

**УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ
«УНИВЕРСИТЕТ УПРАВЛЕНИЯ «ТИСБИ»**

Кафедра информационных технологий

Утверждаю
Зав. кафедрой
О.В.Федорова
Протокол заседания
кафедры № 10
от 06.04.2026

Рабочая программа дисциплины

Наименование дисциплины	Архитектура программного обеспечения
Направление подготовки	09.03.01 Информатика и вычислительная техника
Направленность (профиль) образовательной программы	Инженерия информационных систем
Год набора	2025, 2026

Составители:

Доцент лаборатории сетей и блокчейн технологий Мелехова А.Л.
АНО ВО «Университет Иннополис»

Казань

РАБОЧАЯ ПРОГРАММА ДИСЦИПЛИНЫ (МОДУЛЯ)

Разделы рабочей программы:

1. Краткая характеристика дисциплины (модуля).
2. Перечень планируемых результатов обучения.
3. Место дисциплины (модуля) в структуре основной профессиональной образовательной программы высшего образования (ОПОП ВО).
4. Объем, структура и содержание дисциплины (модуля).
5. Учебно-методическое обеспечение.
6. Материально-техническое обеспечение.
7. Оценочные и методические материалы.
8. Дополнительные сведения.

Программа составлена в соответствии с требованиями ОПОП ВО Университета Иннополис по данному направлению подготовки.

Форма обучения	Семестр	Трудоемкость дисциплины в		Контактная работа		Контроль, час.	Самостоятельная работа, час.	Форма промежуточной аттестации (экз./диф. Зач./зач.)
		Зачетных единиц	академических часов	Лекции, час.	Семинарские (практические) занятия, час.			
очная	8	4	144	30	30	18	66	экзамен

1. КРАТКАЯ ХАРАКТЕРИСТИКА ДИСЦИПЛИНЫ (МОДУЛЯ)

Изучение дисциплины (модуля) «Applied Software Architecture / Архитектура программного обеспечения» обеспечивает формирование и развитие компетенций обучающихся в области программного обеспечения компьютерных вычислительных систем и сетей, автоматизированных систем обработки информации и управления, их применение для решения различных прикладных задач в рамках профессиональной деятельности. В ходе освоения дисциплины обучающиеся формируют способность применения навыков построения проектной деятельности, включая анализ и изменение функциональных и нефункциональных требований, а также их влияние на проектирование решений, создание архитектуры продукта в рамках разработки программного обеспечения с применением классических, инкрементно-итерационных и гибких методологий.

2. ПЕРЕЧЕНЬ ПЛАНИРУЕМЫХ РЕЗУЛЬТАТОВ ОБУЧЕНИЯ

Целью освоения дисциплины (модуля) является формирование знаний, умений и навыков в рамках профессиональной деятельности, позволяющих успешно работать в избранной отрасли, а также обладать компетенциями, способствующими социальной мобильности, устойчивости и конкурентоспособности обучающегося в условиях современных рыночных отношений. Задачами дисциплины являются формирование способности проектирования решений и создания информационных архитектур в области программного обеспечения компьютерных вычислительных систем и сетей, автоматизированных систем обработки и управления.

Компетенции, формируемые в результате освоения дисциплины (модуля) при проведении учебных занятий в форме контактной работы обучающихся с педагогическими работниками Университета и в форме самостоятельной работы обучающихся:

№ п/п	Код компетенции	Наименование компетенции	Индикаторы компетенций	Методы и технологии обучения, способствующие формированию компетенции
1.	ПК-1	Способен разрабатывать требования и проектировать программное обеспечение, создавать и поддерживать среды разработки, автоматизировать и контролировать качество разработки программных решений для	Знать: основы структурного, объектно-ориентированного и функционального программирования; стандартные и прикладные библиотеки для выбранного языка программирования; принципы включения стороннего исходного кода (в том числе открытого) в разрабатываемые решения жизненный цикл и этапы разработки ПО; компилируемые и скриптовые языки	Метод анализа конкретных ситуаций, ситуационные задачи и упражнения; Метод дискуссии; Метод контрольных вопросов; Проектная технология; Информационно –

		информационных систем	программирования; языки разметки и представления/хранения данных; архитектуру баз данных (SQL / NoSQL); языки запросов к базам данных (SQL); паттерны проектирования программного кода; архитектуры приложений; общие принципы работы операционных систем; принципы работы микросервисной архитектуры; принципы взаимодействия с API удаленных сервисов (REST, SOAP, PRC); системы управления проектами и задачами (Jira, YouTrack и др.); методики управления IT-проектами; инструменты и методы прототипирования пользовательского интерфейса. Уметь: писать поддерживаемый код на различных языках программирования; определять архитектуру приложений ИС, их масштабируемость; устанавливать и настраивать серверы баз данных; работать с инструментами и методами обработки BigData; эффективно использовать инструменты и системы контроля версий; работать с контейнерами и виртуальными машинами; управлять изменениями в приложениях и автоматизировать их; оценивать временные и материальные затраты на разработку отдельных компонент информационных систем. Владеть: методами и приемами формализации задач, методами и средствами проектирования программного обеспечения, проектирования программных интерфейсов, проектирования баз данных; навыками автоматизированной проверки и заливки кода, сборки компонентов программного решения.	коммуникационная технология; Групповая технология
2.	ПК-2	Способен выполнять работы по созданию (модификации) и сопровождению информационных систем	Знать: архитектуру, устройство и функционирование вычислительных и информационных систем; устройство и функционирование современных информационных систем; современные стандарты	Метод анализа конкретных ситуаций, ситуационные задачи и упражнения; Метод дискуссии; Метод контрольных

			информационного взаимодействия систем; программные средства и платформы инфраструктуры информационных технологий организаций; современные подходы и стандарты автоматизации организации (например, CRM, MRP, ERP, ITIL, ITSM); методы выявления требований. Уметь: разрабатывать архитектуру, дизайн и прототипы информационной системы; разрабатывать метрики (количественные показатели) работы информационной системы; настраивать и создавать компоненты анализа программного решения информационной системы; исправлять дефекты и несоответствия в архитектуре и дизайне информационной системы, подтверждать исправление дефектов и несоответствий в коде информационной системы и документации к ней; проверять (верифицировать) архитектуру информационной системы, выполнять параметрическую настройку информационной системы; разрабатывать технологии обмена данными; разрабатывать базы данных информационной системы; осуществлять оптимизацию информационной системы; разрабатывать пользовательскую документацию; оценивать временные и материальные затраты на разработку информационной системы при заданных ресурсах. Владеть: методами оптимизации работы информационной системы и её компонентов; приемами интеграции информационной системы с техническим обеспечением заказчика.	вопросов; Проектная технология; Информационно – коммуникационная технология; Групповая технология
--	--	--	---	---

Знания: сформированы систематические знания шаблонов и стилей проектных решений; паттернов проектирования; драйверов, формирующих архитектурные решения; моделей построения проектных решений; типовых тактик соответствия атрибутам качества; методологии определения целей и задач

принятия решений и проведения экспериментальных исследований; способов инсталляции и механизмов администрирования, тенденции их развития; лексики, представляющей нейтральный научный стиль, а также основной терминологии своей специальности.

