

МИНОБРНАУКИ РОССИИ



Федеральное государственное бюджетное образовательное учреждение
высшего образования
«Российский государственный гуманитарный университет»
(ФГБОУ ВО «РГГУ»)

ОТДЕЛЕНИЕ ИНТЕЛЛЕКТУАЛЬНЫХ СИСТЕМ В ГУМАНИТАРНОЙ СФЕРЕ
Учебно-научный центр программного и лингвистического обеспечения
интеллектуальных систем

МЕТОДЫ СОВРЕМЕННОГО ПРОГРАММИРОВАНИЯ

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 2021

«Методы современного программирования»

Рабочая программа дисциплины

Составитель:

доцент

М.Е. Епифанов

УТВЕРЖДЕНО

Протокол совместного заседания ОИС в ГС и кафедры МЛиИС

№ 6 от 03.06.21

ОГЛАВЛЕНИЕ

1. Пояснительная записка

1.1 Цель и задачи дисциплины (*модуля*)

1.2. Формируемые компетенции, соотнесённые с планируемыми результатами обучения 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. Лист изменений

Приложение 3. 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. 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 основных концепций и технологий, применяемых в языках программирования и других средствах программирования, в частности, реализуемых в современных инструментальных системах проектирования и программирования. В рамках курса также рассматриваются тенденции дальнейшего развития выразительных средств и технологий.

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

Коды компетенции	Содержание компетенций	Перечень планируемых результатов обучения дисциплине
УК-1. Способен осуществлять критический анализ проблемных ситуаций на основе системного подхода, вырабатывать стратегию действий	УК-1.1. Знает принципы сбора, отбора и обобщения информации. УК-1.2. Умеет соотносить разнородные явления и систематизировать их в рамках избранных видов профессиональной деятельности. УК-1.3. Имеет практический опыт работы с информационными объектами и сетью Интернет, опыт библиографического разыскания, создания научных текстов.	<i>Знать:</i> современные подходы и технологии, применяемые для разработки программных приложений. <i>Уметь:</i> пользоваться современными образовательными и информационными технологиями для получения знаний. <i>Владеть</i> навыками: - анализа документации к программным системам; - самостоятельного специалиста в области программной реализации интеллектуальных систем.
УК-2. Способен управлять проектом на всех этапах его жизненного цикла	УК-2.1. Знает различные модели жизненного цикла и стандарты на представление этапов работы над проектом. УК-2.2. Умеет использовать современные бизнес- и информационные технологии для реализации проектов на различных этапах	<i>Знать:</i> современные подходы и технологии, применяемые в инструментальных средах (IDE) для совместной разработки программных приложений. <i>Уметь:</i> использовать спецификации проектов программных приложений. <i>Владеть:</i> навыками совместной работы в рабочих группах программистов.

	жизненного цикла. УК-2.2. Имеет практический опыт участия в реализации проектов на разных этапах жизненного цикла.	
УК-3. Способен организовать и руководить работой команды, вырабатывая командную стратегию для достижения поставленной цели	УК-3.1. Знает различные приемы и способы социализации личности и социального взаимодействия. УК-3.2. Умеет строить отношения с окружающими людьми, с коллегами. УК-3.3. Имеет практический опыт участия в командной работе, в социальных проектах, в шефской или волонтерской деятельности, опыт распределения ролей в условиях командного взаимодействия.	<i>Знать:</i> основные положения техники безопасности при работе аппаратным обеспечением. <i>Уметь:</i> применять эти знания в повседневной практике.
ОПК-2. Способен выявлять сущность проблем, возникающих в ходе профессиональной деятельности, привлекать соответствующий математический аппарат и информационные технологии для их решения	ОПК-2.1. Знает примеры решения разных классов задач, возникающих в профессиональной деятельности. ОПК-2.2. Умеет использовать инструментальные средства для построения математических моделей, используемых для решения прикладных задач из разных классов. ОПК-2.3. Имеет практический опыт участия в решении практических задач, предполагающий использование математического аппарата и	<i>Знать:</i> методы доступа к информационным ресурсам; основные принципы разработки прикладных программных систем. <i>Уметь:</i> 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. <i>Владеть:</i> навыками использования поисковых и библиотечных систем.

