

Документ подписан простой электронной подписью

Информация о владельце:

МИНИСТЕРСТВО СЕЛЬСКОГО ХОЗЯЙСТВА РОССИЙСКОЙ ФЕДЕРА-
ЦИИ

ФИО: Хоружий Иннокентий Иванович АПК

Должность: Директор института экономики и управления АПК

Дата подписи: 16.01.2025

Уникальный программный ключ:

1e90b132d9b040c6b7589160b015dddf2cb1e6a9



(ФГБОУ ВО РГАУ - МСХА имени К.А. Тимирязева)

Институт экономики и управления АПК
Кафедра прикладной информатики

УТВЕРЖДАЮ:
Директор института
экономики и управления АПК
Л.И. Хоружий
“ 28 ” 08 2025 г.



РАБОЧАЯ ПРОГРАММА ДИСЦИПЛИНЫ

Б1.В.08 Объектно-ориентированное проектирование и программирование

для подготовки бакалавров

ФГОС ВО

Направление: 09.03.03 Прикладная информатика

Направленность: Системы искусственного интеллекта

Курс 4

Семестр 8

Форма обучения: очная

Год начала подготовки: 2025

Москва, 2025

Разработчик (и): Журавлев М.В., к.т.н., доцент
(ФИО, ученая степень, ученое звание)


(подпись)

(ФИО, ученая степень, ученое звание)

(подпись)

«28» августа 2025 г.

Рецензент: Демичев В.А., к.э.н., доцент
(ФИО, ученая степень, ученое звание)



(подпись)

«28» августа 2025 г.

Программа составлена в соответствии с требованиями ФГОС ВО, профессионального стандарта и учебного плана по направлению подготовки 09.03.03 «Прикладная информатика»

Программа обсуждена на заседании кафедры прикладной информатики
протокол №1 от «28» августа 2025 г.

И.о. заведующего кафедрой
прикладной информатики Худякова Е.В., д.э.н., профессор
(ФИО, ученая степень, ученое звание)



(подпись)

«28» августа 2025 г.

Согласовано:

Председатель учебно-методической комиссии
института экономики и управления АПК
Гупалова Т.Н., к.э.н., доцент
(ФИО, ученая степень, ученое звание)



(подпись)

«28» августа 2025 г.

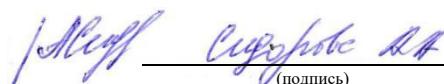
И.о. заведующего выпускающей кафедрой
прикладной информатики Худякова Е.В., д.э.н., профессор
(ФИО, ученая степень, ученое звание)



(подпись)

«28» августа 2025 г.

Заведующий отделом комплектования ЦНБ


(подпись)

Содержание

1. ЦЕЛЬ ОСВОЕНИЯ ДИСЦИПЛИНЫ	4
2. МЕСТО ДИСЦИПЛИНЫ В УЧЕБНОМ ПРОЦЕССЕ	5
3. ПЕРЕЧЕНЬ ПЛАНИРУЕМЫХ РЕЗУЛЬТАТОВ ОБУЧЕНИЯ ПО ДИСЦИПЛИНЕ, СООТНЕСЕННЫХ С ПЛАНИРУЕМЫМИ РЕЗУЛЬТАТАМИ ОСВОЕНИЯ.....	5
ОБРАЗОВАТЕЛЬНОЙ ПРОГРАММЫ	5
4. СТРУКТУРА И СОДЕРЖАНИЕ ДИСЦИПЛИНЫ	8
4.1 РАСПРЕДЕЛЕНИЕ ТРУДОЁМКОСТИ ДИСЦИПЛИНЫ ПО ВИДАМ РАБОТ ПО СЕМЕСТРАМ.....	8
4.2. СОДЕРЖАНИЕ ДИСЦИПЛИНЫ	8
4.3. ЛЕКЦИИ/ПРАКТИЧЕСКИЕ ЗАНЯТИЯ.....	20
4.4. ПЕРЕЧЕНЬ ВОПРОСОВ ДЛЯ САМОСТОЯТЕЛЬНОГО ИЗУЧЕНИЯ ДИСЦИПЛИНЫ....	25
5. ОБРАЗОВАТЕЛЬНЫЕ ТЕХНОЛОГИИ	27
6. ТЕКУЩИЙ КОНТРОЛЬ УСПЕВАЕМОСТИ И ПРОМЕЖУТОЧНАЯ АТТЕСТАЦИЯ ПО ИТОГАМ ОСВОЕНИЯ ДИСЦИПЛИНЫ	30
6.1. ТИПОВЫЕ КОНТРОЛЬНЫЕ ЗАДАНИЯ ИЛИ ИНЫЕ МАТЕРИАЛЫ, НЕОБХОДИМЫЕ ДЛЯ ОЦЕНКИ ЗНАНИЙ, УМЕНИЙ И НАВЫКОВ И (ИЛИ) ОПЫТА ДЕЯТЕЛЬНОСТИ	30
6.2. ОПИСАНИЕ ПОКАЗАТЕЛЕЙ И КРИТЕРИЕВ КОНТРОЛЯ УСПЕВАЕМОСТИ, ОПИСАНИЕ ШКАЛ ОЦЕНИВАНИЯ	60
7. УЧЕБНО-МЕТОДИЧЕСКОЕ И ИНФОРМАЦИОННОЕ ОБЕСПЕЧЕНИЕ ДИСЦИПЛИНЫ	61
7.1 Основная литература	61
7.2. Дополнительная литература.....	62
7.3 Методические указания, рекомендации и другие материалы к занятиям.....	62
7.4 Перечень статей А* и статей журналов белого списка для изучения .	63
8. ПЕРЕЧЕНЬ РЕСУРСОВ ИНФОРМАЦИОННО-ТЕЛЕКОММУНИКАЦИОННОЙ СЕТИ «ИНТЕРНЕТ», НЕОБХОДИМЫХ ДЛЯ ОСВОЕНИЯ ДИСЦИПЛИНЫ	64
9. ПЕРЕЧЕНЬ ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ.....	64
10. ОПИСАНИЕ МАТЕРИАЛЬНО-ТЕХНИЧЕСКОЙ БАЗЫ, НЕОБХОДИМОЙ ДЛЯ ОСУЩЕСТВЛЕНИЯ ОБРАЗОВАТЕЛЬНОГО ПРОЦЕССА ПО ДИСЦИПЛИНЕ	68
11. МЕТОДИЧЕСКИЕ РЕКОМЕНДАЦИИ СТУДЕНТАМ ПО ОСВОЕНИЮ ДИСЦИПЛИНЫ	69
<i>Виды и формы отработки пропущенных занятий</i>	70
12. МЕТОДИЧЕСКИЕ РЕКОМЕНДАЦИИ ПРЕПОДАВАТЕЛЯМ ПО ОРГАНИЗАЦИИ ОБУЧЕНИЯ ПО ДИСЦИПЛИНЕ	70

Аннотация

рабочей программы учебной дисциплины Б1.В.08 «Объектно-ориентированное проектирование и программирование» для подготовки бакалавров по направлению 09.03.03 «Прикладная информатика» направленность «Системы искусственного интеллекта»

Цель освоения дисциплины: приобретение системных знаний по объектно-ориентированной методологии и технологии разработки программного обеспечения, формирование практических навыков применения инструментов, реализующих объектный подход при разработке многопоточных приложений для решения задач в области искусственного интеллекта.

Место дисциплины в учебном плане: дисциплина включена в часть учебного плана, формируемую участниками образовательных отношений, по направлению подготовки 09.03.03 «Прикладная информатика».

Требования к результатам освоения дисциплины: в результате освоения дисциплины формируются следующие компетенции (индикаторы): ПК-16 (ПЛ-3).1; ПК-16 (ПЛ-3).2; ПК-16 (ПЛ-3).3.

Краткое содержание дисциплины: Базовые принципы объектно-ориентированной методологии проектирования автоматизированных информационных систем, основные элементы объектной модели, их применение при разработке программ, базовые принципы объектно-ориентированной технологии программирования, применение для моделирования и проектирования автоматизированных информационных систем унифицированного языка моделирования UML, жизненный цикл программных продуктов, виды и методики тестирования программ, критерии качества программных продуктов, испытания и оценка качества программного обеспечения, задачи из области искусственного интеллекта, для которых целесообразно распараллеливание вычислительных процессов, формирование практических навыков применения принципов объектно-ориентированного программирования и языка программирования C++ при разработке многопоточных приложений для решения задач в области искусственного интеллекта.

Общая трудоёмкость дисциплины: 144 часов/4 зачётные единицы, в том числе 4 часа практической подготовки.

Промежуточный контроль: зачёт с оценкой.

1. Цель освоения дисциплины

Приобретение системных знаний по объектно-ориентированной методологии и технологии разработки программного обеспечения, формирование практических навыков применения инструментов, реализующих объектный подход при разработке многопоточных приложений для решения задач в области искусственного интеллекта.

2. Место дисциплины в учебном процессе

Дисциплина «Объектно-ориентированное проектирование и программирование» включена в часть учебного плана, формируемую участниками образовательных отношений, и реализуется в соответствии с требованиями ФГОС ВО и учебного плана по направлению 09.03.03 «Прикладная информатика».

Предшествующими курсами, на которых непосредственно базируется дисциплина «Объектно-ориентированное проектирование и программирование» являются: Алгоритмизация и программирование, Разработка программного обеспечения для мобильных устройств, Проектирование пользовательских интерфейсов ЭИС АПК, Проектирование информационных систем.

Дисциплина «Объектно-ориентированное проектирование и программирование» является основополагающей для изучения следующих дисциплин: Проектный практикум, Производственная практика, Производственная практика в виде стажировки, Выполнение и защита выпускной квалификационной работы.

Особенностью дисциплины «Объектно-ориентированное проектирование и программирование» является изучение для применения в прикладном программировании возможностей современных объектно-ориентированных средств на примере MS Visual Studio 2019, объектно-ориентированной методологии и технологии разработки программного обеспечения на языке программирования C++: инкапсуляции, иерархии, абстрагирования, полиморфизма, модульности, параллелизма, устойчивости при разработке высокопроизводительных программ, реализующих параллельные вычислительные процессы при решении задач из области искусственного интеллекта.

Рабочая программа дисциплины «Объектно-ориентированное проектирование и программирование» для инвалидов и лиц с ограниченными возможностями здоровья разрабатывается индивидуально с учетом особенностей психофизического развития, индивидуальных возможностей и состояния здоровья таких обучающихся.

3. Перечень планируемых результатов обучения по дисциплине, соотнесенных с планируемыми результатами освоения образовательной программы

Изучение данной учебной дисциплины направлено на формирование у обучающихся компетенций, представленных в таблице 1.

Таблица 1 - Требования к результатам освоения учебной дисциплины (профессиональные компетенции)

№ п/ п	Компе- тенции	Содержание компетенции (или ее части)	Индикатор достижения компетенции и его содер- жание	В результате изучения учебной дисциплины обучающиеся должны:		
				знатъ	уметь	владеть
1	ПК-16 (PL-3)	Способен приме- нять языки про- граммирования C/C++ для решения задач в области ИИ (продвинутый уро- вень)	ПК-16 (PL-3).1 Разраба- тывает и отлаживает эфек- тивные многопоточные ре- шения на C++, тестирует, испытывает и оценивает ка- чество таких решений	Знает методы, инстру- менты и библиотеки для разработки эффективных многопоточных про- граммных решений на C++, а также стандарты, регламентирующие про- цессы тестирования, ис- пытаний и оценки каче- ства программного обес- печения	Умеет разрабатывать мно- гопоточные программные решения в области искус- ственного интеллекта, проводить их тестирова- ние, испытание, оценку качества, грамотно приме- ня员 атомарные операции и механизм блокировок для решения проблем одно- временного доступа к дан- ным из нескольких пото- ков	Владеет навыками раз- работки, отладки, те- стирования, испытания, оценки качества и про- изводительности мно- гопоточных програм- мных решений на C++, навыками устранения найденных узких мест
			ПК-16 (PL-3).2 Разраба- тывает и отлаживает системы ИИ на C++ под конкретные аппаратные платформы с ограничениями по вычис- лительной мощности, в том числе для встроенных си- стем	Знает библиотеки ин- струментальных средств, методы и сред- ства разработки, отладки программных продуктов на C++ с учётом их реа- лизации на конкретных аппаратных платформах	Умеет применять объ- ектно-ориентированную методологию и техноло- гию, методы оптимизации моделей, язык программи- рования C++ и библиотеки инструментальной среды для разработки и отладки систем искусственного интеллекта с учётом вы- числительных мощностей, на которых будет произво- диться их развёртывание и эксплуатация	Владеет навыками раз- работки и отладки си- стем искусственного интеллекта для кон- кретных аппаратных платформ с ограниче- ниями по вычислитель- ной мощности, навы- ками подбора библио- тек, методов и инстру- ментов, соответствую- щих решаемым задачам

		<p>ПК-16 (PL-3).3 Разрабатывает и отлаживает решения на C++, использующие GPU и FPGA для массовой параллелизации вычислений в рамках общей системы ИИ, с применением как готовых решений, так и разработкой своих</p>	<p>Знает методы оптимизации моделей параллельных вычислений и готовые инструменты для их реализации, средства отладки и профилирования кода</p>	<p>Умеет при разработке систем искусственного интеллекта на C++ использовать средства отладки и профилирования кода, возможности GPU и FPGA для массовой параллелизации вычислений</p>	<p>Владеет методами оптимизации моделей и параллельных вычислительных процессов при разработке систем искусственного интеллекта, методами и инструментами отладки и профилирования кода, что позволяет находить участки кода, ограничивающие производительность системы</p>
--	--	---	---	--	---

4. Структура и содержание дисциплины

4.1 Распределение трудоёмкости дисциплины по видам работ по семестрам

Общая трудоёмкость дисциплины составляет **4 зач.ед. (144 часа)**, включая 4 часа практической подготовки, их распределение по видам работ в семестре представлено в таблице 2.

Таблица 2 - Распределение трудоёмкости дисциплины по видам работ по семестрам

Вид учебной работы	Трудоёмкость		
	час./*	в т.ч. по семестрам	
		8/*	
Общая трудоёмкость дисциплины по учебному плану	144/4		144/4
1. Контактная работа:	50,35/4		50,35/4
Аудиторная работа	50,35/4		50,35/4
<i>в том числе:</i>			
лекции (Л)	16		16
практические занятия (ПЗ)	34/4		34/4
консультации перед экзаменом			
контактная работа на промежуточном контроле (КРА)	0,35		0,35
2. Самостоятельная работа (СРС)	93,65		93,65
<i>самостоятельное изучение разделов, самоподготовка (проработка и повторение лекционного материала и материала учебников и учебных пособий, подготовка к практическим занятиям и т.д.)</i>	84,65		84,65
Подготовка к зачёту с оценкой	9		9
Вид промежуточного контроля:		Zачёт с оценкой	

* в том числе практическая подготовка

4.2. Содержание дисциплины

Таблица 3 - Тематический план учебной дисциплины

Наименование разделов и тем дисциплин (укрупнённо)	Всего	Аудиторная работа			Внеаудиторная работа СР
		Л	ПЗ	ПКР	
Раздел 1. Объектно-ориентированная методология и технология разработки программного обеспечения	18,5	2,5	4		12
Тема 1.1. Методология объектно-ориентированного подхода разработки программного обеспечения	5	1			4
Тема 1.2. Базовые положения объектно-ориентированной технологии программирования	4,5	0,5			4

применительно к языку программирования C++					
Тема 1.3. Язык унифицированного моделирования UML, его применение при проектировании автоматизированных информационных систем	9	1	4		4
Раздел 2. Жизненный цикл программных продуктов. Виды работ, выполняемых на отдельных его этапах.	27,5	4,5	6		17
Тема 2.1. Стандарты, регламентирующие жизненный цикл программного обеспечения и работы, выполняемые на отдельных его этапах	3,5	0,5			3
Тема 2.2. Содержание стандартов ГОСТ Р 59793-2021 «Автоматизированные системы. Стадии создания», ГОСТ Р ИСО/МЭК 12207-2010 «Информационная технология. Системная и программная инженерия. Процессы жизненного цикла программных средств»	4	1			3
Тема 2.3. Содержание стандарта ГОСТ Р ИСО/МЭК 25010-2015 «Информационные технологии. Системная и программная инженерия. Требования и оценка качества систем и программного обеспечения (SQuaRE). Модели качества систем и программных продуктов», его использование для оценки качества программного обеспечения	6	1	2		3
Тема 2.4. Определение порядка верификации, валидации и испытаний автоматизированных информационных систем в стандарте ГОСТ Р 59792-2021 «Информационные технологии. Комплекс стандартов на автоматизированные системы. Виды испытаний автоматизированных систем»	7	1	2		4
Тема 2.5. Виды и методики проведения тестирования программного обеспечения	7	1	2		4
Раздел 3. Возможности параллельной обработки в среде MS Visual Studio 2019	45	5	10		30
Тема 3.1. Параллелизм задач в среде MS Visual Studio 2019	9	1	2		6
Тема 3.2. Лямбда-выражения и их применение в параллельных вычислительных процессах	9	1	2		6
Тема 3.3. Библиотека параллельных шаблонов (PPL) MS Visual Studio 2019	9	1	2		6
Тема 3.4. Обработка исключений в среде MS Visual Studio 2019 с параллелизмом	9	1	2		6
Тема 3.5. Рекомендации по работе со средой выполнения программ Visual Studio 2019 с параллелизмом	9	1	2		6
Раздел 4. Задачи из области искусственного интеллекта, для решения которых целесообразно распараллеливание вычислительных процессов и применение многопоточности	43,65	4	14 / 4		25,65

Тема 4.1. Создание сети обработки изображений	11	1	4/ 1		6
Тема 4.2. Программирование неиерархических и иерархических методов кластер-анализа данных с применением параллелизации задач	12,65	1	4/ 1		7,65
Тема 4.3. Применение параллелизации вычислительных процессов для генерации решений в системах поддержки принятия решений	11	1	4/ 1		6
Тема 4.4. Формирование базы знаний и ранжирование альтернатив с применением параллельных вычислительных процессов при разработке экспертных систем	9	1	2/ 1		6
Контактная работа на промежуточном контроле (КРА)	0,35	0	0	0,35	0
Подготовка к зачёту с оценкой	9	0	0	0	9
Всего за 8 семестр	144	16	34/ 4	0,35	93,65

Раздел 1. Объектно-ориентированная методология и технология разработки программного обеспечения

Тема 1.1. Методология объектно-ориентированного подхода разработки программного обеспечения

Базовые принципы методологии объектно-ориентированного подхода: инкапсуляция, модульность, иерархия, параллелизм, устойчивость, абстрагирование. Основное понятие объектно-ориентированного подхода – класс. Класс – тип данных объекта. Структура класса, его составные элементы: поля, методы, конструкторы, деструкторы, встроенные классы, интерфейсы. Инструменты проектирования автоматизированных систем с применением объектно-ориентированного подхода: UML диаграммы, case-средства, поддерживающие язык унифицированного моделирования UML.

Тема 1.2. Базовые положения объектно-ориентированной технологии программирования применительно к языку программирования C++

Технологии программирования, применяемые в рамках объектно-ориентированного подхода: объектно-ориентированное программирование, модульное программирование. Основные типы модулей программ, разработанных на объектно-ориентированном языке программирования C++: классы, пакеты, библиотеки. Наличие обширных библиотек в объектно-ориентированной среде программирования MS Visual Studio 2019, возможности их использования.