Умения: сформированы умения проводить анализ функциональных и нефункциональных требований, а также бизнес и технических ограничений проекта; использовать типовые тактики для соответствия выбранных в ходе анализа атрибутов качества; строить архитектурные виды с учетом физической, динамической и статической перспектив; проверять гипотезы архитектурных решений путем создания прототипов и тестирования данных прототипов; оперировать единым представлением сети в процессе ее создания и дальнейшего сопровождения; проводить экспериментальные исследования, применять методы планирования экспериментов, анализировать результаты экспериментальных исследований; использовать базовые информационные процессы и технологии для проектирования и реализации информационных систем; работать в команде; корректно выражать и аргументированно обосновывать положения предметной области знания, интерпретировать полученные в практической деятельности результаты.

Навыки (владения): сформировано владение навыками выявления и анализа драйверов архитектуры; навыками сравнения различных стилей архитектурных решений для выбора наиболее подходящего архитектурного шаблона в качестве отправной точки; навыками детализации проектных решений с целью их видоизменения в соотношении с эволюцией требований и изменениями контекста пользования программным продуктом; навыками поддержки принятия решений и планирования экспериментов и анализа их результатов; навыками бесконфликтной работы и толерантного поведения при создании профессионально-направленной коммуникации.

3. МЕСТО ДИСЦИПЛИНЫ (МОДУЛЯ) В СТРУКТУРЕ ОСНОВНОЙ ПРОФЕССИОНАЛЬНОЙ ОБРАЗОВАТЕЛЬНОЙ ПРОГРАММЫ ВЫСШЕГО ОБРАЗОВАНИЯ (ОПОП ВО)

Данная учебная дисциплина (модуль) включена в «Блок 1. Дисциплины (модули)» образовательной программы по направлению подготовки 09.03.01 Информатика и вычислительная техника, направленность (профиль) образовательной программы «Инженерия информационных систем» и относится к части, формируемой участниками образовательных отношений, обеспечивает формирование профессиональных компетенций. Дисциплина (модуль) «Applied Software Architecture / Архитектура программного обеспечения» позволяет обучающимся получить знания, умения и навыки для успешной профессиональной деятельности в области программного обеспечения компьютерных вычислительных систем и сетей, автоматизированных систем обработки информации и управления. Дисциплина осваивается студентами при обучении в очной форме на 3 курсе в 8 семестре.

4. ОБЪЕМ, СТРУКТУРА И СОДЕРЖАНИЕ ДИСЦИПЛИНЫ (МОДУЛЯ)

Объем дисциплины (модуля) составляет:

4 зачетные единицы, 144 академических часа, объем контактной работы — 60 академических часов.

Структура дисциплины (модуля):

№ раздела	Наименование раздела дисциплины (модуля)	Трудоемкость, академические часы			
		Очная форма			
		Контактная работа		Контроль, час.	Самостоятельная работа, час.
		Лекции, час.	Семинарские (практические) занятия, час.		
1.	Введение в Архитектуру ПО, структуры проектирования	4	4	0	9
2.	Драйверы Архитектуры	4	4	0	9
3.	Тактики и Паттерны	4	4	0	9
4.	Проектирование архитектуры	4	4	0	9
5.	Аналитические модели	4	4	0	10
6.	Документирование архитектуры.	5	5	0	10
7.	Эволюция архитектуры. Методы гибкой разработки ПО	5	5	0	10
	Промежуточная аттестация - экзамен			18	
ИТОГО:		30	30	18	66

Содержание дисциплины (модуля):

№ раздела	Наименование раздела дисциплины (модуля)	Содержание дисциплины (модуля) по темам	Формируемые компетенции
1.	Введение в Архитектуру ПО, структуры проектирования	Определение архитектуры ПО. Отношение архитектуры ПО к разработке.	
2.	Драйверы Архитектуры	Выявление бизнес-целей и контекста. Выявление инженерных целей и разрабатываемого ПО в контексте Определение архитектурных драйверов.	
3.	Тактики и Паттерны	Определение архитектурных тактик. Виды архитектурных тактик, базирующихся на стандартных нефункциональных требованиях. Стратегии использования архитектурных тактик. Определение архитектурных паттернов.	

		Рассмотрение наиболее популярных архитектурных паттернов. Стратегии комбинирования тактик совместно с паттернами проектирования.	ПК-1; ПК-2
4.	Проектирование архитектуры	Стратегии проектирования. Рассмотрение метода проектирования через атрибуты качества (Attribute-Driven Design Method). Рассмотрение этапов проектирования системы через пример “системы автоматизации заданий”.	
5.	Аналитические модели	Моделирование архитектур для обеспечения анализа атрибутов качества. Составление контрольных списков атрибутов качества; Изучение техники анализа «обратной стороны конверта» (Back-of-the-Envelope Analysis). Проведение экспериментов, моделирования и прототипирования. Проведение анализа на разных этапах жизненного цикла.	
6.	Документирование архитектуры. Архитектурные стили	Обзор принципов и методов хорошей архитектурной документации. Изучение хороших и плохих примеров архитектур; документирования. Рассмотрение документирования архитектуры программного обеспечения с использованием UML. Рассмотрение преимуществ и недостатков при использовании UML. Рассмотрение стилей, базирующихся на семействах: Потока данных (Dataflow) Возврата вызова (Call-Return) Клиент серверных (Client-server) Многоуровневых (Tiered) Событийно-ориентированных (Event Oriented). Рассмотрение архитектур базирующихся на: Атрибуте качества «Доступность» (Availability) Атрибуте качества «Безопасность» (Security) Межплатформном программном обеспечении	
7.	Эволюция архитектуры. Методы гибкой разработки ПО	Рассмотрение факторов эволюции. Метод анализа компромиссов архитектуры (ATAM); Рассмотрение облегченного способа эволюции архитектуры. Рассмотрение архитектуры в разрезе методов, относящихся к гибкой разработке. Рассмотрение практик использования архитектуры ПО. Использование архитектуры для построения облачных технологий.	