	информационных технологий.	
ОПК-4. Способен осуществлять эффективное управление проектными группами в области разработки программных средств	<p>ОПК-4.1. Знает различные модели жизненного цикла программного продукта, требования стандартов и распространенных методов создания программных систем к структуре деятельности на различных этапах разработки. Знает различные методы организации коллективной работы программистов.</p> <p>ОПК-4.2. Умеет использовать инструментальные средства и методики управления разработкой программных средств для практической работы в период создания программного продукта.</p> <p>ОПК-4.3. Имеет практический опыт участия в различных этапах работы над созданием программного средств.</p>	<p><i>Знать:</i></p> <ul style="list-style-type: none"> - современные подходы и технологии, применяемые для разработки программных приложений; - общепринятую классификацию языков и других средств программирования; - основные синтаксические конструкции изучаемых языков программирования и то, как их использовать в разработке программ <p><i>Уметь:</i></p> <ul style="list-style-type: none"> - комплексно разрабатывать программные приложения, реализуя их в архитектуре клиент/сервер; - пользоваться современными образовательными и информационными технологиями для получения знаний. <p><i>Владеть</i> навыками:</p> <ul style="list-style-type: none"> - анализа документации к программным системам; - применения текстовых процессоров и других приложений для подготовки документации по программным продуктам - самостоятельного специалиста в области программной реализации интеллектуальных систем.
ОПК-5. Способен применять новые информационные технологии в гуманитарных областях знаний с использованием средств интеллектуального анализа данных, машинного обучения, компьютерной лингвистики и представления знаний	<p>ОПК-5.1. Знает теоретические основы машинного обучения, задачи, решаемые с помощью машинного обучения и интеллектуального анализа данных.</p> <p>ОПК-5.2. Умеет применять методы интеллектуального анализа данных и машинного обучения для анализа данных в гуманитарных областях.</p> <p>ОПК-5.3. Имеет практический опыт</p>	<p><i>Знать:</i></p> <ul style="list-style-type: none"> - 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; <p><i>Уметь:</i></p> <p>решать на основе вышеперечисленных знаний задачи по программированию компонентов интеллектуальных систем в упрощенном виде.</p>

	применения интеллектуального анализа данных, машинного обучения, компьютерной лингвистики и представления знаний к исследованиям в гуманитарных областях.	
ОПК-6. Способен осваивать, применять и разрабатывать документацию к программным системам в области программирования и информационных систем	ОПК-6.1. Знает стандарты на оформление программной документации. ОПК-6.2. Умеет использовать текстовые и графические редакторы для подготовки программной документации. ОПК-6.3. Владеет инструментальными средствами для разработки документации к программным системам.	<i>Знать:</i> - основные свойства алгоритмов, формы записи алгоритмов, стандартные алгоритмы (сортировки, поиск и т.п.); <i>Уметь:</i> - разрабатывать документацию для создаваемых программ

1.3. Место дисциплины в структуре образовательной программы

Дисциплина «Методы современного программирования» входит в состав базовых дисциплин учебного плана.

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

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

2. Структура дисциплины

Структура дисциплины для очной формы обучения

Структура дисциплины для очной формы обучения

Общая трудоёмкость дисциплины составляет 12 з.е., 456 ч., в том числе контактная работа обучающихся с преподавателем 120 ч., самостоятельная работа обучающихся 318 ч. Дисциплина читается с 1-го по 3-й семестр.

№ п/п	Раздел дисциплины/темы	Се ме	Виды учебной работы (в часах)	Формы текущего
-------	------------------------	-------	-------------------------------	----------------

		ст р	Контактная				Про меж уточ ная атте стац ия	Са мо сто яте ль ная рабо та	контроля успеваемости, форма промежуточной аттестации (<i>по семестрам</i>)
			Лек ции	Се ми нар	Прак тичес кие занят ия	Лабо ратор ные занят ия			
1	Средства и языки программирования: концепции, стили и реализация	1	2					4	Опрос
2	Сравнительный обзор языков программирования C, C++, Java и C#	1	2					4	Опрос
3	Интегрированная среда разработки Visual Studio и платформа .NET	1	4			6		10	Оценка выполнения практических заданий
4	Язык C++: основные средства.	1				4		8	Оценка выполнения практических заданий, контрольная работа
5	Средства объектно-ориентированного программирования в C++	1				6		24	Оценка выполнения практических заданий, контрольная работа
6	Язык C++: шаблоны и их применение	1	2			6		16	Оценка выполнения практических заданий
7	Стандартная библиотека C++	1	2			4		16	Оценка выполнения практических заданий
8	Язык C#	1	8			14		86	Оценка выполнения практических заданий
	Промежуточная аттестация	1							зачет
9	Библиотека классов .NET Framework	2				12		40	Оценка выполнения