Базовые принципы объектно-ориентированной технологии программирования: инкапсуляция, наследование, полиморфизм. Реализация механизма инкапсуляции за счёт применения модификаторов доступа классов в MS Visual Studio 2019. Перегрузка методов и конструкторов в объектно-ориентированных средах и в MS Visual Studio 2019. Правила реализации механизма полиморфизма. Преимущества использования полиморфизма. Наследование в объектно-ориентированных средах и в MS Visual Studio 2019, его достоинства.

Единичное и множественное наследование в различных объектно-ориентированных языках программирования. Единичное наследование в языке C++, его возможности при разработке программ.

Тема 1.3. Язык унифицированного моделирования UML, его применение при проектировании автоматизированных информационных систем

Моделирование и проектирования программных продуктов с применением унифицированного языка моделирования UML, включающего UML-диаграммы вариантов использования, UML-диаграммы классов, UML-диаграммы взаимодействия, UML-диаграммы деятельности, UML-диаграммы состояний, UML-диаграммы компонентов, UML-диаграммы размещения, UML-диаграммы последовательности, кооперативные UML-диаграммы, UML-диаграммы реализации.

Применение графических case-средств Visual Paradigm Онлайн (VP Онлайн) Free Edition, Diagrams.net для разработки UML-диаграмм.

Раздел 2. Жизненный цикл программных продуктов. Виды работ, выполняемых на отдельных его этапах.

Тема 2.1. Стандарты, регламентирующие жизненный цикл программного обеспечения и работы, выполняемые на отдельных его этапах

Модели жизненного цикла программного обеспечения: каскадная модель, модель с контролем на промежуточных стадия, спиральная модель, V-образная модель. Стадии проектирования автоматизированных информационных систем, отражённые в стандарте ГОСТ Р 59793-2021 «Автоматизированные системы. Стадии создания». Процессы жизненного цикла программного обеспечения, указанные в стандарте ГОСТ Р ИСО/МЭК 12207-2010 «Информационная технология. Системная и программная инженерия. Процессы жизненного цикла программных средств».

Стандарт ГОСТ Р ИСО/МЭК 90003-2014 «Разработка программных продуктов. Руководящие указания по применению ИСО 9001: 2008 при разработке программных продуктов», его применение для управления процессом разработки и оценки качества программного обеспечения.

Стандарты, регламентирующие разработку документации на программные продукты: ГОСТ Р ИСО/МЭК 15910-2002 «Процесс создания документации пользователя программного средства», ГОСТ Р 59795-2021 «Требования к содержанию документов», ГОСТ 34.201-2020 «Информационные технологии. Комплекс стандартов на автоматизированные системы. Виды, комплектность и обозначение документов при создании автоматизированных систем». Стандарты, определяющие порядок верификации, валидации и испытаний автоматизированных информационных систем - ГОСТ Р 59792-2021 «Информационные технологии. Комплекс стандартов на автоматизированные системы. Виды испытаний автоматизированных систем».

Тема 2.2. Содержание стандартов ГОСТ Р 59793-2021 «Автоматизированные системы. Стадии создания», ГОСТ Р ИСО/МЭК 12207-2010 «Информационная технология. Системная и программная инженерия. Процессы жизненного цикла программных средств»

Стадии жизненного цикла программного обеспечения, определённые в стандарте ГОСТ Р 59793-2021 «Автоматизированные системы. Стадии создания»: формирование требований к автоматизированной системе, разработка концепции автоматизированной системы, техническое задание, эскизный проект, технический проект, рабочая документация, ввод в действие, сопровождение автоматизированной системы.

Процессы жизненного цикла программного обеспечения, указанные в стандарте ГОСТ Р ИСО/МЭК 12207-2010 «Информационная технология. Системная и программная инженерия. Процессы жизненного цикла программных средств»: процессы соглашения, процессы проекта, технические процессы, процессы организационного обеспечения проекта, процессы реализации программного средства, процессы поддержки программного средства, процессы повторного применения программных средств.

Тема 2.3. Содержание стандарта ГОСТ Р ИСО/ МЭК 25010-2015 «Информационные технологии. Системная и программная инженерия. Требования и оценка качества систем и программного обеспечения (SQuaRE). Модели качества систем и программных продуктов», его использование для оценки качества программного обеспечения

Модели качества программного обеспечения. Структура, используемая в моделях качества программных продуктов, которая позволяет представлять их в виде набора характеристик или свойств. Показатели качества автоматизированной системы, программного продукта: функциональная пригодность, уровень производительности, совместимость, удобство использования, надежность, защищённость, сопровождаемость, переносимость. Характеристики показателя качества «Функциональная пригодность»: функциональная полнота, функциональная корректность, функциональная целесообразность. Характеристики показателя качества «Уровень производительности»: временные характеристики, использование ресурсов, потенциальные возможности. Характеристики показателя качества «Совместимость»: существование, интероперабельность. Характеристики показателя качества «Удобство использования»: определимость пригодности, изучаемость, управляемость, защищённость от ошибки пользователя, эстетика пользовательского интерфейса, доступность. Характеристики показателя качества «Надёжность»: завершённость, готовность, отказоустойчивость, восстанавливаемость. Характеристики показателя качества «Защищённость»: конфиденциальность, целостность, неподдельность, отслеживаемость, подлинность. Характеристики показателя качества «Сопровождаемость»: модульность, возможность многократного использования, анализируемость, модифицируемость, тестируемость. Характеристики показателя качества «Переносимость»: адаптируемость, устанавливаемость,

взаимозаменяемость. Цели использования моделей качества программного обеспечения. Требования пользователей при оценке качества программных продуктов: эффективность, производительность, удовлетворённость, свобода от рисков, надёжность, защищённость, покрытие контекста, изучаемость, доступность.

Тема 2.4. Определение порядка верификации, валидации и испытаний автоматизированных информационных систем в стандарте ГОСТ Р 59792-2021 «Информационные технологии. Комплекс стандартов на автоматизированные системы. Виды испытаний автоматизированных систем»

Цели и задачи верификации программного обеспечения. Методы верификации программного обеспечения: экспертиза, анализ требований, ревью кода, формальные методы (методы Флойда и Хоора), статический анализ кода, динамические методы, синтетические методы, моделирование и прототипирование.

Этапы проведения верификации программного обеспечения: верификация требований к программному обеспечению, верификация проекта программного обеспечения, верификация кода программных модулей, верификация интегрированного программного обеспечения.

Цель и задачи валидации программного обеспечения. Методы валидации программ: функциональное тестирование, интеграционное тестирование, приемочное тестирование, usability тестирование, нагрузочное и стресс-тестирование, системное тестирование.

Виды испытаний автоматизированных систем: предварительные, опытная эксплуатация, приёмочные.

Автономные и комплексные испытания автоматизированных систем. Документация, сопровождающая автономные испытания автоматизированных систем. Разработка программы автономных испытаний. Разделы программы автономных испытаний: перечень функций, подлежащих испытаниям, описание взаимосвязей объекта испытаний с другим частями автоматизированной системы, уровень, порядок и методы проведения испытаний и обработка результатов, критерии приёмки модулей по результатам испытаний.

Документация, сопровождающая комплексные испытания автоматизированных систем. Разработка программы комплексных испытаний. Разделы программы комплексных испытаний: перечень объектов испытаний, состав предъявленной документации, описание проверяемых взаимосвязей между объектами испытаний, очерёдность испытаний модулей программного продукта, порядок и методы проведения испытаний.

Необходимые составляющие комплексных испытаний: программа комплексных испытаний, заключения по испытаниям отдельных модулей испытуемого программного продукта, комплексные тесты, программные и технические средства, соответствующие им.

Опытная эксплуатация автоматизированной системы. Ведение журнала опытной эксплуатации автоматизированной системы. Информация, помещаемая в журнал опытной эксплуатации.

Приёмочные испытания программного обеспечения. Документация, сопровождающая приёмочные испытания автоматизированной системы. Разработка программы приёмочных испытаний. Основные разделы программы приёмочных испытаний: перечень объектов, выделенных для испытаний, критерии приёмки частей автоматизированной системы, средства для проведения испытаний, методика испытаний и обработки их результатов.

Тема 2.5. Виды и методики проведения тестирования программного обеспечения

Цель и принципы тестирования программного обеспечения. Классификация тестирования программного обеспечения по запуску кода на исполнение, классификация тестирования программного обеспечения по доступу к коду и архитектуре, классификация тестирования программного обеспечения по уровню детализации приложения. Классификация тестирования программного обеспечения по принципам работы с приложением: позитивное тестирование, негативное тестирование. Классификация тестирования программного обеспечения по уровню функционального тестирования: дымовое тестирование, тестирование критического пути, расширенное тестирование.

Классификация тестирования программного обеспечения в зависимости от целей тестирования: функциональное тестирование, нефункциональное тестирование, тестирование производительности, нагрузочное тестирование, тестирование масштабируемости, объёмное тестирование, стрессовое тестирование, инсталляционное тестирование, тестирование интерфейса, тестирование удобства использования, тестирование локализации, тестирование безопасности, тестирование надёжности, регрессионное тестирование.

Раздел 3. Возможности параллельной обработки в среде MS Visual Studio 2019

Тема 3.1. Параллелизм задач в среде MS Visual Studio 2019

Характеристика среды программирования MS Visual Studio 2019 с возможностями организации параллельных вычислений. Историческая справка развития среды выполнения параллелизма. concrt140.dll — среда выполнения параллелизма, включённая в Visual Studio, требуется для параллельных контейнеров и алгоритмов. Использование средой выполнения программ с параллелизмом совместного планировщика задач, реализующего алгоритм переноса нагрузки для эффективного распределения ресурсов. Включение в среду выполнения с параллелизмом примитивов синхронизации, использующих совместное блокирование для синхронизации доступа к ресурсам.

Архитектура программной среды MS Visual Studio 2019 с возможностями организации параллельных вычислений. Основные компоненты среды

MS Visual Studio 2019 с возможностями организации параллельных вычислений: библиотеки параллельных шаблонов (PPL), библиотеки асинхронных агентов, планировщика задач и диспетчера ресурсов. Взаимодействие компонентов среды выполнения программ с параллелизмом с операционной системой и приложениями.

Библиотека параллельных шаблонов (PPL), её содержимое - алгоритмы и контейнеры общего назначения для выполнения детального параллелизма. Возможности PPL по обеспечению параллелизма императивных данных, обеспечению параллелизма задач.

Библиотека асинхронных агентов как модель программирования на основе интерфейсов и субъектов передачи сообщений для задач конвейера. Использование библиотеки асинхронных агентов для асинхронно взаимодействующих процессов.

Планировщик задач среды программирования MS Visual Studio 2019 с возможностями организации параллельных вычислений, его функции: планирование и координирование задач при выполнении, использование алгоритмов распределения нагрузки.

Менеджер ресурсов среды программирования MS Visual Studio 2019 с возможностями организации параллельных вычислений, его основная функция – управление вычислительными ресурсами при выполнении программ. Использование менеджера ресурсов для настройки производительности библиотек и приложений. Использование менеджером ресурсов библиотеки параллельных шаблонов, библиотеки агентов и планировщика задач для динамической балансировки ресурсов при изменении рабочей нагрузки.

Тема 3.2. Лямбда-выражения и их применение в параллельных вычислительных процессах

Понятие лямбда-выражения в C++. Синтаксис лямбда-выражения: [spisok](parameters) -> return_type { // тело функции }, где spisok — список переменных, которые будут захвачены из окружающего контекста (может быть пустым или содержать переменные по значению или по ссылке), parameters — параметры, которые принимает лямбда (аналогично обычной функции), return_type — тип возвращаемого значения, тело функции — код, который будет выполнен, когда лямбда будет вызвана. Ограничения на использование лямбда-выражений. Использование лямбда-выражений при программировании параллельных вычислений совместно с параллельными шаблонами.

Тема 3.3. Библиотека параллельных шаблонов (PPL) MS Visual Studio 2019

Представление параллельных алгоритмов в виде существующих функций в среде параллелизма. Алгоритм «parallel_for», его назначение – много-кратное выполнение одной и той же параллельной задачи. Поддержка алгоритмом «parallel_for» вложенного параллелизма. Отличия алгоритма от кон-

структурой `for`: выполняет задачи в предопределенном порядке, не поддерживает произвольные условия завершения, параметр `_Index_type` типа должен быть целочисленным типом, механизм обработки исключений для `parallel_for` алгоритма отличается от механизма обработки исключений `for`. Алгоритм параллелизма «`parallel_for_each`». Назначение алгоритма «`parallel_for_each`» - одновременное выполнение задач в итеративном контейнере. Алгоритм параллелизма «`parallel_invoke`». Его назначение - выполнение набора задач параллельно. Алгоритм «`parallel_transform`». Назначение алгоритма «`parallel_transform`» - выполнение множества операций параллелизации данных, к которым относятся: настройка яркости изображения и другие операции обработки изображений, суммирование или получение скалярного произведения двух векторов и выполнение других векторных вычислений, трансформировка трехмерных лучей, при которой каждая итерация относится к одному пикселю, подлежащему отображению. Перегрузки алгоритма «`parallel_transform`». Алгоритм «`parallel_reduce`». Назначение алгоритма «`parallel_reduce`»: умножение последовательности матриц для получения матрицы, умножение вектора на последовательность матриц для получения вектора, вычисление длины последовательности строк, объединение списка элементов в один элемент.

Алгоритмы параллельной сортировки: «`parallel_sort`», «`parallel_buffered_sort`», «`parallel_radixsort`». Механизм сортировки, реализованный в алгоритмах «`parallel_sort`», «`parallel_buffered_sort`» - выполнение сравнений. Сравнительный анализ алгоритмов сортировки «`parallel_sort`», «`parallel_buffered_sort`», «`parallel_radixsort`» по механизму сортировки, стабильности сортировки, требованиям к памяти, временной сложности, доступу итераторов. Критерии выбора алгоритма сортировки: размер набора данных, объем работы, выполняемой функцией сравнения или хэш-функцией, количество доступных вычислительных ресурсов, характеристики набора данных, размер блока.

Тема 3.4. Обработка исключений в среде MS Visual Studio 2019 с параллелизмом

Необходимость использования механизма исключений при разработке программ, реализующих параллельные вычислительные процессы на C++.

Ошибки при выполнении программ с параллельными вычислительными процессами, при которых целесообразно реализовать собственные механизмы обработки исключений - ошибки исполняющей среды: сбой в получении ресурса, а также ошибки в рабочих функциях, предоставляемых задачам и группам задач. Необходимость написания конструкции `try-catch` для каждого блока параллельных вычислений.

Механизм обработки исключений, создаваемых параллелизмом `::task`, организация очередей для исполнения обработчиков исключений, если задача имеет несколько продолжений. Тип исключения среды выполнения

`concurrency::task_canceled`, относящийся к задачам при параллельной обработке (`task`). Создание средой выполнения `task_canceled` при вызове `task::get` для отменённой задачи. Возможности перехвата и обработки исключения `concurrency::task_canceled` из продолжения на основе задач или при вызове `task::get`. Завершение средой выполнения приложения при создании исключения задачей и его перехватом одним из ее продолжений или основным приложением.

Обработка исключений, создаваемых группами задач, в среде MS Visual Studio 2019 с параллелизмом. Остановка средой выполнения MS Visual Studio 2019 всех активных задач при возникновении исключения в тексте исполняемой функции.

Исключения, создаваемые средой выполнения программ с параллельными процессами. Природа исключений, создаваемых средой выполнения программ с параллельными процессами, - ошибки программирования. Необходимость применения методик валидации и тестирования для ликвидации ошибок в программах и устранения исключений, создаваемых средой выполнения программ с параллельными процессами. Исключения, создаваемые асинхронными агентами.

Тема 3.5. Рекомендации по работе со средой выполнения программ Visual Studio 2019 с параллелизмом

Нецелесообразность распараллеливания тел небольших циклов. Необходимость тщательной оценки задачи перед применением модели ветвления-соединения. Неверно использованная модель ветвления-соединения может привлечь существенные затраты на синхронизацию. Использование алгоритма параллелизма «`parallel_invoke`» для решения проблем разделения и завоевания, который использует рекурсию для декомпозиции задачи на подзадачи. Использование отмены или обработки исключений для разрыва с параллельным циклом. Внимательное выполнение повторной блокировки в параллельном цикле. Возможные последствия блокировки тела параллельного цикла: нехватка памяти, неэффективное использование аппаратных ресурсов. Оценка целесообразности выполнения блокирующих операций при отмене параллельной работы. Рекомендация избегать записывать общие данные в параллельном цикле.

Понятие ложного общего доступа. Целесообразность отказа от ложного общего доступа при разработке программ. Необходимость и способы проверки действительности переменных на протяжении всего времени существования каждой задачи.

Раздел 4. Задачи из области искусственного интеллекта, для решения которых целесообразно распараллеливание вычислительных процессов и применение многопоточности

Тема 4.1. Создание сети обработки изображений

Определение операций, выполняемых над изображениями на основе его характеристик. Использование модели потока данных для маршрутизации образов через сеть. Создание сетью, основанной на потоке данных, конвейера задач. Организация в модели потока данных взаимодействия независимых компонентов программы друг с другом за счёт отправки сообщений. Выполнение действий компонентом, получившем сообщение, передача результата действия другому компоненту. Сравнение с моделью потока управления, в которой приложение использует структуры управления, например условные операторы, циклы и т. д., для управления порядком операций в программе.

Этапы построения сети обработки изображений: определение функций обработки изображений, создание сети асинхронных блоков сообщений, выполняющих обработку изображений по правилам: для любого изображения, созданного заданным автором, преобразуйте в серое масштабирование, для любого изображения, имеющего доминирующий красный цвет, удалите зеленые и синие компоненты, а затем затемните его, для любого другого изображения примените тонинг сепии. Все изображения JPEG (.jpg) хранятся в определенном каталоге, отладка, тестирование программы.

Элементы сети обработки изображений:

- объект параллелизма::преобразователя, который загружает объект с диска и добавляет запись в объект для связывания изображения с исходным именем файла,
- объект параллелизма::unbounded_buffer, который отправляет загруженные изображения в фильтры обработки изображений,
- объект, который преобразует изображения, созданные конкретным автором в серый масштаб, он использует метаданные изображения для определения его автора,
- объект, удаляющий компоненты зеленого и синего цвета из изображений с красным цветом в качестве доминирующего цвета,
- объект, который затемняет изображения, имеющие доминирующий красный цвет,
- объект, который применяет сепию к изображениям, не созданным конкретным автором и не являющимися преимущественно красными,
- объект, который сохраняет обработанный image на диск в виде растрового изображения, извлекает исходное имя файла из другого объекта и изменяет расширение имени файла на .bmp,
- объект, который освобождает память для изображений,
- объект параллелизма::call, который является узлом терминала в сети, сообщает основному приложению о том, что изображение было обработано.

Тема 4.2. Программирование неиерархических и иерархических методов кластер-анализа данных с применением параллелизации задач

Постановка задачи кластеризации данных. Меры близости, применяемые в разных алгоритмах кластеризации. Классификация алгоритмов кластеризации: эвристические графовые алгоритмы, статистические алгоритмы, методы иерархической кластеризации.

Неиерархические алгоритмы кластеризации. Методы k-средних, метод главных компонент. Алгоритм кластеризации данных методом k-средних. Алгоритм кластеризации данных k ближайших соседей. Различия этих алгоритмов.

Иерархические методы кластеризации больших данных: характеристика, примеры. Классы иерархических методов кластеризации: агломеративная кластеризация, дивизивная кластеризация. Метрики, применяемые в иерархических методах классификации: Евклидово расстояние, Манхэттенское расстояние, расстояние Чебышева, процент несогласия.

