на JavaScript?  
(function(x){  
 delete x;  
 return x;  
})(1);  
+1;  
null;  
undefined;  
Error.

20. Что получится в результате выполнения следующего кода на JavaScript?  
var y = 1, x = y = typeof x;  
 x;  
1;  
"number";  
undefined;  
"undefined".

21. Что получится в результате выполнения следующего кода на JavaScript?  
(function f(f){  
 return typeof f();  
})(function(){ return 1; });  
+"number";

```
"undefined";  
"function";  
Error.
```

22. Что получится в результате выполнения следующего кода на JavaScript?

```
var foo = {  
  bar: function() { return this.baz; },  
  baz: 1  
};  
(function(){  
  return typeof arguments[0]();  
})(foo.bar);  
+"undefined";  
"object";  
"number";  
"function".
```

23. Что получится в результате выполнения следующего кода на JavaScript?

```
var foo = {  
  bar: function() { return this.baz; },  
  baz: 1  
}  
typeof (f = foo.bar());  
+"undefined";  
"object";  
"number";  
"function".
```

24. Что получится в результате выполнения следующего кода на JavaScript?

```
var f = (function f(){ return "1"; }, function g(){ return 2; }());  
typeof f;  
"string";  
+"number";  
"function";  
"undefined".
```

25. Что получится в результате выполнения следующего кода на JavaScript?

```
var x = 1;  
if (function f(){} ) {  
  x += typeof f;  
}  
x;  
1;  
"1function";  
+"1undefined";  
NaN.
```

26. Что получится в результате выполнения следующего кода на JavaScript?

```
var x = [typeof x, typeof y][1];  
typeof typeof x;  
"number";  
+"string";
```

"undefined";  
"object".

27. Что получится в результате выполнения следующего кода на JavaScript?

```
(function(foo){  
    return typeof foo.bar;  
})({ foo: { bar: 1 } });  
+"undefined";  
"object";  
"number";  
Error.
```

28. Что получится в результате выполнения следующего кода на JavaScript?

```
(function f(){  
    function f(){ return 1; }  
    return f();  
    function f(){ return 2; }  
})();  
1;  
+2;  
Error (e.g. "Too much recursion");  
undefined.
```

29. Что получится в результате выполнения следующего кода на JavaScript?

```
function f(){ return f; }  
new f() instanceof f;  
true;  
+false;  
null;  
one.
```

30. Что получится в результате выполнения следующего кода на JavaScript?

```
with (function(x, undefined){}) length;  
1;  
+2;  
undefined;  
Error.
```

### **Типовые практические задания:**

#### ***Задание 1***

При помощи цикла for выведите чётные числа от 2 до 10.

#### ***Критерии выполнения задания 1***

Задание считается выполненным, если: обучающийся разработал программу на языке JavaScript, корректно решающую поставленную задачу, и может объяснить принцип ее работы.

#### ***Задание 2***

Натуральное число, большее 1, называется простым, если оно ни на что не делится, кроме себя и 1. Другими словами,  $n > 1$  – простое, если при делении на любое число от 2 до  $n-1$  есть остаток. Создайте код, который выводит все простые числа из интервала от 2 до 10. Результат должен быть: 2,3,5,7.

### **Критерии выполнения задания 2**

Задание считается выполненным, если: обучающийся разработал программу на языке JavaScript, корректно решающую поставленную задачу, и может объяснить принцип ее работы.

### **Задание 3**

Реализовать функцию `square(array)`, возвращающую массив, состоящий из квадратов всех элементов исходного массива. Попробуйте использовать для этого метод массива `map`.

### **Критерии выполнения задания 3**

Задание считается выполненным, если: обучающийся разработал программу на языке JavaScript, корректно решающую поставленную задачу, и может объяснить принцип ее работы.

### **Задание 4**

Удалите из строки 'Я очень люблю программировать на Javascript' все гласные буквы.

### **Критерии выполнения задания 4**

Задание считается выполненным, если: обучающийся разработал программу на языке JavaScript, корректно решающую поставленную задачу, и может объяснить принцип ее работы.

### **Задание 5**

Замените `for` на `while`

```
for (var i = 0; i < 3; i++) {  
  console.log( "номер " + i + "!" );  
}
```

### **Критерии выполнения задания 5**

Задание считается выполненным, если: обучающийся разработал программу на языке JavaScript, корректно решающую поставленную задачу, и может объяснить принцип ее работы.

### **Задание 6**

Найти максимальный и минимальный элементы в массиве  
`var arr = [2,11,4,6,1,2,9,12,3].`

### **Критерии выполнения задания 6**

Задание считается выполненным, если: обучающийся разработал программу на языке JavaScript, корректно решающую поставленную задачу, и может объяснить принцип ее работы.