									практических заданий
10	Разработка и применение сценариев. Языки Javascript .NET и TypeScript	2				10		20	Оценка выполнения практических заданий
11	Язык Python	2				8		24	Оценка выполнения практических заданий
	Промежуточная аттестация	2							зачет
12	Разработка многопоточных приложений на платформе .NET	3				8		12	Оценка выполнения практических заданий, опрос оценка совместной работы со студентом, как с исполнителем
13	Разработка распределенных приложений на платформе .NET	3				22		54	Оценка выполнения практических заданий, опрос оценка совместной работы со студентом, как с исполнителем
	Промежуточная аттестация	3						18	экзамен
	Итого		20			100	18	318	

3. Содержание дисциплины

В курсе предлагается обзор основных концепций, стилей и реализующих их средств программирования, изучаются

- средства объектно-ориентированного программирования (ООП) в языках C++ и C#;
- средства разработки серверных приложений в распределенных системах, в частности, средства удаленного доступа к базам данных;
- методы реализации базовых алгоритмов (сортировки, поиск и т.п.);
- методы проектирования и программной реализации сложных структур данных;
- методы программной реализации некоторых упрощенных компонентов интеллектуальных систем.

№ 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	Сравнительный обзор языков программирования C, C++, Java и C#	Синтаксис языков, реализуемая ими основная функциональность. В частности: - указатели (ссылки – pointers) явно в C и C++; - C как «чисто» процедурный язык без ООП; - расширение C до C++, в частности, встроенная система ООП в C++; - Java и C# как объектноориентированные языки.
3	Интегрированная среда разработки Visual Studio и платформа .NET	Состав Microsoft Visual Studio (VS). Платформа .NET Framework. Инструментальные средства программиста: поддержка проектов, контекстный редактор, отладчик и т.п
4	Язык C++: основные средства.	Синтаксис, виды инструкций, управляющие инструкции. Типы и объявления. Указатели и работа с ними. Массивы и структуры. Обзор операторов, выражения. Функции. Перегруженные функции. Пространства имен. Исключения. Разработка исходного кода, компиляция, сборка программы.
5	Средства объектно-ориентированного программирования в C++	Парадигма ООП, инкапсуляция, полиморфизм, наследование. Объекты и классы в C++. Перегрузка операторов. Классы – производные, абстрактные; виртуальные функции. Иерархии классов и их проектирование. Обработка исключений.
6	Язык C++: шаблоны и их применение	Определение шаблона. Шаблоны строк. Шаблоны и типы. Шаблоны функций. Специализация. Наследование и шаблоны.
7	Стандартная библиотека C++	Организация библиотеки, контейнеры, их проектирование. Стандартные контейнеры. Алгоритмы и объекты-

		функции. Итераторы и распределители памяти. Потoki.
8	Язык C#	Структура программы C#. Типы. Массивы. Строки Инструкции, выражения и операторы. Классы и структуры. Интерфейсы. Индексаторы. Типы перечислений. Делегаты. События. Универсальные шаблоны. Итераторы. Выражения запросов LINQ. Лямбда-выражения. Пространства имен. Типы, допускающие значения NULL. Небезопасный код и указатели. Коллекции. Исключения и обработка исключений
9	Библиотека классов .NET Framework	Обзор пространств имен, типов (классов) и их методов, (обще)доступных из входящих в состав VS языков. Выполнение практических заданий, в которых некоторые из этих классов применяются в программных образцах
10	Разработка и применение сценариев. Языки Javascript .NET и TypeScript	Синтаксис языков. Интерпретация сценариев. Обзор операторов, выражения. Виды инструкций. Встроенные функции. Встроенные объекты и их иерархия. Объекты, классы и прототипы.
11	Язык Python	Подключаемая среда IDE Python, Python Tools for Visual Studio (PTVS). Описание языка, программа на Python. Императивный, функциональный и объектно-ориентированный стили в Python. ООП в Python. Библиотеки Python. Разработка сценариев на языке Python.
12	Разработка многопоточных приложений на платформе .NET	Выполнение учебного проекта малыми (2-3 человека) группами студентов под руководством преподавателя в качестве менеджера проекта
13	Разработка распределенных приложений на платформе .NET	Использование ADO.NET для связи с другими приложениями. Общее представление о двухуровневой (клиент/сервер) и трехуровневой архитектурах распределенных приложений, о Windows DNA на основе COM+. Технологии ASP.NET, Ajax, формат JSON и их применение. Службы IIS.NET. Реализация серверных приложений на