Применение для практической реализации на языке C++ различных методов кластеризации данных параллелизации вычислительных процессов, особенно для данных больших объёмов.

Тема 4.3. Применение параллелизации вычислительных процессов для генерации решений в системах поддержки принятия решений

Когнитивные карты, правила их построения и применение для представления причинно-следственных связей в исследуемой предметной области. Применение когнитивных карт в экспертных системах и автоматизированных системах поддержки принятия решений.

Методы принятия решений при реализации игровых стратегий, применяемые в компьютерных играх, генерации решений в системах поддержки принятия решений: правила и сценарии, конечные автоматы, поведенческие деревья, планирование, эвристические алгоритмы, адаптивные системы, вероятностные модели.

Методы генерации решений, применяемые для управления передвижением объектов: алгоритмы поиска пути, навигационные сетки, векторное управление, навигационные управляющие системы, процедурные методы генерации пути, интерполяция и сглаживание.

Метод анализа иерархий для выбора решения. Этапы решения задачи методом анализа иерархий: 1) построение качественной модели проблемы в виде иерархии, включающей цель, альтернативные варианты достижения цели и критерии для оценки качества альтернатив, 2) определение приоритетов всех элементов иерархии с использованием метода парных сравнений, 3) синтез глобальных приоритетов альтернатив путем линейной свертки приоритетов элементов иерархии, 4) проверка суждений на согласованность, 5) принятие решения на основе полученных результатов. Целесообразность распараллеливания вычислительных процессов при программной реализации когнитивных карт, метода анализа иерархий, конечных автоматов, поведенческих деревьев.

Тема 4.4. Формирование базы знаний и ранжирование альтернатив с применением параллельных вычислительных процессов при разработке экспертных систем

Модели знаний: продукционная, предикатная, фреймовая, объектно-ориентированная модели, семантические сети. Представление знаний продукциями. Вывод в продукционных системах. Представление знаний фреймами. Слоты как составные части фреймов. Технологические аспекты организации логического вывода на сети фреймов. Представление знаний семантическими сетями. Вывод на основе семантических сетей. Представление знаний на языке исчисления предикатов первого порядка. Логический вывод на основе метода резолюций. Объектно-ориентированные модели знаний и их практическая реализация в современных инструментальных программных средах. Представление и использование метазнаний. Интеграция различных способов представления знаний.

Методы экспертной оценки: ранжирование, парное сравнивание, множественные сравнения, непосредственная оценка, Черчмена-Акоффа, метод Терстоуна, метод фон Неймана-Моргенштерна. Применение методов экспертной оценки для сбора знаний с последующей формализации в базу знаний. Оценка согласованности мнений экспертов для дальнейшего использования знаний. Вычисление коэффициента конкордации. Критерий согласования Пирсона. Применение параллельных вычислительных процессов для сбора и структурирования знаний с помощью метода ранжирования альтернатив и продукционной модели знаний.

Нечётная логика, нечёткие множества и нечёткие отношения. Нечёткие выводы. Проверка согласованности мнений экспертов при представлении знаний.

Целесообразность программирования алгоритмов формирования баз знаний и выводов на знаниях с применением параллельных вычислительных процессов.

4.3. Лекции/практические занятия

Таблица 4 - Содержание лекций/ практических занятий и контрольные мероприятия

№ п/п	№ темы	№ и название лекций/ практических занятий	Формируемые компетенции (индикаторы)	Вид контрольного мероприятия	Кол-во часов
Раздел 1. Объектно-ориентированная методология и технология разработки программного обеспечения					

№ п/п	№ темы	№ и название лекций/ практических занятий	Формируемые компетенции (индикаторы)	Вид контрольного мероприятия	Кол-во часов
1	Тема 1.1. Методология объектно-ориентированного подхода разработки программного обеспечения	Лекция № 1. Основные принципы объектно-ориентированной методологии разработки программного обеспечения	ПК-16 (PL-3).2	Устный опрос, тестирование	1
2	Тема 1.2. Базовые положения объектно-ориентированной технологии программирования применительно к языку программирования C++	Лекция № 2. Базовые положения объектно-ориентированной технологии программирования применительно к языку программирования C++	ПК-16 (PL-3).2	Устный опрос, тестирование	0,5
3	Тема 1.3. Язык унифицированного моделирования UML, его применение при проектировании автоматизированных информационных систем	Лекция № 3. Нотации языка унифицированного моделирования UML. Его применение при проектировании программных продуктов	ПК-16 (PL-3).2	Устный опрос, тестирование	1
		Практическая работа № 1. Проектирование автоматизированной интеллектуальной информационной системы с применением языка унифицированного моделирования UML.	ПК-16 (PL-3).2	Задание практической работы	4

Раздел 2. Жизненный цикл программных продуктов. Виды работ, выполняемых на отдельных его этапах.

4	Тема 2.1. Стандарты, регламентирующие жизненный цикл программного обеспечения и работы, выполняемые на отдельных его этапах	Лекция № 4. Стандарты, регламентирующие жизненный цикл программного обеспечения и работы, выполняемые на отдельных его этапах	ПК-16 (PL-3).1	Устный опрос, тестирование	0,5
5	Тема 2.2. Содержание стандартов ГОСТ Р 59793-2021 «Автоматизированные системы. Стадии создания», ГОСТ Р ИСО/МЭК 12207-2010 «Информационная технология. Системная и программ-	Лекция № 5. Содержание стандартов ГОСТ Р 59793-2021 «Автоматизированные системы. Стадии создания», ГОСТ Р ИСО/МЭК 12207-2010	ПК-16 (PL-3).1	Устный опрос, тестирование	1

№ п/п	№ темы	№ и название лекций/ практических занятий	Формируемые компетенции (индикаторы)	Вид контрольного мероприятия	Кол-во часов
	ная инженерия. Процессы жизненного цикла программных средств»				
6	Тема 2.3. Содержание стандарта ГОСТ Р ИСО/ МЭК 25010-2015 «Информационные технологии. Системная и программная инженерия. Требования и оценка качества систем и программного обеспечения (SQuaRE). Модели качества систем и программных продуктов», его использование для оценки качества программного обеспечения	Лекция № 6. Содержание стандарта ГОСТ Р ИСО/ МЭК 25010-2015 «Информационные технологии. Системная и программная инженерия. Требования и оценка качества систем и программного обеспечения (SQuaRE). Модели качества систем и программных продуктов»	ПК-16 (PL-3).1	Устный опрос, тестирование	1
		Практическая работа № 2. Адаптация и корректировка сформулированных в ГОСТ Р ИСО/ МЭК 25010-2015 модели и критериев оценки качества программ применительно к интеллектуальным информационным системам, реализующим параллельные вычислительные процессы	ПК-16 (PL-3).1	Задание практической работы	2
7	Тема 2.4. Определение порядка верификации, валидации и испытаний автоматизированных информационных систем в стандарте ГОСТ Р 59792-2021 «Информационные технологии. Комплекс стандартов на автоматизированные системы. Виды испытаний автоматизированных систем»	Лекция № 7. Верификация, валидация и испытания программных продуктов, рекомендации ГОСТ Р 59792-2021 «Информационные технологии. Комплекс стандартов на автоматизированные системы. Виды испытаний автоматизированных систем»	ПК-16 (PL-3).1	Устный опрос, тестирование	1
		Практическая работа № 3. Разработка программы автономных испытаний программного продукта, решающего задачи в области искусственного интеллекта с применением парализации вычислительных процессов	ПК-16 (PL-3).1	Задание практической работы	2

№ п/п	№ темы	№ и название лекций/ практических занятий	Формируемые компетенции (индикаторы)	Вид контрольного мероприятия	Кол-во часов
8	Тема 2.5. Виды и методики проведения тестирования программного обеспечения	Лекция № 8. Виды и методики проведения тестирования программного обеспечения	ПК-16 (PL-3).1	Устный опрос, тестирование	1
		Практическая работа № 4. Знакомство с инструментом функционального тестирования TestComplete	ПК-16 (PL-3).1	Задание практической работы	2
Раздел 3. Возможности параллельной обработки в среде MS Visual Studio 2019					
9	Тема 3.1. Параллизм задач в среде MS Visual Studio 2019	Лекция № 9. Характеристика и архитектура программной среды MS Visual Studio 2019 с возможностями организации параллельных вычислений	ПК-16 (PL-3).2, ПК-16 (PL-3).3	Устный опрос, тестирование	1
		Практическая работа № 5. Знакомство с возможностями организации параллельных вычислений в среде MS Visual Studio 2019	ПК-16 (PL-3).2, ПК-16 (PL-3).3	Задание практической работы	2
10	Тема 3.2. Лямбда-выражения и их применение в параллельных вычислительных процессах	Лекция № 10. Лямбда-выражения и их применение в параллельных вычислительных процессах	ПК-16 (PL-3).2, ПК-16 (PL-3).3	Устный опрос, тестирование	1
		Практическая работа № 6. Разработка лямбда-выражений при практической реализации параллельных вычислений	ПК-16 (PL-3).2, ПК-16 (PL-3).3	Задание практической работы	2
11	Тема 3.3. Библиотека параллельных шаблонов (PPL) MS Visual Studio 2019	Лекция № 11. Библиотека параллельных шаблонов (PPL) MS Visual Studio 2019	ПК-16 (PL-3).2, ПК-16 (PL-3).3	Устный опрос, тестирование	1
		Практическая работа № 7. Практическое применение алгоритмов «parallel_for», «parallel_for_each», «parallel_transform», «parallel_sort» для реализации сортировки записей, содержащих информацию об оцениваемых альтернативах.	ПК-16 (PL-3).2, ПК-16 (PL-3).3	Задание практической работы	2
12	Тема 3.4. Обработка исключений в среде MS Visual Studio	Лекция № 12. Обработка исключений в среде MS Visual Studio 2019 с параллелизмом	ПК-16 (PL-3).2, ПК-16 (PL-3).3	Устный опрос, тестирование	1

№ п/п	№ темы	№ и название лекций/ практических занятий	Формируемые компетенции (индикаторы)	Вид контрольного мероприятия	Кол-во часов
	2019 с параллелизмом	Практическая работа № 8. Создание собственных исключений при реализации параллельных вычислений	ПК-16 (PL-3).2, ПК-16 (PL-3).3	Задание практической работы	2
13	Тема 3.5. Рекомендации по работе со средой выполнения программы Visual Studio 2019 с параллелизмом	Лекция № 13. Рекомендации по работе со средой выполнения программы Visual Studio 2019 с параллелизмом	ПК-16 (PL-3).2, ПК-16 (PL-3).3	Устный опрос, тестирование	1
		Практическая работа № 9. Оценка возможностей использования модели ветвление-соединение при организации параллельных вычислений	ПК-16 (PL-3).2, ПК-16 (PL-3).3	Задание практической работы	2
Раздел 4. Задачи из области искусственного интеллекта, для решения которых целесообразно распараллеливание вычислительных процессов и применение многопоточности					
14	Тема 4.1. Создание сети обработки изображений	Лекция № 14. Применение возможностей параллельных вычислительных процессов для создания сети обработки изображений	ПК-16 (PL-3).2, ПК-16 (PL-3).3	Устный опрос, тестирование	1
		Практическая работа № 10. Создание сети обработки изображений	ПК-16 (PL-3).2, ПК-16 (PL-3).3	Задание практической работы	4/ 1
15	Тема 4.2. Программирование неиерархических и иерархических методов кластер-анализа данных с применением параллелизации задач	Лекция № 15. Целесообразность практической реализации неиерархических и иерархических методов кластер-анализа данных с применением параллелизации задач	ПК-16 (PL-3).2, ПК-16 (PL-3).3	Устный опрос, тестирование	1
		Практическая работа № 11. Практическая реализация иерархического алгоритма кластер-анализа данных	ПК-16 (PL-3).2, ПК-16 (PL-3).3	Задание практической работы	4/ 1
16	Тема 4.3. Применение параллелизации вычислительных процессов для генерации решений в системах поддержки принятия решений	Лекция № 16. Применение параллелизации вычислительных процессов для генерации решений в системах поддержки принятия решений	ПК-16 (PL-3).2, ПК-16 (PL-3).3	Устный опрос, тестирование	1
		Практическая работа № 12. Применение параллелизации вычислительных процессов для реализации модуля оценки	ПК-16 (PL-3).2, ПК-16 (PL-3).3	Задание практической работы	4/ 1

№ п/п	№ темы	№ и название лекций/ практиче- ских занятий	Форми- руемые компе- тенции (индика- торы)	Вид кон- троль- ного ме- роприя- тия	Кол- во часов
		принимаемого решения с ис- пользованием множества кри- териев (показателей)			
17	Тема 4.4. Формиро- вание базы знаний и ранжирование аль- тернатив с примене- нием параллельных вычислительных процессов при разра- ботке экспертных си- стем	Лекция № 17. Формирование базы знаний и ранжирование альтернатив с применением параллельных вычислитель- ных процессов при разработке экспертных систем	ПК-16 (PL-3).2, ПК-16 (PL-3).3	Устный опрос, те- стирова- ние	1
		Практическая работа № 13. Формирование базы знаний и ранжирование альтернатив с применением параллельных вычислительных процессов при разработке экспертных си- стем	ПК-16 (PL-3).2, ПК-16 (PL-3).3	Задание практиче- ской ра- боты	2 / 1

4.4. Перечень вопросов для самостоятельного изучения дисциплины

Таблица 5

№ п/п	№ раздела и темы	Перечень рассматриваемых вопросов для самостоятельного изучения
Раздел 1. Объектно-ориентированная методология и технология разработки про- граммного обеспечения		
1	Тема 1.1. Методология объектно-ориен- тированного подхода разработки про- граммного обеспечения	Историческая справка о появлении и этапах развития объектно-ориентированной мето- дологии разработки программного обес- печения. ПК-16 (PL-3).1
2	Тема 1.2. Базовые положения объектно- ориентированной технологии програм- мирования применительно к языку про- граммирования C++	Особенности реализации в языке програм- мирования C++ инкапсуляции, полимор- физма и наследования. ПК-16 (PL-3).1
3	Тема 1.3. Язык унифицированного мо- делирования UML, его применение при проектировании автоматизированных информационных систем	Различные виды представлений классов на UML-диаграммах классов. Представление на диаграммах классов отношений между классами: наследование, ассоциации. ПК-16 (PL-3).1
Раздел 2. Жизненный цикл программных продуктов. Виды работ, выполняемых на отдельных его этапах.		
4	Тема 2.1. Стандарты, регламентирую- щие жизненный цикл программного обеспечения и работы, выполняемые на отдельных его этапах	Содержание и применение стандартов, ре- гламентирующих оформление техниче- ского задания на автоматизированную си- стему и программный продукт.

№ п/п	№ раздела и темы	Перечень рассматриваемых вопросов для самостоятельного изучения
		ПК-16 (PL-3).1
5	Тема 2.2. Содержание стандартов ГОСТ Р 59793-2021 «Автоматизированные системы. Стадии создания», ГОСТ Р ИСО/МЭК 12207-2010 «Информационная технология. Системная и программная инженерия. Процессы жизненного цикла программных средств»	Эволюция содержания стандарта ГОСТ Р ИСО/МЭК 12207. Возможный причины такой эволюции. Как изменились группы процессов, указанных в стандарте? ПК-16 (PL-3).1
6	Тема 2.3. Содержание стандарта ГОСТ Р ИСО/ МЭК 25010-2015 «Информационные технологии. Системная и программная инженерия. Требования и оценка качества систем и программного обеспечения (SQuaRE). Модели качества систем и программных продуктов», его использование для оценки качества программного обеспечения	Изучить основные положения ГОСТ Р ИСО/МЭК 25001 2017 «Информационные технологии. Системная и программная инженерия. Требования и оценка качества систем и программного обеспечения (SQuaRE) Планирование и управление» ПК-16 (PL-3).1
7	Тема 2.4. Определение порядка верификации, валидации и испытаний автоматизированных информационных систем в стандарте ГОСТ Р 59792-2021 «Информационные технологии. Комплекс стандартов на автоматизированные системы. Виды испытаний автоматизированных систем»	Уточнить понятия верификации и валидации программного обеспечения. В чём их различие? В чём отличие понятий «тестирование» и «испытание» программного продукта? ПК-16 (PL-3).1
8	Тема 2.5. Виды и методики проведения тестирования программного обеспечения	Рассмотреть наиболее широко применяемые автоматизированные средства тестирования программного обеспечения. Провести анализ их функций, выявить достоинства и недостатки. ПК-16 (PL-3).1
Раздел 3. Возможности параллельной обработки в среде MS Visual Studio 2019		
9	Тема 3.1. Параллелизм задач в среде MS Visual Studio 2019	Визуализаторы параллелизма задач в среде MS Visual Studio 2019, их функции, применение. ПК-16 (PL-3).2, ПК-16 (PL-3).3
10	Тема 3.2. Лямбда-выражения и их применение в параллельных вычислительных процессах	Универсальные лямбда-выражения. Составные части лямбда-выражений. ПК-16 (PL-3).2, ПК-16 (PL-3).3
11	Тема 3.3. Библиотека параллельных шаблонов (PPL) MS Visual Studio 2019	Использование при параллельном программировании лямбда-выражений. Создание продолжений задач при параллельном программировании. ПК-16 (PL-3).2, ПК-16 (PL-3).3
12	Тема 3.4. Обработка исключений в среде MS Visual Studio 2019 с параллелизмом	Порядок обработки нескольких исключений, возникающих в программе, реализующей параллельный вычислительных процесс. ПК-16 (PL-3).2, ПК-16 (PL-3).3

№ п/п	№ раздела и темы	Перечень рассматриваемых вопросов для самостоятельного изучения
13	Тема 3.5. Рекомендации по работе со средой выполнения программ Visual Studio 2019 с параллелизмом	Применение планировщика заданий для настройки производительности исполнения программного кода ПК-16 (PL-3).2, ПК-16 (PL-3).3
Раздел 4. Задачи из области искусственного интеллекта, для решения которых целесообразно распараллеливание вычислительных процессов и применение многопоточности		
14	Тема 4.1. Создание сети обработки изображений	Классы, методы и инструменты обработки изображений библиотеки C++ STL, библиотеки OpenCV, библиотеки System.Drawing. ПК-16 (PL-3).2, ПК-16 (PL-3).3
15	Тема 4.2. Программирование неиерархических и иерархических методов кластер-анализа данных с применением параллелизации задач	Сравнение агломеративных и дивизивных методов кластер-анализа. Критерии оценки результатов иерархического кластер-анализа информации. ПК-16 (PL-3).2, ПК-16 (PL-3).3
16	Тема 4.3. Применение параллелизации вычислительных процессов для генерации решений в системах поддержки принятия решений	Практическая реализация в программных средах конечных автоматов, деревьев решений. Методики приведения любых деревьев к бинарным. Методы обхода деревьев. ПК-16 (PL-3).2, ПК-16 (PL-3).3
17	Тема 4.4. Формирование базы знаний и ранжирование альтернатив с применением параллельных вычислительных процессов при разработке экспертных систем	Анализ различных методов экспертной оценки с позиции их возможного использования для формирования баз знаний. Нечёткая логика и нечёткие выводы в экспертных системах. ПК-16 (PL-3).2, ПК-16 (PL-3).3

5. Образовательные технологии

При реализации программы дисциплины используются следующие современные методики и технологии обучения:

- гибкая архитектура программ – 25% содержания ежегодно обновляется при участии индустриальных партнёров с учетом отраслевой направленности;
- адаптивные технологии взаимодействия с профессионалами из индустрии (наставничество, кейсы от индустриальных партнеров);
- проектно-соревновательный подход – хакатоны и командные решения отраслевых задач;
- проблемно-ориентированное обучение – работа над кейсами от индустриальных партнёров;
- решение практических задач на практических занятиях в лабораториях центра «Институт цифровой трансформации в АПК».