### **Задание 7**

Реализуйте функцию `sum(array)`, возвращающую сумму всех элементов массива. Попробуйте использовать для этого метод массива `reduce`.

### **Критерии оценки задания 7**

Задание считается выполненным, если: обучающийся разработал программу на языке JavaScript, корректно решающую поставленную задачу, и может объяснить принцип ее работы.

### **Задание 8**

Создайте функцию `clone()`, позволяющую создать копию объекта `person1` так, чтобы при изменении одного объекта второй оставался прежним. Например:

```
var person1 = {  
  hair: "brown",  
  eyes: "green",  
  age: 15  
}
```



```
var person2 = clone(person1);

person2.hair = "red";
person1.hair === "brown"; // true
```

### ***Критерии оценки задания 8***

Задание считается выполненным, если: обучающийся разработал программу на языке JavaScript, корректно решающую поставленную задачу, и может объяснить принцип ее работы.

### ***Задание 9***

Найти первое число в массиве: `var arr = [2,11,4,6,1,2,9,12,3];` больше 10 и прекратить выполнение цикла.

### ***Критерии выполнения задания 9***

Задание считается выполненным, если: обучающийся разработал программу на языке JavaScript, корректно решающую поставленную задачу, и может объяснить принцип ее работы.

### ***Задание 10***

С помощью методов массива `split`, `join`, `push`, `pop`, `shift`, `unshift`, `splice` превратите строку 'Не люблю программировать на Pascal' в строку 'Я очень люблю программировать на Javascript'.

Вы должны получить следующий код:

```
var string = 'Не люблю программировать на Pascal';
// ваш код
console.log(string); // выведет 'Я очень люблю программировать на Javascript'
```

### ***Критерии выполнения задания 10***

Задание считается выполненным, если: обучающийся разработал программу на языке JavaScript, корректно решающую поставленную задачу, и может объяснить принцип ее работы.

### ***Задание 11***

Реализовать рекурсивную факториал-функцию `factorial(n)`.

### ***Критерии выполнения задания 11***

Задание считается выполненным, если: обучающийся разработал программу на языке JavaScript, корректно решающую поставленную задачу, и может объяснить принцип ее работы.

### ***Задание 12***

Создайте функцию `merge()`, объединяющую свойства двух объектов. Если попадают свойства с одинаковым именем, отдавайте приоритет второму объекту:

```
var person1 = {
  hair: "brown",
  eyes: "green",
  age: 15
}
```

```
var person2 = {
  hair: "red",
  height: 180,
  isMale: true
}
```

```
var result = merge(person1, person2);
```

```
/*  
result = {  
  hair: "red",  
  eyes: "green",  
  age: 15,  
  height: 180,  
  isMale: true  
}  
*/
```

### ***Критерии выполнения задания 12***

Задание считается выполненным, если: обучающийся разработал программу на языке JavaScript, корректно решающую поставленную задачу, и может объяснить принцип ее работы.

#### **Типовые теоретические вопросы:**

- 1) Сущность функционального программирования.
- 2) Декларативный характер функционального программирования.
- 3) Чистые функции.
- 4) Ссылочная прозрачность.
- 5) Сохранение данных неизменяемыми.
- 6) Преимущества функционального программирования.
- 7) Функциональное программирование в сравнении с объектно-ориентированным программированием.
- 8) Функции, замыкания и области видимости.
- 9) Концепции функционального программирования.
- 10) Функциональное программирование на языке JavaScript.
- 11) Локальная область видимости.
- 12) Операции с объектом в JavaScript.
- 13) Асинхронная событийная модель работы в NodeJS.
- 14) Соединение пользователей с ресурсами.
- 15) Открытость.
- 16) Технологии масштабирования.
- 17) Гомогенные мультикомпьютерные системы.
- 18) Распределенные операционные системы.
- 19) Мультикомпьютерные операционные системы.
- 20) Программное обеспечение промежуточного уровня.
- 21) Варианты архитектуры клиент-сервер.
- 22) Уровень обработки.
- 23) Типы данных в JavaScript.
- 24) Глобальная область видимости.
- 25) Доступ к свойству объекта в JavaScript через переменную.
- 26) Модульная структура основных библиотек в NodeJS.
- 27) Прозрачность в распределенных системах.
- 28) Отделение правил от механизмов.
- 29) Общие концепции аппаратных решений.
- 30) Гетерогенные мультикомпьютерные системы.
- 31) Операционные системы для однопроцессорных компьютеров.
- 32) Системы с распределенной разделяемой памятью.
- 33) Клиенты и серверы.
- 34) Многозвенные архитектуры.

- 35) Современные варианты архитектуры.
- 36) Уровень данных.
- 37) Переменные и идентификаторы в JavaScript.
- 38) Рекурсия.
- 39) Встроенные объекты.
- 40) Работа с файловой системой в NodeJS.