		платформе Windows средствами VS. Выполнение учебного проекта малыми (2-3 человека) группами студентов под руководством преподавателя в качестве менеджера проекта
--	--	---

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 технологические элементы и/или применяемые средства программирования, скажем, что 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 12 и 13 – 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;
- 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.

В период временного приостановления посещения обучающимися помещений и территории РГГУ. для организации учебного процесса с применением электронного

обучения и дистанционных образовательных технологий могут быть использованы следующие образовательные технологии:

- видео-лекции;
- онлайн-лекции в режиме реального времени;
- электронные учебники, учебные пособия, научные издания в электронном виде и доступ к иным электронным образовательным ресурсам;
- системы для электронного тестирования;
- консультации с использованием телекоммуникационных средств.

5. Оценка планируемых результатов обучения

5.1. Система оценивания

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

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	зачтено	A
83 – 94			B
68 – 82	хорошо		C
56 – 67	удовлетворительно		D
50 – 55			E
20 – 49	неудовлетворительно	не зачтено	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»	Выставляется обучающемуся, если он: - грамотно выполнил большинство (в том числе и все обязательные) практические задания (программные образцы – samples); - глубоко и прочно усвоил теоретический и

		<p>практический материал, может продемонстрировать это на занятиях и в ходе промежуточной аттестации;</p> <ul style="list-style-type: none"> - 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; - свободно ориентируется в учебной и профессиональной литературе. <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/ С	«хорошо»/ «зачтено»	<p>Выставляется обучающемуся, если он:</p> <ul style="list-style-type: none"> - грамотно, возможно с помощью преподавателя, выполнил достаточное количество практических заданий (программные образцы – samples), в том числе и все обязательные; - знает теоретический и практический материал, грамотно и по существу излагает его на занятиях и в ходе промежуточной аттестации, не допуская существенных неточностей; - правильно применяет теоретические положения при решении практических задач профессиональной направленности разного уровня сложности, владеет необходимыми для этого навыками и приёмами; - достаточно хорошо ориентируется в учебной и профессиональной литературе. <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"> - выполнил (возможно с помощью преподавателя) достаточное количество практических заданий (программные образцы – samples); - знает на базовом уровне теоретический и практический материал, допускает отдельные ошибки при его изложении на занятиях и в ходе промежуточной аттестации; - испытывает определённые затруднения в применении теоретических положений при решении практических задач профессиональной направленности стандартного уровня сложности, владеет необходимыми для этого базовыми навыками и приёмами; - демонстрирует достаточный уровень знания учебной литературы по дисциплине.

		<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"> - не выполнил (несмотря на возможную помощь преподавателя) достаточное количество практических заданий (программные образцы – samples); - не знает на базовом уровне теоретический и практический материал, допускает грубые ошибки при его изложении на занятиях и в ходе промежуточной аттестации. - испытывает серьёзные затруднения в применении теоретических положений при решении практических задач профессиональной направленности стандартного уровня сложности, не владеет необходимыми для этого навыками и приёмами; - демонстрирует фрагментарные знания учебной литературы по дисциплине. <p>Оценка по дисциплине выставляются 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.

