

МИНОБРНАУКИ РОССИИ



Федеральное государственное автономное образовательное учреждение  
высшего образования

**«Российский государственный гуманитарный университет»  
(ФГАОУ ВО «РГГУ»)**

*ОТДЕЛЕНИЕ ИНТЕЛЛЕКТУАЛЬНЫХ СИСТЕМ В ГУМАНИТАРНОЙ СФЕРЕ*

*Учебно-научный центр программного и лингвистического обеспечения  
интеллектуальных систем*

***ПРОГРАММНОЕ И ЛИНГВИСТИЧЕСКОЕ ОБЕСПЕЧЕНИЕ ИНТЕЛЛЕКТУАЛЬНЫХ СИСТЕМ***

РАБОЧАЯ ПРОГРАММА ДИСЦИПЛИНЫ

45.04.04 Интеллектуальные системы в гуманитарной среде

Когнитивное и программное обеспечение интеллектуальных роботов и программирование интеллектуальных систем

Уровень квалификации выпускника: магистр

Форма обучения: очная

РПД адаптирована для лиц

с ограниченными возможностями

здоровья и инвалидов

Москва 2025

«Программное и лингвистическое обеспечение интеллектуальных систем»

Рабочая программа дисциплины

Составитель:

доцент

М.Е. Елифанов,

УТВЕРЖДЕНО

Протокол заседания кафедры МЛИИС

№ 3 от 10.12.2024.

## **ОГЛАВЛЕНИЕ**

### **1. Пояснительная записка**

- 1.1 Цель и задачи дисциплины (*модуля*)
- 1.2. Формируемые компетенции, соотнесённые с планируемыми результатами обучения по дисциплине
- 1.3. Место дисциплины в структуре образовательной программы

### **2. Структура дисциплины**

### **3. Содержание дисциплины**

### **4. Образовательные технологии**

### **5. Оценка планируемых результатов обучения**

- 5.1. Система оценивания
- 5.2. Критерии выставления оценок
- 5.3. Оценочные средства (материалы) для текущего контроля успеваемости, промежуточной аттестации обучающихся по дисциплине

### **6. Учебно-методическое и информационное обеспечение дисциплины**

- 6.1. Список источников и литературы
- 6.2. Перечень ресурсов информационно-телекоммуникационной сети «Интернет»

### **7. Материально-техническое обеспечение дисциплины**

### **8. Обеспечение образовательного процесса для лиц с ограниченными возможностями здоровья и инвалидов**

### **9. Методические материалы**

- 9.1. Планы семинарских занятий
- 9.2. Методические указания для обучающихся по освоению дисциплины
- 9.3. Иные материалы

## **Приложения**

Приложение 1. Аннотация дисциплины

Приложение 2. Фонд оценочных средств для проведения промежуточной аттестации обучающихся по дисциплине

## 1. Пояснительная записка

### 1.1. Цели и задачи дисциплины

**Цели дисциплины.** Главная цель курса – дать студенту целостное представление о концепциях, технологиях и средствах современного программирования, а также методов их применения к разработке интеллектуальных систем, включая такой специфический их компонент, как подсистема интеллектуального диалога с интеллектуальной системой. Другими целями курса можно считать обучение слушателей работе с научной и технической литературой, технической документацией в области программирования, способствовать формированию у студентов навыков работы самостоятельного программиста.

**Задачи дисциплины:** освоение средств объектно-ориентированного и функционального программирования, средств разработки реляционных баз данных и удаленного доступа к ним, методов автоматического анализа текста на естественном (русском) языке и программной реализации «больших» словарей.

1.2. Формируемые компетенции, соотнесённые с планируемыми результатами обучения по дисциплине:

### Перечень планируемых результатов обучения по дисциплине, соотнесенных с индикаторами достижения компетенций

Компетенция (код и наименование)	Индикаторы компетенций (код и наименование)	Результаты обучения
УК-2 Способен управлять проектом на всех этапах его жизненного цикла.	<p>УК-2.1 Знает различные модели жизненного цикла и стандарты на представление этапов работы над проектом</p> <p>УК-2.2 Умеет использовать современные бизнес- и информационные технологии для реализации проектов на различных этапах жизненного цикла</p> <p>УК-2.3 Имеет практический опыт участия в реализации проектов на разных этапах жизненного цикла</p>	<p><i>Знать:</i> современные подходы и технологии, применяемые для разработки программных приложений.</p> <p><i>Уметь:</i> пользоваться современными образовательными и информационными технологиями для получения знаний.</p> <p><i>Владеть</i> навыками: - анализа документации к программным системам; - самостоятельного специалиста в области программной реализации интеллектуальных систем.</p>
ОПК-1 Способен применять в профессиональной деятельности методы математического анализа, логики и моделирования, теоретического и экспериментального исследования в информатике, лингвистике и гуманитарных науках	<p>ОПК-1.1 Знает основы математического анализа, логики и математического моделирования</p> <p>ОПК-1.2 Умеет использовать математические методы для построения моделей в информатике, лингвистике и некоторых гуманитарных дисциплинах</p> <p>ОПК-1.3 Владеет методами теоретического и экспериментального исследования в информатике, лингвистике и гуманитарных науках</p>	<p><i>Знать:</i> современные подходы и технологии, применяемые в инструментальных средах (IDE) для совместной разработки программных приложений.</p> <p><i>Уметь:</i> использовать спецификации проектов программных приложений; методы доступа к информационным ресурсам; основные принципы разработки прикладных программных систем; перерабатывать большие объемы информации и вычленять главное (анализ информации); приобретать с большой степенью самостоятельности новые знания, используя современные образовательные и информационные технологии; использовать документацию к программным системам и стандартам в области программирования и информационных систем в практической деятельности.</p> <p><i>Владеть:</i> навыками совместной работы в рабочих группах программистов</p>
ОПК-3 Способен использовать фундаментальные знания в области гуманитарных, социальных и лингвистических наук, а также в сфере техники и технологии информатики для совершенствования профессиональной деятельности	<p>ОПК-3.1 Знает основы лингвистики, социологии и математической статистики</p> <p>ОПК-3.2 Умеет пользоваться инструментальными средствами, библиотеками и фреймворками для анализа социологических данных и текстов</p> <p>ОПК-3.3 Имеет практический опыт работы с инструментальными средствами для</p>	<p><i>Знать:</i> - современные подходы и технологии, применяемые для разработки программных приложений; - базовые типы данных и возможности для определения новых типов данных в изучаемых языках; стандартные структуры данных (в том числе списки, стеки, очереди, деки, деревья, графы) и варианты их представления в программах, методы построения сложных многоссылочных сетевых структур; - основы теории реляционных баз данных (БД), в частности,</p>

	анализа данных, в том числе анализа текстов	принципы построения и проектирования реляционных БД; - различные подходы к реализации автоматического морфологического, синтаксического и семантического анализа текста на естественном (русском) языке. <i>Уметь:</i> - решать на основе вышеперечисленных знаний задачи по программированию компонентов интеллектуальных систем в упрощенном виде; - эксплуатировать БД реляционного типа как пользователь и как системный администратор; комплексно разрабатывать базы данных, реализуя их, в частности, в архитектуре клиент/сервер <i>Владеть:</i> самостоятельного специалиста в области программной реализации интеллектуальных систем
<i>ОПК-5</i> Способен применять новые информационные технологии в гуманитарных областях знаний с использованием средств интеллектуального анализа данных, машинного обучения, компьютерной лингвистики и представления знаний	<i>ОПК-5.1</i> Знает теоретические основы машинного обучения, задачи, решаемые с помощью машинного обучения и интеллектуального анализа данных <i>ОПК-5.2</i> Умеет применять методы интеллектуального анализа данных и машинного обучения для анализа данных в гуманитарных областях <i>ОПК-5.3</i> Имеет практический опыт применения интеллектуального анализа данных, машинного обучения, компьютерной лингвистики и представления знаний к исследованиям в гуманитарных областях	

### 1.3. Место дисциплины в структуре образовательной программы

Дисциплина «Программное и лингвистическое обеспечение интеллектуальных систем» входит как обязательная в состав вариативной части учебного плана.

Для освоения дисциплины необходимы компетенции, сформированные (в рамках бакалавриата) в ходе изучения следующих дисциплин: «Алгебра», «Математическая логика», «Программирование», «Базы данных», «Технологии и средства разработки Интернет-приложений», «Функциональное программирование» и «Язык программирования Java».

Курс согласуется с читаемыми параллельно в этих же семестрах курсами «Методы современного программирования», «Логика интеллектуальных систем» и «Проектирование интеллектуальных систем».

В результате освоения дисциплины формируются знания, умения и владения, необходимые для изучения следующих дисциплин и прохождения практик: выполнение ВКР (магистерской диссертации).

## 2. Структура дисциплины

### Структура дисциплины для очной формы обучения

Общая трудоёмкость дисциплины составляет 9 з.е., 324 ч., в том числе контактная работа обучающихся с преподавателем 90 ч., самостоятельная работа обучающихся 216 ч.

Дисциплина читается с 1-го по 3-й семестр.

