



Автономная некоммерческая образовательная организация
высшего образования
«Воронежский экономико-правовой институт»
(АНОО ВО «ВЭПИ»)

УТВЕРЖДАЮ



Проректор
по учебно-методической работе
А.Ю. Жильников

15 октября 2025 г.

ФОНД ОЦЕНОЧНЫХ СРЕДСТВ ПО ДИСЦИПЛИНЕ (МОДУЛЮ)

Б1.О.14 Программная инженерия
(наименование дисциплины (модуля))

27.03.05 Инноватика
(код и наименование направления подготовки)

Направленность (профиль) Инновационные технологии
(наименование направленности (профиля))

Квалификация выпускника Бакалавр
(наименование квалификации)

Форма обучения Очная, заочная
(очная, заочная)

Рекомендован к использованию Филиалами АНОО ВО «ВЭПИ»

Воронеж 2025

Фонд оценочных средств по дисциплине (модулю) рассмотрен и одобрен на заседании кафедры прикладной информатики.

Протокол от «08» октября 2025 г. № 2

Фонд оценочных средств по дисциплине (модулю) согласован со следующими представителями работодателей или их объединений, направление деятельности которых соответствует области профессиональной деятельности, к которой готовятся обучающиеся:

Директор ООО ПСК «НСКОМ» Петров Р.А.

(должность, название организации, ФИО, печать)



Директор ООО «Сфера АйТи» Павлов С.В.

(должность, название организации, ФИО, печать)



Заведующий кафедрой

М.С. Агафонова

Разработчики:

Ст. преподаватель

Д.В. Байбеков

1. Перечень компетенций с указанием этапов их формирования в процессе освоения ОП ВО

Целью проведения дисциплины Б1.О.14 Программная инженерия является достижение следующих результатов обучения:

Код компетенции	Наименование компетенции
УК-1	способен осуществлять поиск, критический анализ и синтез информации, применять системный подход для решения поставленных задач
УК-2	Способен определять круг задач в рамках поставленной цели и выбирать оптимальные способы их решения, исходя из действующих правовых норм и имеющихся ресурсов и ограничений

В формировании данных компетенций также участвуют следующие дисциплины (модули), практики образовательной программы (по семестрам (курсам) их изучения):

- для очной формы обучения:

Наименование дисциплин (модулей), практик	Этапы формирования компетенций по семестрам изучения							
	1 сем.	2 сем.	3 сем.	4 сем.	5 сем.	6 сем.	7 сем.	8 сем.
Математика	УК-1	УК-1	УК-1					
Теория вероятностей и математическая статистика				УК-1				
Информатика и программирование	УК-1	УК-1						
Системный анализ и принятие решений				УК-1				
Информационные системы и технологии		УК-1	УК-1					
Компьютерная и инженерная графика	УК-1							
Стратегический менеджмент в инновационных организациях							УК-1	
Применение Excel в экономических расчетах		УК-1						
Системы искусственного интеллекта в профессиональной деятельности					УК-1			
Исследование операций и методы оптимизации					УК-1	УК-1		
Технические измерения и приборы			УК-1					
Теория алгоритмов		УК-1						
Документирование инновационной деятельности				УК-1				
Делопроизводство				УК-1				
Экономическая теория	УК-2	УК-2						
Программная инженерия						УК-2	УК-2	
Инновационный менеджмент							УК-2	
Маркетинг в инновационной сфере					УК-2	УК-2		
Управление качеством					УК-2			
Создание и реализация общественных проектов			УК-2					
Проектная деятельность					УК-2	УК-2	УК-2	
Методы принятия управленческих решений					УК-2			
Методы оптимальных решений в инновационных процессах						УК-2		
Учебная практика (ознакомительная практика)		УК-1						
Производственная практика (преддипломная практика)								УК-2

Подготовка к сдаче и сдача государственного экзамена									УК-1; УК-2
Выполнение, подготовка к процедуре защиты и защита выпускной квалификационной работы									УК-1; УК-2

- для заочной формы обучения:

Наименование дисциплин (модулей), практик	Этапы формирования компетенций по семестрам изучения				
	1 курс	2 курс	3 курс	4 курс	5 курс
Математика	УК-1	УК-1			
Теория вероятностей и математическая статистика		УК-1			
Информатика и программирование	УК-1	УК-1			
Системный анализ и принятие решений			УК-1		
Информационные системы и технологии		УК-1			
Компьютерная и инженерная графика		УК-1			
Стратегический менеджмент в инновационных организациях				УК-1	
Применение Excel в экономических расчетах		УК-1			
Системы искусственного интеллекта в профессиональной деятельности			УК-1		
Исследование операций и методы оптимизации			УК-1		
Технические измерения и приборы		УК-1			
Теория алгоритмов		УК-1			
Документирование инновационной деятельности		УК-1			
Делопроизводство		УК-1			
Экономическая теория	УК-2				
Программная инженерия				УК-2	
Инновационный менеджмент				УК-2	
Маркетинг в инновационной сфере			УК-2		
Управление качеством			УК-2		
Создание и реализация общественных проектов		УК-2			
Проектная деятельность			УК-2		
Методы принятия управленческих решений				УК-2	
Методы оптимальных решений в инновационных процессах					УК-2
Учебная практика (ознакомительная практика)		УК-1			
Производственная практика (преддипломная практика)					УК-1; УК-2
Подготовка к сдаче и сдача государственного экзамена					УК-1; УК-2
Выполнение, подготовка к процедуре защиты и защита выпускной квалификационной работы					УК-1; УК-2

Этап дисциплины (модуля) Б1.О.14 Программная инженерия в формировании компетенций соответствует:

- для очной формы обучения – 6 и 7 семестру;
- для заочной формы обучения – 3 и 4 курсу.

2. Показатели и критерии оценивания компетенций на различных этапах их формирования, шкалы оценивания

Показателями оценивания компетенций являются следующие результаты обучения:

Код компетенции	Код и наименование индикатора достижения	Планируемые результаты обучения по
-----------------	--	------------------------------------

	компетенции	дисциплине (модулю)
УК-1. Способен осуществлять поиск, критический анализ и синтез информации, применять системный подход для решения поставленных задач	ИУК-1.1. Выполняет поиск, критический анализ и синтез информации для решения поставленных задач.	<p>знать:</p> <ul style="list-style-type: none"> - стадии жизненного цикла создания программных средств; <p>уметь:</p> <ul style="list-style-type: none"> -осуществлять анализ архитектуры программных продуктов; <p>владеть:</p> <ul style="list-style-type: none"> - навыками проектирования и синтеза программных продуктов.
	ИУК-1.2. Использует системный подход для решения поставленных задач.	<p>знать:</p> <ul style="list-style-type: none"> - основные этапы разработки программных продуктов; <p>уметь:</p> <ul style="list-style-type: none"> - создавать концепцию проекта с учетом анализа его жизненного цикла; <p>владеть:</p> <ul style="list-style-type: none"> - системным подходом к анализу проектного продукта.
УК-2. Способен определять круг задач в рамках поставленной цели и выбирать оптимальные способы их решения, исходя из действующих правовых норм и имеющихся ресурсов и ограничений	ИУК 2.1 Формулирует в рамках поставленной цели проекта совокупность взаимосвязанных задач, обеспечивающих ее достижение. Определяет ожидаемые результаты решения выделенных задач	<p>Знать: основные понятия и категории экономики, экономические законы и закономерности; теорию рыночного спроса и рыночного предложения, рыночные структуры, теорию издержек производства, макроэкономические показатели, основные виды налогов, виды ценных бумаг, факторы экономического роста</p> <p>Уметь: приводить примеры: факторов производства и факторных доходов, российских предприятий разных организационных форм, глобальных экономических проблем</p> <p>Владеть: навыками выбора из предлагаемого перечня оптимальных источников экономической информации; навыками оценки экономических действий субъекта в качестве потребителя, члена семьи и гражданина</p>
	ИУК 2.2 Проектирует решение конкретной задачи проекта, выбирая оптимальный способ ее решения, исходя из действующих правовых норм и имеющихся ресурсов и ограничений	<p>Знать: способы определения задач в рамках поставленной цели, выбирать оптимальные способы их решения, исходя из действующих правовых норм и имеющихся ресурсов и ограничений. Формулировать в рамках поставленной цели проекта совокупность взаимосвязанных задач, обеспечивающих ее достижение.</p> <p>Определять ожидаемые результаты решения выделенных задач</p> <p>Уметь: проектировать решение конкретной задачи проекта, выбирая оптимальный способ ее решения, исходя из действующих правовых норм и имеющихся ресурсов и ограничений</p> <p>Владеть: методами решения конкретных задач проекта заявленного качества и за установленное время. Публично представлять результаты решения конкретной задачи проекта</p>
	ИУК 2.3 Решает конкретные задачи проекта заявленного качества и за установленное время	<p>Знать: как публично представлять результаты решения конкретной задачи проекта</p> <p>Уметь: решать конкретные задачи проекта заявленного качества и за установленное время</p> <p>Владеть: навыками формулирования в рамках</p>