Перечень компетенций, формируемых дисциплиной с указанием этапов их формирования

<i>№ n/ n</i>	<i>Код компетенции</i>	<i>Описание этапов формирования компетенции</i>
1.	УК-1, УК-2, УК-3,	<p>Знать: рассмотренные в рамках данного курса - языки программирования, другие средства программирования; - структуры данных, алгоритмы и возможные способы их реализации.</p> <p>Уметь: применять перечисленные выше знания к решению практических задач сформулированных в рамках изучения данного курса</p> <p>Владеть: навыками самостоятельного выбора и применения рассмотренных методов и средств к решению задач</p>
2	ОПК-2, ОПК-4,.	<p>Знать: рассмотренные в рамках данного курса - современные подходы и технологии, применяемые для разработки программного обеспечения; - методы разработки программных приложений, использующих сложные структуры данных</p> <p>Уметь: - 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</p> <p>Владеть: навыками самостоятельного выбора и применения вышеперечисленных знаний к разработке программных приложений</p>
3	ОПК-5,	<p>Знать: 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>Уметь: грамотно декомпозировать задачи в процессе проектирования программных</p>

		приложений, применяя перечисленные выше знания; разбираться в кодах, структуре и логике программ, разрабатываемых своими коллегами
		Владеть: навыками совместной работы (по разработке ПО) в коллективе
4	ОПК-6	Знать: основные принципы разработки технической документации программным продуктам
		Уметь: пользоваться справочными системами, разбираться в другой технической документации
		Владеть: навыками документирования результатов собственной реализации программного продукта

К разделу 4 описания ФОС.

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

- обзоры выразительных и функциональных средств изучаемых языков и библиотек (применительно к разделам 3-11);
- программные образцы (samples) изучаемых элементов технологий и средств разработки ПО;
- традиционно «жестко», четко сформулированные задачи;
- совместное решение некоторых задач в рамках так называемых учебных проектов (разделы 12, 13).

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.

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 12 и 13, – см. п. 3) 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, «недоделок», 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.

Примеры зачетных практических заданий (программных образцов).

1. Для изучаемых в рамках данного курса языков программирования обзоры:

- базовых «процедурных» средств языка;
- представленных в языке средств ООП.

2. Выполняемые на C# с использованием системной библиотеки поддерживаемых .NET Framework классов учебные проекты.

- Управление выполнением внешнего процесса.

По сценарию sample-a запускается внешний исполняемый модуль (exe-файл), который, например, что-то пишет в известный «выходной» файл. «Обнаруживается» момент окончания выполнения этого процесса, после чего результат его работы считывается из его выходного файла и предъявляется пользователю образца.

- Пример взаимодействия клиент-сервер.

В качестве сервера используется директория (папка), определенная как localhost. Это

позволяет выполнять `sample` на единственном (локальном) компьютере.

На сервере размещается база данных и ее программная оболочка. Простая БД реализована в IDE MS Access. Это упрощение обусловлено невозможностью (из-за предоставленного для него объема времени) в рамках данного курса устанавливать на компьютере учащегося и осваивать MS SQL Server.

Простое приложение (html-страница) – поддерживаемое на стороне клиента браузером MS Internet Explorer или MS Edge запрашивает серверную БД и получает результат запроса через ASP.NET. Результат запроса предьявляется на странице пользователю клиента. На сервере запрос к БД выполняется через ADO.NET.

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. Страуструп, Б. Дизайн и эволюция C++ [Электронный ресурс] / Б. Страуструп; Пер. с англ. - М.: ДМК Пресс, 2007. - 448 с.: ил. - (Серия «Для программистов»). - ISBN 5-94074-005-7.

<http://znanium.com/catalog.php?bookinfo=409529>

2. Мейерс С. Наиболее эффективное использование C++. 35 новых рекомендаций по улучшению ваших программ и проектов [Электронный ресурс]: учебное пособие. — Электрон. дан. — М.: ДМК Пресс, 2007. — 294 с.

<http://znanium.com/bookread2.php?book=407506>

К разделу 2

2. Книга (документация) по Visual Studio 2015 (содержит ссылку на документацию по Visual Studio 2017) на сайте технической документации фирмы Microsoft для разработчиков ПО (MSDN):

<https://msdn.microsoft.com/ru-ru/library/dd831853.aspx>

К разделам 3-7

3. *Страуструп Б.* Язык программирования C++. Специальное издание. – М.: Издательство БИНОМ, 2011.– 1136 с.