№ п/п	Раздел дисциплины/темы	Семестр	Виды учебной работы (в часах)					Формы текущего контроля успеваемости, форма промежуточной аттестации (по семестрам)	
			Контактная				Промежуточная аттестация		Самостоятельная работа
			Лекции и	Семинары	Практические занятия	Лабораторные занятия			
1	Средства и языки программирования: концепции, стили и реализация	1				2	2	Оценка выполнения упражнений, опрос	
2	Язык Java	1				4	4	Оценка выполнения практических заданий	
3	Библиотека Java	1				6	10	Оценка выполнения практических заданий	
4	Язык Common Lisp.	1				4	8	Оценка выполнения	

	Рекурсивное программирование, «чистый» Лисп								практических заданий, контрольная работа
5	Язык Common Lisp. Общий обзор средств	1				8		20	Оценка выполнения практических заданий, контрольная работа
6	CLOS (Common Lisp Object System)	1				6		16	Оценка выполнения практических заданий контрольная работа
	Промежуточная аттестация	1					<b>12</b>		экзамен
7	Практикум: сетевая объектная модель	2				4		10	Оценка совместной работы со студентом, как с исполнителем
8	Реляционные формализмы ч.1: реляционная алгебра и основы теории проектирования реляционных БД	2				4		6	Оценка выполнения упражнений, контрольная работа опрос блиц-опрос
9	Практические вопросы разработки реляционных БД	2				4		6	Оценка выполнения практических заданий, опрос в форме обсуждения некоторых вопросов проектирования схемы данных и реализации интерфейса к БД
10	Языки разметки и языки сценариев	2				2		8	Оценка выполнения практических заданий
11	Разработка приложений на основе DHTML и XML.	2				6		8	Оценка выполнения практических заданий
12	Архитектуры распределенных приложений.	2				2		6	Оценка выполнения практических заданий, опрос блиц-опрос
13	Практикум: реляционное моделирование и выполнение индивидуального учебного проекта БД с удаленным доступом	2				6		12	Оценка совместной работы со студентом, как с исполнителем
14	Реляционные формализмы ч.2: исчисления кортежей и доменов	2				2		4	Оценка выполнения упражнений, опрос
	Промежуточная аттестация	2					<b>12</b>		экзамен
15	Практикум: сетевая объектная модель (окончание)	3				6		12	Оценка совместной работы со студентом, как с исполнителем
16	Практикум: реляционное моделирование и выполнение индивидуального учебного проекта БД с удаленным доступом (окончание)	3				10		24	Оценка совместной работы со студентом, как с исполнителем
17	Архитектура интеллектуальной системы (ИС). Обзор	3				4		6	Опрос, опрос в форме обсуждения,

	возможных методов и средств реализации ИС.								блиц-опрос
18	Автоматический анализ русскоязычного текста. Обзор подходов к решению проблемы.	3				2		6	Опрос, опрос в форме обсуждения, блиц-опрос
19	Методы реализации словарей и морфологический анализ текста	3				4		20	Оценка выполнения практических заданий, опрос, опрос в форме обсуждения, блиц-опрос
20	Система поверхностно-синтаксического анализа русского предложения	3				4		10	Опрос, опрос в форме обсуждения, блиц-опрос, домашнее контрольное задание
	Итоговая аттестация	3					12	18	зачет с оценкой
	Итого					<b>90</b>	<b>36</b>	<b>216</b>	

### 3. Содержание дисциплины

№ п/п	Наименование раздела дисциплины	Содержание раздела
1.	Средства и языки программирования: концепции, стили и реализация	Средства программирования, языки программирования, краткая история их развития. Стили. Основные стили: процедурный, функциональный, логический. От модульного программирования к ООП. Виды ООП. Компонентное программирование.
2.	Язык Java	Кроссплатформенные приложения. Виртуальная Java-машина (JVM). «Устройство» (файлы) приложения на языке Java. Интегрированная среда разработки NetBeans. Компилятор. Обзор языка Java: - парадигма ООП, - переменные и типы данных, операции, управляющие операторы, - классы, структура и методы, инкапсуляция, полиморфизм, наследование, - пакеты и интерфейсы, - обработка исключений, - ввод-вывод, представление о многопоточном программировании в Java.
3.	Библиотека Java	Работа со строками. Пакет java.lang. Пакет java.util – коллекции и сервисные классы. Ввод-вывод – пакет java.io. Обработка событий. Системы AWT и Swing. Компоненты Java Beans. Средства для реализации распределенных приложений (включая класс Applet, сервлеты, ...)
4.	Язык Common Lisp. Рекурсивное программирование, «чистый» Лисп	Введение в программирование на языке Common Lisp: - синтаксис: основной тип данных – списки; атомы, их типы; S-выражения, функции; - основные встроенные функции: -- «чистый Лисп» и интерпретатор (eval, apply, funcall) Интегрированная среда разработки Cormon Common Lisp
5.	Язык Common Lisp. Общий обзор средств	Макросы и их определение. Функции. Символы. Функционалы (map- и другие функции). Управляющие конструкции: ветвления, циклы, блоки (функции и макросы их реализующие). Обзор типов атомов.
6	CLOS (Common Lisp Object System)	Определения классов. Объекты. Множественное наследование. Методы и обобщенные функции.

		Системные функции для работы с объектами и определениями классов
7, 15	Практикум: сетевая объектная модель (начало – 2 семестр – раздел 7, окончание – 3 семестр – раздел 15)	Основные «сложные» динамические структуры данных (списки, динамические массивы, множества, стеки, очереди, деки, графы (в частности, деревья и сети) и возможные способы их реализации (процедурные языки, Common Lisp). Объектные модели. Реализация многосвязной сетевой объектной модели, функциональность которой включает поиск узла, редактирование сети и навигацию по сети (блуждание по ее узлам). Разработка входных языков описания такой модели. Сериализация и десериализация модели. Сохранение состояния модели в файле и в БД.
8	Реляционные формализмы ч.1: реляционная алгебра и основы теории проектирования реляционных БД	Формализация отношений. Определения и свойства основных операторов реляционной алгебры; классификация различных операторов выбора (селекции) и соединения. Определение реляционной алгебры, ее выражения; ограничение множества ее операторов (независимость). Зависимости между данными. Виды зависимостей (функциональные, многозначные, вида «проекция-соединение») и соответствующие им отношения логического следования с адекватными ему системами вывода. Соответствующие нормальные формы. Методы декомпозиции и синтеза для построения 3-ей нормальной формы
9	Практические вопросы разработки реляционных БД	Обзор инструментальных средств в интегрированных средах разработки (IDE) реляционных БД MS Access и MySQL Workbench. Разработка таблиц и запросов, визуальное программирование форм и отчетов. Программирование на языке VBA (Visual Basic for Applications) Access с использованием объектных моделей доступа к данным (ADO и DAO) и объектной модели приложения Access (объекты форм, элементов управления, etc). Различные подходы к разработке интерфейса БД. Представление о SQL Server.
10	Языки разметки и языки сценариев	XML – синтаксис; типы документов и прикладные («предметные») XML-языки, средства определения типа документа (грамматики прикладных языков): XDR- и XSD-схемы, DTD. HTML – синтаксис и основные элементы. Общий обзор технологии и средств DHTML. Документ HTML и его структура. CSS (какадные таблицы стилей). Сценарии, языки JavaScript и VBScript. Окно браузера и фрейм. Объекты DHTML, работа с DOM из кода JavaScript. Модель событий DHTML. HTML компоненты. ООП в JavaScript: прототипы (в JavaScript и JScript), кратко – о классах в JavaScript.NET Особенности HTML 5
11	Разработка приложений на основе DHTML и XML.	Программирование индивидуальных элементов HTML. Динамические стили, абсолютное и относительное позиционирование, анимация. Изменение содержания документа HTML. Изменение поведения элементов. XML DOM работа с ним из кода JavaScript. Применение XSL и XSLT для отображения XML документа в документ DHTML. Внедрение ActiveX элементов и Java апплетов на страницы DHTML. Библиотека jQuery и ее применение. Представление об архитектуре MVC, JavaScriptMVC framework
12	Архитектуры распределенных приложений.	Использование ADO для связи с другими приложениями. Общее представление о двухуровневой (клиент/сервер) и трехуровневой архитектурах распределенных приложений, о Windows DNA на основе COM+. Технологии ASP, Ajax, формат JSON и их применение. Реализация серверных приложений на платформе Windows и средствами Java.
13, 16	Практикум: реляционное моделирование и выполнение индивидуального учебного проекта БД с удаленным доступом (начало – 2 семестр – раздел 13, окончание – 3 семестр – раздел 16)	Некоторые примеры концептуальных схем, отражающие различные «трудности» моделирования предметной области (разрабатываются всей группой совместно с преподавателем). Примеры удачных учебных проектов прошлых лет. Разработка проекта в архитектуре «клиент/сервер» (БД на сервере, удаленный интерфейс как клиентское приложение) на локальном компьютере (с использованием localhost). Контроль преподавателем основных стадий разработки БД, выполняемой студентом в рамках его индивидуального проекта.
14	Реляционные формализмы ч.2: исчисления кортежей и доменов	Формальные системы запросов, их эквивалентность и выразительная полнота. Реляционные исчисления кортежей и исчисления доменов, определение их синтаксиса и семантики (как для логического языка). Их эквивалентность (с учетом ограничений) реляционной алгебре. Языки SQL и

		Query by Example.
15	Практикум: сетевая объектная модель (окончание)	См. п. 7 наст. таблицы
16	Практикум: реляционное моделирование и выполнение индивидуального учебного проекта БД с удаленным доступом (окончание)	См. п. 13 наст. таблицы
17	Архитектура интеллектуальной системы (ИС). Обзор возможных методов и средств реализации ИС.	Основные подсистемы ИС: базы фактов и знаний, решатель и его компоненты, интерфейса с пользователем. Краткий обзор истории развития систем (методов) искусственного интеллекта и подходов к их программной реализации. Проблемы реализации интеллектуального диалога с пользователем.
18	Автоматический анализ русскоязычного текста. Обзор подходов к решению проблемы.	Графематический, морфологический, синтаксический, семантический анализ текста. Модель «Смысл-Текст». Система Этап. Последовательное выполнение этапов синтаксического и семантического анализа. Другие подходы. Выделение понятий. Тезаурусы. Онтологии.
19	Методы реализации словарей и морфологический анализ текста	Лексические и грамматические словари. Вспомогательные словари «частных случаев». Применение БД для представления словарей. Организация словаря в виде файловой системы. Индексы и способы их реализации. Объектная модель лексики языка флективного типа. Обзор стратегий морфоанализа. Морфоанализатор «Скобки»
20	Система поверхностно-синтаксического анализа русского предложения	Свойства проективности и рекурсивности применительно к линейной структуре простого предложения. Общие принципы организации системы. Иерархия модулей системы и их задачи. Анализ сегментной структуры русского предложения. Объектное моделирование системы поверхностно-синтаксического анализа русского предложения. Интерпретатор лингвистических алгоритмов.

#### 4. Образовательные технологии

Применительно к данной дисциплине, прежде всего, следует отметить, что почти весь образовательный процесс осуществляется с использованием компьютера.

При преподавании дисциплины используются следующие информационные технологии.

- Преподаватель или подготовивший доклад (или краткое сообщение) студент могут представить изучаемый материал посредством демонстрации презентаций, а также сопутствующих программных образцов и методических материалов.
- Возможна коллективная работа студентов в рабочих группах по некоторым заданиям.
- Предварительный прием программных заданий «доверенными» студентами (т.е. у которых преподаватель уже принял соответствующее задание).
- Обращения к представленным в сети справочным Интернет-ресурсам по языкам (и другим средствам) программирования.
- Электронная почта и другие средства Интернет используется студентами для общения с преподавателем.
- При необходимости занятия проводятся в режиме on-line конференций в Zoom.
- Сервер учебных материалов отделения используется студентами для создания словаря по курсу и, самостоятельной работы.