5. УЧЕБНО-МЕТОДИЧЕСКОЕ ОБЕСПЕЧЕНИЕ

Университет Иннополис обеспечен необходимым комплектом лицензионного и свободно распространяемого программного обеспечения, в том числе отечественного производства.

Каждый обучающийся в течение всего периода обучения обеспечен индивидуальным неограниченным доступом к электронно-библиотечной системе (электронной библиотеке) и к электронной информационно-образовательной

среде Университета (<https://my.university.innopolis.ru>). Электронно-библиотечная система и электронная информационно-образовательная среда обеспечивают возможность доступа, обучающегося из любой точки, в которой имеется доступ к информационно-телекоммуникационной сети «Интернет» (далее — сеть «Интернет»), как на территории Университета, так и вне его.

Также обучающимся обеспечен доступ (удаленный доступ) к современным профессиональным базам данных и информационным справочным системам.

Перечень учебно-методического обеспечения дисциплины (модуля):

Основная литература:

1. Гунько, А. В. Системное программное обеспечение: конспект лекций / А. В. Гунько. — Новосибирск: Новосибирский государственный технический университет, 2011. — 138 с. — Режим доступа: <http://www.iprbookshop.ru/45020.html> — ЭБС «IPRbooks».
2. Сеницын, С. В. Верификация программного обеспечения: учебное пособие / С. В. Сеницын, Н. Ю. Налютин. — 3-е изд. — Москва: Интернет-Университет Информационных Технологий (ИНТУИТ), Ай Пи Ар Медиа, 2020. — 367 с. — Режим доступа: <https://www.iprbookshop.ru/97540.html> — ЭБС «IPRbooks».

Дополнительная литература:

1. Малявко, А. А. Системное программное обеспечение. Формальные языки и методы трансляции. Часть 1: учебное пособие / А. А. Малявко. — Новосибирск: Новосибирский государственный технический университет, 2010. — 104 с. — Режим доступа: <http://www.iprbookshop.ru/45017.html> — ЭБС «IPRbooks».
2. Перл, И. А., Каленова, О. В. Введение в методологию программной инженерии. — СПб.: Университет ИТМО, 2017. — 53 с. — Режим доступа: <https://reader.lanbook.com/book/136397#3> — ЭБС на платформе «Лань».

Дистанционная поддержка дисциплины (модуля)

Для дистанционной поддержки дисциплины используется система управления образовательным контентом Moodle, развернутая на портале АНО ВО «Университет Иннополис». Во время обучения все студенты получают индивидуальные пароли для входа в систему, в которой размещаются: программа курса, контрольные вопросы, задания, темы лекций с презентационными материалами и т. д.

6. МАТЕРИАЛЬНО-ТЕХНИЧЕСКОЕ ОБЕСПЕЧЕНИЕ

1. Университет располагает материально-технической базой, соответствующей действующим противопожарным правилам и нормам и обеспечивающей проведение всех видов дисциплинарной и междисциплинарной подготовки, практической и научно-исследовательской работ обучающихся, предусмотренных учебным планом.

2. Помещения, которые представляют собой учебные аудитории для проведения занятий лекционного типа, занятий семинарского (практического) типа,

курсового проектирования (выполнения курсовых работ), групповых и индивидуальных консультаций, текущего контроля и промежуточной аттестации, а также помещения для самостоятельной работы и помещения для хранения и профилактического обслуживания учебного оборудования. Помещения укомплектованы специализированной мебелью и техническими средствами обучения, служащими для представления учебной информации большой аудитории.

Изучение дисциплины обеспечивается в учебных аудиториях, оснащенных:

- столами и стульями;
- компьютерной техникой;
- специализированным оборудованием, включая демонстрационное оборудование.

Перечень материально-технического оснащения учебных аудиторий конкретизируется приказом директора или уполномоченного им лица ежегодно и размещается на официальном сайте Университета в информационно-коммуникационной сети «Интернет» в подразделе «Материально-техническое обеспечение и оснащённость образовательного процесса. Доступная среда» раздела «Сведения об образовательной организации» по ссылке: <https://innopolis.university/sveden/objects>.

3. Для проведения занятий лекционного и семинарского (практического) типа предлагаются наборы демонстрационного оборудования и учебно-наглядных пособий, обеспечивающие тематические иллюстрации, соответствующие программе дисциплины (модуля).

4. Лаборатории Университета, в том числе Кибер-физическая лаборатория информационной безопасности «Иннокиберполигон» (<https://engineerschool.innopolis.university/innocyberpolygon>), Цифровая лаборатория искусственного интеллекта «InnoDataHub» (<https://engineerschool.innopolis.university/innodatahub>), Цифровая образовательная песочница для разработки передовых систем «DevOps Playground» (<https://engineerschool.innopolis.university/devops>), оснащены лабораторным и специализированным оборудованием. Перечень материально-технического оснащения и программного обеспечения лабораторий размещается на официальном сайте Университета по ссылке: <https://innopolis.university/sveden/objects>.

5. Помещения для самостоятельной работы обучающихся, в том числе приспособленные для использования инвалидами и лицами с ограниченными возможностями здоровья, оснащены компьютерной техникой с возможностью подключения к сети «Интернет» и обеспечением доступа к в электронную информационно-образовательную среду Университета.