(ЭБ УМК ОИС: [d:_ois_lib\2cpp\B_Stroustrup_Yazyk_programirovaniya_C++ (2011).djvu])

4. Visual C++ в Visual Studio 2015. Книга (документация) на сайте технической документации фирмы Microsoft для разработчиков ПО (MSDN):

<https://msdn.microsoft.com/ru-ru/library/60k1461a.aspx>

К разделу 8

5. С#. Книга (документация) на сайте технической документации фирмы Microsoft для разработчиков ПО (MSDN):

<https://msdn.microsoft.com/ru-ru/library/kx37x362.aspx>

6. *Шилдт Г.* C# 4.0: полное руководство. – М.: ООО «И. Д. Вильямс», 2011 – 1056 с. (ЭБ УМК ОИС: [d:_ois_lib\C-sh\GShildt-C#4.0.Polnoe_rukov-2011.pdf])

К разделам 9, 12, 13

7. .NET Framework 4.6 и 4.5. Книга (документация) на сайте технической документации фирмы Microsoft для разработчиков ПО (MSDN):
[https://msdn.microsoft.com/ru-ru/library/w0x726c2\(v=vs.110\).aspx](https://msdn.microsoft.com/ru-ru/library/w0x726c2(v=vs.110).aspx)

К разделам 10, 13

8. Справочник по языку JavaScript. Книга (документация) на сайте технической документации фирмы Microsoft для разработчиков ПО (MSDN):
[https://msdn.microsoft.com/ru-ru/library/d1et7k7c\(v=vs.94\).aspx](https://msdn.microsoft.com/ru-ru/library/d1et7k7c(v=vs.94).aspx)

К разделам 11, 12, 13

9. Начало работы с Python. Книга (документация) на сайте технической документации фирмы Microsoft для разработчиков ПО (MSDN):
<https://msdn.microsoft.com/ru-ru/library/dn705848.aspx>

б) Дополнительная литература

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

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

12. *Прата С.* Язык программирования C++. – М.: Издательский дом «Вильямс», 2007. (ЭБ УМК ОИС: [d:_ois_lib\2cpp\Prata_YAzyk-C++2007.pdf.djvu])

К разделу 13

13. *Горин М. А.* 2 в 1: Как создать Web-сайт. – М.: Триумф, 2009

14. *Дунаев В. В.* HTML, скрипты и стили. – СПб: ВHV-СПб, 2008

15. *Коэн И.* Полный справочник по HTML, CSS и JavaScript. Серия: Справочник профессионала. – М.: Эком Паблишера, 2007

16. *Крейн Д., Пакарелло, Э., Джеймс, Д.* Аjax в действии. – М.: Издательский дом «Вильямс», 2006.

17. *Айзекс С.* Dynamic HTML. – СПб.: ВHV-Санкт-Петербург, 1998. (ЭБ УМК ОИС: – с примерами в папке [d:_ois_lib\3ip\DHTML-book(byS_Isaacs)])

18. *Бибо, Б., Кау, 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])

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

20. XML Schema., – рабочий документ (рекомендация) консорциума W3C, опубликован (регулярно обновляется) на сайте консорциума (<http://www.w3.org/>):
<http://www.w3.org/TR/xmlschema/>.

6.2 Перечень ресурсов информационно-телекоммуникационной сети "Интернет", необходимых для освоения дисциплины

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

<https://msdn.microsoft.com/ru-ru/library/dd831853.aspx> – книга (документация) по Visual Studio 2015 (содержит ссылку на документацию по Visual Studio 2017) на сайте технической документации фирмы Microsoft для разработчиков ПО (MSDN)

<https://msdn.microsoft.com/ru-ru/library/60k1461a.aspx> – Visual C++ в Visual Studio 2015. Книга (документация) на сайте технической документации фирмы Microsoft для разработчиков ПО (MSDN)

<https://msdn.microsoft.com/ru-ru/library/kx37x362.aspx> – C#. Книга (документация) на сайте технической документации фирмы Microsoft для разработчиков ПО (MSDN)

[https://msdn.microsoft.com/ru-ru/library/w0x726c2\(v=vs.110\).aspx](https://msdn.microsoft.com/ru-ru/library/w0x726c2(v=vs.110).aspx) – .NET Framework 4.6 и 4.5. Книга (документация) на сайте технической документации фирмы Microsoft для разработчиков ПО (MSDN)