Таким образом, рассматривая новые технологические элементы и/или средства программирования, преподаватель или подготовивший доклад (или краткое сообщение) студент представляют изучаемый материал посредством демонстрации

- презентаций;
- программных образцов, показывающих соответствующие элементы функциональности;
- обзоров выразительных средств изучаемых языков программирования и библиотек.

В последних двух случаях показывается исходный код и демонстрируемый эффект или протокол вычисления.

Подавляющее большинство домашних заданий являются практическими и требуют программной реализации. Студенты выполняют их индивидуально, но, в некоторых случаях небольших учебных проектов и в случае двух основных (разделы 7, 15 и 13, 16 – см. п. 3) проектов объединяются для их выполнения в небольшие рабочие группы (аналоги временных коллективов разработчиков на предприятии, производящем ПО). При этом они используют

- справочные системы применяемых программных средств (в частности, IDE – интегрированных программных сред для разработки ПО);
- представленные в сети Интернет ресурсы, содержащие справочную информацию и техническую документацию (см. п. 6);
- предоставляемые IDE средства редактирования исходных кодов и их отладки;

- электронную почту и другие средства Интернет для общения с преподавателем и совместной работы в сети в рамках рабочей группы;
- GitHub – веб-сервис для хостинга IT-проектов и их совместной разработки.

## 5. Оценка планируемых результатов обучения

### 5.1. Система оценивания

Форма контроля	Макс. количество баллов	
	За одну работу	Всего
Текущий контроль:		
- опрос	4 балла	20 баллов
- участие в дискуссии на практическом занятии	2 балла	10 баллов
- выполнение задания (программные образцы – samples)	От 1 до 8 баллов	30 баллов
Промежуточная аттестация зачет/экзамен		40 баллов
<b>Итого за семестр (дисциплину)</b> зачет/экзамен		<b>100 баллов</b>

Полученный совокупный результат конвертируется в традиционную шкалу оценок и в шкалу оценок Европейской системы переноса и накопления кредитов (European Credit Transfer System; далее – ECTS) в соответствии с таблицей:

100-балльная шкала	Традиционная шкала	Шкала ECTS
95 – 100	отлично	A
83 – 94		B
68 – 82	хорошо	C
56 – 67		D
50 – 55		E
20 – 49	неудовлетворительно	FX
0 – 19		F

### 5.2. Критерии выставления оценки по дисциплине

Баллы/ Шкала ECTS	Оценка по дисциплине	Критерии оценки результатов обучения по дисциплине
100-83/ A,B	«отлично»/ «зачтено»	<p>Выставляется обучающемуся, если он:</p> <ul style="list-style-type: none"> <li>- грамотно выполнил большинство (в том числе и все обязательные) практические задания (программные образцы – samples);</li> <li>- глубоко и прочно усвоил теоретический и практический материал, может продемонстрировать это на занятиях и в ходе промежуточной аттестации;</li> <li>- исчерпывающе и логически стройно излагает учебный материал, умеет увязывать теорию с практикой, справляется с решением задач профессиональной направленности высокого уровня сложности, правильно обосновывает принятые решения;</li> <li>- свободно ориентируется в учебной и профессиональной литературе.</li> </ul> <p>Оценка по дисциплине выставляется обучающемуся с учётом результатов текущей и промежуточной аттестации.</p> <p>Компетенции, закреплённые за дисциплиной, сформированы на уровне – «высокий».</p>
82-68/ C	«хорошо»/ «зачтено»	<p>Выставляется обучающемуся, если он:</p> <ul style="list-style-type: none"> <li>- грамотно, возможно с помощью преподавателя, выполнил достаточное количество практических заданий (программные образцы – samples), в том числе и все обязательные;</li> <li>- знает теоретический и практический материал, грамотно и по существу излагает его на занятиях и в ходе промежуточной аттестации, не допуская существенных неточностей;</li> <li>- правильно применяет теоретические положения при решении практических задач</li> </ul>

		<p>профессиональной направленности разного уровня сложности, владеет необходимыми для этого навыками и приёмами;</p> <p>- достаточно хорошо ориентируется в учебной и профессиональной литературе.</p> <p>Оценка по дисциплине выставляется обучающемуся с учётом результатов текущей и промежуточной аттестации.</p> <p>Компетенции, закреплённые за дисциплиной, сформированы на уровне – «хороший».</p>
67-50/ D,E	«удовлетворительно»/ «зачтено»	<p>Выставляется обучающемуся, если он:</p> <ul style="list-style-type: none"> <li>- выполнил (возможно с помощью преподавателя) достаточное количество практических заданий (программные образцы – samples);</li> <li>- знает на базовом уровне теоретический и практический материал, допускает отдельные ошибки при его изложении на занятиях и в ходе промежуточной аттестации;</li> <li>- испытывает определённые затруднения в применении теоретических положений при решении практических задач профессиональной направленности стандартного уровня сложности, владеет необходимыми для этого базовыми навыками и приёмами;</li> <li>- демонстрирует достаточный уровень знания учебной литературы по дисциплине.</li> </ul> <p>Оценка по дисциплине выставляется обучающемуся с учётом результатов текущей и промежуточной аттестации.</p> <p>Компетенции, закреплённые за дисциплиной, сформированы на уровне – «достаточный».</p>
49-0/ F,FX	«неудовлетворительно»/ не зачтено	<p>Выставляется обучающемуся, если он не явился по неуважительной/неизвестной причине на аттестацию или:</p> <ul style="list-style-type: none"> <li>- не выполнил (несмотря на возможную помощь преподавателя) достаточное количество практических заданий (программные образцы – samples);</li> <li>- не знает на базовом уровне теоретический и практический материал, допускает грубые ошибки при его изложении на занятиях и в ходе промежуточной аттестации.</li> <li>- испытывает серьёзные затруднения в применении теоретических положений при решении практических задач профессиональной направленности стандартного уровня сложности, не владеет необходимыми для этого навыками и приёмами;</li> <li>- демонстрирует фрагментарные знания учебной литературы по дисциплине.</li> </ul> <p>Оценка по дисциплине выставляется обучающемуся с учётом результатов текущей и промежуточной аттестации.</p> <p>Компетенции на уровне «достаточный», закреплённые за дисциплиной, не сформированы.</p>

### 5.3. Оценочные средства (материалы) для текущего контроля успеваемости, промежуточной аттестации обучающихся по дисциплине

Фонд оценочных средств (ФОС) для проведения промежуточной аттестации по дисциплине состоит из 4 разделов.

1 раздел. Перечень компетенций с указанием этапов их формирования в процессе освоения образовательной программы;

2 раздел. Описание показателей и критериев оценивания компетенций на различных этапах их формирования, описание шкал оценивания;

3 раздел. Типовые контрольные задания или иные материалы, необходимые для оценки знаний, умений, навыков и (или) опыта деятельности, характеризующих этапы формирования компетенций в процессе освоения образовательной программы;

4 раздел. Методические материалы, определяющие процедуры оценивания знаний, умений, навыков и (или) опыта деятельности, характеризующих этапы формирования компетенций.

Для каждого результата обучения по дисциплине (модулю) или практике определяются показатели и критерии оценивания усвоения компетенций на различных этапах их формирования, шкалы и процедуры оценивания.

Разделы 1-3 фонда оценочных средств для проведения промежуточной аттестации обучающихся по дисциплине приведены в Приложении 3 с тем же названием.

#### **К разделу 4 описания ФОС.**

В целом, все содержание курса можно разделить на а) теоретическую часть (разделы 1, 8, 14, 17, 18) или в большей степени теоретическую (как, например, разделы 19, 20, некоторые части разделов 2, с 4 по 6 и с 9 по 12) и б) практическую часть.

Теоретический материал осваивается студентами последовательно, в соответствии с планом (см. 2). При этом, прорабатывая каждую тему, преподаватель вначале фокусирует внимание учащихся на наиболее важных вопросах изучаемой темы, подробнее рассматривая лишь «наиболее трудные места». Затем студенты самостоятельно разбираются с изучаемым материалом, стараются выполнить заданные упражнения и т.п. На следующих занятиях, при необходимости, этот материал уже рассматривается подробно, преподаватель дает необходимые пояснения. В процессе изучения темы и по ее завершении применяются различные формы контроля (опросы, обсуждения и т.п.)

Основным видом практической деятельности студентов является программное решение задач. Их можно разделить на следующие виды:

- обзоры выразительных и функциональных средств изучаемых языков и библиотек (только применительно к разделам 2, 3, 5, 9, 10);
- программные образцы (samples) изучаемых элементов технологий и средств разработки ПО;
- традиционно «жестко», четко сформулированные задачи (как правило, в разделах 4-6 и, в редких случаях, остальных, связанных с практикой программирования, темах);
- совместное решение некоторых задач в рамках так называемых учебных проектов.

Студент выполняет обзор, следуя

- плану руководства пользователя (User Guide), которое, как правило, чаще всего является частью встроенной в IDE справочной подсистемы, или иного, аналогичного по жанру технического документа или литературного источника;
- указаниям преподавателя.

(Руководство пользователя прежде всего предназначено для освоения соответствующего средства программирования, в отличие от справочника по языку (language reference) или справочника (manual) иного программного средства.)

Обзор представляет собой программу, выводящую в виде протокола результаты применения команд к данным различных встроенных типов, использования различных управляющих конструкций и т.п.

Программные образцы отличаются от обычно рассматриваемых в курсах программирования четко сформулированных задач тем, что учащийся должен сам придумать «сюжет», сценарий выполнения программы, стремясь к тому, чтобы как можно проще, но, в то же время ясно, полно, «выпукло» показать эффект, результат демонстрируемого элемента технологии и/или средства программирования, ориентируясь на подобные образцы известных справочных систем.

Важную роль в формировании у студента навыков профессионального программиста играет такой вид деятельности, как проверка правильности выполнения достаточно сложных образцов своими коллегами. Конечно, в этом случае студент уже должен был решить соответствующую задачу и получить по ней у преподавателя зачет. Преподаватель же потом проверяет как правильность выполнения образца исполнителем, так и качество проверки проверяющим.

В рамках учебных проектов студенты или индивидуально, или объединенные на время в небольшие рабочие группы (аналоги малых коллективов разработчиков ПО) решают некоторые задачи программной реализации совместно с преподавателем. Здесь преподаватель, в зависимости от ситуации, может выступать в различных ролях: заказчика, менеджера проекта, режиссера – системного аналитика или системного архитектора. В роли менеджера проекта он контролирует выполнение отдельных частей проекта, следит за своевременностью выполнения последовательных его стадий (этапов). Студент же выступает здесь в качестве исполнителя – разработчика ПО и QA-инженера (от Quality Assurance - специалист по качеству ПО, «тестировщик»), а иногда – также в роли системного аналитика или системного архитектора.