6. Обучающимся предоставляется доступ (в том числе удаленный) к ресурсам информационно-телекоммуникационной сети «Интернет», электронным ресурсам (в том числе электронным библиотечным системам, современным профессиональным базам данных и информационным справочным системам):

№	Ссылка на информационный ресурс	Наименование разработки в электронной форме	Доступность
Электронно-библиотечные системы:			

1.	http://www.iprbookshop.ru	Русскоязычная ЭБС на платформе «IPR Smart». Учебники и учебные пособия для университетов издательства «IPRbooks». Учебники и учебные пособия для лиц с ОВЗ	Индивидуальный неограниченный доступ
2.	http://e.lanbook.com	Русскоязычная ЭБС на платформе «Лань». Учебники и учебные пособия для университетов издательства «Лань». Учебники и учебные пособия для лиц с ОВЗ	Индивидуальный неограниченный доступ
Профессиональные базы данных:			
3.	http://search.ebscohost.com	Платформа EBSCOHost. Доступ к англоязычным полнотекстовым книгам	Индивидуальный неограниченный доступ
4.	https://portal.university.innopolis.ru/reading_hall	Электронный каталог научно-технической библиотеки АНО ВО «Университет Иннополис»	Для студентов, зарегистрированных в системе
5.	https://habr.com	База данных для IT-специалистов	Доступ свободный
6.	https://www.sciencedirect.com	База данных ScienceDirect	Доступ свободный
7.	https://elibrary.ru	Научная электронная библиотека	Доступ свободный
8.	https://link.springer.com	Полнотекстовая англоязычная БД Springer	Доступ из локальной сети УИ. Для удаленного доступа требуется регистрация из локальной сети УИ
9.	https://ar.cnki.net	База Данных Academic Reference. База данных полнотекстовых англоязычных ресурсов по всем академическим дисциплинам, опубликованных в Китае	Доступ из локальной сети УИ
10.	https://www.ams.org/journals	База данных англоязычных полнотекстовых журналов AMS Journals (2017–2022 гг.)	Доступ из локальной сети УИ
11.	https://www.dl.begellhouse.com/collections/6764f0021c05bd10.html	База данных англоязычных полнотекстовых журналов Begell	Доступ из локальной сети УИ
12.	https://saemobilus.sae.org/	База данных англоязычных полнотекстовых журналов SAE International SAE eJournals eBooks	Доступ из локальной сети УИ
13.	https://www.worldscientific.com	База данных англоязычных полнотекстовых журналов WorldScientific 2001–2022 гг.	Доступ из локальной сети УИ
14.	https://www.scitation.org/ebooks	AIPP E-Book Collection I + Collection II. Англоязычная База данных полнотекстовых электронных книг.	Доступ из локальной сети УИ
15.	https://ufn.ru/	Автономная некоммерческая организация Редакция журнала "Успехи физических наук". Электронный журнал на русском языке.	Доступ из локальной сети УИ
16.	https://sk.sagepub.com/books/discipline	SAGE Publications Ltd eBook Collections. Англоязычная База данных полнотекстовых электронных книг	Доступ из локальной сети УИ
17.	https://onlinelibrary.wiley.com/	The Wiley Journals Database Англоязычная База данных полнотекстовых электронных журналов	Доступ из локальной сети УИ
18.	https://www.mathnet.ru/	Полнотекстовая коллекция русскоязычных электронных математических журналов	Доступ из локальной сети УИ
19.	https://quantum-electron.lebedev.ru/arhiv/	Электронная версия журнала «Квантовая электроника»	Доступ из локальной сети УИ

20.	http://journals.rcsi.science/	Полнотекстовая коллекция журналов Российской академии наук включает 140 наименований журналов, охватывающих различные научные специальности.	Доступ из локальной сети УИ
21.	https://www.orbit.com/	База данных патентного поиска, объединяющая информацию о более чем 122 миллионах патентных публикаций, полученную из 120 международных патентных ведомств, включая РосПатент, Всемирную организацию интеллектуальной собственности (ВОИС), Европейскую патентную организацию.	Доступ из локальной сети УИ
Информационные справочные системы:			
22.	https://minobrnauki.gov.ru/	Официальный сайт Министерства науки и высшего образования Российской Федерации	Доступ свободный
23.	https://www.edu.ru/	Федеральный портал «Российское образование»	Доступ свободный
24.	http://window.edu.ru/	Информационная система "Единое окно доступа к образовательным ресурсам"	Доступ свободный
25.	http://school-collection.edu.ru/	Единая коллекция цифровых образовательных ресурсов	Доступ свободный
26.	http://fcior.edu.ru/	Федеральный центр информационно-образовательных ресурсов	Доступ свободный

7. Университет Иннополис обеспечен необходимым комплектом лицензионного и свободно распространяемого программного обеспечения, в том числе отечественного производства, в который входят:

Наименование ПО	Производство	Лицензионное / свободно распространяемое
Операционные системы:		
Microsoft Imagine (Windows Client, Server)	зарубежное	лицензионное
Браузеры:		
Яндекс.Браузер	отечественное	свободно распространяемое
Google Chrome	зарубежное	свободно распространяемое
Офисные приложения:		
Microsoft Imagine (Visio, OneNote)	зарубежное	лицензионное
TeXstudio	зарубежное	свободно распространяемое
Adobe Acrobat Reader	зарубежное	свободно распространяемое
Программное обеспечение для планирования и учета времени:		
Toggle app	зарубежное	свободно распространяемое
Системы управления проектами:		
Microsoft Imagine (Project)	зарубежное	лицензионное
Системы управления базами данных:		
Microsoft Imagine (SQL Server)	зарубежное	лицензионное
Системы резервного копирования (backup):		
Acronis Backup Advanced for HyperV	зарубежное	лицензионное
Справочно-правовые системы:		
КонсультантПлюс: справочно-правовая система	отечественное	лицензионное
Системы управления обучением:		
Система LMS Moodle	зарубежное	свободно распространяемое
Средства антивирусной защиты:		
Kaspersky Endpoint Security для бизнеса Стандартный Russian Edition	отечественное	лицензионное
Среды разработки:		
Visual Studio Code	зарубежное	свободно распространяемое
Bash (Unix shell)	зарубежное	свободно распространяемое

Anaconda	зарубежное	свободно распространяемое
Robotic Operating System	зарубежное	свободно распространяемое
CopelliaSim	зарубежное	свободно распространяемое
Google Colaboratory	зарубежное	свободно распространяемое
Пакеты программных средств и библиотек:		
AutoPsy	зарубежное	свободно распространяемое
Interactive Disassembler (IDA)	зарубежное	свободно распространяемое
Системы управления библиографической информацией:		
Zotero	зарубежное	свободно распространяемое
Сервисы и службы:		
Bind	зарубежное	свободно распространяемое
Docker	зарубежное	свободно распространяемое

7. МЕТОДИЧЕСКИЕ И ОЦЕНОЧНЫЕ МАТЕРИАЛЫ

Задания для семинарских (практических) занятий:

№ п/п	Наименование раздела дисциплины (модуля)	Перечень рассматриваемых тем (вопросов)
1.	Введение в Архитектуру ПО, структуры проектирования	Рассмотрим систему, которая принимает упорядоченный набор строк; каждая строка - упорядоченное множество слов; и каждое слово является упорядоченным набором символов. Любая строка может быть «циклически сдвинута», повторно удаляя первое слово и добавляя его в конце строки. Система выводит список всех круговых сдвигов всех строк в алфавитном порядке. Для решения этой проблемы воспользуйтесь следующим архитектурным решением. Предлагаемая архитектура: Существует четыре основные функции: ввод, циклический сдвиг, алфавитный и выходной. Они координируются как подпрограммы по основной программе - мастер-контроль. Данные передаются между компонентами через общую память (то есть символы, индекс и алфавитный индекс находятся в общем хранилище, к которому могут обращаться все функции).
2.	Драйверы Архитектуры	Рассмотрим систему, которая принимает упорядоченный набор строк; каждая строка - упорядоченное множество слов; и каждое слово является упорядоченным набором символов. Любая строка может быть «циклически сдвинута», повторно удаляя первое слово и добавляя его в конце строки. Система выводит список всех круговых сдвигов всех строк в алфавитном порядке. Для решения этой проблемы воспользуйтесь следующим архитектурным решением. Предлагаемая архитектура: Абстрактный интерфейс к данным. Вместо того, чтобы размещать форматы хранения в вычислительных модулях, доступ к данным осуществляется абстрактно (например, в виде списка или набора). Вычисления вызывают неявно при изменении данных. Таким образом, взаимодействие основано на активной модели данных. Например, действие добавления новой строки в хранилище строк приводит к отправке события в модуль сдвига. Это позволяет создавать круговые сдвиги (в отдельном абстрактном хранилище общих данных). И в свою очередь, приводит к тому, что алфавитный указатель неявно вызывается так, чтобы он мог алфавитировать строки.
3.	Тактики и Паттерны	Рассмотрим систему, которая принимает упорядоченный набор строк; каждая строка - упорядоченное множество слов; и каждое слово является упорядоченным набором символов. Любая строка может быть «циклически

		сдвинута», повторно удаляя первое слово и добавляя его в конце строки. Система выводит список всех круговых сдвигов всех строк в алфавитном порядке. Для решения этой проблемы воспользуйтесь следующим архитектурным решением. Предлагаемая архитектура: Есть четыре фильтра: вход, круговой сдвиг, алфавит и вывод. Каждый фильтр обрабатывает данные и отправляет их следующему фильтру. Управление распределяется: каждый фильтр может запускаться всякий раз, когда у него есть данные для вычисления. Обмен данными между фильтрами строго ограничивается передачей данных по каналам.
4.	Проектирование архитектуры	Используйте структуру, рассмотренную на лабораторном занятии в качестве основы для создания трех новых вариантов системы. Каждая новая система имеет одно или несколько требований. В каждой системе, как можно ближе, придерживайтесь предоставленной структуры (архитектурный шаблон каналов и фильтров). Убедитесь, что вы используете хорошие методы программирования, включая комментарии, описывающие роль и функции любых новых модулей, а также описание того, как вы изменили базовые системные модули. Если вы не будете придерживаться разумной практики программирования, ваши баллы за выполненное задание будут снижены.
5.	Аналитические модели	Используйте существующую структуру, описанную на лабораторном занятии в качестве основы для создания 3 новых систем. Каждая новая система должна иметь одно или несколько требований. Для каждой системы, пожалуйста, придерживайтесь ориентированной на сообщения архитектуры как можно ближе. Убедитесь, что вы используете хорошие методы программирования, включая комментарии, которые указывают, как вы изменили базовую систему. Если вы не следуете разумно хорошей практике программирования, ваши баллы за выполненное задание будут снижены.
6.	Документирование архитектуры. Архитектурные стили	Дайте краткое резюме, в котором приводятся доказательства сильных и слабых сторон предлагаемых архитектур в отношении следующего: Простота изменений в алгоритмах обработки: например, сдвиг строк может выполняться на каждой строке, поскольку он считывается с устройства ввода, на всех строках после их чтения или по требованию, когда для алфавита требуется новый набор сдвинутых линий. Простота изменений в представлении данных: например, строки, слова и символы могут храниться различными способами. Аналогично, круговые сдвиги могут храниться явно или неявно. Производительность: как в пространстве, так и во времени. Что такое планирование в реальном времени и каковы основные проблемы с расписанием? Почему Мюллер утверждает, что производительность конкретной системной функции не может быть получена из фрагмента кода, принадлежащего функции? Объясните, почему вы согласны или не согласны? В чем разница между анализом на основе модели и анализом на основе измерения для анализа надежности и доступности? Какой подход вы использовали бы для анализа доступности сервера данных - почему? Как продвижение доступности в системе влияет на атрибуты качества, ориентированные на изменение качества (например, изменчивость, масштабируемость, расширяемость).
7.	Эволюция архитектуры. Методы гибкой разработки ПО	Опишите ситуацию, когда шаблон, ориентированный на события, может быть лучшим выбором, чем обратного вызова и почему. Опишите ситуацию, когда шаблон обратного вызова может быть лучшим выбором, чем ориентированный на события и почему. Какие качества запрашиваются с помощью чистой клиент-серверной системы? Какие качества запрещены? Какие отличия клиент-серверной системы от 3-уровневой системы?

		Модифицируйте систему, рассмотренную на лабораторном занятии. Краткосрочные приоритеты управления для системы заключаются в следующем: Объедините системы управления запасами Leaf Tech и EEP под одним интерфейсом с постоянным внешним видом. Измените систему, чтобы приложение для доставки также включало продукты Leaf Tech. Обеспечьте аутентификацию для всех пользователей. Следите за всеми действиями входа / выхода (имя пользователя, время входа / выхода). Вам не нужно использовать существующие интерфейсы. Когда вы включаете эти изменения в систему, предположите, что пользователи будут использовать исходную систему до тех пор, пока вы не перенесите их в новые приложения. С точки зрения пользователя, миграция — это событие, один момент времени, когда они используют одно из устаревших приложений, в следующий момент они используют ваше улучшенное приложение. Это включает обеспечение согласованности данных из старого приложения в новое приложение.
--	--	---

Текущий контроль успеваемости обучающихся по дисциплине (модулю):