[https://msdn.microsoft.com/ru-ru/library/d1et7k7c\(v=vs.94\).aspx](https://msdn.microsoft.com/ru-ru/library/d1et7k7c(v=vs.94).aspx) – Справочник по языку JavaScript. Книга (документация) на сайте технической документации фирмы Microsoft для разработчиков ПО (MSDN)

<https://msdn.microsoft.com/ru-ru/library/dn705848.aspx> – Начало работы с Python. Книга (документация) на сайте технической документации фирмы Microsoft для разработчиков ПО (MSDN)

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

<http://www.w3.org/TR/xmlschema/> – XML Schema (XSD) – рабочий документ (рекомендация) консорциума W3C

Перечень БД и ИСС

№п /п	Наименование
1	Международные реферативные наукометрические БД, доступные в рамках национальной подписки в 2021 г. Web of Science Scopus
2	Профессиональные полнотекстовые БД, доступные в рамках национальной подписки в 2021 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.

1. Перечень ПО

№п /п	Наименование ПО	Производитель	Способ распространения (лицензионное или свободно распространяемое)
1	Adobe Master Collection CS4	Adobe	лицензионное
2	ОС «Альт Образование» 8	ООО «Базальт СПО	лицензионное
3	Windows 10 Pro	Microsoft	лицензионное
4	Kaspersky Endpoint Security	Kaspersky	лицензионное
5	Microsoft Office 2016	Microsoft	лицензионное
6	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 лекциях 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. Примеры планов лабораторных занятий

1. Изучение средств программирования (темы 1-11).

Тема4 (4ч.) Язык C++: основные средства

Цель занятий: изучение базовых «процедурных» средств языка (т.е. исключая встроенную в язык подсистему ООП)

Форма проведения – опрос, дискуссии, решение задач.

Вопросы для обсуждения:

Синтаксис, виды команд, управляющие инструкции.

Типы и объявления.

Указатели и работа с ними.

Массивы и структуры.

Обзор операторов, выражения. Функции.

Перегруженные функции.

Пространства имен. Исключения.

Разработка исходного кода, компиляция, сборка программы.)

Примеры контрольных вопросов:

1. Парадигма процедурного программирования, в том числе на примере C++.
2. Подмножество C в C++ и расширения C к C++.
3. Применение рассматриваемых типов данных.
4. Элементы программирования динамических структур данных с использованием указателей и структур.

Задание:

программно реализовать обзорный образец базовых «процедурных» средств языка.

Список источников и литературы:

1. Страуструп, Б. Дизайн и эволюция C++ [Электронный ресурс] / Б. Страуструп; Пер. с англ. - М.: ДМК Пресс, 2007. - 448 с.: ил. - (Серия «Для программистов»). - ISBN 5-94074-005-7.
<http://znanium.com/catalog.php?bookinfo=409529>
2. Мейерс С. Наиболее эффективное использование C++. 35 новых рекомендаций по улучшению ваших программ и проектов [Электронный ресурс]: учебное пособие. — Электрон. дан. — М.: ДМК Пресс, 2007. — 294 с.
<http://znanium.com/bookread2.php?book=407506>
3. *Страуструп Б.* Язык программирования C++. Специальное издание. – М.: Издательство БИНОМ, 2011.– 1136 с.
(ЭБ УМК ОИС: [d:_ois_lib\2cpp\B_Stroustrup_Yazyk_programirovaniya_C++ (2011).djvu])
4. Visual C++ в Visual Studio 2015. Книга (документация) на сайте технической документации фирмы Microsoft для разработчиков ПО (MSDN):
<https://msdn.microsoft.com/ru-ru/library/60k1461a.aspx>

2. Выполняемые на C# с использованием системной библиотеки поддерживаемых .NET Framework классов учебные проекты. (темы 12-13).

Тема13 (24ч.) Разработка распределенных приложений на платформе .NET

Цель занятий: изучение архитектуры «клиент-сервер» и пример уу реализации с использованием соответствующего подмножества классов из системной библиотеки.NET.

Форма проведения – опрос, дискуссии, решение задач.

Вопросы для обсуждения:

Классы ADO.NET.