Выполнение некоторых проектов разными коллективами обсуждается затем на занятии всей группой.

Центральное место в практической части курса занимают 2 индивидуальных учебных проекта: реляционная БД с удаленным доступом к ней в архитектуре «клиент-сервер» и многосвязная сетевая объектная модель, которые выполняются студентом от этапа постановки задачи до стадии реализации основных компонентов (со 2-го и до конца 3-го семестра обучения настоящей дисциплине).

Почти все практические задания выполняются студентами в процессе самостоятельной работы.

Самостоятельная работа студента включает

- усвоение нового материала предыдущих занятий;
- подготовку к следующему занятию (в том числе самостоятельный предварительный разбор некоторой части его материала);
- подготовку докладов и кратких сообщений;
- выполнение домашних заданий (в основном это программные образцы изучаемых элементов технологий и средств разработки ПО);
- выполнение некоторых небольших учебных проектов и двух основных (разделы 7, 15 и 13, 16 – см. п. 2) в составе рабочих групп;
- подготовку к контрольным мероприятиям и промежуточной аттестации (экзамену).

Все эти виды образовательной деятельности учащегося *обеспечиваются*

- изучением источников из списка учебной литературы (см. п. 6 в котором список разбит по тематически схожим разделам), соответствующие разделы которых задаются преподавателем и усвоение которых контролируется преподавателем в ходе обсуждений и опросов на последующих занятиях;
- использованием справочных подсистем, встроенных в применяемые программные средства (в частности, в IDE – интегрированные программные среды для разработки ПО);
- использованием представленных в сети Интернет ресурсов, содержащих справочную информацию и техническую документацию.

Кроме того, студенты, по мере необходимости, получают указания преподавателя в виде планов выполнения практических заданий или фиксации в них ошибок, «неделю», и т.п. Учащиеся также могут обращаться к преподавателю за получением консультаций. Такого рода контакты студента с преподавателем осуществляются как в аудитории, так и по электронной почте. При необходимости проводятся on-line конференции в Zoom.

#### Типовые примеры используемых оценочных средств

##### Примеры оценочных средств, используемых в теоретической части курса.

###### **Блиц-опросы**

В процессе проведения блиц-опросов (см. раздел 4 настоящего приложения) выясняется знание учащимися основных понятий рассматриваемых тем. Например, при «прохождении» тем к раздела 8 проверяется знание «базовых» понятий высшей алгебры, математической логики и рассматриваемых основных понятий реляционного формализма, формулировки некоторых важных теорем, умение решать простые задачи. Приведем начало списка вопросов, предлагаемого студентам для подготовки:

###### **Предварительные сведения**

Множество. Элемент мн-ва. Подмножество. Мн-ва равны. Универсальное мн-во. Операции над множествами. Кортеж. Прямое (декартово) произведение. Отношение. Отношения рефлексивные, антисимметричные, транзитивные.

Частичный и линейный порядки. Лексикографический порядок.

Функция (Ф). Область определения Ф. Обл. измен. Ф. Полные и частичные Ф. Ф в, Ф на. Инъекция, сюръекция, биекция. Равномощность мн-в. Конечные и счетные мн-ва. Сравнение мн-в: простые задачи.

Основные комбинаторные схемы: прямое произведение мн-в (правило произведения), перестановки (порядки), сочетания, размещения, размещения с повторениями.

###### **Отношения и схемы отношений.**

Атрибут. Универсум атрибутов. Домены атрибутов. Схемы (С) отношений. Отношения (О). Кортеж. Сужение кортежа. Суперключ О. Ключ О. Ключи: выделенные, неявные, первичные. Операции изменения О.

###### **Реляционная алгебра.**

Определение основных операторов. Расширения основных операторов. Определение реляционной алгебры. С алгебраического выражения. Алгебраическое выражение как отображение. Теорема об ограничении мн-ва операторов.

###### **Функциональные зависимости.**

Функциональные зависимости (ФЗ). ФЗ над С. О удовлетворяет ФЗ. Логическое следование для ФЗ. Вывод для ФЗ. Аксиомы (правила) вывода. Аксиомы F1–F6. Аксиомы Армстронга. Замыкание мн-ва ФЗ. Теоремы полноты в логике. Полнота F1–F6. RAP–вывод и его аксиомы. Покрытие для мн-ва ФЗ. ... <...>

Блиц-опросы к разделу 12 и разделам 18–20 устроены аналогичным образом, но в этих разделах не рассматриваются математические понятия, т.е. теорем и «строгих» определений здесь нет. Примеры вопросов к разделу 20:

- Что такое частеречная омонимия? На каком этапе анализа текста она может быть разрешена? (На этапе синтаксического анализа).

И наоборот:

- *Какой вид омонимии может быть разрешен на этапе синтаксического анализа?* (Частеречная омонимия)

Или: *Привести пример частеречной омонимии вида «существительное vs глагол».* (Например, *сталь/стали – Выплавляли сталь ... / Лошади стали ...*)

И т.п.

#### **Контрольная работа к разделу 8 «Системы вывода для функциональных зависимостей».**

В каждом из вариантов предлагается три задачи, в каждой из которых задается множество исходных функциональных зависимостей F и некоторая «целевая» функциональная зависимость. В задаче (1) применяется система вывода F1-F6, в (2) – система Армстронга, в (3) – RAP-вывод. Например: пусть  $F = \{AB \rightarrow C, B \rightarrow D, CD \rightarrow E, CE \rightarrow GH, G \rightarrow A\}$  во всех трех задачах,

(1) составить последовательность вывода из F для  $AB \rightarrow E$  используя правила вывода F1-F6,

(2) составить последовательность вывода из F для  $AB \rightarrow G$  используя правила вывода Армстронга,

(3) составить RAP-последовательность вывода из F для  $AB \rightarrow G$ .

#### **Домашнее контрольное задание к разделу 20 «Построение синтаксического графа предложения».**

Для не слишком сложного русского предложения S строятся в виде синтаксического графа все варианты результата синтаксического анализа S. (Варианты возникают из-за семантических неоднозначностей, например, для *Мать любит дочь*. будут построены два варианта, когда дочь любима матерью и когда наоборот, мать любима дочерью. А в случае *Мать любит сына* мы имеем единственный вариант разбора предложения.)

#### **Выполнение упражнений и решение задач к разделам 8 и 14**

В качестве основного учебного пособия по этим разделам используется [7] *Мейер Д.* Теория реляционных баз данных. – М.: Мир, 1987. – 608 с. (О доступе к ней см. раздел 6 наст. рабочей программы.)

Некоторые задачи и упражнения предназначены для усвоения текущего материала, как правило, их следует решить «к следующему занятию». Т.е. они являются одной из форм оперативного, текущего контроля. Приведем примеры таких задач.

- Несложные доказательства некоторых алгебраических свойств операторов реляционной алгебры (например, дистрибутивность оператора выбора относительно булевых операторов:  $\sigma A = a(r \cap s) = \sigma A = a(r) \cap \sigma A = a(s)$  – для пересечения). Для оператора пересечения это упражнение выполняется «в классе», в начале изучения темы, «не глядя в книгу» [7], а для объединения и вычитания – «дома» (в [7] отсутствуют доказательства для этих случаев).

- Примеры или контрпримеры рассматриваемых понятий. (Например: а) привести пример отношения, показывающего, что его схема находится во второй нормальной форме, но не находится в третьей нормальной форме относительно заданного множества функциональных зависимостей.

Другие задачи (из [7] – некоторая часть задач после глав с 1-ой по 7-ю и 10-й) решаются студентами в течение всего второго семестра с целью более глубокого понимания материала и повышения математической культуры учащегося.

#### **Примеры оценочных средств, используемых в практической части курса.**

##### **Контрольные работы к разделам 4, 5 и 6**

Все три контрольных выполняются программно в IDE Comman Common Lisp в компьютерном классе.

В контрольной работе «Рекурсивное программирование на “чистом” Лиспе» к разделу 4 каждому студенту предлагается индивидуальный вариант, включающий две задачи – с рекурсией только по хвосту списка (более простой вариант рекурсии) и с рекурсией и по хвосту и по голове. Время выполнения ограничивается преподавателем в зависимости от сложности задач.

Пример. По заданному списку S построить другой, показывающий число элементов каждого его подсписка на каждом уровне вложения:

$(a\ 23\ ("fff"\ (b\ c\ d\ 7))\ e\ (a\ d\ z)\ ((k)\ q\ w\ (1\ m\ 77\ dd\ b))) \rightarrow (6\ (2\ (4))\ (3)\ (4\ (1)\ (5)))$

$((((a)))) \rightarrow (1\ (1\ (1\ (1))))$

$((((a)))\ b\ (d\ d\ d\ d\ (c\ c\ c\ (b\ (a)\ b)))) \rightarrow (3\ (1\ (1\ (1)))\ (4\ (3\ (2\ (1))))$

$(()) \rightarrow (1\ (0))$

(рекурсия по хвосту и по голове списка)

В контрольной работе к разделу 5 дается несколько простых задач на применение функционалов (map-функций и некоторых других), функций интерпретаторов (eval, apply и funcall), lambda-выражений, блоковых выражений, различных способов задания формальных параметров функций.

В контрольной работе к разделу проверяется успешность освоения студентами CLOS (Common Lisp Object System). Каждому учащемуся индивидуально предлагается всего одна задача, в которой, исходя из условия, надо определить некоторый класс и/или обобщенную функцию, класс и суперкласс с наследованием и т.п.

Пример.

##### **Предварительные сведения.**

Кватернионы — система гиперкомплексных чисел, образующая векторное пространство размерностью четыре над полем вещественных чисел.

Кватернионы можно определить как формальную сумму  $a+bi+cj+dk$ , где  $a, b, c, d$  — вещественные числа, а  $i, j, k$  — мнимые единицы со следующим свойством:  $i^2=j^2=k^2=ijk=-1$ . Таким образом, таблица умножения базисных кватернионов  $1, i, j, k$  — выглядит так:

×	1	i	j	k
1	1	i	j	k
i	i-1	k	-j	
j	j-k	-1	i	
k	k	j	-i	-1

Например,  $ij=k$ , а  $ji=-k$ .

Как вектор и скаляр, кватернион представляет собой пару  $(a, \mathbf{u})$ , где  $\mathbf{u}$  — вектор трёхмерного пространства, а  $a$  — скаляр, то есть вещественное число.