Таблица 6

№ п/ п	Тема и форма занятия		Наименование используемых активных и интерактивных образовательных технологий
1	Тема 1.1. Методология объектно-ориентированного подхода разработки программного обеспечения	Лекция	Мультимедиа технологии
2	Тема 1.2. Базовые положения объектно-ориентированной технологии программирования применительно к языку программирования C++	Лекция	Мультимедиа технологии
3	Тема 1.3. Язык унифицированного моделирования UML, его применение при проектировании автоматизированных информационных систем	Лекция	Мультимедиа технологии
		Практическая работа	Проектные технологии
4	Тема 2.1. Стандарты, регламентирующие жизненный цикл программного обеспечения и работы, выполняемые на отдельных его этапах	Лекция	Мультимедиа технологии
5	Тема 2.2. Содержание стандартов ГОСТ Р 59793-2021 «Автоматизированные системы. Стадии создания», ГОСТ Р ИСО/МЭК 12207-2010 «Информационная технология. Системная и программная инженерия. Процессы жизненного цикла программных средств»	Лекция	Мультимедиа технологии
6	Тема 2.3. Содержание стандарта ГОСТ Р ИСО/ МЭК 25010-2015 «Информационные технологии. Системная и программная инженерия. Требования и оценка качества систем и программного обеспечения (SQuaRE). Модели качества систем и программных продуктов», его использование для оценки качества программного обеспечения	Лекция	Мультимедиа технологии
		Практическая работа	Проблемно-ориентированные технологии
7	Тема 2.4. Определение порядка верификации, валидации и испытаний автоматизированных ин-	Лекция	Мультимедиа технологии

	формационных систем в стандарте ГОСТ Р 59792-2021 «Информационные технологии. Комплекс стандартов на автоматизированные системы. Виды испытаний автоматизированных систем»	Практическая работа	Проблемно-ориентированные технологии
8	Тема 2.5. Виды и методики проведения тестирования программного обеспечения	Лекция	Мультимедиа технологии
		Практическая работа	Проблемно-ориентированные технологии
9	Тема 3.1. Параллелизм задач в среде MS Visual Studio 2019	Лекция	Мультимедиа технологии
		Практическая работа	Проблемно-ориентированные технологии
10	Тема 3.2. Лямбда-выражения и их применение в параллельных вычислительных процессах	Лекция	Мультимедиа технологии
		Практическая работа	Проблемно-ориентированные технологии
11	Тема 3.3. Библиотека параллельных шаблонов (PPL) MS Visual Studio 2019	Лекция	Мультимедиа технологии
		Практическая работа	Проблемно-ориентированные технологии
12	Тема 3.4. Обработка исключений в среде MS Visual Studio 2019 с параллелизмом	Лекция	Мультимедиа технологии
		Практическая работа	Проблемно-ориентированные технологии
13	Тема 3.5. Рекомендации по работе со средой выполнения программы Visual Studio 2019 с параллелизмом	Лекция	Мультимедиа технологии
		Практическая работа	Проблемно-ориентированные технологии
14	Тема 4.1. Создание сети обработки изображений	Лекция	Мультимедиа технологии
		Практическая работа	Проектные технологии
15	Тема 4.2. Программирование неиерархических и иерархических методов кластер-анализа данных с применением параллелизации задач	Лекция	Мультимедиа технологии
		Практическая работа	Исследовательские технологии
16	Тема 4.3. Применение параллелизации вычислительных процессов для генерации решений в системах поддержки принятия решений	Лекция	Мультимедиа технологии
		Практическая работа	Исследовательские технологии
17	Тема 4.4. Формирование базы знаний и ранжирование альтернатив с применением параллельных вычислительных процессов при разработке экспертных систем	Лекция	Мультимедиа технологии
		Практическая работа	Исследовательские технологии

6. Текущий контроль успеваемости и промежуточная аттестация по итогам освоения дисциплины

6.1. Типовые контрольные задания или иные материалы, необходимые для оценки знаний, умений и навыков и (или) опыта деятельности

Вопросы для устного опроса и для защиты практических работ

Тема 1.1. Методология объектно-ориентированного подхода разработки программного обеспечения

- 1 Какие в настоящее время существуют методологии разработки программного обеспечения? В чём их принципиальное отличие?
- 2 Что составляет основу объектно-ориентированной методологии разработки программного обеспечения?
- 3 Какие базовые принципы объектно-ориентированной методологии разработки программного обеспечения?
- 4 Что означает принципы инкапсуляции?
- 5 Что означает принципы иерархии?
- 6 Что означает принцип абстрагирования?
- 7 Какие элементы объектной модели Вам известны?

Тема 1.2. Базовые положения объектно-ориентированной технологии программирования применительно к языку программирования C++

- 1 Какие базовые принципы объектно-ориентированной технологии программирования?
- 2 Что означает принцип инкапсуляции, каково его прикладное значение при разработке программ?
- 3 Что означает принцип полиморфизма, каково его прикладное значение при разработке программ?
- 4 Что означает принцип наследования, каково его прикладное значение при разработке программ?
- 5 Какие функции выполняют конструкторы классов?
- 6 Какие функции выполняют деструкторы классов?

Тема 1.3. Язык унифицированного моделирования UML, его применение при проектировании автоматизированных информационных систем

- 1 Какие типы диаграмм входят в унифицированный язык моделирования UML?
- 2 Для чего применяется каждый типов UML-диаграмм при моделировании и проектировании автоматизированных информационных систем?
- 3 Какие нотации и возможности использования имеют UML-диаграммы вариантов использования?
4. Какие нотации и возможности использования имеют UML-диаграммы классов?
- 5 Какие отношения могут быть между классами на UML-диаграммах классов?

6 Какие нотации и возможности использования имеют UML-диаграммы взаимодействия?

7 Какие нотации и возможности использования имеют UML-диаграммы состояний?

8 Какие инструментальные средства можно использовать для построения UML-диаграмм?

Тема 2.1. Стандарты, регламентирующие жизненный цикл программного обеспечения и работы, выполняемые на отдельных его этапах

1 Что излагается в стандарте ГОСТ Р 59793-2021 «Автоматизированные системы. Стадии создания»?

2 Что изложено в стандарте ГОСТ Р ИСО/МЭК 12207-2010 «Информационная технология. Системная и программная инженерия. Процессы жизненного цикла программных средств»?

3 Какие разделы технического задания определяются в ГОСТ 34.602-2020?

4 Какие виды испытаний обозначены в ГОСТ Р 59792-2021 «Информационные технологии. Комплекс стандартов на автоматизированные системы. Виды испытаний автоматизированных систем»?

5 В чём заключается оценка качества программного обеспечения в соответствии со стандартом ГОСТ Р ИСО/ МЭК 25010-2015 «Информационные технологии. Системная и программная инженерия. Требования и оценка качества систем и программного обеспечения (SQuaRE). Модели качества систем и программных продуктов»?

Тема 2.2. Содержание стандартов ГОСТ Р 59793-2021 «Автоматизированные системы. Стадии создания», ГОСТ Р ИСО/МЭК 12207-2010 «Информационная технология. Системная и программная инженерия. Процессы жизненного цикла программных средств»

1 Какие стадии создания автоматизированных систем определены в стандарте ГОСТ Р 59793-2021 «Автоматизированные системы. Стадии создания»?

2 Чем регламентируется форма технического задания на разработку автоматизированной системы?

3 Какие группы процессов рассмотрены в ГОСТ Р ИСО/МЭК 12207-2010 «Информационная технология. Системная и программная инженерия. Процессы жизненного цикла программных средств»?

4 Какие процессы включены в группу процессов соглашения ГОСТ Р ИСО/МЭК 12207-2010?

5 Какие процессы включены в группу процессов организационного обеспечения проекта ГОСТ Р ИСО/МЭК 12207-2010?

6 Какие процессы включены в группу процессов проекта ГОСТ Р ИСО/МЭК 12207-2010?

7 Какие процессы включены в группу технических процессов ГОСТ Р ИСО/МЭК 12207-2010?

8 Какие процессы включены в группу процессов реализации программных средств ГОСТ Р ИСО/МЭК 12207-2010?

9 Какие процессы включены в группу процессов повторного применения программных средств ГОСТ Р ИСО/МЭК 12207-2010?

Тема 2.3. Содержание стандарта ГОСТ Р ИСО/ МЭК 25010-2015 «Информационные технологии. Системная и программная инженерия. Требования и оценка качества систем и программного обеспечения (SQuaRE). Модели качества систем и программных продуктов», его использование для оценки качества программного обеспечения

- 1 Что означает термин «качество программного обеспечения»?
- 2 Как можно охарактеризовать модель качества программного продукта?
- 3 Что представляет собой модель качества, ориентированная на применение программного продукта?
- 4 Какие показатели характеризуют качество программного продукта?
- 5 Как выражается влияние показателей качества программного продукта, обозначенных в ГОСТ Р ИСО/ МЭК 25010-2015, на удобство эксплуатации пользователя программы?
- 6 Какие описатели содержатся в ГОСТ Р ИСО/ МЭК 25010-2015 для уточнения показателей качества программных продуктов?

Тема 2.4. Определение порядка верификации, валидации и испытаний автоматизированных информационных систем в стандарте ГОСТ Р 59792-2021 «Информационные технологии. Комплекс стандартов на автоматизированные системы. Виды испытаний автоматизированных систем»

- 1 Что означает «верификация» программного продукта?
- 2 Что означает «валидация» программного продукта?
- 3 В чём отличие терминов «верификация», «валидация», «тестирование» программного обеспечения?
- 4 Какие виды испытаний регламентированы ГОСТ Р 59792-2021 «Информационные технологии. Комплекс стандартов на автоматизированные системы. Виды испытаний автоматизированных систем»?
- 5 С чего начинается процесс проведения любого вида испытаний программного продукта?
- 6 Какой вид испытаний программного продукта в случае получения положительного заключения позволяет начинать эксплуатацию программного продукта?

Тема 2.5. Виды и методики проведения тестирования программного обеспечения

- 1 Какие виды тестирования программных продуктов по запуску кода на исполнение применяются в практике тестирования?
- 2 Какие виды тестирования программных продуктов, зависящие от целей тестирования, применяются в практике тестирования программ?
- 3 Какие программные средства могут использоваться для автоматизации процесса тестирования программ?

- 4 Какие виды функционального тестирования программных продуктов, зависящие от уровня тестирования, Вам известны?
- 5 Какие классификации методик и видов тестирования программного обеспечения Вам известны?

Тема 3.1. Параллелизм задач в среде MS Visual Studio 2019

- 1 За счёт каких компонентов программной среды MS Visual Studio 2019 реализуются параллельные вычисления в программах?
- 2 Какую архитектуру имеет программная среда MS Visual Studio 2019 с возможностями организации параллельных вычислений?
- 3 Какие компоненты имеет библиотека параллельных шаблонов (PPL) MS Visual Studio 2019?
- 4 Что такое библиотека асинхронных агентов MS Visual Studio 2019, какое она имеет назначение?
- 5 Что такое менеджер ресурсов среды программирования MS Visual Studio 2019 с возможностями организации параллельных вычислений, какова его основная функция?
- 6 Какие функции выполняет менеджер ресурсов среды программирования MS Visual Studio 2019?

Тема 3.2. Лямбда-выражения и их применение в параллельных вычислительных процессах

- 1 Какая конструкция объектной среды называется лямбда-выражением?
- 2 Какой синтаксис принят для определения лямбда-выражений?
- 3 Для чего используются лямбда-выражения при разработке программ?

Тема 3.3. Библиотека параллельных шаблонов (PPL) MS Visual Studio 2019

- 1 Какие алгоритмы включаются в библиотеку параллельных шаблонов (PPL) MS Visual Studio 2019?
- 2 Для чего предназначен алгоритм «parallel_for»?
- 3 Для чего предназначен алгоритм «parallel_for_each»?
- 4 Для чего предназначен алгоритм «parallel_transform»?
- 5 Для чего предназначен алгоритм «parallel_reduce»?
- 6 Для чего предназначены алгоритмы «parallel_sort», «parallel_buffered_sort», «parallel_radixsort»? Какие классы задач они решают?

Тема 3.4. Обработка исключений в среде MS Visual Studio 2019 с параллелизмом

- 1 Почему необходимо использовать механизм исключений при разработке программ, реализующих параллельные вычислительные процессы на C++?
- 2 Для каких типов ошибок, возникающих при выполнении программ с параллельными вычислительными процессами, необходимо разрабатывать собственные механизмы обработки исключений?

3 Какие исключения создаются при выполнении параллельных вычислительных процессов в программах?

4 Как осуществляется обработка исключений средой выполнения MS Visual Studio 2019 программ с параллельными процессами?

Тема 3.5. Рекомендации по работе со средой выполнения программ Visual Studio 2019 с параллелизмом

1 Почему при разработке программ с параллельными процессами необходима тщательная оценка решаемой задачи перед применением модели ветвления-соединения?

2 Для чего используется алгоритм «parallel_invoke»?

3 Что такое ложный доступ, почему от него отказываются при разработке программ?

Тема 4.1. Создание сети обработки изображений

1 Какие инструменты среды программирования Visual Studio 2019 позволяют обрабатывать изображения?

2 Какие операции могут выполняться над изображениями?

3 Как происходит в модели потока данных взаимодействие независимых компонентов программы друг с другом?

4 Какие действия выполняют на этапах создания сети изображений?

5 Какие Вы знаете элементы сети обработки изображений?

Тема 4.2. Программирование неиерархических и иерархических методов кластер-анализа данных с применением

1 Что такое кластеризация данных?

2 Чем отличаются задачи классификации и кластеризации данных?

3 Как можно классифицировать алгоритмы кластеризации?

4 Как можно охарактеризовать неиерархические алгоритмы кластеризации? Приведите примеры таких алгоритмов.

5 Как можно охарактеризовать иерархические алгоритмы кластеризации? Приведите примеры таких алгоритмов.

6 Приведите примеры практической реализации алгоритмов кластеризации с применением параллельных вычислений в MS Visual Studio 2019?

7 Какие метрики для оценки степени близости кластеризуемых объектов и степени сплочённости кластер-групп, различия различных кластер-групп Вы знаете?

Тема 4.3. Применение параллелизации вычислительных процессов для генерации решений в системах поддержки принятия решений

1 Какие методы генерации решений при реализации игровых стратегий в компьютерных играх Вы знаете?

2 Как можно использовать распараллеливание вычислительных процессов для представления конечных автоматов и генерации решений с их помощью?

3 Как можно использовать распараллеливание вычислительных процессов для представления деревьев решений (метода иерархий) и генерации решений с их помощью?

4 Как можно использовать распараллеливание вычислительных процессов для представления когнитивных карт и генерации решений с их помощью?

5 Как можно использовать распараллеливание вычислительных процессов для нахождения оптимальных управлеченческих решений с помощью эволюционных алгоритмов?

Тема 4.4. Формирование базы знаний и ранжирование альтернатив с применением параллельных вычислительных процессов при разработке экспертных систем

1 Какие модели можно использовать для представления знаний в экспертных системах?

2 Какие методы экспертной оценки можно использовать для сбора знаний с целью наполнения базы знаний экспертной системы?

3 Как производится оценка согласованности мнений экспертов при формировании базы знаний?

4 Что такое нечёткая логика и как она используется для формализации знаний и вывода на знаниях в экспертных системах?

5 Как с помощью параллельных вычислений можно реализовать вывод на знаниях?

Практические задания

Тема 1.3. Язык унифицированного моделирования UML, его применение при проектировании автоматизированных информационных систем

Практическая работа № 1. Проектирование автоматизированной интеллектуальной информационной системы с применением языка унифицированного моделирования UML.

Задание: выполнить проектирование выбранного варианта интеллектуальной информационной системы, применив для этого UML-диаграммы вариантов использования, классов, взаимодействия, состояний.

Варианты задания для выполнения практической работы:

Тема № 1. Проектирование автоматизированной системы поддержки принятия решений для управления качеством производимой продукции.

Тема № 2. Проектирование автоматизированной системы поддержки принятия решений для управления производством на предприятии.

Тема № 3. Проектирование экспертной системы для повышения эффективности деятельности технологической службы предприятия.

Тема № 4. Проектирование автоматизированной системы поддержки принятия решений для повышения эффективности деятельности отдела маркетинга и сбыта предприятия.

Тема № 5. Проектирование автоматизированной системы поддержки принятия решений для повышения эффективности деятельности финансовых

служб предприятия.

Тема № 6. Проектирование автоматизированной системы поддержки принятия решений для повышения эффективности оперативного планирования деятельности предприятия.

Тема № 7. Проектирование экспертной системы для оценки качества работы сотрудников вуза.

Тема № 8. Проектирование автоматизированной системы поддержки принятия решений для повышения эффективности стратегического планирования деятельности предприятия.

Тема № 9. Проектирование автоматизированной системы поддержки принятия решений для повышения эффективности вырабатываемых управлениемских решений по бизнес-планированию и/ или инвестиционному проектированию.

Тема № 10. Проектирование автоматизированной системы поддержки принятия решений для совершенствования процессов управления персоналом предприятия (организации).

Тема № 11. Проектирование экспертной системы для оценки конкурентоспособности предприятия.

Тема № 12. Проектирование экспертной системы для оценки инвестиционной привлекательности предприятия.

Тема № 13. Проектирование экспертной системы для оценки финансовой устойчивости предприятия.

Тема № 14. Проектирование экспертной системы для оценки производственного потенциала предприятия.

Тема № 15. Проектирование экспертной системы для оценки кадровых ресурсов предприятия.

Тема № 16. Проектирование модуля распознавания и верификации лиц, посещающих банк (кейс № 2 «Интеграция модуля компьютерного зрения в банковскую антифрод-систему»).

Тема № 17. Проектирование автоматизированной системы поддержки принятия решений для агрономов (кейс № 3 «Система поддержки принятия решений для агрономов»).

Тема № 18. Проектирование мультиагентной автоматизированной системы управления теплицей (кейс № 4 «Мультиагентная система управления теплицей»).

Тема № 19. Проектирование автоматизированной интеллектуальной системы анализа клиентских обращений (кейс № 5 «Интеллектуальная система анализа клиентских обращений»).

Тема № 20. Проектирование автоматизированной интеллектуальной системы для распознавания текста, классификации документов и проверки их полей (кейс № 10 «Автоматизация документооборота: OCR + валидация»).

Тема № 21. Проектирование интеллектуального ассистента оператора банка (кейс № 12 «Интеграция LLM в сервисы РСХБ (ассистент оператора)»).

Тема № 22. Проектирование автоматизированной интеллектуальной системы подбора оптимальных страховых пакетов для фермеров (кейс № 14 «Рекомендательная система агрострахования»).

Тема № 23. Проектирование автоматизированной интеллектуальной системы управления качеством данных (кейс № 15 «Управление качеством данных (Data Quality)»).

Тема 2.3. Содержание стандарта ГОСТ Р ИСО/ МЭК 25010-2015 «Информационные технологии. Системная и программная инженерия. Требования и оценка качества систем и программного обеспечения (SQuaRE). Модели качества систем и программных продуктов», его использование для оценки качества программного обеспечения

Практическая работа № 2. Адаптация и корректировка сформулированных в ГОСТ Р ИСО/ МЭК 25010-2015 модели и критериев оценки качества программ применительно к интеллектуальным информационным системам, реализующим параллельные вычислительные процессы.