Классы ASP.NET.

HTTP-серверы, служба IIS и организация localhost.

Общая архитектура образца и сценарий его выполнения.

Пример (зачетного) задания:

Пример взаимодействия клиент-сервер.

В качестве сервера используется директория (папка), определенная как localhost. Это позволяет выполнять sample на единственном (локальном) компьютере.

На сервере размещается база данных и ее программная оболочка. Простая БД реализована в IDE MS Access. Это упрощение обусловлено невозможностью (из-за предоставленного для него объема времени) в рамках данного курса устанавливать на компьютере учащегося и осваивать MS SQL Server.

Простое приложение (html-страница) – поддерживаемое на стороне клиента браузером MS Internet Explorer или MS Edge запрашивает серверную БД и получает результат запроса через ASP.NET. Результат запроса предьявляется на странице пользователю клиента. На сервере запрос к БД выполняется через ADO.NET.

Список источников и литературы:

1. С#. Книга (документация) на сайте технической документации фирмы Microsoft для разработчиков ПО (MSDN):
<https://msdn.microsoft.com/ru-ru/library/kx37x362.aspx>
2. Шилдт Г. С# 4.0: полное руководство. – М.: ООО «И. Д. Вильямс», 2011 – 1056 с.
(ЭБ УМК ОИС: [d:_ois_lib\C-sh\GShildt-C#4.0.Polnoe_rukov-2011.pdf])
3. \.NET Framework 4.6 и 4.5. Книга (документация) на сайте технической документации фирмы Microsoft для разработчиков ПО (MSDN):
[https://msdn.microsoft.com/ru-ru/library/w0x726c2\(v=vs.110\).aspx](https://msdn.microsoft.com/ru-ru/library/w0x726c2(v=vs.110).aspx)
4. MSDN Microsoft Developer Network – техническая документация фирмы Microsoft для разработчиков ПО: <http://msdn.microsoft.com>

АННОТАЦИЯ ДИСЦИПЛИНЫ

Дисциплина «Методы современного программирования» 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. 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 основных концепций и технологий, применяемых в языках программирования и других средствах программирования, в частности, реализуемых в современных инструментальных системах проектирования и программирования. В рамках курса также рассматриваются тенденции дальнейшего развития выразительных средств и технологий.

Дисциплина направлена на формирование следующих компетенций:

- УК-1. Способен осуществлять критический анализ проблемных ситуаций на основе системного подхода, выработать стратегию действий;
- УК-2. Способен управлять проектом на всех этапах его жизненного цикла;
- УК-3. Способен организовать и руководить работой команды, выработывая командную стратегию для достижения поставленной цели;
- ОПК-2. Способен выявлять сущность проблем, возникающих в ходе профессиональной деятельности, привлекать соответствующий математический аппарат и информационные технологии для их решения;
- ОПК-4. Способен осуществлять эффективное управление проектными группами в области разработки программных средств;
- ОПК-5. Способен применять новые информационные технологии в гуманитарных областях знаний с использованием средств интеллектуального анализа данных, машинного обучения, компьютерной лингвистики и представления знаний;
- ОПК-6. Способен осваивать, применять и разрабатывать документацию к программным системам в области программирования и информационных систем;

В результате освоения дисциплины обучающийся должен:

Знать

- общепринятую классификацию языков и других средств программирования;
- основные свойства алгоритмов, формы записи алгоритмов, стандартные алгоритмы (сортировки, поиск и т.п.);
- основные синтаксические конструкции изучаемых языков программирования и то, как их использовать в разработке программ;
- 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
- современные подходы и технологии, применяемые для разработки программных приложений.

Уметь

- решать на основе вышеперечисленных знаний задачи по программированию компонентов интеллектуальных систем в упрощенном виде;

- комплексно разрабатывать программные приложения, реализуя их в архитектуре клиент/сервер.

Владеть навыками

- самостоятельного специалиста в области программной реализации интеллектуальных систем;
- применения текстовых процессоров и других приложений для подготовки документации по программным продуктам.

По дисциплине предусмотрены промежуточные аттестации в форме зачета и экзамен.

Общая трудоемкость освоения дисциплины составляет 12 зачетных единиц.

Приложение 2

ЛИСТ ИЗМЕНЕНИЙ