Операция сложения определена следующим образом:  $(a, \mathbf{u}) + (b, \mathbf{v}) = (a+b, \mathbf{u}+\mathbf{v})$

Произведение определяется следующим образом:  $(a, \mathbf{u}) (b, \mathbf{v}) = (ab - \mathbf{u} \cdot \mathbf{v}, a\mathbf{v} + b\mathbf{u} + \mathbf{u} \times \mathbf{v})$ , где  $\cdot$  обозначает скалярное произведение, а  $\times$  — векторное произведение.

**Задача.**

Определить класс для представления кватернионов, метод to-string его «печати в строку» (для удобного вывода в протокол) в виде  $a+bi+cj+dk$ , «распространить» операции сложения и умножения на кватернионы.

Процедуры оценивания знаний, умений, навыков и (или) опыта деятельности, характеризующих этапы формирования компетенций описывается процедура оценивания этапов формирования компетенции.

**Теоретическая часть курса.**

***Блиц-опросы***

Проводятся индивидуально, от обычных опросов отличаются тем, что на ответ студенту дается фиксированное время, в зависимости от сложности вопроса/задачи, например, не более минуты (в самых простых случаях).

Попытка сдачи студентом продолжается до первой ошибки или незнания ответа. Блиц-опрос считается сданным, если студент без ошибок ответит на определенное число вопросов (10-12). В противном случае студент «занимает очередь» для следующей попытки, или «уходит» готовиться к следующему сроку сдачи.

***Контрольная работа к разделу 8 «Системы вывода для функциональных зависимостей».***

Проводится первый раз на занятии, отводится фиксированное время – не более получаса. В случае ошибок – неправильно выполненные задачи переписываются в дополнительное время (даются, конечно другие варианты). При оценке учитывается количество попыток. Оценка снижается, если студенту так и не удалось уложиться в положенное время.

***Домашнее контрольное задание к разделу 20 «Построение синтаксического графа предложения».***

Выполняется студентом индивидуально, каждому предлагается свой пример русского предложения. Оценивается полнота анализа (все ли варианты разбора учтены), правильность проведения связей «хозяин-слуга» между словами предложения.

***Выполнение упражнений и решение задач к разделам 8 и 14***

Выполнение всех задач и упражнений контролируется и оценивается преподавателем.

Часть задач «на доказательство» принимаются у студентов в форме устного собеседования, при этом в случае ошибочных суждений учащегося ему явно на ошибки не указывается, а развивая эти суждения и рассматривая примеры преподаватель старается подвести собеседника к ситуации, когда он сам осознает ошибку.

**Практическая часть курса.**

В рамках данного курса преподавателем контролируется и оценивается выполнение всех видов практических заданий, см. начало настоящего раздела.

## 6. Учебно-методическое и информационное обеспечение дисциплины

### 6.1. Список источников и литературы

*Замечание:* многие источники полностью или в виде актуальных фрагментов имеются в электронной библиотеке (ЭБ) Учебно-методического кабинета (УМК) Отделения интеллектуальных систем (ОИС), папка [d:\\_ois\\_lib] (либо [c:\\_ois\\_lib] в случае единственного логического диска) на компьютерах деканата.

#### а) Основная литература

1. Информационные технологии и системы: Учебное пособие / Е.Л. Федотова. - М.: ИД ФОРУМ: НИЦ Инфра-М, 2013. - 352 с.: ил.; 60x90 1/16. - (Высшее образование).

<http://znanium.com/catalog.php?bookinfo=374014>

2. Проектирование информационных систем: Учебное пособие / Н.З. Емельянова, Т.Л. Партыка, И.И. Попов. - М.: Форум: НИЦ ИНФРА-М, 2014. - 432 с.: ил.; 60x90 1/16. - (Профессиональное образование)

<http://znanium.com/catalog.php?bookinfo=419815>

#### К разделам 2, 3, 7, 12, 15

3. *Монахов В.В.* Язык программирования Java и среда NetBeans. – СПб: БХВ-Петербург, 2009. – 718 с.

(ЭБ УМК ОИС: [d:\\_ois\\_lib\3Java\monahov\_v\_v\_yazyk\_programirovaniya\_java\_i\_sreda\_netbeans.doc], имеется также на сайте znanium.com)

4. *Нютон П., Шилдт Г.* Java2. Полное руководство. – СПб.: БХВ-Петербург, 2008. – 1055 с. (ЭБ УМК ОИС: [d:\\_ois\\_lib\3Java\Java\_2\_Naughton-Schildt.pdf])

5. Раздел портала Oracle, содержащий техническую документацию (онлайн и в виде свободно распространяемых электронных книг) о платформе Java Standard Edition (Java SE): <http://docs.oracle.com/javase/>

#### К разделам 4, 5, 6, 7, 15

6. *Грэм, П.* ANSI Common Lisp – Пер. с англ. – СПб.: Символ-Плюс, 2012. – 448 с. (ЭБ УМК ОИС: [d:\\_ois\\_lib\3fp\ANSI\_Common\_Lisp\_(Rus).pdf])

#### К разделам 8, 14

7. *Мейер Д.* Теория реляционных баз данных. – М.: Мир, 1987. – 608 с.

(ЭБ УМК ОИС: [d:\\_ois\\_lib\3db\Maier(Rus).djvu])

*Замечание.* Книга на русском языке больше не переиздавалась, однако она является единственным источником, отражающем основные результаты теории РБД с необходимой для студентов ОИС степенью полноты и формализации.

#### К разделам 9, 13, 16

8. *Коселев В.Е.* Access 2007 – М.: ООО «Бином-Пресс», 2008.

(ЭБ УМК ОИС: [d:\\_ois\\_lib\3db\Koshelev\_Access2007.djvu])

9. *Мартинин С.А.* Проектирование и реализация баз данных в СУБД MySQL с использованием MySQL Workbench: Учебное пособие / С.А. Мартинин и др. - М.: ИД ФОРУМ: НИЦ Инфра-М, 2012. - 160 с.

<http://znanium.com/catalog.php?bookinfo=556449>

10. Официальный учебный курс Microsoft. Microsoft Office Access 2003. – М.: Эком, Бином. Лаборатория знаний, 2006 (издание 2004 г - ЭБ УМК ОИС: [d:\\_ois\\_lib\3db\MicrosoftAccess2003\_Shag-za-shagom.djvu])

#### К разделам 10, 11, 12, 13, 16

11. *Айзекс С.* Dynamic HTML. – СПб.: ВHV-Санкт-Петербург, 1998.

(ЭБ УМК ОИС: \_ с примерами в папке [d:\\_ois\\_lib\3ip\DHTML-book(byS\_Isaacs)])

12. *Бибо, Б., Кац, И.* jQuery. Подробное руководство по продвинутому JavaScript – Пер. с англ. – СПб.: Символ-Плюс, 2011. – 448 с.

(ЭБ УМК ОИС: [d:\\_ois\\_lib\3ip\jQuery\_PodrobRukovodPoProdvinutomuJavaScript\_2ed.pdf])

13. *Оберг Р. Дж.* Технология COM+. Основы и программирование. – М.: Издательский дом «Вильямс», 2000. (ЭБ УМК ОИС: [d:\\_ois\\_lib\3wp\complus.djvu])

14. MSDN Microsoft Developer Network – техническая документация фирмы Microsoft для разработчиков ПО: <http://msdn.microsoft.com>

15. XML (Extensible Markup Language) – техническая документация Microsoft XML Parser SDK

16. XML Schema., – рабочий документ (рекомендация) консорциума W3C, опубликован (регулярно обновляется) на сайте консорциума

(<http://www.w3.org/>): <http://www.w3.org/TR/xmlschema/>.

К разделам 18, 20

17. Кобзарева Т.Ю. В поисках синтаксической структуры: автоматический анализ русского предложения с опорой на сегментацию. – М.: РГГУ, 2015.  
(ЭБ УМК ОИС: [d:\\_ois\\_lib\\_PiLOIS\KobzarevaTYu\_V\_poiskah\_syntaksicheskoi\_strukturny.doc])

2.1. **Дополнительная литература**

К разделам 2, 3, 7, 15

18. Даконта М., Саганич А. XML и JAVA 2 (серия "Библиотека программиста"). – СПб.: Питер, 2001.  
19. Флэнаган Д. Java в примерах. – СПб.: Символ-Плюс, 2003.  
20. Хорстманн К.С., Корнелл Г. Java 2. Библиотека профессионала. Т. 1,2. Тонкости программирования. 8-е издание. – М.: ООО "И. Д. Вильямс", 2009.  
21. Шилдт Г. Полный справочник по Java. – М.: Издательский дом «Вильямс», 2007  
22. Шилдт Г. Java 8. Полное руководство. – М.: Издательский дом «Вильямс», 2016

К разделам 4, 5, 6, 7, 15

23. Филд А., Харрисон П. Функциональное программирование. – М.: Мир, 1993.  
24. Steele, Guy L., Common Lisp the Language, 2nd edition - Digital Press, 1990 – в электронном виде: <http://www-prod-gif.supelec.fr/docs/cltl/cltl2.html>

К разделам 8, 14

25. Дейт К. Введение в системы баз данных. – М.; СПб.; Киев: Издательский дом «Вильямс», 2008. (имеется в библиотеке РГГУ)  
26. Ульман Дж. Основы систем баз данных. – М.: Финансы и статистика, 1983  
(ЭБ УМК ОИС: [d:\\_ois\\_lib\3db\Ullman.djvu])

К разделам 9, 13, 16

27. Гарнаев А.Ю. Самоучитель VBA. – 2-е изд., перераб. и доп. – СПб.: БХВ-Петербург, 2007.  
28. Гурвиц Г.А. Microsoft Access 2007. Разработка приложений на реальном примере. – СПб.: БХВ-Петербург, 2007.  
29. Дженнингс Р. Использование Microsoft Office Access 2003. Специальное издание. – М.: Издательский дом «Вильямс», 2006.  
30. Кузин А.В., Демин В.М. Разработка баз данных в системе Microsoft Access: учебник. – М.: Форум, 2009

К разделам 10, 11, 12, 13, 16

31. Горин М. А. 2 в 1: Как создать Web-сайт. – М.: Триумф, 2009  
32. Дунаев В. В. HTML, скрипты и стили. – СПб.: BHV-СПб, 2008  
33. Козн И. Полный справочник по HTML, CSS и JavaScript. Серия: Справочник профессионала. – М.: Эком Паблишера, 2007  
34. Крейн Д., Пакарелло, Э., Джеймс, Д. Ажак в действии. – М.: Издательский дом «Вильямс», 2006.