Задание практической работы: Выполнить анализ показателей оценки качества программного обеспечения, сформулированных в ГОСТ Р ИСО/ МЭК 25010-2015, принимая во внимание основные классы интеллектуальных информационных систем – экспертные системы, автоматизированные системы поддержки принятия решений, нейронные сети и интеллектуальные сетевые сервисы дополнить набор показателей, указанных в ГОСТ Р ИСО/ МЭК 25010-2015.

Тема 2.4. Определение порядка верификации, валидации и испытаний автоматизированных информационных систем в стандарте ГОСТ Р 59792-2021 «Информационные технологии. Комплекс стандартов на автоматизированные системы. Виды испытаний автоматизированных систем»

Практическая работа № 3. Разработка программы автономных испытаний программного продукта, решающего задачи в области искусственного интеллекта с применением парализации вычислительных процессов.

Задание практической работы: Использовать приложение создания сети обработки изображений для разработки его программы автономных испытаний в соответствии с ГОСТ Р 59792-2021.

Тема 2.5. Виды и методики проведения тестирования программного обеспечения

Практическая работа № 4. Знакомство с инструментом функционального тестирования TestComplete.

Задание практической работы: За счёт применением любого скрипта, поддерживаемого TestComplete сгенерировать тест для функционального тестирования любого программного продукта, разработанного ранее или в рамках выполнения заданий практических работ № 7 -14.

Тема 3.1. Параллелизм задач в среде MS Visual Studio 2019

Практическая работа № 5. Знакомство с возможностями организации параллельных вычислений в среде MS Visual Studio 2019.

Задание практической работы: Имеется массив больших данных о клиентах банка, содержащих ошибки. Необходимо организовать параллельные вычислительные процессы в среде MS Visual Studio 2019 для анализа массива больших данных на наличие пустых полей, определения противоречивых значений полей (когда одни и те же поля, например, номер телефона гражданина, в разных записях с данными гражданина имеет разные значения), определение неверных или явно некорректных значений полей. (кейс № 15 «Управление качеством данных (Data Quality)»).

Тема 3.2. Лямбда-выражения и их применение в параллельных вычислительных процессах

Практическая работа № 6. Разработка лямбда-выражений при практической реализации параллельных вычислений (кейс № 15 «Управление качеством данных (Data Quality)»).

Задание практической работы: Разработать лямбда-выражения для решения задачи анализа массива больших данных на наличие пустых полей, определения противоречивых значений полей (когда одни и те же поля, например, номер телефона гражданина, в разных записях с данными гражданина имеет разные значения), определение неверных или явно некорректных значений полей.

Тема 3.3. Библиотека параллельных шаблонов (PPL) MS Visual Studio 2019

Практическая работа № 7. Практическое применение алгоритмов «parallel_for», «parallel_for_each», «parallel_transform», «parallel_sort» для реализации сортировки записей, содержащих информацию об оцениваемых альтернативах при подборе оптимальных страховых пакетов для фермеров (кейс № 14 «Рекомендательная система агрострахования»).

Задание практической работы: Имеется информация о финансовых возможностях фермеров и типам кредитов, выдаваемых банками, которые они могут получить в рамках агрострахования. Применить встроенные алгоритмы сортировки среды MS Visual Studio 2019 для одновременного проведения необходимых расчётов и упорядочения кредитов каждого фермера по срокам погашения, по размеру выплаченных финансовых средств.

Тема 3.4. Обработка исключений в среде MS Visual Studio 2019 с параллелизмом

Практическая работа № 8. Создание собственных исключений при реализации параллельных вычислений (кейс № 15 «Управление качеством данных (Data Quality)»).

Задание практической работы: Имеется массив больших данных о клиентах банка, содержащих ошибки. Необходимо организовать параллельные вычислительные процессы в среде MS Visual Studio 2019 для анализа массива больших данных на наличие пустых полей, определения противоречивых значений полей (когда одни и те же поля, например, номер телефона гражданина, в разных записях с данными гражданина имеет разные значения), определение неверных или явно некорректных значений полей. (кейс № 15 «Управление качеством данных (Data Quality)»). На каждую проблему исходных данных необходимо создать отдельный тип исключения, который будет её обрабатывать.

Тема 3.5. Рекомендации по работе со средой выполнения программ Visual Studio 2019 с параллелизмом

Практическая работа № 9. Оценка возможностей использования модели ветвление-соединение при организации параллельных вычислений.

Задание практической работы: выполнить анализ возможностей применения модели ветвление-соединение при решении задания практической работы № 5.

Тема 4.1. Создание сети обработки изображений

Практическая работа № 10. Создание сети обработки изображений.

Задание практической работы: Набор изображений в виде файлов формата .bmp хранится на диске.

Необходимо считать все доступные изображения, выполнить их анализ и преобразование следующим образом:

1. у любого изображения, созданного заданным автором, необходимо удалить зелёные и синие цвета,
2. у любого изображения, имеющего красный цвет, необходимо заменить его на черный,
3. любое другое изображение сделать чёрно-белым.

Тема 4.2. Программирование неиерархических и иерархических методов кластер-анализа данных с применением параллелизации задач

Практическая работа № 11. Практическая реализация иерархического алгоритма кластер-анализа данных (кейс № 10 «Автоматизация документооборота: OCR + валидация»).

Задание: Необходимо выполнить кластеризацию документов, сопровождающих процесс получения кредитов, поступающих вместе с кредитными заявками. В качестве признаков используются: наименование документа, автор документа, географическое размещение получателя кредита и его предприятия, сумма кредита, срок погашения кредита, размер предприятия – получателя кредита и ряд других. Применить для решения задачи ниже представленный алгоритм кластеризации (разработчик алгоритма ЦЭМИ РАН). Использовать параллельные вычисления для сокращения времени работы алгоритма.

Для иллюстрации прикладных аспектов кластер – анализа рассмотрим следующий алгоритм кластеризации. Алгоритм позволяет решить задачу классификации N объектов множества $O = \{O_1, O_2, \dots, O_N\}$, каждый из которых характеризуется подмножеством из множества $P = \{P_1, P_2, \dots, P_M\}$, состоящего из M признаков. Каждый объект O_i $i = \overline{1, N}$ описывается подмножеством признаков $\{P_k\} \in P$, являющимися качественным признаком образом. Предполагается, что каждый признак оценивается «хорошо» - 1, «удовлетворительно» - 0,5, «плохо» - 0. Все образы объектов систематизируются в матрицу образов:

$$V = \begin{pmatrix} V_{11} & V_{12} & \cdots & V_{1M} \\ V_{21} & V_{22} & \cdots & V_{2M} \\ \cdots & \cdots & \cdots & \cdots \\ V_{N1} & V_{N2} & \cdots & V_{NM} \end{pmatrix},$$

где V_{ij} – j -й признак i -го объекта;

$\vec{P} = (P_1, P_2, \dots, P_M)^T$ – вектор относительной значимости признаков, причем

$$\sum_{j=1}^M P_j = 1.$$

Пусть задан порог близости $DEL: 0 < DEL < 1$. Будем считать мерой близости между q -ым и l -ым объектами функцию $\rho(q, l) = 1 - \sum_{j=1}^M |V_{qj} - V_{lj}| \cdot P_j$.

Предполагаем, что функция

$$RAST(q, l) = \begin{cases} 1 & \text{если } \rho(q, l) \geq DEL \\ 0 & \text{если } \rho(q, l) < DEL \end{cases}$$

характеризует расстояние между q и l объектами.

Пусть N объектов распределены на группы, количество которых KG . Произведенное разбиение может быть охарактеризовано следующими показателями.

1) Мера взаимной близости i -ой и k -ой группы объектов

$$MI_{ik} = \frac{1}{DLIN(i)DLIN(k)} \sum_{r=1}^{DLIN(i)} \sum_{q=1}^{DLIN(k)} RAST(S_r^{(i)}, S_q^{(k)}), \text{ где}$$

$S_r^{(i)}$ – номер r -го объекта из i -й группы; $S_q^{(k)}$ – номер q -го объекта из k -й группы; $DLIN(i)$ – число объектов в i -й группе; $DLIN(k)$ – число объектов в k -й группе.

2) Мера взаимосвязи i -й группы с остальными группами:

$$SVAZ(i) = \frac{1}{KG - 1} \sum_{K=1, k \neq i}^{KG} MI(i, k).$$

3) Мера сплоченности (компактности) i -й группы:

$$KOMP(i) = MI(i, i) - SVAZ(i).$$

$$4) \text{ Усредненная мера взаимосвязи групп } ISR = \frac{1}{KG} \sum_{i=1}^{KG} MI(i,i).$$

$$5) \text{ Усредненная мера сплочённости групп } VSR = \frac{1}{KG} \sum_{i=1}^{KG} SVAZ(i).$$

Качество произведённого разбиения можно определить по критерию: KRIT = ISR – VSR, KRIT → max, для которого должно быть найдено соответствующее KG – количество групп.

Будем получать каждое новое разбиение из старого объединением каких-либо двух групп, для этого объединяем такие группы i и k, у которых мера близости MI(i,k) → max. На каждом шаге подсчитывается значение критерия и считается то разбиение (того шага), для которого KRIT → max. Требуется постоянно хранить матрицу и меры близостей групп. На первом шаге объединяют в группы те объекты, признаки которых с точностью заданного порога близости DEL одинаковые. Решение задачи состоит из 2-х этапов: 1) образование начального разбиения и 2) получение оптимального разбиения.

Для успешной работы описанного выше алгоритма кластеризации выполняется предварительная классификация (первый этап) – объекты объединяют в группы таким образом, чтобы $RAST(p,l) = 1$ для любой пары p, l ; если p и l принадлежат одной группе.

1) Пусть SK и PK – массивы длиной N. Если PK(i)=0, то i-й объект есть представитель i-й группы. Если PK(i) ≠ 0 =k, то i-й объект входит в k-ю группу.

Вначале все PK(i)= 0, i=1, 2 ..., N, то есть каждый объект есть представитель своей собственной группы.

2) Организуем цикл по $i = 1, 2, \dots, N$

если PK(i) ≠ 0, то переход к следующему i;

если PK(i) = 0, то строим массив связей SK i-го (элемента) объекта со всеми следующими:

$SK(j) = RAST(i, j) \quad j = i+1, i+2, \dots, N$

3) Затем движемся от (i+1) - го элемента массива SK до тех пор, пока не встретим первую единицу. Организуем цикл $j = i+1, i+2, \dots, N$. Если SK(j)=1 и PK(j)=0, то j-й объект объединяем в одну группу с i-м, делая для этого PK(j)=i.

4) Организуем далее цикл $l = j+1, j+2, \dots, N$ и осуществляем корректировку массива SK:

$SK(l) = SK(l) \times RAST(j, l)$ Таким образом, информация об образованных группах сосредотачивается в массиве PK. Количество групп $KG < N$, чем меньше DEL тем меньше KG.

Вторым этапом является оптимальная классификация.

Вектор PK, полученный на этапе предварительной классификации преобразуется следующим образом: элементы, входящие в одну группу, связываются в циклическую цепь (каждый последующий элемент ссылается на предыдущий, а последний элемент ссылается на первый).

Объединение i и k группы состоит из двух шагов 1) Изменение указателей для образования цепи, связывающей элементы объединенной группы. При

этом последний элемент i -й группы ссылается на первый элемент k -й группы, а последний элемент k -й группы на первый элемент в i -й группе.

2) Значение $DLIN(k) = 0$, то есть k -я группа ликвидируется.

Все характеристики вновь полученного разбиения пересчитываются. Элементы матрицы MI все пересчитывать не требуется.

1) Для пересчета элементов $MI(1, i), MI(2, i), \dots, MI(i-1, i)$

Используется формула:

$$MI(j,i) = \frac{DLIN(i)MI(j,i) + DLIN(k)MI(j,k)}{DLIN(i) + DLIN(k)}, \quad j = 1, 2, \dots, i-1;$$

2) Для $MI(i,i)$

$$MI(i,i) = \frac{[DLIN(i)]^2 MI(i,j) + [DLIN(k)]^2 MI(k,k) + 2DLIN(i)DLIN(k)MI(i,k)}{[DLIN(i) + DLIN(k)]^2};$$

3) Для пересчета элементов

$MI(i, i+1), MI(i, i+2), \dots, MI(i, KG)$

$$MI(i,j) = \frac{DLIN(i)MI(i,j) + DLIN(k)MI(k,j)}{DLIN(i) + DLIN(k)}, \quad j = i+1, i+2, \dots, KG.$$

Тема 4.3. Применение параллелизации вычислительных процессов для генерации решений в системах поддержки принятия решений

Практическая работа № 12. Применение параллелизации вычислительных процессов для реализации модуля оценки принимаемого решения с использованием множества критериев (показателей). Практическая работа проводится в форме хакатона.

Ведущий конкурса-хакатона распределяет участников по командам методом жеребьевки или по желанию студентов. Все участники делятся на 5 команд по 3 человека. Команды выбирают своего капитана, который будет координировать работу команды и представлять готовый проект. Капитан команды распределяет роли внутри членов команды (например, все участники занимаются разработкой алгоритма принятия решений и соответствующего ему дерева решений, один участник – формализацией исходных данных, второй – формализацией алгоритма принятия решений, третий – разработкой пользовательского интерфейса).

Наставники из числа преподавателей выбирают подшефную команду и сопровождают её на протяжении всего конкурса. Наставники также помогают командам выбрать капитана, могут давать советы своим командам по реализации проекта, а также помогают капитанам делегировать обязанности внутри команды.

Каждая команда должна разработать программный модуль, способный оказывать помощь в подборе кандидатов на должности менеджеров высшего звена и специалистов среднего звена банка. На реализацию проекта командам отводится 4 часа.

Этапы работы:

1. Формализация исходных данных:

- разработка критериев отбора, шкал и методов оценки данных кандидатов или их резюме,
- представление исходных данных задачи в виде базы данных или типизированного файла;
- определение структуры портфолио кандидатов, анкет, предлагаемых тестов и результатов тестирования, собеседования, форм, заполняемых по результатам собеседования с кандидатами, в качестве ориентира следует использовать формы резюме, предлагаемые hh.ru, superjob.ru.

2. Разработка алгоритма принятия решения:

- разработка алгоритма анализа данных и подбора менеджеров высшего звена,
- разработка алгоритм анализа данных и подбора специалистов среднего звена,
- реализация на языке программирования алгоритма и дерева принятия решений.

3. Разработка интерфейса программного продукта:

- разработка форм ввода исходных данных,
- разработка интерфейса для взаимодействия с лицом, принимающим решение.

4. Апробация программного средства:

- ввод исходных данных, соответствующих кандидатам-менеджерам высшего звена,
- ввод данных о вакансиях,
- ввод исходных данных, соответствующих кандидатам-специалистам среднего звена,
- формирование программой управленческого решения, его анализ разработчиками.

Ожидаемый результат:

На основе введённых данных кандидатов на замещение вакантных должностей, результатов анализа их портфолио, результатов собеседования, формального отбора по квалификационным требованиям необходимо подобрать кандидатов на две вакансии для менеджеров высшего звена или специалистов высокой квалификации, пять кандидатов для менеджеров среднего звена и специалистов.

Задание практической работы (хакатона): Автоматизировать процесс формирования управленческих решений по подбору персонала банка. Для этого разработать дерево принятия решений (метод иерархий), использовать его для генерации решений программой, реализующей параллельные вычисления.

Персонал любой организации играет важную роль в результатах её деятельности. Поэтому руководство Россельхозбанка уделяет должное внимание вопросам подбора персонала и привлечения молодых специалистов, в том

числе выпускников и студентов вуза. В банке используют следующие методы набора персонала:

1. Разработаны анкеты-резюме, которые каждый работник может заполнить и опустить в ящик, находящийся в любом отделении банка и принадлежащий службе управления кадров.

2. Объявления о вакансиях помещаются в профильных изданиях –«Коммерсанте», «Ведомостях». Эта информация предназначена не для людей, активно ищущих работу, а для работающих профессионалов, которые по тем или иным причинам готовы поменять свое рабочее место.

3. Информация о вакансиях размещается на Интернет-платформах онлайн рекрутинга. Ежедневно приходит до 100 анкет.

4. Организуется стажировка студентов высших учебных заведений.

5. Размещается информация о вакансиях на баннерах на сайтах других компаний, специализирующихся на различных направлениях бизнеса (в зависимости от потребности в специалистах той или иной квалификации).

6. Для кандидатов проводятся «дни карьеры», предусматривающие собеседования с несколькими руководителями подразделений банка.

7. В «днях карьеры» участвуют кадровые агентства и вузы.

8. Для подбора топ-менеджеров банк пользуется услугами рекрутёрских агентств, так как на такие позиции привлекаются в том числе западные специалисты.

Отбор персонала проходит по следующей схеме:

- первичный отбор проводит отдел подбора кадров (по резюме);
- профессиональный отбор кандидатов проводят служба управления кадров и специалисты подразделений банка (включая тесты и опросники, разработанные ведущими специалистами структурных подразделений банка, и собеседование);
- наконец, проводится интервью руководителем подразделения, в которое подбирается кандидат.

Для решения задачи следует рассмотреть все методы, которыми осуществляется набор персонала на разные должности с учётом квалификации, разработать алгоритм и дерево принятия решений. Алгоритм принятия решений апробировать для подбора персонала на замещение вакансий. На данный момент имеется две вакансии для менеджеров высшего звена или специалистов высокой квалификации, пять вакансий для менеджеров среднего звена и специалистов, которых необходимо принять для работы в банке. С учётом известного количества вакансий по конкретным должностям сформировать на основе алгоритма и дерева принятия решений управлеченческое решение о наборе персонала. Считается, что в отборе на должности менеджеров высшего звена претендуют не менее двадцати человек, на должности специалистов среднего звена – не менее ста человек.

Тема 4.4. Формирование базы знаний и ранжирование альтернатив с применением параллельных вычислительных процессов при разработке экспертных систем

Практическая работа № 13. Формирование базы знаний и ранжирование альтернатив с применением параллельных вычислительных процессов при разработке экспертных систем (кейс № 3 «Система поддержки принятия решений для агрономов»).

Задание: В соответствии с заданием кейса разрабатывается автоматизированная система поддержки принятия решений (АСПР или DSS - Decision Support System), которая помогает агрономам принимать решения по посевам и удобрениям. По техническому заданию АСПР должна в своей структуре интегрировать ряд модулей, выполняющих следующие функции: прогноз урожайности (ML-модели на исторических и климатических данных), данные с IoT-сенсоров о состоянии почвы и рекомендационные алгоритмы.

В рамках настоящей практической работы необходимо разработать модуль сбора знаний экспертов, набор правил и алгоритмов принятия решений по вопросам полива растений, внесения удобрений, пестицидов и гербицидов. Причём в рамках модуля должен быть реализован алгоритм выбора альтернатив по видам удобрений, пестицидов и гербицидов, проводимым полеводческим работам.

Вопросы к зачёту по дисциплине

- 1 Каковы базовые принципы методологии объектно-ориентированного подхода?
- 2 Что означает и какие элементы включает класс как структура программирования в C++?
- 3 Какие инструменты проектирования автоматизированных систем с применением объектно-ориентированного подхода Вы знаете?
- 4 Что означает UML?
- 5 Какие типы UML-диаграмм Вам известны? Охарактеризуйте их нотации.
- 6 Какие case-средства поддерживают язык унифицированного моделирования UML?
- 7 Какие технологии программирования существуют в рамках объектно-ориентированной методологии программирования и проектирования программного обеспечения?
- 8 Какие базовые положения объектно-ориентированной технологии программирования применительно к языку программирования C++?
- 9 Какие типы модулей у программ, разработанных на объектно-ориентированном языке программирования C++?
- 10 Какие модификаторы доступа языка программирования C++ позволяют реализовать механизм инкапсуляции?
- 11 Как осуществляется перегрузка методов и конструкторов в объектно-ориентированных средах и в MS Visual Studio 2019?
- 12 Как реализуется механизм полиморфизма в C++?