№ п/п	Наименование раздела дисциплины (модуля)	Форма текущего контроля	Материалы текущего контроля
1.	Введение в Архитектуру ПО, структуры проектирования	Проверка выполнения домашних заданий; Устный / письменный опрос	Устный / письменный опрос: Что такое «вид» и как он отличается от структуры? Что такое «перспектива» и как это влияет на наш анализ структур? Какова роль архитектуры в сокращении рисков проекта?
2.	Драйверы Архитектуры	Проверка выполнения домашних заданий; Устный / письменный опрос; Коллоквиум	Устный / письменный опрос: Что такое драйверы архитектуры? Перечислите и опишите каждый. Какие из архитектурных драйверов имеют наибольшее влияние на дизайн и структуру системы? Каковы шесть частей сценария атрибута качества? Опишите каждую часть. Какова связь между вариантом использования и сценарием атрибута качества? Если вы хотите добавить информацию об атрибутах качества в прецедент, как бы вы это сделали?
3.	Тактики и Паттерны	Проверка выполнения домашних заданий; Устный / письменный опрос	Устный / письменный опрос: Применительно к Многослойной Архитектуры (Layer Architecture) поясните, каковы элементы и отношения в ней? Перечислите три вещи, в которых люди часто ошибаются при документировании Многослойной Архитектуры. Опишите, как модель уровней использует тактики: абстрактные общие службы, инкапсулирование и использование посредника?
4.	Проектирование архитектуры	Проверка выполнения домашних заданий; Устный / письменный опрос; Защита проекта;	Тематика дискуссий: Наличествует утверждение: «Что отличает ограничение от других (даже высокоприоритетных) требований, так это то, что оно не подлежит обсуждению». Должны ли мы руководствоваться этим утверждением во время проектирования? Например, было бы разумно спроектировать систему так, чтобы вначале удовлетворить всем ограничениям, прежде чем

		Коллоквиум	беспокоиться о других требованиях? Задания, в том числе, для групповых проектов: В тематическом исследовании Билса и Хэйри обсуждаются различные методы настройки производительности для серверов IBM Websphere Application. Какие общие архитектурные подходы (выберите 2 или 3) для максимизации производительности ИТ-сервера можно выделить из конкретных технологий, основанных на технологии, представленных в указанном примере?
5.	Аналитические модели	Проверка выполнения домашних заданий; Устный / письменный опрос; Тестирование (письменное или компьютерное)	Тестирование: Подведите итог концепции мышления организации. Сашими – это технология проектирования архитектуры, как способ создания архитектуры в гибком жизненном цикле. Обобщите принципы Сашими. Опишите все сильные и слабые стороны, которые вы видите в этом подходе к архитектурному проектированию. Подведите итог Концентрического подхода. Опишите все сильные и слабые стороны, которые вы видите в этом подходе к архитектурному проектированию.
5.	Документирование архитектуры. Архитектурные стили	Проверка выполнения домашних заданий; Устный / письменный опрос; Коллоквиум	Устный / письменный опрос: Перечислите три категории стилей документации и кратко объясните, какие перспективы предназначены для каждого из них. Соответствует ли архитектурная документация к системам, разработанным с использованием гибких методов или систем, чья архитектура часто изменяется? Вкратце объясните, почему соответствует или нет. В случае, если соответствует, то объясните, как это необходимо адаптировать к рассматриваемому контексту. Как шаблон каналов и фильтров отличается от последовательной структуры? Предположим, что вы хотите определить систему, которая структурирована как канал и фильтр, однако у вас есть только механизмы возврата вызова для работы: Используя только компоненты и разъемы обратного вызова, как вы могли бы имитировать канал? Каковы некоторые проблемы с использованием механизмов обратного вызова для представления систем канал и фильтр?
7.	Эволюция архитектуры. Методы гибкой разработки ПО	Проверка выполнения домашних заданий; Устный / письменный опрос; Эссе; Коллоквиум	Устный / письменный опрос: Своими словами опишите преимущества и недостатки в использовании COTS? Предположим, что вы ограничены использованием платформы Eclipse SOA для разработчиков Java и SOA для новой системы на основе SOA, которую вы создаете для клиента. Какие риски вы могли бы ожидать при использовании этой платформы с открытым исходным кодом? Когда необходимо производить оценку архитектуры и по какой причине? Опишите принцип наименьших привилегий. Как это способствует безопасности в системе? Какое влияние это может оказать на системный дизайн? Опишите типы индикаторов вторжения. Тема для эссе: "Преимущества и недостатки принятия облачных

			вычислений с точки зрения бизнеса".
--	--	--	-------------------------------------

Контрольные вопросы для подготовки к промежуточной аттестации:

№ п/п	Наименование раздела дисциплины (модуля)	Наименование вопроса
1.	Введение в Архитектуру ПО, структуры проектирования	Что такое архитектура программного обеспечения? Основные компоненты архитектуры ПО. Отношение архитектуры ПО к разработке. Что такое "вид", его значение. Основные отличия "вида" от структуры. Роль архитектуры в сокращении рисков проекта.
2.	Драйверы Архитектуры	Как архитектура программного обеспечения связана и чем отличается от системной и корпоративной архитектуры? Виды архитектурных драйверов. Выявление бизнес-целей и контекста. Выявление инженерных целей и разрабатываемого ПО в контексте. Описание основных видов архитектурных драйверов. Шесть частей сценария атрибута качества. Их описание. Связь между вариантом использования и сценарием атрибута качества. Способы добавления информации об атрибутах качества в прецедент.
3.	Тактики и Паттерны	Какие решения принимаются, базируясь на архитектуре? Определение архитектурных тактик. Определение архитектурных паттернов. Виды архитектурных тактик базирующихся на стандартных нефункциональных требованиях. Стратегии использования архитектурных тактик. Виды основных архитектурных паттернов. Стратегии комбинирования тактик совместно с паттернами проектирования. Элементы и отношения в Многослойной Архитектуре (Layer Architecture). Опишите, как модель уровней использует тактики: абстрактные общие службы, инкапсулирование и использование посредника.
4.	Проектирование архитектуры	Учитывая следующее требование: «Система должна иметь возможность обнаруживать ошибки программного обеспечения и автоматически перезапускать соответствующий компонент», - поясните, существуют ли потенциальные проблемы с этим требованием? Кажется ли это разумным требованием или нет? Если нет, опишите проблемы, связанные с этим требованием. Стратегии проектирования; Методы проектирования через атрибуты качества (Attribute-Driven Design Method). Этапы проектирования системы через пример "системы автоматизации заданий". Наличествует утверждение: «Что отличает ограничение от других (даже высокоприоритетных) требований, так это то, что оно не подлежит обсуждению». Стоит ли руководствоваться этим утверждением во время проектирования? Было бы разумно спроектировать систему так, чтобы вначале удовлетворить всем ограничениям, прежде чем беспокоиться о других требованиях? В тематическом исследовании Билса и Хэйри обсуждаются различные методы настройки производительности для серверов IBM Websphere Application. Какие общие архитектурные подходы (выберите 2 или 3) для максимизации производительности ИТ-сервера можно выделить из конкретных технологий, основанных на технологии, представленных в указанном примере?
5.	Аналитические модели	Опишите, как использовались шаблоны автора «Практическое исследование:

		ранняя эксплуатация производственных линий». Каковы были ключевые компромиссы, связанные со структурами, выбранными архитектором? Моделирование архитектур для обеспечения анализа атрибутов качества. Составление контрольных списков атрибутов качества. Изучение техники анализа «обратной стороны конверта» (Back-of-the-Envelope Analysis). Эксперименты, моделирование и прототипирование. Анализ на разных этапах жизненного цикла. Концепция мышления организации. Что такое Сашими в дисциплине? Основные принципы Сашими. Сильные и слабые стороны Сашими. Концентрический подход. Сильные и слабые стороны.
6.	Документирование архитектуры. Архитектурные стили	Можете ли вы охарактеризовать шаблон каналов и фильтров в терминах перспективы, топологического расположения, семантики, качества. Соответствует ли архитектурная документация к системам, разработанным с использованием гибких методов или систем, чья архитектура часто изменяется? Вкратце объясните, почему соответствует или нет. В случае, если соответствует, то объясните, как это необходимо адаптировать к рассматриваемому контексту. Как шаблон каналов и фильтров отличается от последовательной структуры? Предположим, что вы хотите определить систему, которая структурирована как канал и фильтр, однако у вас есть только механизмы возврата вызова для работы: Используя только компоненты и разъемы обратного вызова, как вы могли бы имитировать канал? Каковы некоторые проблемы с использованием механизмов обратного вызова для представления систем канал и фильтр? Сильные и слабые стороны предлагаемых архитектур в отношении следующего: Простота изменений в алгоритмах обработки: например, сдвиг строк может выполняться на каждой строке, поскольку он считывается с устройства ввода, на всех строках после их чтения или по требованию, когда для алфавита требуется новый набор сдвинутых линий. Простота изменений в представлении данных: например, строки, слова и символы могут храниться различными способами. Производительность: как в пространстве, так и во времени. Что такое планирование в реальном времени и каковы основные проблемы с расписанием? Почему Мюллер утверждает, что производительность конкретной системной функции не может быть получена из фрагмента кода, принадлежащего функции? Объясните, почему вы согласны или не согласны? В чем разница между анализом на основе модели и анализом на основе измерения для анализа надежности и доступности? Какой подход вы использовали бы для анализа доступности сервера данных - почему?
7.	Эволюция архитектуры. Методы гибкой разработки ПО	Подумайте о людях, которые прочтут архитектурную документацию. Перечислите три типа заинтересованных сторон и определите конкретно ту информацию, которую им нужно будет извлечь из документа. Какие взгляды, вероятно, содержат эту информацию? В каких условиях клиентский сервер обеспечит лучшую производительность, чем трехуровневая система, и наоборот, при каких условиях трехуровневая система обеспечит лучшую производительность, чем клиентский сервер? Факторы эволюции. Метод анализа компромиссов архитектуры (АТАМ). Облегченный способа эволюции архитектуры. Архитектура в разрезе методов, относящихся к гибкой разработке. Рассмотрение практик использования архитектуры ПО. Использование архитектуры для построения облачных технологий. Преимущества и недостатки в использовании COTS?

		Преимущества и недостатки принятия облачных вычислений с точки зрения бизнеса. Типы индикаторов вторжения. Принцип наименьших привилегий.
--	--	---

Вопросы/Задания к промежуточной аттестации в устной/письменной форме:

1. Что такое архитектура программного обеспечения?
2. Основные компоненты архитектуры ПО.
3. Как архитектура программного обеспечения связана и чем отличается от системной и корпоративной архитектуры?
4. Какие решения принимаются, базируясь на архитектуре?
5. Опишите, как модель уровней использует тактики: абстрактные общие службы, инкапсулирование и использование посредника.
6. Соответствует ли архитектурная документация к системам, разработанным с использованием гибких методов или систем, чья архитектура часто изменяется? Вкратце объясните, почему соответствует или нет. В случае, если соответствует, то объясните, как это необходимо адаптировать к рассматриваемому контексту.
7. Как шаблон каналов и фильтров отличается от последовательной структуры?
8. Предположим, что вы хотите определить систему, которая структурирована как канал и фильтр, однако у вас есть только механизмы возврата вызова для работы. Используя только компоненты и разъемы обратного вызова, как вы могли бы имитировать канал?
9. Каковы некоторые проблемы с использованием механизмов обратного вызова для представления систем канал и фильтр?
10. Что такое планирование в реальном времени и каковы основные проблемы с расписанием?
11. Почему Мюллер утверждает, что производительность конкретной системной функции не может быть получена из фрагмента кода, принадлежащего функции? Объясните, почему вы согласны или не согласны?
12. В чем разница между анализом на основе модели и анализом на основе измерения для анализа надежности и доступности? Какой подход вы использовали бы для анализа доступности сервера данных? Почему?

Критерии оценивания сформированности компетенций

Оценивание уровня учебных достижений, обучающихся по дисциплине (модулю), осуществляется в виде текущего контроля успеваемости и промежуточной аттестации.

В рамках внутренней системы оценки качества образовательной деятельности по программе обучающимся предоставляется возможность оценивания условий, содержания, организации и качества образовательного процесса по данной дисциплине (модулю).

Промежуточная аттестация по дисциплине (модулю) осуществляется в форме экзамена, при этом проводится оценка компетенций, сформированных по дисциплине.

Система оценивания результатов обучения по дисциплине (модулю)

Знания, умения и навыки обучающихся при промежуточной аттестации определяются в следующей форме: «отлично» (А), «хорошо» (В), «удовлетворительно» (С), «неудовлетворительно» (D).