К разделам 18, 19, 20

35. Зализняк А.А. Грамматический словарь русского языка. – М.: Рус.яз., 1980.  
36. Леонтьева Н.Н. Автоматическое понимание текста: системы, модели, ресурсы. – М., 2006.  
37. Мельчук И.А. Опыт теории лингвистических моделей «Смысл–Текст». Семантика, синтаксис. – М.: Яз. рус. культуры, 1999.  
38. Рубашкин В.Ш. Онтологическая семантика. – М.: Физматлит, 2012.

6.2 **Перечень ресурсов информационно-телекоммуникационной сети**

**"Интернет", необходимых для освоения дисциплины**

- <http://api.jquery.com/> – Справочная документация по библиотеке jQuery  
<http://www.cormanlisp.com/> – официальный сайт Corman Technologies Inc. (скачивание IDE Corman Common Lisp и технической документации к ней)  
<http://dev.mysql.com/doc/> – Справочная документация по MySQL  
<http://docs.oracle.com/javase/> – Раздел портала Oracle, содержащий техническую документацию (онлайн и в виде свободно распространяемых электронных книг) о платформе Java Standard Edition (Java SE).

<https://github.com/> – GitHub – веб-сервис для хостинга IT-проектов и их совместной разработки

<http://msdn.microsoft.com> – MSDN Microsoft Developer Network – техническая документация фирмы Microsoft для разработчиков ПО:

<https://netbeans.org/downloads/> – Скачивание IDE NetBeans и JDK

<http://severe-programmer.com/manual/ustanovka-i-nastrojka-apache-mysql-na-os-x-10-9-mavericks/> – Рекомендации по установке и настройке Apache и MySQL

<https://tproger.ru/translations/java-json-library-comparison/> – Сравнение четырёх популярных библиотек Java для работы с JSON

[http://vana.nvte.ee:89/evara/Программа\\_MySQL\\_Workbench\\_ru.pdf](http://vana.nvte.ee:89/evara/Программа_MySQL_Workbench_ru.pdf) – Руководство по разработке БД при помощи MySQL Workbench

<http://www.oracle.com/technetwork/java/index-jsp-142942.html> – Справочная документация по Oracle Java Enterprise Edition

<http://www.oracle.com/technetwork/java/javae/downloads/index.html> – Руководство по установке и настройке Apache Tomcat, регистрации в NetBeans

<http://www.ruscorpора.ru/> – Сайт Национального корпуса русского языка

#### Перечень БД и ИСС

№п/п	Наименование
1	Международные реферативные наукометрические БД, доступные в рамках национальной подписки в 2025 г. Web of Science Scopus
2	Профессиональные полнотекстовые БД, доступные в рамках национальной подписки в 2025 г. Журналы Cambridge University Press ProQuest Dissertation & Theses Global SAGE Journals Журналы Taylor and Francis
3	Профессиональные полнотекстовые БД JSTOR Издания по общественным и гуманитарным наукам Электронная библиотека Grebennikon.ru
4	Компьютерные справочные правовые системы Консультант Плюс, Гарант

## 7. Материально-техническое обеспечение дисциплины

Занятия проводятся в компьютерных классах ауд. 307 и 706, расположенных по адресу 125993, Москва, Миусская пл., д. 6, стр.2.,

Этот компьютерный класс оснащен

- достаточным количеством объединенных в локальную сеть рабочих станций,
- медиапроектором и экраном,
- меловой доской.

В классе имеются возможности

- подключения ноутбука к медиапроектору,
- одновременного доступа в Интернет для преподавателя и студентов.

### Состав программного обеспечения (ПО)

№п/п	Наименование ПО	Производитель	Способ распространения (лицензионное или свободно распространяемое)
1	Adobe Master Collection CS4	Adobe	лицензионное
2	Kaspersky Endpoint Security	Kaspersky Lab	лицензионное
3	ОС MS Windows 7 Pro	Microsoft	лицензионное
4	ОС MS Windows 10 Pro	Microsoft	лицензионное
5	Internet Explorer 10	Microsoft	устанавливался в Windows 7
6	Internet Explorer 11	Microsoft	устанавливался в Windows 7-10
7	MS Edge	Microsoft	устанавливается в Windows 10
8	Microsoft Share Point 2010	Microsoft	лицензионное
9	MS Visual Studio 2017	Microsoft	лицензионное
10	MS Visual Studio 2019	Microsoft	лицензионное
11	MS Visual Studio 2022		
12	Microsoft Office 2003	Microsoft	лицензионное
13	Microsoft Office 2013	Microsoft	лицензионное
14	Microsoft Office 2016	Microsoft	лицензионное
15	IIS	Microsoft	встроенный в Windows <a href="http://cevvpr">http://cevvpr</a>
16	IDE NetBeans 8.2 + JDK 7 (и более поздних версий)	Oracle, Apache в наст. время	свободно распространяется
17	IDE Corman Common Lisp v. 3.1	<a href="http://www.corman.com">Corman Technologies Inc.</a>	лицензионное
18	Zoom	Zoom	лицензионное

## 8. Обеспечение образовательного процесса для лиц с ограниченными возможностями здоровья и инвалидов

В ходе реализации дисциплины используются следующие дополнительные методы обучения, текущего контроля успеваемости и промежуточной аттестации обучающихся в зависимости от их индивидуальных особенностей:

- для слепых и слабовидящих:
  - лекции оформляются в виде электронного документа, доступного с помощью компьютера со специализированным программным обеспечением;
  - письменные задания выполняются на компьютере со специализированным программным обеспечением, или могут быть заменены устным ответом;
  - обеспечивается индивидуальное равномерное освещение не менее 300 люкс;
  - для выполнения задания при необходимости предоставляется увеличивающее устройство; возможно также использование собственных увеличивающих устройств;
  - письменные задания оформляются увеличенным шрифтом;
  - экзамен и зачёт проводятся в устной форме или выполняются в письменной форме на компьютере.
- для глухих и слабослышащих:
  - лекции оформляются в виде электронного документа, либо предоставляется звукоусиливающая аппаратура индивидуального пользования;
  - письменные задания выполняются на компьютере в письменной форме;
  - экзамен и зачёт проводятся в письменной форме на компьютере; возможно проведение в форме тестирования.
- для лиц с нарушениями опорно-двигательного аппарата:
  - лекции оформляются в виде электронного документа, доступного с помощью компьютера со специализированным программным обеспечением;

- письменные задания выполняются на компьютере со специализированным программным обеспечением;
- экзамен и зачёт проводятся в устной форме или выполняются в письменной форме на компьютере.

При необходимости предусматривается увеличение времени для подготовки ответа.

Процедура проведения промежуточной аттестации для обучающихся устанавливается с учётом их индивидуальных психофизических особенностей.

Промежуточная аттестация может проводиться в несколько этапов.

При проведении процедуры оценивания результатов обучения предусматривается использование технических средств, необходимых в связи с индивидуальными особенностями обучающихся. Эти средства могут быть предоставлены университетом, или могут использоваться собственные технические средства.

Проведение процедуры оценивания результатов обучения допускается с использованием дистанционных образовательных технологий.

Обеспечивается доступ к информационным и библиографическим ресурсам в сети Интернет для каждого обучающегося в формах, адаптированных к ограничениям их здоровья и восприятия информации:

- для слепых и слабовидящих:
  - в печатной форме увеличенным шрифтом;
  - в форме электронного документа;
  - в форме аудиофайла.
- для глухих и слабослышащих:
  - в печатной форме;
  - в форме электронного документа.
- для обучающихся с нарушениями опорно-двигательного аппарата:
  - в печатной форме;
  - в форме электронного документа;
  - в форме аудиофайла.

Учебные аудитории для всех видов контактной и самостоятельной работы, научная библиотека и иные помещения для обучения оснащены специальным оборудованием и учебными местами с техническими средствами обучения:

- для слепых и слабовидящих:
  - устройством для сканирования и чтения с камерой SARA CE;
  - дисплеем Брайля PAC Mate 20;
  - принтером Брайля EmBraille ViewPlus;
- для глухих и слабослышащих:
  - автоматизированным рабочим местом для людей с нарушением слуха и слабослышащих;
  - акустический усилитель и колонки;
- для обучающихся с нарушениями опорно-двигательного аппарата:
  - передвижными, регулируемые эргономическими партами СИ-1;
  - компьютерной техникой со специальным программным обеспечением.

## 9. Методические материалы

Лабораторные занятия по «блокам» схожих тем однотипны, они проводятся по единой схеме. На занятиях объясняются основные теоретические положения изучаемых тем, формулируются постановки для части задач, предлагаются методы, возможные сценарии и технологические элементы для их решения.

Часть задач объясняется непосредственно на практических занятиях.

Углубленно текущий материал изучается студентами самостоятельно.

Почти все практические задания также выполняются студентами в процессе самостоятельной работы.

На практических занятиях при необходимости проводится разбор текущего материала и контроль в форме опросов и дискуссий его усвоения.

Наконец, на практических занятиях осуществляется контроль за выполнением практических заданий, при необходимости проводятся индивидуальные или групповые консультации.

(Конкретнее эти положения изложены в п. 5.3.)

### 9.1. Пример планов лабораторных занятий

#### 1. Изучение средств программирования (темы 1-6, 9-12).

Тема6 (6ч.) CLOS (Common Lisp Object System)

*Цель занятий:* изучение встроенной в язык подсистемы ООП и программирование классов.

*Форма проведения* – опрос, дискуссии, решение задач.

*Вопросы для обсуждения:*

Определения классов. Объекты.

Множественное наследование.

Методы и обобщенные функции.

Системные функции для работы с объектами и определениями классов.

*Примеры контрольных вопросов:*

1. Парадигма объектно-ориентированного программирования, в том числе на примере CLOS.
2. Сравнение CLOS и ООП в C++, C#, Java языке сценариев JavaScript.
3. Применение CLOS для программирования динамических структур данных.

*Задания:*

см. в п.5.3 настоящего документа.

*Список источников и литературы:*

1. *Грэм, П.* ANSI Common Lisp – Пер. с англ. – СПб.: Символ-Плюс, 2012. – 448 с. (ЭБ УМК ОИС: [d:\\_ois\\_lib\3fp\ANSI\_Common\_Lisp\_(Rus).pdf])
2. *Steele, Guy L.*, Common Lisp the Language, 2nd edition - Digital Press, 1990 – в электронном виде: <http://www-prod-gif.supelec.fr/docs/clt/clt12.html>

## АННОТАЦИЯ ДИСЦИПЛИНЫ

Дисциплина «Программное и лингвистическое обеспечение интеллектуальных систем» реализуется на Отделении интеллектуальных систем в гуманитарной сфере Учебнонаучным центром программного и лингвистического обеспечения интеллектуальных систем.