- 13 В чём заключаются преимущества использования полиморфизма при разработке программ?
- 14 Как реализуется наследование в объектно-ориентированных средах и в MS Visual Studio 2019?
- 15 В чём заключаются достоинства наследования при разработке программ?
- 16 Какого типа наследование реализовано в языке C++, его возможности при разработке программ?
- 17 Как можно охарактеризовать UML, какие программные средства позволяют строить UML-диаграммы?
- 18 Какие типы диаграмм включает унифицированный язык моделирования UML?
- 19 Какие нотации имеют UML-диаграммы вариантов использования, для чего они применяются?
- 20 Какие нотации имеют UML-диаграммы классов, для чего они могут использоваться?
- 21 Какие нотации имеют UML-диаграммы взаимодействия, для чего они могут использоваться?
- 22 Какие нотации имеют UML-диаграммы деятельности, для чего они могут использоваться?
- 23 Какие нотации имеют UML-диаграммы состояний, для чего они могут использоваться?
- 24 Какие нотации имеют UML-диаграммы компонентов, для чего они могут использоваться?
- 25 Какие нотации имеют UML-диаграммы размещения, для чего они могут использоваться?
- 26 Какие нотации имеют UML-диаграммы последовательности, для чего они могут использоваться?
- 27 Какие нотации имеют кооперативные UML-диаграммы, для чего они могут использоваться?
- 28 Какие нотации имеют UML-диаграммы реализаций, для чего они могут использоваться?
- 29 Что такое «модель жизненного цикла программного обеспечения», какие модели Вам известны?
- 30 Какие стадии и этапы жизненного цикла автоматизированных систем отражены в стандарте ГОСТ Р 59793 «Автоматизированные системы. Стадии создания»?
- 31 Какие процессы жизненного цикла программных продуктов отражаются в стандарте ГОСТ Р ИСО/МЭК 12207-2010 «Информационная технология. Системная и программная инженерия. Процессы жизненного цикла программных средств»?
- 32 Какие основные положения стандарта ГОСТ Р ИСО/ МЭК 25010-2015 «Информационные технологии. Системная и программная инженерия. Требования и оценка качества систем и программного обеспечения (SQuaRE). Модели качества систем и программных продуктов»?

- 33 С какими этапами жизненного цикла программных продуктов связана группа процессов соглашения, обозначенная в ГОСТ Р ИСО/МЭК 12207-2010?
- 34 С какими этапами жизненного цикла программных продуктов связана группа процессов проекта, обозначенная в ГОСТ Р ИСО/МЭК 12207-2010?
- 35 С какими этапами жизненного цикла программных продуктов связана группа технических процессов, обозначенная в ГОСТ Р ИСО/МЭК 12207-2010?
- 36 С какими этапами жизненного цикла программных продуктов связана группа процессов организационного обеспечения проекта, обозначенная в ГОСТ Р ИСО/МЭК 12207-2010?
- 37 С какими этапами жизненного цикла программных продуктов связана группа процессов реализации программного средства, обозначенная в ГОСТ Р ИСО/МЭК 12207-2010?
- 38 С какими этапами жизненного цикла программных продуктов связана группа процессов поддержки программного средства, обозначенная в ГОСТ Р ИСО/МЭК 12207-2010?
- 39 С какими этапами жизненного цикла программных продуктов связана группа процессов повторного применения программных средств, обозначенная в ГОСТ Р ИСО/МЭК 12207-2010?
- 40 Как можно охарактеризовать модель качества программного обеспечения?
- 41 Какие показатели качества программных продуктов указаны в ГОСТ Р ИСО/ МЭК 25010-2015?
- 42 Что такое верификация программного продукта, каковы её цели и задачи?
- 43 Какие методы верификации программного обеспечения Вам известны?
- 44 Какие известны этапы проведения верификации программного обеспечения?
- 45 Что такое валидация программного продукта, каковы её цели и задачи?
- 46 Какие методы валидации программного обеспечения Вам известны?
- 47 Какие виды испытаний программного обеспечения определены в ГОСТ 34.603-92 «Информационная технология. Виды испытаний автоматизированных систем»?
- 48 Как разрабатывается программа автономных испытаний, каковы её разделы?
- 49 Как разрабатывается программа комплексных испытаний, каковы её разделы?
- 50 Как разрабатывается программа приёмочных испытаний, каковы её разделы?
- 51 В чём заключаются цели, задачи и принципы тестирования программного обеспечения?
- 52 Как можно классифицировать методики тестирования программного обеспечения?
- 53 Как реализована технология параллельного программирования MS Visual Studio 2019?

54 Какие программные компоненты включает среда программирования MS Visual Studio 2019 для программирования параллельных вычислительных процессов?

55 Какую архитектуру имеет программная среда MS Visual Studio 2019 с возможностями организации параллельных вычислений?

56 Какие компоненты имеет библиотека параллельных шаблонов (PPL) MS Visual Studio 2019?

57 Что такое библиотека асинхронных агентов MS Visual Studio 2019, какое она имеет назначение?

58 Что такое менеджер ресурсов среды программирования MS Visual Studio 2019 с возможностями организации параллельных вычислений, какова его основная функция?

59 Какие функции выполняет менеджер ресурсов среды программирования MS Visual Studio 2019?

60 Что означает лямбда-выражение?

61 Какой синтаксис у лямбда-выражений?

62 Для сего используются лямбда-выражения?

63 Какие алгоритмы включаются в библиотеку параллельных шаблонов (PPL) MS Visual Studio 2019?

64 Какое назначение у алгоритма «parallel_for»?

65 Какое назначение у алгоритма «parallel_for_each»?

66 Какое назначение у алгоритма «parallel_transform»?

67 Какое назначение у алгоритма «parallel_reduce»?

68 К какой группе относятся алгоритмы «parallel_sort», «parallel_buffered_sort», «parallel_radixsort»? Какие классы задач они решают?

69 Почему необходимо использовать механизм исключений при разработке программ, реализующих параллельные вычислительные процессы на C++?

70 Для каких типов ошибок, возникающих при выполнении программ с параллельными вычислительными процессами, необходимо разрабатывать собственные механизмы обработки исключений?

71 Какие исключения создаются при выполнении параллельных вычислительных процессов в программах?

72 Как осуществляется обработка исключений средой выполнения MS Visual Studio 2019 программ с параллельными процессами?

73 Почему при разработке программ с параллельными процессами необходима тщательная оценка решаемой задачи перед применением модели ветвления-соединения?

74 Для чего используется алгоритм «parallel_invoke»?

75 Что такое ложный доступ, почему от него отказываются при разработке программ?

76 Какие инструменты MS Visual Studio 2019 используются для обработки изображений?

77 Какие операции могут выполняться над изображениями?

- 78 Как происходит в модели потока данных взаимодействие независимых компонентов программы друг с другом?
- 79 Какие действия выполняют на этапах создания сети изображений?
- 80 Какие Вы знаете элементы сети обработки изображений?
- 81 Что такое кластеризация данных?
- 82 Как можно классифицировать алгоритмы кластеризации?
- 83 Как можно охарактеризовать неиерархические алгоритмы кластеризации? Приведите примеры таких алгоритмов.
- 84 Как можно охарактеризовать иерархические алгоритмы кластеризации? Приведите примеры таких алгоритмов.
- 85 Приведите примеры практической реализации алгоритмов кластеризации с применением параллельных вычислений в MS Visual Studio 2019?
- 86 Какие метрики для оценки степени близости кластеризуемых объектов и степени сплочённости кластер-групп, различия различных кластер-групп Вы знаете?
- 87 Какие методы генерации решений при реализации игровых стратегий в компьютерных играх Вы знаете?
- 88 Как можно использовать распараллеливание вычислительных процессов для представления конечных автоматов и генерации решений с их помощью?
- 89 Как можно использовать распараллеливание вычислительных процессов для представления деревьев решений (метода иерархий) и генерации решений с их помощью?
- 90 Как можно использовать распараллеливание вычислительных процессов для представления когнитивных карт и генерации решений с их помощью?
- 91 Как можно использовать распараллеливание вычислительных процессов для нахождения оптимальных управлеченческих решений с помощью эволюционных алгоритмов?
- 92 Какие модели можно использовать для представления знаний в экспертных системах?
- 93 Какие методы экспертной оценки можно использовать для сбора знаний с целью наполнения базы знаний экспертной системы?
- 94 Как производится оценка согласованности мнений экспертов при формировании базы знаний?
- 95 Как с помощью параллельных вычислений можно реализовать вывод на знаниях?
- 96 Что такое нечёткая логика и как она используется для формализации знаний и вывода на знаниях в экспертных системах?

Примеры тестовых заданий

- 1). Установите соответствие между названием стадий жизненного цикла автоматизированной системы по ГОСТ Р 59793-2021 «Автоматизированные системы. Стадии создания» и содержанием работ, на них выполняемых:
 - Формирование требований к автоматизированной системе - Сбор данных об

объекте автоматизации и осуществляемых видах деятельности, проведение оценки качества функционирования объекта, выявление проблем, решение которых возможно средствами автоматизации, оценка (технико-экономическая, социальная и т. и.) целесообразности создания автоматизированной системы, подготовка исходных данных для формирования требований к автоматизированной системе с характеристикой объекта автоматизации, описанием требований к системе, ограничениями допустимых затрат на разработку, формулировку и оформление требований пользователей (заказчика) к автоматизированной системе.

- Разработка концепции автоматизированной системы – Организация-разработчик проводит детальное изучение объекта автоматизации и необходимые научно-исследовательские работы (НИР), оформляют и утверждают отчеты о НИР, проводят разработку альтернативных вариантов концепции создаваемой автоматизированной системы и планов их реализации, проводят оценку необходимых ресурсов на их реализацию и обеспечение функционирования, а также оценку преимуществ и недостатков каждого варианта. Далее выполняют сопоставление требований пользователя и характеристик предлагаемой системы, выбор оптимального варианта, определение порядка оценки качества и условий приемки системы, проводят оценку эффектов, получаемых от системы.

- Техническое задание – Разработка и утверждение заказчиком и исполнителем работ технического задания на проектирование автоматизированной системы.

Эскизный проект - Разработка предварительных проектных решений по системе и ее частям, в рамках которой определяются функции автоматизированной системы, функции подсистем, их цели и назначение, состав комплексов задач и отдельных задач, решаемых автоматизированной системой, концепции информационной базы, её укрупненная структура, подбор системы управления базой данных, определение состава вычислительной системы, функций и параметров основных программных средств.

2). Установите соответствие между группой процессов, определённых ГОСТ Р ИСО/МЭК 12207—2010 и их характеристиками:

Процессы соглашения – Группа процессов жизненного цикла программного продукта определяет действия, необходимые для выработки соглашений между двумя организациями. Если реализуется процесс приобретения, то он обеспечивает средства для проведения деловой деятельности с поставщиком продуктов, предоставляемых для применения в функционирующей системе, услугах поддержки этой системы или элементах системы, разработанных в рамках проекта.

Процессы организационного обеспечения проекта – Группа процессов жизненного цикла программного продукта позволяет осуществлять менеджмент возможностей организаций приобретать и поставлять продукты или услуги через инициализацию, поддержку и управление проектами. Эти процессы обеспечивают ресурсы и инфраструктуру, необходимые для поддержки проектов,

и гарантируют удовлетворение организационных целей и установленных соглашений. Они не претендуют на роль полной совокупности деловых процессов, реализующих менеджмент деловой деятельности организации.

Процессы проекта – Данная группа процессов жизненного цикла программного продукта включает процессы менеджмента проекта, которые используются для планирования, выполнения, оценки и управления продвижением проекта, и процессы поддержки проекта, которые обеспечивают выполнение специализированных целей менеджмента. В рамках данной группы процессов выполняется создание и развитие планов проекта, оценка фактического выполнения и продвижения относительно плановых заданий и управление выполнением проекта вплоть до полного его завершения.

3). Комплекс характеристик программного продукта, определяющих способность выполнять возложенные на него функции, называется программного обеспечения (ответ указать в именительном падеже)

4). К показателям качества программного обеспечения можно отнести следующие три показателя:

- функциональность,
- надежность,
- удобство использования,
- лёгкость,
- отказоустойчивость.

5). Поставьте в соответствие характеристикам показателей качества программного обеспечения, определённым в стандарте ISO/IEC 25010:2011, название показателей:

- *Функциональность* – Этот показатель означает, что программный продукт выполняет возложенные на него задачи, отвечает заданным потребностям пользователей. Данный аспект предполагает правильную и точную работу, совместимость всех входящих в состав компонентов.
- *Надежность* – Этот показатель означает бесперебойное выполнение программным продуктом возлагаемых на него задач на заданных условиях в течение установленного времени.
- *Юзабилити (удобство использования)* - Этот параметр характеризует степень удобства программного обеспечения для пользователей, его наглядность, легкость эксплуатации и изучения.
- *Эффективность* – Этому показателю соответствует степень обеспечения программным продуктом необходимой производительности при заданных условиях.

6). В зависимости от возможностей запуска кода на исполнение при тестировании программного обеспечения его можно классифицировать на (укажите два метода):

- статическое,
- динамическое,
- детерминированное,
- стохастическое,

- предметное.

7). Установите соответствие между процедурой тестирования и её содержанием при рассмотрении классификации видов тестирования программного обеспечения в зависимости от целей тестирования:

- *Функциональное тестирование* (functional testing) — Направлено на проверку корректности выполнения функций, реализованных в программном продукте.
- *Нефункциональное тестирование* (non-functional testing) — Тестирование атрибутов компонентов или автоматизированной системы, не относящихся к обеспечению её функциональности.
- *Тестирование производительности* (performance testing) — Определение стабильности и потребления ресурсов в условиях различных сценариев использования и нагрузок, возникающих при работе программного продукта.
- *Нагрузочное тестирование* (load testing) — Определение или сбор показателей производительности и времени отклика программно-технической системы или устройства в ответ на внешний запрос с целью установления соответствия требованиям, предъявляемым к данной автоматизированной системе (устройству).
- *Тестирование масштабируемости* (scalability testing) — Тестирование программного продукта или автоматизированной системы, которое измеряет производительность компьютерной сети или автоматизированной системы в условиях, когда количество пользовательских запросов увеличивается или уменьшается.

8). Обследование объекта и обоснование необходимости создания автоматизированной системы, а также формирование требований пользователей (заказчика) к автоматизированной системе осуществляется на этапе «..... к автоматизированной системе», определённом в стандарте ГОСТ Р 59793-2021 Автоматизированные системы. Стадии создания. (В ответе указать название стадии жизненного цикла автоматизированной системы по контексту фразы)

9). Изучение объекта автоматизации, проведение необходимых научно-исследовательских работ, разработка вариантов концепции автоматизированной системы, выбор варианта концепции автоматизированной системы, удовлетворяющего требованиям пользователя, для последующего проектирования, выполняется на этапе «..... автоматизированной системы», определённом в стандарте ГОСТ Р 59793-2021 Автоматизированные системы. Стадии создания. (В ответе указать название стадии жизненного цикла автоматизированной системы по контексту фразы)

10). Разработка и утверждение заказчиком и исполнителем работ технического задания на проектирование автоматизированной системы осуществляется на этапе «.....», определённом в стандарте ГОСТ Р 59793-2021 Автоматизированные системы. Стадии создания. (В ответе указать название стадии жизненного цикла автоматизированной системы)

11). Разработка проектных решений автоматизированной системы, её обеспечивающих подсистем и программных модулей, разработка заданий на проектирование в смежных частях проекта объекта автоматизации, разработка и оформление документации на поставку изделий для комплектования автоматизированной системы и (или) технических требований (технических заданий) на их разработку осуществляется на этапе «.....», определённом в стандарте ГОСТ Р 59793-2021 Автоматизированные системы. Стадии создания. (В ответе указать название стадии жизненного цикла автоматизированной системы)

12). Разработка рабочей документации на автоматизированную систему и ее части, разработка или адаптация программного обеспечения и других подсистем автоматизированной системы осуществляется на этапе «.....», определённом в стандарте ГОСТ Р 59793-2021 Автоматизированные системы. Стадии создания. (В ответе указать название стадии жизненного цикла автоматизированной системы)

13). Подготовка объекта автоматизации к вводу автоматизированной системы в действие, подготовка и обучение персонала, проведение строительно-монтажных, пусконаладочных работ, необходимых для эксплуатации автоматизированной системы, проведение предварительных испытаний, опытной эксплуатации и приёмочных испытаний автоматизированной системы осуществляется на этапе, определённом в стандарте ГОСТ Р 59793-2021 Автоматизированные системы. Стадии создания. (В ответе указать название стадии жизненного цикла автоматизированной системы с основным словом в именительном падеже)

14). Установите соответствие между понятиями:

- *Тестирование программного обеспечения* (*Software Testing*) — Проверка соответствия программного продукта предъявляемым требованиям, обеспечение уверенности в качестве программного обеспечения, поиск очевидных ошибок в программном обеспечении, которые должны быть выявлены до того, как их обнаружат пользователи программы.

- *Верификация* (*verification*) — Процесс оценки автоматизированной системы, чтобы понять, удовлетворяют ли результаты текущего этапа разработки условиям, которые были сформулированы в его начале.

- *Валидация* (*validation*) — Определение соответствия разрабатываемого программного обеспечения ожиданиям и потребностям пользователя, его требованиям к системе.

15). Упорядочите этапы тестирования программного продукта в логической и временной последовательности их выполнения:

Этап 1 - Анализ разработанного программного продукта

Этап 2 – Изучение требований к разработке программного продукта

Этап 3 - Разработка стратегии тестирования и планирование процедур контроля качества программного продукта

Этап 4 - Создание тестовой документации и тестов для программного продукта

Этап 5 - Тестирование прототипа программного продукта

Этап 6 - Основное тестирование программного продукта

Этап 7 – Стабилизация программного продукта

Этап 8 – Передача в эксплуатацию программного продукта.

16) Основные принципы объектно-ориентированной методологии проектирования программного обеспечения:

- инкапсуляция,
- декомпозиция,
- иерархия,
- робастность.

17). Объектную модель определяют:

- инкапсуляция,
- модульность,
- иерархия,
- структурированность,
- абстрагирование,
- однотипные объекты.

18). Дополнительные элементы объектной модели:

- типизация,
- абстрагирование,
- параллелизм,
- устойчивость,
- эквивалентность.

19) Основные принципы объектно-ориентированной технологии программирования:

- инкапсуляция,
- полиморфизм,
- устойчивость,
- наследование,
- структурированность.

20). Инкапсуляция - это

- подход, позволяющий изолировать интерфейс объекта, отражающий его внешнее поведение, от внутренней реализации объекта,
- подход, базирующийся на функциональной декомпозиции,
- подход, позволяющий создать иерархию классов,
- это ограничение, накладываемое на класс объектов и препятствующее взаимозаменяемости различных классов.

21). Объектно-ориентированный подход базируется на ...

- функциональной декомпозиции исследуемого объекта или системы,
- объектной декомпозиции исследуемого объекта или системы,
- декомпозиции системы на классы, объединяющие однотипные объекты.

22). Инкапсуляцию классов регулируют специальные ключевые слова, именуемые ...

- литералами,

- модификаторами доступа,

- ключами,

- управляемыми конструкциями.

23). Однотипные объекты группируются в ...

- поля,

- методы,

- классы,

- модули.

24). Классы группируются в

- файлы,

- пакеты,

- группы,

- подсистемы.