Критерии оценивания результатов обучения по дисциплине (модулю)

Результат обучения по дисциплине (модулю)	Общая характеристика результата обучения по дисциплине (модулю)
А «Отлично»	Обучающийся владеет знаниями предмета в полном объеме рабочей программы, достаточно глубоко осмысливает дисциплину; самостоятельно, в логической последовательности и исчерпывающе отвечает на все вопросы, подчеркивая при этом самое существенное, умеет анализировать, сравнивать, классифицировать, обобщать, конкретизировать и систематизировать изученный материал, выделять в нем главное: устанавливать причинно-следственные связи; четко формирует ответы, свободно читает результаты анализов и других исследований и решает ситуационные задачи повышенной сложности; хорошо знаком с основной литературой и методами исследования большого объема, необходимым для практической деятельности; увязывает теоретические аспекты предмета с задачами практического значения.
В «Хорошо»	Обучающийся владеет знаниями предмета практически в полном объеме рабочей программы; самостоятельно, в логической последовательности и исчерпывающе отвечает на все вопросы, подчеркивая при этом самое существенное, умеет анализировать, сравнивать, классифицировать, обобщать, конкретизировать и систематизировать изученный материал, выделять в нем главное: устанавливать причинно-следственные связи; четко формирует ответы, свободно читает результаты анализов и других исследований и решает ситуационные задачи повышенной сложности; хорошо знаком с основной литературой и методами исследования большого объема, необходимым для практической деятельности; увязывает теоретические аспекты предмета с задачами практического значения.
С «Удовлетворительно»	Обучающийся владеет основным объемом знаний по дисциплине; проявляет затруднения в самостоятельных ответах, оперирует неточными формулировками; в процессе ответов допускаются ошибки по существу вопросов. Обучающийся способен решать лишь наиболее легкие задачи, владеет только обязательным минимумом методов исследований.
D «Неудовлетворительно»	Обучающийся не освоил обязательного минимума знаний предмета, не способен ответить на вопросы даже при дополнительных наводящих вопросах преподавателя.

Методические указания для обучающихся по освоению дисциплины (модуля)

Вид учебных занятий/деятельности	Деятельность обучающегося
Лекция	Написание конспекта лекций: кратко, схематично, последовательно фиксировать основные положения лекции, выводы, формулировки, обобщения; пометить важные мысли, выделять ключевые слова, термины. Обозначить вопросы, термины или другой материал, который вызывает трудности, пометить и попытаться найти ответ в рекомендуемой литературе. Если самостоятельно не удастся разобраться в материале, необходимо сформулировать вопрос и задать преподавателю на консультации, во время семинарского (практического) занятия.
Практическое (семинарское) занятие	При подготовке к семинарскому (практическому) занятию необходимо проработать материалы лекций, основной и дополнительной литературы по заданной теме. На

	основании обработанной информации постараться сформировать собственное мнение по выносимой на обсуждение тематике. Обосновать его аргументами, сформировать список источников, подтверждающих его. Во время семинарского (практического) занятия активно участвовать в обсуждении вопросов, высказывать аргументированную точку зрения на проблемные вопросы. Приводить примеры из источниковой базы и научной и/или исследовательской литературы.
Устный/письменный опрос	Отвечать, максимально полно, логично и структурировано, на поставленный вопрос. Основная цель – показать всю глубину знаний по конкретной теме или ее части.
Эссе	Написание прозаического сочинения небольшого объема и свободной композиции, выражающего индивидуальные впечатления и соображения по конкретному поводу или вопросу и заведомо не претендующего на определяющую или исчерпывающую трактовку предмета. При работе над эссе следует четко и грамотно формулировать мысли, структурировать информацию, использовать основные понятия, выделять причинно-следственные связи. Как правило эссе имеет следующую структуру: вступление, тезис и аргументация его, заключение. В качестве аргументов могут выступать исторические факты, явления общественной жизни, события, жизненные ситуации и жизненный опыт, научные доказательства, ссылки на мнение ученых и др.
Подготовка к промежуточной аттестации	При подготовке к промежуточной аттестации необходимо проработать вопросы по темам, которые рекомендуются для самостоятельной подготовки. При возникновении затруднений с ответами следует ориентироваться на конспекты лекций, семинаров, рекомендуемую литературу, материалы электронных и информационных справочных ресурсов, статей. Если тема вызывает затруднение, четко сформулировать проблемный вопрос и задать его преподавателю.
Практические (лабораторные) занятия	Практические занятия предназначены прежде всего для разбора отдельных сложных положений, тренировки аналитических навыков, а также для развития коммуникационных навыков. Поэтому на практических занятиях необходимо участвовать в тех формах обсуждения материала, которые предлагает преподаватель: отвечать на вопросы преподавателя, дополнять ответы других студентов, приводить примеры, задавать вопросы другим выступающим, обсуждать вопросы и выполнять задания в группах. Работа на практических занятиях подразумевает домашнюю подготовку и активную умственную работу на самом занятии. Работа на практических занятиях в форме устного опроса заключается прежде всего в тренировке навыков применять теоретические положения к самому разнообразному материалу. В ходе практических занятий студенты работают в группах для обсуждения предлагаемых вопросов.
Самостоятельная работа	Самостоятельная работа состоит из следующих частей: 1) чтение учебной, справочной, научной литературы; 2) повторение материала лекций; 3) составление планов устных выступлений; 4) подготовка видеопрезентации. При чтении учебной литературы нужно разграничивать для себя материал на отдельные проблемы, концепции, идеи. Учебную литературу можно найти в электронных библиотечных системах, на которые подписан АНО Университет Иннополис.
Дискуссия	Публичное обсуждение спорного вопроса, проблемы. Каждая сторона должна оппонировать мнение собеседника, аргументируя свою позицию.
Тестирование (устное/письменное)	При подготовке к тестированию необходимо проработать материалы лекций, семинаров, основной и дополнительной литературы по заданной теме. Основная цель тестирования – показать уровень сформированности знаний по конкретной теме или ее части.
Выполнение домашних заданий и групповых проектов	Для выполнения домашних заданий и групповых проектов необходимо получить формулировку задания от преподавателя и убедиться в понимании задания. При выполнении домашних заданий и групповых проектов необходимо проработать

	материалы лекций, основной и дополнительной литературы по заданной теме.
--	--

Для подготовки к занятиям рекомендуется использовать представленные источники в электронных форматах и дополнительную литературу.

8. ДОПОЛНИТЕЛЬНЫЕ СВЕДЕНИЯ

Данная программа и/или ее фрагменты не могут быть использованы другими образовательными организациями без соответствующего разрешения АНО ВО «Университет Иннополис».