**Цели дисциплины.** Главная цель курса – дать студенту целостное представление о концепциях, технологиях и средствах современного программирования, а также методов их применения к разработке интеллектуальных систем, включая такой специфический их компонент, как подсистема интеллектуального диалога с интеллектуальной системой. Другими целями курса можно считать обучение слушателей работе с научной и технической литературой, технической документацией в области программирования, способствовать формированию у студентов навыков работы самостоятельного программиста.

**Задачи дисциплины:** освоение средств объектно-ориентированного и функционального программирования, средств разработки реляционных баз данных и удаленного доступа к ним, методов автоматического анализа текста на естественном (русском) языке и программной реализации «больших» словарей.

Дисциплина направлена на формирование следующих компетенций:

- УК-2 способностью управлять проектом на всех этапах его жизненного цикла.
- ОПК-1 способностью применять в профессиональной деятельности методы математического анализа, логики и моделирования, теоретического и экспериментального исследования в информатике, лингвистике и гуманитарных науках
- ОПК-3 способностью использовать фундаментальные знания в области гуманитарных, социальных и лингвистических наук, а также в сфере техники и технологии информатики для совершенствования профессиональной деятельности
- ОПК-5 способностью применять новые информационные технологии в гуманитарных областях знаний с использованием средств интеллектуального анализа данных, машинного обучения, компьютерной лингвистики и представления знаний

В результате освоения дисциплины обучающийся должен:

Знать

- современные подходы и технологии, применяемые для разработки программных приложений;
- современные подходы и технологии, применяемые в инструментальных средах (IDE) для совместной разработки программных приложений;
- базовые типы данных и возможности для определения новых типов данных в изучаемых языках; стандартные структуры данных (в том числе списки, стеки, очереди, деки, деревья, графы) и варианты их представления в программах, методы построения сложных многоссылочных сетевых структур;
- основы теории реляционных баз данных (БД), в частности, принципы построения и проектирования реляционных БД;
- различные подходы к реализации автоматического морфологического, синтаксического и семантического анализа текста на естественном (русском) языке.

Уметь

- пользоваться современными образовательными и информационными технологиями для получения знаний;
- использовать спецификации проектов программных приложений;
- методы доступа к информационным ресурсам; основные принципы разработки прикладных программных систем;
- перерабатывать большие объемы информации и вычленять главное (анализ информации);
- приобретать с большой степенью самостоятельности новые знания, используя современные образовательные и информационные технологии;
- решать на основе вышеперечисленных знаний задачи по программированию компонентов интеллектуальных систем в упрощенном виде;
- эксплуатировать БД реляционного типа как пользователь и как системный администратор; комплексно разрабатывать базы данных, реализующих, в частности, в архитектуре клиент/сервер.

Владеть навыками

- анализа документации к программным системам;
- самостоятельного специалиста в области программной реализации интеллектуальных систем;
- совместной работы в рабочих группах программистов.

По дисциплине предусмотрены промежуточные аттестации в форме зачета с оценкой и экзамена.

Общая трудоемкость освоения дисциплины составляет 9 зачетных единиц.

Приложение 2

**Фонд оценочных средств для проведения промежуточной аттестации обучающихся по дисциплине**

**Глоссарий ФОС**

1. Компетенция - способность применять знания, умения и навыки в конкретном виде профессиональной деятельности, к которой готовятся выпускники, освоившие образовательную программу;
2. Критерии оценивания компетенции – признаки, на основании которых происходит оценка по показателям компетенций;
3. Показатели – это уровни освоения компетенции, выражаемые в обобщенной характеристике процесса и результата освоения компетенции;
4. Фонд оценочных средств – комплекс оценочных средств, контрольно-измерительных и методических материалов, предназначенных для определения уровня освоения компетенций обучающихся и качества результатов обучения в ходе освоения образовательной программы высшего образования;
5. Шкала оценивания - система оценивания качества освоения образовательных программ через систему балльных, цифровых или буквенных обозначений;
6. Этапы формирования компетенции – определенная стадия процесса формирования компетенции (знать, уметь, владеть).

Приложение выполнено в соответствии с пунктом 21 приказа МОН от 19.12.2013 №1367

**ФОС для проведения промежуточной аттестации по дисциплине состоит из 4 разделов:**

1 раздел. Перечень компетенций с указанием этапов их формирования в процессе освоения образовательной программы;

2 раздел. Описание показателей и критериев оценивания компетенций на различных этапах их формирования, описание шкал оценивания;

3 раздел. Типовые контрольные задания или иные материалы, необходимые для оценки знаний, умений, навыков и (или) опыта деятельности, характеризующих этапы формирования компетенций в процессе освоения образовательной программы;

4 раздел. Методические материалы, определяющие процедуры оценивания знаний, умений, навыков и (или) опыта деятельности, характеризующих этапы формирования компетенций.

Для каждого результата обучения по дисциплине (модулю) или практике определяются показатели и критерии оценивания сформированности компетенций на различных этапах их формирования, шкалы и процедуры оценивания.

В 1 разделе ФОС указываются этапы формирования компетенций, формируемых данной дисциплиной.

Таблица П1.1

Перечень компетенций, формируемых дисциплиной с указанием этапов их формирования

<i>№ п/п</i>	<i>Код компетенции</i>	<i>Описание этапов формирования компетенции</i>
1.	УК-2	<b>Знать:</b> рассмотренные в рамках данного курса - языки программирования, другие средства программирования; - структуры данных, алгоритмы и возможные способы их реализации.
		<b>Уметь:</b> применять перечисленные выше знания к решению практических задач, сформулированных в рамках изучения данного курса
		<b>Владеть:</b> навыками самостоятельного выбора и применения рассмотренных методов и средств к решению задач
2	ОПК-3	<b>Знать:</b> рассмотренные в рамках данного курса - современные подходы и технологии, применяемые для разработки программного обеспечения; - методы разработки программных приложений, использующих сложные структуры данных
		<b>Уметь:</b> - разрабатывать ПО в архитектуре клиент/сервер - программно реализовывать приложения на основе сложных структур данных - решать на основе вышеперечисленных знаний задачи по программированию компонентов интеллектуальных систем в упрощенном виде
		<b>Владеть:</b> навыками самостоятельного выбора и применения вышеперечисленных знаний к разработке программных приложений
3	ОПК-1	<b>Знать:</b> принципы объектно-ориентированного программирования (ООП), изучаемые в рамках данного курса средства ООП и другие средства (программные интерфейсы компонентов приложения и методы их спецификации, пакеты, пространства имен и т.п.) обеспечивающие отдельную реализацию частей приложения
		<b>Уметь:</b> грамотно декомпозировать задачи в процессе проектирования программных приложений, применяя перечисленные выше знания; разбираться в кодах, структуре и логике программ, разрабатываемых своими коллегами
		<b>Владеть:</b> навыками совместной работы (по разработке ПО) в коллективе

**2 раздел ФОС.**

Заполняется таблица № 2, содержащая информацию о показателях и критериях оценивания компетенции на различных этапах ее формирования, описание шкал оценивания.

Таблица П1.2

Показатели, критерии и шкалы оценивания компетенций на этапах их формирования

В таблице ниже традиционные оценки: 5 – отлично, 4 – хорошо, 3 - удовлетворительно

Код компетенции	Описание этапов формирования компетенции	Показатели	Критерии оценивания (признаки, на основании которых происходит оценка по показателям)	Шкалы оценивания		
				Традиционный	Баллы	ECTS
УК-2	<p><b>Знать:</b> рассмотренные в рамках данного курса</p> <ul style="list-style-type: none"> <li>- языки программирования, другие средства программирования;</li> <li>- структуры данных, алгоритмы и возможные способы их реализации.</li> </ul>	Базовый	Теоретическое содержание дисциплины освоено частично, некоторые учебные задания не выполнены.	3	55-67	D, E
		Продвинутый	Теоретическое содержание дисциплины освоено частично, без пробелов, все учебные задания выполнены, качество выполнения ни одного из них не оценено минимальным числом баллов.	4	68-82	C
		Высокий	Теоретическое содержание дисциплины освоено полностью, без пробелов.	5	83-100	A, B

	<b>Уметь:</b> применять перечисленные выше знания к решению практических задач, сформулированных в рамках изучения данного курса	Базовый	Теоретическое содержание дисциплины освоено частично, но пробелы не носят существенного характера, необходимые практические навыки работы с освоенным материалом в основном сформированы.	3	55-67	D, E
		Продвинутый	Теоретическое содержание дисциплины освоено полностью, без пробелов, некоторые практические навыки работы с освоенным материалом сформированы недостаточно.	4	68-82	C
		Высокий	Теоретическое содержание дисциплины освоено полностью, без пробелов, необходимые практические навыки работы с освоенным материалом сформированы, все предусмотренные рабочей программой дисциплины учебные задания выполнены.	5	83-100	A, B
	<b>Владеть:</b> навыками самостоятельного выбора и применения рассмотренных методов и средств к решению задач	Базовый	Теоретическое содержание дисциплины освоено частично, но пробелы не носят существенного характера, необходимые практические навыки работы с освоенным материалом в основном сформированы, большинство предусмотренных рабочей программой дисциплины учебных заданий выполнено, некоторые из выполненных заданий, возможно, содержат ошибки.	3	55-67	D, E
		Продвинутый	Теоретическое содержание дисциплины освоено полностью, без пробелов, некоторые практические навыки работы с освоенным материалом сформированы недостаточно, все предусмотренные рабочей программой дисциплины учебные задания выполнены, качество выполнения ни одного из них не оценено минимальным числом баллов, некоторые виды заданий выполнены с ошибками.	4	68-82	C
		Высокий	Теоретическое содержание дисциплины освоено полностью, без пробелов, необходимые практические навыки работы с освоенным материалом сформированы, все предусмотренные рабочей программой дисциплины учебные задания выполнены, качество их выполнения оценено числом баллов, близким к максимальному значению.	5	83-100	A, B
ОПК-5	<b>Знать:</b> рассмотренные в рамках данного курса - современные подходы и технологии, применяемые для разработки программного обеспечения; - методы разработки программных приложений, использующих сложные структуры данных	Базовый	Рассмотренные в рамках данного курса подходы, методы, технологии и средства программирования освоены частично, некоторые практические задания не выполнены.	3	55-67	D, E