25). Класс состоит из ...

- тело класса,

- модуль класса,

- пакет,

- заголовок класса.

26). Класс может включать

- поля,

- методы,

- конструкторы,

- атрибуты,

- показатели.

27). UML – это

- язык программирования класса 4GL,

- язык манипуляции реляционными данными,

- унифицированный язык моделирования,

- средство управления проектами по разработке программного обеспечения.

28). Классы, доступные из любых классов и любых пакетов имеют модификатор доступа ...

- private,

- public,

- native.

29). Классы, доступные в пределах данного пакета имеют модификатор доступа ...

- пустой (без модификатора),

- private,

- abstract,

- public.

30). Классы, к которым имеют доступ только объекты этого же класса, имеют модификатор доступа

- пустой (без модификатора),

- abstract,

- public,
- private,
- protected.

31). Классы, к которым имеют доступ только классы-наследники, имеют модификатор доступа

- пустой (без модификатора),
- abstract,
- public,
- private,
- protected.

32). Свойство классов и объектов, позволяющее заимствовать черты других классов, называется

- инкапсуляция,
- модульность,
- наследование,
- подобие.

33). Механизм объектно-ориентированных сред, позволяющий разработчикам использовать одноимённые методы и конструкторы, называется

- перегрузка,
- абстрагирование,
- полиморфизм,
- устойчивость.

34). Одноимённые методы и конструкторы, содержащиеся в классе, называются ...

- абстрактные,
- перегруженные,
- финальные,
- динамические.

35). Механизм, позволяющий повысить надёжность программ, разрабатываемых в объектно-ориентированных средах, называется

- механизм виртуальной памяти,
- механизм исключений,
- механизм двойной защиты,
- механизм интерфейсов.

36). Все ошибки, возникающие в программах, разработанных на языке C++, относятся к

- классу Exception,
- классу Error,
- классу RuntimeException,
- классу IOException,
- разным классам.

37). Все исключения, возникающие в программах, разработанных на языке C++, относятся к классу

- exception,

- error,
- RuntimeException,
- IOException.

38). Для выявления участков программы, на которых могут возникать ошибочные и сбойные ситуации, используется конструкция ..., которая позволяет обозначить охраняемый участок программы:

- try – catch,
- do – while,
- for-break,
- break – continue.

39). Установите соответствие между описанием механизма объектно-ориентированного программирования в C++ и их названием

- *Полиморфизм времени компиляции* (статический полиморфизм или перегрузка методов) – Механизм объектно-ориентированного программирования, при котором функция или оператор могут быть перегружены несколькими реализациями, которые отличаются по количеству или типам параметров. Подходящая реализация выбирается на основе типов аргументов во время компиляции.

- *Полиморфизм времени выполнения* (динамический полиморфизм или переопределение методов) - Механизм объектно-ориентированного программирования, при котором виртуальная функция, определённая в базовом классе, переопределяется функцией с теми же именем и сигнатурой в производном классе. Подходящая функция для вызова определяется во время выполнения на основе фактического типа объекта, на который указывается.

- *Одиночное (единственное) наследование* – Механизм объектно-ориентированного программирования, при котором класс–наследник создаётся от одного родительского класса.

- *Множественное наследование* – Механизм объектно-ориентированного программирования, при котором класс–наследник создаётся от двух родительских классов.

- *Виртуальное наследование* - Механизм объектно-ориентированного программирования, при котором класс–наследник создаётся из класса, в котором есть виртуальные методы.

40). В языке программирования C++ существуют следующие виды полиморфизма:

- полиморфизм времени компиляции,
- полиморфизм времени выполнения,
- полиморфизм времени сборки программы,
- полиморфизм времени исполнения программы.

41). Уничтожение объектов в программах, разработанных на языке C++, выполняется с помощью:

- конструкторов;
- деструкторов;
- специального метода System.gc();

- методом, разработанным программистом,
- фрагментом кода, написанным программистом.

42). Имя конструктора, разработанного в классах программ на языке С++

- совпадает с именем класса, в котором он содержится,
- совпадает с именем одного из методов класса,
- совпадает с именем пакета, включающего содержащий его класс,
- имеет уникальное имя в пределах класса,
- имеет уникальное имя в пределах пакета.

43). Диаграммы классов отображают

- информационные процессы, протекающие в исследованной предметной области,
- отношения между объектами и классами,
- статическую структуру предметной области (объектной модели) или программы,
- динамические процессы, протекающие в предметной области и отражённые в объектной модели.

44). Диаграммы взаимодействия показывают

- процессы обмена сообщениями (информацией) между классами,
- порядок взаимодействия классов и объектов,
- хронологию взаимодействия объектов классов во времени,
- отдельные процессы обмена информацией.

45). Нотации диаграмм взаимодействия включают:

- прямоугольники, обозначающие классы,
- направленные стрелки,
- овалы, обозначающие сообщения,
- вертикальные прямые, обозначающие линии жизни объектов.

46). Диаграммы вариантов использования позволяют представить

- бизнес-процессы, протекающие в предметной области,
- потоки перемещения информации,
- процессы обработки информации,
- структуру исследуемой системы.

47). Имя деструктор любого класса, созданного в программе на языке программирования С++, будет начинаться с:

- имени класса, для которого он создан,
- символа «~»,
- служебного слова «destr»,
- служебного слова «Destructor»,
- любого разрешённого символа для имён идентификаторов, заданного программистом.

48). Концептуальной основой объектно-ориентированного подхода является (указать ответ в виде словосочетания в именительном падеже)

49). Унифицированный язык моделирования, применяемый для моделирования и проектирования сложных систем в рамках объектно-ориентированного подхода, обозначается оригинальной аббревиатурой

50). Механизм объектно-ориентированной технологии программирования, который связывает код вместе с обрабатываемыми им данными и сохраняет их в безопасности как от внешнего влияния, так от ошибочного использования, являющийся защитной оболочкой, предохраняющей код и данные от произвольного доступа из других кодов, определенных вне этой оболочки, называется (слово указать в именительном падеже)

51). Механизм, реализованный в объектно-ориентированных средах программирования, который позволяет использовать один интерфейс для общего класса действий, причём специфическое действие определяется точной природой ситуации, этот механизм также характеризуют как «один интерфейс, много методов», называется (указать ответ в именительном падеже)

52). Конструкторы в C++ не наследуются, они вызываются, когда

- дочерний класс создаёт свой объект,
- базовый класс создаёт свой объект,
- базовый класс инициализирует свой объект,
- упоминается в программе имя базового класса,
- упоминается в программе имя дочернего класса.

53). Дан фрагмент программы на языке C++:

```
class Complex {  
public:  
    double re, im;  
    double abs() { return sqrt(re*re+im*im); } };
```

В классе Second создали объект A класса Complex: Complex A;

Затем решили вызвать метод abs() объекта A. Укажите верный синтаксис вызова метода.

- double pp=A.abs();
- A.abs();
- double pp = A::abs();
- double pp = A..abs();
- int ff = A.abs();

54). Установите соответствие между шаблонами параллельных алгоритмов и реализуемыми ими функциями:

- Алгоритм «parallel_for» – Многократное выполнение одной и той же параллельной задачи,
- Алгоритм «parallel_for» - Поддержка вложенного параллелизма,

- Алгоритм параллелизма «parallel_for_each» - Одновременное выполнение задач в итеративном контейнере,
 - Алгоритм параллелизма «parallel_invoke» - Выполнение набора задач параллельно,
 - Алгоритм «parallel_buffered_sort» - Выполнение параллельной сортировки.
- 55). Установите соответствие между шаблонами параллельных алгоритмов и реализуемыми ими функциями:
- Алгоритм «parallel_transform» - Выполнение множества операций параллелизации данных,
 - Алгоритм «parallel_reduce» - Умножение последовательности матриц, умножение вектора на последовательность матриц,
 - Алгоритм «parallel_sort» - Выполнение параллельной сортировки,
 - Алгоритм «parallel_radixsort» - Выполнение параллельной сортировки.

6.2. Описание показателей и критериев контроля успеваемости, описание шкал оценивания

Для оценки знаний, умений, навыков и формирования компетенций по дисциплине применяется традиционная система контроля и оценки успеваемости студентов.

В основу традиционной системы положены принципы, в соответствии с которыми происходит формирование оценки за ответ (решение теста), осуществляется в ходе текущего и промежуточного контроля знаний обучающихся.

Таблица 7 - Критерии оценки успеваемости студентов при приёме дифференцированного зачёта

Критерии оценки	Оценка
Отлично	Студент в учебном семестре выполнил все практические работы, уверенно, правильно, полно, без неточностей и ошибок ответил подавляющее большинство теоретических вопросов, предложенных преподавателем при их проверке, сдал зачётный тест на оценку «отлично» (правильно выполнил более 90 % тестовых заданий).
Хорошо	Студент в учебном семестре выполнил все практические работы, правильно, полно и с небольшими неточностями ответил подавляющее большинство теоретических вопросов, предложенных преподавателем при их проверке, сдал зачётный тест на оценку «хорошо» (правильно выполнил более 75 % и менее 90 % тестовых заданий).
Удовлетворительно	Студент в учебном семестре выполнил все практические работы, правильно ответил большинство теоретических

	вопросов, предложенных преподавателем при их проверке, сдал зачётный тест на оценку «удовлетворительно» (правильно выполнил более 60 % и менее 75 % тестовых заданий).
Неудовлетворительно	Студент в учебном семестре не выполнил все практические работы, неправильно, неполно и/или с ошибками ответил большинство теоретических вопросов, предложенных преподавателем при их проверке, либо вообще не мог сформулировать ответы на вопросы преподавателя, сдал зачётный тест на оценку «неудовлетворительно» (правильно выполнил менее 60 % тестовых заданий).

7. Учебно-методическое и информационное обеспечение дисциплины

7.1 Основная литература

- 1 Антонов, А. С. Параллельное программирование с использованием OpenMP. — М.: Изд-во МГУ, 2009. — 128 с.
- 2 Объектно-ориентированное программирование: учебное пособие / В. В. Лозовский, Е. Н. Штрекер, Е. С. Данилович [и др.]. — Москва: РТУ МИРЭА, 2025. — 484 с. — ISBN 978-5-7339-2498-4. — Текст: электронный // Лань : электронно-библиотечная система. — URL: <https://e.lanbook.com/book/493547> — Режим доступа: для авториз. пользователей.
- 2 Восс, М. Параллельное программирование на C++ с помощью библиотеки TBB: руководство / М. Восс, Р. Асенхо, Д. Рейндерс ; перевод с английского А. А. Слинкина. — Москва: ДМК Пресс, 2020. — 674 с. — ISBN 978-5-97060-864-7. — Текст : электронный // Лань: электронно-библиотечная система. — URL: <https://e.lanbook.com/book/179500> — Режим доступа: для авториз. пользователей.
- 3 Юрина, Т. А. Объектно-ориентированное программирование: учебно-методическое пособие / Т. А. Юрина. — Омск: СибАДИ, 2023. — 72 с. — Текст : электронный // Лань: электронно-библиотечная система. — URL: <https://e.lanbook.com/book/338576> — Режим доступа: для авториз. пользователей.
- 4 Пай, П. Реактивное программирование на C++ / П. Пай, П. Абрахам; перевод с английского В. Ю. Винника. — Москва : ДМК Пресс, 2019. — 324 с. — ISBN 978-5-97060-778-7. — Текст: электронный // Лань: электронно-библиотечная система. — URL: <https://e.lanbook.com/book/131698> — Режим доступа: для авториз. пользователей.
- 5 Уильямс, Э. Параллельное программирование на C++ в действии. Практика разработки многопоточных программ. — М.: ДМК-Пресс, 2016. — 672 с. — ISBN 978-5-94074-537-2, 978-5-97060-194-5
- 6 Пош, М. Программирование встроенных систем на C++17. — М.: ДМК

Пресс, 2020. — 394 с. — ISBN 978-5-97060-785-5

7 Куссурум Д. Modern Parallel Programming with C++ and Assembly Language.
— Apress, 2022. — 642 с. — ISBN 978-1-4842-7917-5

7.2. Дополнительная литература

- 1 Унгер, А. Ю. Объектно-ориентированное программирование: учебник / А. Ю. Унгер. — Москва: РТУ МИРЭА, 2023. — 102 с. — ISBN 978-5-7339-2051-1. — Текст: электронный// Лань: электронно-библиотечная система. — URL: <https://e.lanbook.com/book/398276> — Режим доступа: для авториз. пользователей.
- 2 Мурлин, А. Г. Объектно-ориентированное программирование: учебное пособие/ А. Г. Мурлин, В. А. Мурлина, М. В. Янаева. — Краснодар: КубГТУ, 2021. — 151 с. — ISBN 978-5-8333-1059-5. — Текст: электронный // Лань: электронно-библиотечная система. — URL: <https://e.lanbook.com/book/231569> — Режим доступа: для авториз. пользователей.
- 3 Лозовский, В. В. Объектно-ориентированное программирование : учебно-методическое пособие / В. В. Лозовский, В. А. Морозов, А. А. Гололобов. — Москва: РТУ МИРЭА, 2024. — 107 с. — ISBN 978-5-7339-2120-4. — Текст: электронный // Лань: электронно-библиотечная система. — URL: <https://e.lanbook.com/book/405227>—Режим доступа: для авториз. пользователей.
- 4 Гримм, Р. Параллельное программирование на современном C++. Что каждый профессионал должен знать о параллельном программировании / Р. Гримм; перевод с английского В. Ю. Винника. — Москва: ДМК Пресс, 2022. — 616 с. — ISBN 978-5-97060-957-6. — Текст: электронный // Лань: электронно-библиотечная система. — URL: <https://e.lanbook.com/book/314870> — Режим доступа: для авториз. пользователей.
- 5 Просайз Д. Прикладное машинное обучение и искусственный интеллект для инженеров. — М.: ДМК, 2024. — 480 с. — ISBN 978-5-93700-198-6
- 6 Карап Б. TensorFlow.NET в действии. — СПб.: Питер, 2023. — 352 с. — ISBN 978-5-4461-1926-4
- 7 Lo K. PyTorch.NET: глубокое обучение без Python. — Apress, 2024. — 290 с. — ISBN 978-1-4842-9275-0
- 8 Джанджуа М. TinyML Cookbook. — Packt, 2023. — 280 с. — ISBN 978-1-80461-512-0
- 9 Карпов А. Гетерогенные вычисления в embedded AI. — СПб.: БХВ, 2024. — 352 с. — ISBN 978-5-9775-2031-7
- 10 Itsubo T. et al. An FPGA-based optimizer design for distributed deep learning with multiple GPUs // IEICE Transactions on Information and Systems. — 2021. — Vol. E104D, № 12. — P. 2057–2067. — DOI: 10.1587/transinf.2021EDL8032
- 11 Optimizing Neural Networks for Efficient FPGA Implementation: A Survey. — 2021. — Vol. 28. — P. 4537–4547

7.3 Методические указания, рекомендации и другие материалы к

занятиям

- 1 Программирование. Сборник задач: учебное пособие для вузов / О. Г. Архипов, В. С. Батасова, П. В. Гречкина [и др.]; под редакцией М. М. Маран. — 3-е изд., стер. — Санкт-Петербург: Лань, 2025. — 140 с. — ISBN 978-5-507-50516-6. — Текст : электронный // Лань: электронно-библиотечная система. — URL: <https://e.lanbook.com/book/443291> — Режим доступа: для авториз. пользователей.
- 2 Энтони, У. Параллельное программирование на C++ в действии. Практика разработки многопоточных программ: учебное пособие / У. Энтони; перевод с английского А. А. Слинкин. — Москва : ДМК Пресс, 2012. — 672 с. — ISBN 978-5-94074-448-1. — Текст: электронный // Лань: электронно-библиотечная система. — URL: <https://e.lanbook.com/book/4813> — Режим доступа: для авториз. пользователей.

7.4 Перечень статей А* и статей журналов белого списка для изучения

- 1 Никифоров, А. И., Садовников, Р. В. Применение методов параллельного программирования для моделирования потокоотклоняющих технологий на компьютерах с гибридной архитектурой// Программирование. – 2019. – № 1. – С. 43-51. URL: https://www.elibrary.ru/download/elibrary_36849065_27491280.pdf (свободный доступ)
- 2 Кий, К. И., Анохин, Д. А., Подопросветов, А. В. Программная система обработки изображений с параллельными вычислениями// Программирование. – 2020. – № 6. – С. 41-54. URL: https://www.elibrary.ru/download/elibrary_44007975_85964639.pdf (свободный доступ)
- 3 Выокова, Н. И., Галатенко, В. А., Павлов, А. Н., Самборский, С. В. Отображение параллельных вычислений на распределенные системы, использующие технологию RAPIDIO// Программирование. – 2020. – № 6. – С. 55-66. URL: https://www.elibrary.ru/download/elibrary_44007976_67048928.pdf (свободный доступ)
- 4 Катаев, Н. А., Колганов, А. С. Автоматизированное распараллеливание программ для гетерогенных кластеров с помощью системы SAPFOR// Вычисильные методы и программирование. – 2022. – Т. 23. – № 4. – С. 379-394. URL: https://www.elibrary.ru/download/elibrary_50104881_51324746.pdf (свободный доступ)
- 5 Материалы конференции International Conference on Machine Learning (ICML). – URL <https://dblp.uni-trier.de/db/conf/icml/index.html>
- 6 Материалы конференции ACM SIGKDD Conference on Knowledge Discovery and Data Mining (KDD). – URL: <https://dblp.uni-trier.de/db/conf/kdd/index.html>
- 7 Материалы конференции Conference on Neural Information Processing Systems (NeurIPS). – URL: <https://dblp.uni-trier.de/db/conf/nips/index.html>

- 8 Материалы конференции Conference on Empirical Methods in Natural Language Processing (EMNLP). – URL: <https://dblp.uni-trier.de/db/conf/emnlp/index.html>
- 9 Материалы конференции European Conference on Computer Vision (ECCV). – URL: <https://dblp.uni-trier.de/db/conf/emnlp/index.html>
- 10 Материалы конференции IEEE International Conference on Data Mining (ICDM). – URL: <https://dblp.uni-trier.de/db/conf/icdm/index.html>
- 11 Материалы симпозиума ACM-SIGPLAN по проектированию и реализации языков программирования (PLDI). – URL: <https://dblp.uni-trier.de/db/conf/pldi>
- 12 Международная конференция по программной инженерии (ICSE).
- <https://dblp.uni-trier.de/db/conf/icse>.

8. Перечень ресурсов информационно-телекоммуникационной сети «Интернет», необходимых для освоения дисциплины

- 1 Многопоточность и Thread Pool в C++ URL: <https://habr.com/ru/articles/738250/> (открытый доступ)
- 2 Параллельное программирование в Visual C++ URL: <https://learn.microsoft.com/ru-ru/cpp/parallel/concrt/parallel-algorithms?view=msvc-170> (открытый доступ)
- 3 Параллельное программирование в C++ URL: <https://assets.ctfassets.net/9pcn2syx7zns/l41QeznoklOLamcZY6ZoA/dac493bca993c510fe1dd1aebd62e302/parallel-programming.pdf> (открытый доступ)
- 4 Программирование на C, C# и Java URL: <https://vscode.ru/prog-lessons/potoki-v-visual-cpp-primer-realizatsii-i-ispolzovaniya.html> (открытый доступ)
- 5 Itsubo T. et al. An FPGA-based optimizer design for distributed deep learning with multiple GPUs // IEICE Transactions on Information and Systems. — 2021. — Vol. E104D, № 12. — P. 2057–2067. — DOI: 10.1587/transinf.2021EDL8032
- 15 Optimizing Neural Networks for Efficient FPGA Implementation: A Survey. — 2021. — Vol. 28. — P. 4537–4547

9. Перечень программного обеспечения

Таблица 8 - Перечень программного обеспечения

№ п/ п	Наименование раздела учебной дисциплины	Наименова- ние про- граммы	Тип программы	Автор	Год разра- ботки
1	Тема 1.1. Методология объектно-ориентированного подхода разработки программного обеспечения	MS Windows MS Office (MS Power- Point) Yandex Chrome	Операционная си- стема Редактор слайдов Веб-браузер Веб-браузер	Microsoft Yandex Google	2010 и позже 2025 2025

№ п/ п	Наименование раздела учебной дисциплины	Наименова- ние про- граммы	Тип программы	Автор	Год разра- ботки
2	Тема 1.2. Базовые по- ложения объектно- ориентированной тех- нологии программи- рования примени- тельно к языку про- граммирования C++	MS Windows MS Office (MS Power- Point) Yandex Chrome	Операционная си- стема Редактор слайдов Веб-браузер Веб-браузер	Microsoft Yandex Google	2010 и позже 2025 2025
3	Тема 1.3. Язык унифи- цированного модели- рования UML, его применение при про- ектировании автома- тизированных инфор- мационных систем	MS Windows MS Office (MS Power- Point) Yandex Chrome Visual Para- digm Онлайн (VP Онлайн) Free Edition Diagrams.net	Операционная си- стема Редактор слайдов Веб-браузер Веб-браузер Case-средство Case-средство	Microsoft Yandex Google Visual Par- adigm Он- лайн JGraph	2010 и позже 2025 2025 2025 2025
4	Тема 2.1. Стандарты, регламентирующие жизненный цикл про- граммного обеспече- ния и работы, выпол- няемые на отдельных его этапах	MS Windows MS Office (MS Power- Point) Yandex Chrome	Операционная си- стема Редактор слайдов Веб-браузер Веб-браузер	Microsoft Yandex Google	2010 и позже 2025 2025
5	Тема 2.2. Содержание стандартов ГОСТ Р 59793-2021 «Автома- тизированные си- стемы. Стадии созда- ния», ГОСТ Р ИСО/МЭК 12207- 2010 «Информацион- ная технология. Си- стемная и програм- мная инженерия. Про- цессы жизненного цикла программных средств»	MS Windows MS Office (MS Power- Point) Yandex Chrome	Операционная си- стема Редактор слайдов Веб-браузер Веб-браузер	Microsoft Yandex Google	2010 и позже 2025 2025

№ п/ п	Наименование раздела учебной дисциплины	Наименова- ние про- граммы	Тип программы	Автор	Год разра- ботки
6	Тема 2.3. Содержание стандарта ГОСТ Р ИСО/ МЭК 25010-2015 «Информационные технологии. Системная и программная инженерия. Требования и оценка качества систем и программного обеспечения (SQuaRE). Модели качества систем и программных продуктов», его использование для оценки качества программного обеспечения	MS Windows MS Office (MS Word, MS Excel, MS PowerPoint, MS Access) Yandex Chrome	Операционная система Текстовый редактор Табличный процессор Редактор слайдов СУБД Веб-браузер Веб-браузер	Microsoft Yandex Google	2010 и позже 2025 2025
7	Тема 2.4. Определение порядка верификации, валидации и испытаний автоматизированных информационных систем в стандарте ГОСТ Р 59792-2021 «Информационная технология. Виды испытаний автоматизированных систем»	MS Windows MS Office (MS Word, MS Excel, MS PowerPoint, MS Access) Yandex Chrome	Операционная система Текстовый редактор Табличный процессор Редактор слайдов СУБД Веб-браузер Веб-браузер	Microsoft Yandex Google	2010 и позже 2025 2025
8	Тема 2.5. Виды и методики проведения тестирования программного обеспечения	MS Windows MS Office (MS Word, MS Excel, MS PowerPoint, MS Access) Yandex Chrome TestComplete	Операционная система Текстовый редактор Табличный процессор Редактор слайдов СУБД Веб-браузер Веб-браузер Автоматизированная тестовая система	Microsoft Yandex Google SmartBear	2010 и позже 2025 2025 2025
9	Тема 3.1. Параллелизм задач в среде MS Visual Studio 2019	MS Windows MS Office (MS Word, MS Excel, MS PowerPoint, MS Access) Yandex Chrome MS Visual Studio 2019	Операционная система Текстовый редактор Табличный процессор Редактор слайдов СУБД Веб-браузер Веб-браузер	Microsoft Yandex Google Microsoft	2010 и позже 2025 2025 2019

№ п/ п	Наименование раздела учебной дисциплины	Наименова- ние про- граммы	Тип программы	Автор	Год разра- ботки
			Инструментальная среда разработки про- грамм		
10	Тема 3.2. Лямбда-вы- ражения и их приме- нение в параллельных вычислительных про- цессах	MS Windows MS Office (MS Word, MS Excel, MS PowerPoint, MS Access) Yandex Chrome MS Visual Stu- dio 2019	Операционная си- стема Текстовый редактор Табличный процессор Редактор слайдов СУБД Веб-браузер Веб-браузер Инструментальная среда разработки про- грамм	Microsoft Yandex Google Microsoft	2010 и позже 2025 2025 2019
11	Тема 3.3. Библиотека параллельных шабло- нов (PPL) MS Visual Studio 2019	MS Windows MS Office (MS Word, MS Excel, MS PowerPoint, MS Access) Yandex Chrome MS Visual Stu- dio 2019	Операционная си- стема Текстовый редактор Табличный процессор Редактор слайдов СУБД Веб-браузер Веб-браузер Инструментальная среда разработки про- грамм	Microsoft Yandex Google Microsoft	2010 и позже 2025 2025 2019
12	Тема 3.4. Обработка исключений в среде MS Visual Studio 2019 с параллелизмом	MS Windows MS Office (MS Word, MS Excel,	Операционная си- стема Текстовый редактор Табличный процессор	Microsoft	2010 и позже
13	Тема 3.5. Рекоменда- ции по работе со сре- дой выполнения про- грамм Visual Studio 2019 с параллелизмом	MS PowerPoint, MS Access) Yandex Chrome MS Visual Stu- dio 2019	Редактор слайдов СУБД Веб-браузер Веб-браузер Инструментальная среда разработки про- грамм	Yandex Google Microsoft	2025 2025 2019
14	Тема 4.1. Создание сети обработки изоб- ражений	MS Windows MS Office (MS Word, MS Excel,	Операционная си- стема Текстовый редактор	Microsoft	2010 и позже
15	Тема 4.2. Программи- рование неиерархи- ческих и иерархиче- ских методов кластер-ана- лиза данных с приме- нением параллелиза- ции задач	MS PowerPoint, MS Access) Yandex Chrome MS Visual Stu- dio 2019	Табличный процессор Редактор слайдов СУБД Веб-браузер Веб-браузер Инструментальная среда разработки про- грамм	Yandex Google Microsoft	2025 2025 2019

№ п/п	Наименование раздела учебной дисциплины	Наименование программы	Тип программы	Автор	Год разработки
16	Тема 4.3. Применение параллелизации вычислительных процессов для генерации решений в системах поддержки принятия решений	MS Windows MS Office (MS Word, MS Excel, MS PowerPoint, MS Access) Yandex Chrome MS Visual Studio 2019	Операционная система Текстовый редактор Табличный процессор Редактор слайдов СУБД Веб-браузер Веб-браузер Инструментальная среда разработки программ	Microsoft Yandex Google Microsoft	2010 и позже 2025 2025 2019
17	Тема 4.4. Формирование базы знаний и ранжирование альтернатив с применением параллельных вычислительных процессов при разработке экспертных систем				

10. Описание материально-технической базы, необходимой для осуществления образовательного процесса по дисциплине

Лекции проводятся в специализированной аудитории, оборудованной мультимедийным проектором для демонстрации компьютерных презентаций.

Для проведения практических занятий по дисциплине «Объектно-ориентированное проектирование и программирование» необходима компьютерная аудитория.

Таблица 9 - Сведения об обеспеченности специализированными аудиториями, кабинетами, лабораториями

Наименование специальных помещений и помещений для самостоятельной работы (№ учебного корпуса, № аудитории)	Оснащенность специальных помещений и помещений для самостоятельной работы
Лаборатория «Искусственный интеллект в АПК» (№ 201, учебный корпус № 1)	17 профессиональных рабочих станций с процессорами Intel i9 и графическими ускорителями NVIDIA GeForce RTX 4090 128 ГБ оперативной памяти, 1 ТБ SSD накопителей Серверное оборудование: - 2 модуля с суммарным количеством 772 потоков; - 262 ГБ оперативной памяти, 87 ТБ SSD хранилища; - Высокопроизводительные процессоры Intel Xeon Gold и Platinum; Вычислительный кластер на базе NVIDIA H100; - 7168 ГБ оперативной памяти; - 110 производительных ядер, 220 высокоеффективных потоков; - 400 ГБ видеопамяти, 84480 ядер CUDA;

Наименование специальных помещений и помещений для самостоятельной работы (№ учебного корпуса, № аудитории)	Оснащенность специальных помещений и помещений для самостоятельной работы
	<ul style="list-style-type: none"> - 72 ТБ высокоскоростного хранилища; - 10 Гбит сеть с резервированием. <p>Программная часть лаборатории включает:</p> <ul style="list-style-type: none"> - экосистему инструментов разработки и анализа данных (Python, R, TensorFlow, PyTorch); - библиотеки и фреймворки для глубокого обучения и AI-разработки; - инструменты визуализации и мониторинга производительности моделей, - программные средства поддержки высокопроизводительных вычислений компьютерных классов и лаборатории: фреймворки TensorFlow, PyTorch, Keras, MS Visual Studio 2019 и MXNet
Компьютерный класс (корпус 1, аудитория 213)	<p>Количество рабочих мест: 24</p> <p>Встроенные сетевые адаптеры (Intel I219-V или Realtek RTL8111H), интерфейс RJ-45, скорость 10/100/1000 Мбит/с. Точки доступа: Ubiquiti UniFi AP AC Pro, стандарты IEEE 802.11a/b/g/n/ac, частоты 2.4 ГГц (450 Мбит/с) и 5 ГГц (1300 Мбит/с), поддержка MU-MIMO, питание PoE.</p>
Лекционная аудитория, аудитория Планетарий 1, корпус 12	Количество мест — 180, мультимедийное оборудование — да
Центральная научная библиотека имени Н. И. Железнова	Читальные залы библиотеки
Общежитие № 7	Комната для самоподготовки

11. Методические рекомендации студентам по освоению дисциплины

Посещение лекционных (с конспектированием рассматриваемых вопросов) и практических занятий (с выполнением практических работ), а также проработка рекомендуемой литературы являются необходимым и достаточным условием для получения необходимых знаний, практических умений и навыков по изучаемой дисциплине.

Подготовка студентов к занятиям носит индивидуальный характер, но такая подготовка должна включать изучение конспектов лекций и рекомендуемой литературы, что позволяет усвоить необходимые знания по изучаемой теме. Для получения консультаций по вопросам, ответы на которые студент не смог найти в процессе проработки материалов, предусмотрено внеаудиторное время.

Самостоятельная работа студентов организуется в соответствии с методическими указаниями и должна быть выполнена в объеме, предусмотренном

данной рабочей программой. Самостоятельная работа формирует навыки поиска необходимой информации и способствует лучшему усвоению материала.

Виды и формы отработки пропущенных занятий

Студент, пропустивший занятие лекционного типа, обязан отработать его в одной из следующих форм:

- индивидуальная консультация по инициативе студента (рекомендуемая форма);
- индивидуальная проработка студентом лекционного материала по рекомендуемой литературе, компьютерным презентациям и конспектам, выполненным другими студентами, с последующим устным опросом;
- реферат на тему, предложенную преподавателем.

Трудоемкость реферата не может превышать количества часов лекционных занятий, пропущенных студентом. Рекомендуемый объем реферата – не более 10 страниц. Оригинальность реферата проверяется. По требованию преподавателя студент должен быть готов представить доказательства оригинальности реферата (например, ксерокопии использованных источников, сайты в сети Интернет, копии библиотечных абонентских карточек и др.), а также объяснить значения терминов, встречающихся в реферате.

С разрешения преподавателя студент имеет право отработать пропущенное практическое задание самостоятельно и отчитаться по нему на ближайшем практическом занятии (если это не противоречит его плану) либо во время, назначенное преподавателем для индивидуальных консультаций.

Если самостоятельная отработка практической работы невозможна по техническим причинам либо в связи с недостаточной подготовленностью студента, то кафедра прикладной информатики организует дополнительное практическое занятие для всех студентов, не выполнивших практические работы в срок и не отработавших их самостоятельно.

Пропуск занятия по документально подтвержденной деканатомуважительной причине не является основанием для снижения оценки выполненной практической работы.

12. Методические рекомендации преподавателям по организации обучения по дисциплине

Для обеспечения большей наглядности лекционные занятия должны проводиться в аудиториях, оборудованных проекционной аппаратурой для демонстрации компьютерных презентаций. По каждой теме (вопросу) преподаватель должен сформировать список рекомендуемой литературы.

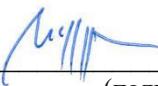
Начало практических занятий следует отводить под обсуждение вопросов студентов по содержанию и методике выполнения практических работы. Допускается при таком обсуждении использование одной из технологий интерактивного обучения. Для проведения индивидуальных консультаций должно быть предусмотрено внеаудиторное время.

При проведении практических занятий для формирования необходимых компетенций следует использовать активные и интерактивные образовательные технологии, описанные в п. 5 данной рабочей программы.

Невыполнение требований к практическим заданиям является основанием для повторного выполнения практической работы с измененным вариантом заданий и снижения оценки.

Контроль знаний студентов проводится в формах текущей аттестаций. Текущая аттестация студентов проводится постоянно на практических занятиях с помощью контроля результатов выполнения практических и тестовых заданий, устного опроса, а также на контрольной неделе. Промежуточная аттестация студентов проводится в форме экзамена (8 семестр).

Программу разработала:
Журавлев М.В., к.т.н., доцент


(подпись)

РЕЦЕНЗИЯ

на рабочую программу дисциплины Б1.В.08 Объектно-ориентированное проектирование и программирование ОПОП ВО по направлению 09.03.03 «Прикладная информатика», направленность «Системы искусственного интеллекта» (квалификация выпускника – бакалавр)

Щедриной Е. А., кандидатом педагогических наук, доцентом кафедры систем автоматизированного проектирования инженерных расчетов ФГБОУ ВО РГАУ-МСХА имени К.А. Тимирязева, (далее по тексту рецензент), проведено рецензирование рабочей программы дисциплины «Объектно-ориентированное проектирование и программирование» ОПОП ВО по направлению 09.03.03 «Прикладная информатика», направленность «Системы искусственного интеллекта» (бакалавриат) разработанной в ФГБОУ ВО «Российский государственный аграрный университет – МСХА имени К.А. Тимирязева», на кафедре прикладной информатики (разработчик – Журавлев М.В., доцент кафедры прикладной информатики, кандидат ф.-м. наук)

Рассмотрев представленные на рецензирование материалы, рецензент пришел к следующим выводам:

1. Предъявленная рабочая программа дисциплины «Объектно-ориентированное проектирование и программирование» (далее по тексту Программа) соответствует требованиям ФГОС ВО по направлению 09.03.03 «Прикладная информатика», компетентностно-ролевым моделям в сфере искусственного интеллекта. Программа содержит все основные разделы, соответствует требованиям к нормативно-методическим документам.

2. Представленная в Программе актуальность учебной дисциплины в рамках реализации ОПОП ВО не подлежит сомнению – дисциплина относится к части, формируемой участниками образовательных отношений.

3. Представленные в Программе цели дисциплины соответствуют требованиям ФГОС ВО направления 09.03.03 «Прикладная информатика» и компетентностно-ролевыми моделями в сфере искусственного интеллекта..

4. В соответствии с учебным планом и компетентностно-ролевыми моделями в сфере искусственного интеллекта, Программой за дисциплиной «Объектно-ориентированное проектирование и программирование» закреплена одна компетенция (3 индикатора). Дисциплина «Объектно-ориентированное проектирование и программирование» и представленная Программа способна реализовать их в заявленных требованиях.

5. **Результаты обучения**, представленные в Программе в категориях знать, уметь, владеть соответствуют специфике и содержанию дисциплины и демонстрируют возможность получения заявленных результатов.

6. Общая трудоёмкость дисциплины «Объектно-ориентированное проектирование и программирование» составляет 144 часа / 4 зач.ед.

7. Информация о взаимосвязи изучаемых дисциплин и вопросам исключения дублирования в содержании дисциплин соответствует действительности. Дисциплина «Объектно-ориентированное проектирование и программирование» взаимосвязана с другими дисциплинами ОПОП ВО и учебного плана по направлению 09.03.03 «Прикладная информатика» и возможность дублирования в содержании отсутствует.

8. Представленная Программа предполагает использование современных образовательных технологий, используемые при реализации различных видов учебной работы. Формы образовательных технологий соответствуют специфике дисциплины.

9. Программа дисциплины «Объектно-ориентированное проектирование и программирование» предполагает проведение занятий в интерактивной форме.

10. Виды, содержание и трудоёмкость самостоятельной работы студентов, представленные в Программе, соответствуют требованиям к подготовке выпускников, содержащимся во ФГОС ВО направления 09.03.03 «Прикладная информатика».

11. Представленные и описанные в Программе формы *текущей* оценки знаний (устный опрос, как в форме обсуждения отдельных вопросов, так и защиты практических работ), соответствуют специфике дисциплины и требованиям к выпускникам.

Форма промежуточного контроля знаний студентов, предусмотренная Программой, осуществляется в форме дифференцированного зачёта (зачёта с оценкой), что соответствует статусу дисциплины, как дисциплины части, формируемой участниками образовательных отношений ФГОС ВО направления 09.03.03 «Прикладная информатика».

12. Формы оценки знаний, представленные в Программе, соответствуют специфике дисциплины и требованиям к выпускникам.

13. Учебно-методическое обеспечение дисциплины представлено: основной литературой – 4 источника, дополнительной литературой – 4 наименования и соответствует требованиям ФГОС ВО направления 09.03.03 «Прикладная информатика» и компетентностно-ролевыми моделями в сфере искусственного интеллекта.

14. Материально-техническое обеспечение дисциплины соответствует специфике дисциплины «Объектно-ориентированное проектирование и программирование» и обеспечивает использование современных образовательных, в том числе интерактивных методов обучения.

15. Методические рекомендации студентам и методические рекомендации преподавателям по организации обучения по дисциплине дают представление о специфике обучения по дисциплине «Объектно-ориентированное проектирование и программирование».

ОБЩИЕ ВЫВОДЫ

На основании проведенного рецензирования можно сделать заключение, что характер, структура и содержание рабочей программы дисциплины «Объектно-ориентированное проектирование и программирование» ОПОП ВО по направлению 09.03.03 «Прикладная информатика», направленность «Системы искусственного интеллекта» (квалификация выпускника – бакалавр), разработанная Журавлевым М.В., к.ф.-м.н., доцентом кафедры прикладной информатики, соответствует требованиям ФГОС ВО, компетентностно-ролевых моделей в сфере искусственного интеллекта, современным требованиям экономики, рынка труда и позволит при её реализации успешно обеспечить формирование заявленных компетенций.

Рецензент: Щедрина Е. А., кандидат педагогических наук, доцент кафедры систем автоматизированного проектирования инженерных расчетов



«28» августа 2025 г.