		Продвинутый	Рассмотренные в рамках данного курса подходы, методы, технологии и средства программирования освоены частично, без пробелов, все практические задания выполнены, качество выполнения ни одного из них не оценено минимальным числом баллов.	4	68-82	C
		Высокий	Рассмотренные в рамках данного курса подходы, методы, технологии и средства программирования освоены полностью, без пробелов.	5	83-100	A, B
		Базовый	Рассмотренные в рамках данного курса подходы, методы, технологии и средства программирования освоены частично, но пробелы не носят существенного характера, необходимые практические навыки применения этих знаний в основном сформированы.	3	55-67	D, E
		Продвинутый	Рассмотренные в рамках данного курса подходы, методы, технологии и средства программирования освоены полностью, без пробелов, некоторые практические навыки применения этих знаний сформированы недостаточно.	4	68-82	C
		Высокий	Рассмотренные в рамках данного курса подходы, методы, технологии и средства программирования освоены полностью, без пробелов, необходимые практические навыки применения этих знаний сформированы, все предусмотренные рабочей программой дисциплины практические задания выполнены.	5	83-100	A, B
	<b>Владеть:</b> навыками самостоятельного выбора и применения вышеперечисленных знаний к разработке программных приложений	Базовый	Рассмотренные в рамках данного курса подходы, методы, технологии и средства программирования освоены частично, но пробелы не носят существенного характера, необходимые практические навыки применения этих знаний в основном сформированы, большинство предусмотренных рабочей программой дисциплины практических заданий выполнено, некоторые из выполненных заданий, возможно, содержат ошибки.	3	55-67	D, E
		Продвинутый	Рассмотренные в рамках данного курса подходы, методы, технологии и средства программирования освоены полностью, без пробелов, некоторые практические навыки применения этих знаний сформированы недостаточно, все предусмотренные рабочей программой дисциплины практические задания выполнены, качество выполнения ни одного из них не оценено минимальным числом баллов, некоторые	4	68-82	C

			виды заданий выполнены с ошибками.			
		Высокий	Рассмотренные в рамках данного курса подходы, методы, технологии и средства программирования освоены полностью, без пробелов, необходимые практические навыки применения этих знаний сформированы, все предусмотренные рабочей программой дисциплины практические задания выполнены, качество их выполнения оценено числом баллов, близким к максимальному значению.	5	83-100	A,B
ОПК-3	<p><b>Знать:</b> принципы объектно-ориентированного программирования (ООП), изучаемые в рамках данного курса средства ООП и другие средства (программные интерфейсы компонентов приложения и методы их спецификации, пакеты, пространства имен и т.п.) обеспечивающие раздельную реализацию частей приложения</p>	Базовый	Рассмотренные в рамках данного курса средства, облегчающие декомпозицию задачи разработки программного продукта, освоены частично, некоторые учебные проекты по совместной программной реализации не выполнены.	3	55-67	D, E
		Продвинутый	Рассмотренные в рамках данного курса средства, облегчающие декомпозицию задачи разработки программного продукта, освоены частично, без пробелов, все учебные проекты по совместной программной реализации выполнены, качество выполнения ни одного из них не оценено минимальным числом баллов.	4	68-82	C
		Высокий	Рассмотренные в рамках данного курса средства, облегчающие декомпозицию задачи разработки программного продукта, освоены полностью, без пробелов.	5	83-100	A, B
	<p><b>Уметь:</b> грамотно декомпозировать задачи в процессе проектирования программных приложений, применяя перечисленные выше знания; разбираться в кодах, структуре и логике программ, разрабатываемых своими коллегами</p>	Базовый	Рассмотренные в рамках данного курса средства, облегчающие декомпозицию задачи разработки программного продукта, освоены частично, но пробелы не носят существенного характера, необходимые практические навыки применения этих знаний в основном сформированы.	3	55-67	D, E
		Продвинутый	Рассмотренные в рамках данного курса средства, облегчающие декомпозицию задачи разработки программного продукта, освоены полностью, без пробелов, некоторые практические навыки применения этих знаний сформированы недостаточно.	4	68-82	C
		Высокий	Рассмотренные в рамках данного курса средства, облегчающие декомпозицию задачи разработки программного продукта, освоены полностью, без пробелов, необходимые практические навыки применения этих знаний сформированы, все предусмотренные рабочей программой дисциплины учебные проекты по совместной программной реализации выполнены.	5	83-100	A, B
	<p><b>Владеть:</b> навыками совместной работы (по разработке ПО) в коллективе</p>	Базовый	Рассмотренные в рамках данного курса средства, облегчающие декомпозицию задачи разработки программного продукта, освоены частично, но пробелы не носят существенного характера, необходимые практические навыки применения этих знаний в основном сформированы, большинство предусмотренных рабочей программой дисциплины учебных проектов по совместной программной реализации	3	55-67	D, E

			выполнено, но с ошибками или недоотлажены.			
		Продвинутый	Рассмотренные в рамках данного курса средства, облегчающие декомпозицию задачи разработки программного продукта, освоены полностью, без пробелов, некоторые практические навыки применения этих знаний сформированы недостаточно, все предусмотренные рабочей программой дисциплины учебные проекты по совместной программной реализации выполнены, качество выполнения ни одного из них не оценено минимальным числом баллов, но некоторые из них содержат ошибки или недоотлажены	4	68-82	C
		Высокий	Рассмотренные в рамках данного курса средства, облегчающие декомпозицию задачи разработки программного продукта, освоены полностью, без пробелов, необходимые практические навыки применения этих знаний сформированы, все предусмотренные рабочей программой дисциплины учебные проекты по совместной программной реализации выполнены, качество их выполнения оценено числом баллов, близким к максимальному значению.	5	83-100	A,B

### 3 раздел ФОС.

Заполняется таблица № 3, содержащая виды оценочных средств для каждого этапа формирования компетенции.

Таблица П1.3

Оценочные средства, соответствующие конкретным этапам формирования компетенций

Код компетенции	Описание этапов формирования компетенции	Наименование оценочных средств
ОПК-3	<p><b>Знать:</b> рассмотренные в рамках данного курса</p> <ul style="list-style-type: none"> <li>- языки программирования, другие средства программирования;</li> <li>- структуры данных, алгоритмы и возможные способы их реализации.</li> </ul>	<p>Регулярные опросы на занятиях</p> <p>Контрольные работы к разделам 4, 5</p> <p>Практические задания: самостоятельно выполненные обзоры языков и других средств программирования</p> <p>Обзорные доклады по библиотекам</p> <p>Зачеты и экзамен</p>
	<p><b>Уметь:</b> применять перечисленные выше знания к решению практических задач, сформулированных в рамках изучения данного курса</p>	<p>Блиц-опрос к разделу 8 (решение простых задач)</p> <p>Контрольные работы к разделам 4, 5</p> <p>Практические задания: задачи по разделам 2-11; программные образцы по разделам 4-11</p> <p>Зачеты и экзамен</p>
	<p><b>Владеть:</b> навыками самостоятельного выбора и применения рассмотренных методов и средств к решению задач</p>	<p>Контрольные работы к разделам 5 и 6</p> <p>Практические задания: задачи по разделам 2-6; программные образцы по разделам 9-11</p> <p>Блиц-опросы к разделу 12 и разделам 18-20 (знание области применения рассматриваемых средств)</p> <p>Зачеты и экзамен</p>
УК-2	<p><b>Знать:</b> рассмотренные в рамках данного курса</p> <ul style="list-style-type: none"> <li>- современные подходы и технологии, применяемые для разработки программного обеспечения;</li> <li>- методы разработки программных приложений, использующих сложные структуры данных</li> </ul>	<p>Регулярные опросы на занятиях</p> <p>Опросы в форме обсуждения,</p> <p>Практические задания: самостоятельно выполненные обзоры средств программирования [разделы 11, 12]; программные образцы по разделам 11-12</p> <p>Обзорные доклады по средствам программирования [раздел 12]</p> <p>Зачеты и экзамен</p>
	<p><b>Уметь:</b></p> <ul style="list-style-type: none"> <li>- разрабатывать ПО в архитектуре клиент/сервер</li> </ul>	<p>Программные образцы по разделам 11-12;</p> <p>Учебные проекты, в том числе два основных индивидуальных учебных</p>

	<p>- программно реализовывать приложения на основе сложных структур данных</p> <p>- решать на основе вышеперечисленных знаний задачи по программированию компонентов интеллектуальных систем в упрощенном виде</p>	<p>проекта: реляционная БД с удаленным доступом к ней в архитектуре «клиент-сервер» и многосвязная сетевая объектная модель</p> <p>Зачеты и экзамен</p>
	<p><b>Владеть:</b> навыками самостоятельного выбора и применения вышеперечисленных знаний к разработке программных приложений</p>	<p>Программные образцы по разделам 11-12;</p> <p>Учебные проекты, в том числе два основных индивидуальных учебных проекта: реляционная БД с удаленным доступом к ней в архитектуре «клиент-сервер» и многосвязная сетевая объектная модель</p> <p>Зачеты и экзамен</p>
ОПК-5	<p><b>Знать:</b> принципы объектно-ориентированного программирования (ООП), изучаемые в рамках данного курса средства ООП и другие средства (программные интерфейсы компонентов приложения и методы их спецификации, пакеты, пространства имен и т.п.) обеспечивающие отдельную реализацию частей приложения</p>	<p>Указанные здесь аспекты в оценочных средствах для пунктов «Знать» двух групп компетенций выше</p>
	<p><b>Уметь:</b></p> <p>грамотно декомпозировать задачи в процессе проектирования программных приложений, применяя перечисленные выше знания;</p> <p>разбираться в кодах, структуре и логике программ, разрабатываемых своими коллегами</p>	<p>Программные образцы;</p> <p>Учебные проекты, в том числе два основных индивидуальных учебных проекта (разделы 12 и 13)</p>
	<p><b>Владеть:</b> навыками совместной работы (по разработке ПО) в коллективе</p>	<p>Программные образцы;</p> <p>Учебные проекты, в том числе два основных индивидуальных учебных проекта (разделы 12 и 13)</p> <p>Проверка правильности выполнения достаточно сложных образцов своими коллегами.</p>

**4 раздел ФОС** содержит методические материалы, определяющие процедуры оценивания знаний, умений, навыков и (или) опыта деятельности, характеризующих этапы формирования компетенций описывается процедура оценивания этапов формирования компетенции.

*Например:* методические материалы о промежуточном контроле успеваемости и сроках и формах его проведения.

В рамках данного курса преподавателем контролируется и оценивается выполнение всех видов практических заданий, см. раздел 5-3 настоящей рабочей программы.