		поставленной цели проекта совокупность взаимосвязанных задач, обеспечивающих ее достижение
ОПК-3.Способен использовать фундаментальные знания для решения базовых задач управления в технических системах с целью совершенствования в профессиональной деятельности	ИОПК 3.1 Применяет фундаментальные знания в области управления технических систем с целью их совершенствования	Знать: программное управление, управление с обратной связью Уметь: использовать методы управления в своей профессиональной деятельности Владеть: навыками моделирования процессов управления
	ИОПК 3.2 Решает базовые задачи управления в технических системах	Знать: способы и методы решения базовых задач управления инновационными проектами Уметь: решать базовые задачи при управлении инновационными проектами Владеть: навыками решения базовых задач при управлении инновационными проектами

Порядок оценки освоения обучающимися учебного материала определяется содержанием следующих разделов дисциплины (модуля):

№ п/п	Наименование раздела дисциплины (модуля)	Код компетенции, код индикатора достижения компетенции	Критерии оценивания	Оценочные средства текущего контроля успеваемости	Шкала оценивания
1	Тема 1. Введение	УК-1 (ИУК-1.1, ИУК-1.2) УК-2 (ИУК-2.1, ИУК-2.2, ИУК-2.3) ОПК-3 (ИОПК-3.1, ИОПК-3.2.)	Знать: - основные понятия программной инженерии Уметь: - решать задачи Владеть: - характеристикой программного продукта и его спецификой	Опрос	«Зачтено» «Не зачтено»
2	Тема 2. Модели и профили жизненного цикла программных средств	УК-1 (ИУК-1.1, ИУК-1.2) УК-2 (ИУК-2.1, ИУК-2.2, ИУК-2.3) ОПК-3 (ИОПК-3.1, ИОПК-3.2.)	Знать: - понятие жизненного цикла программы и его этапы Уметь: - эксплуатировать Владеть: - этапами жизненного цикла программного продукта и его спецификой	Доклад	«Зачтено» «Не зачтено»
3	Тема 3. Модели и процессы управления проектами программных средств	УК-1 (ИУК-1.1, ИУК-1.2) УК-2 (ИУК-2.1, ИУК-2.2,	Знать: - зачем нужны проверки: пассивные и активные данные Уметь: - поэтапно вести учет результатов	Сообщение	«Зачтено» «Не зачтено»

		ИУК-2.3) ОПК-3 (ИОПК-3.1, ИОПК-3.2.)	Владеть: - методом прибавочной стоимости		
4	Тема 4. Управление требованиями к программному обеспечению	УК-1 (ИУК-1.1, ИУК-1.2) УК-2 (ИУК-2.1, ИУК-2.2, ИУК-2.3) ОПК-3 (ИОПК-3.1, ИОПК-3.2.)	Знать: - место дисциплины в разработке программного обеспечения Уметь: - планировать процесс управления согласно требованиям. Владеть: - корректировкой требований и управлением	Доклад	«Зачтено» «Не зачтено»
5	Тема 5. Проектирование программного обеспечения	УК-1 (ИУК-1.1, ИУК-1.2) УК-2 (ИУК-2.1, ИУК-2.2, ИУК-2.3) ОПК-3 (ИОПК-3.1, ИОПК-3.2.)	Знать: - методы проектирования программных продуктов и признаки их классификации Уметь: - модульное проектировании Владеть: - объектно-ориентированной технологией и ее преимуществами	Опрос	«Зачтено» «Не зачтено»
6	Тема 6. Конструирование (детальное проектирование) программного обеспечения	УК-1 (ИУК-1.1, ИУК-1.2) УК-2 (ИУК-2.1, ИУК-2.2, ИУК-2.3) ОПК-3 (ИОПК-3.1, ИОПК-3.2.)	Знать: - языки программирования и их классификация Уметь: - выбрать и обосновать язык программирования Владеть: - языками программирования для решения экономических, инженерных, научных задач	Опрос	«Зачтено» «Не зачтено»
7	Тема 7. Тестирование программного обеспечения	УК-1 (ИУК-1.1, ИУК-1.2) УК-2 (ИУК-2.1, ИУК-2.2, ИУК-2.3) ОПК-3 (ИОПК-3.1, ИОПК-3.2.)	Знать: - сущность и необходимость тестирования программного обеспечения Уметь: - решать задачи Владеть: - комбинированным методом тестирования	Доклад	«Зачтено» «Не зачтено»
8	Тема 8. Сопровождение программного обеспечения	УК-1 (ИУК-1.1, ИУК-1.2) УК-2 (ИУК-2.1,	Знать: - сопровождение программных продуктов Уметь: - вносить изменения Владеть:	Доклад	«Зачтено» «Не зачтено»

		ИУК-2.2, ИУК-2.3) ОПК-3 (ИОПК-3.1, ИОПК-3.2.)	- необходимая документация и предпродажная подготовка программных средств		
9	Тема 9. Конфигурационное управление	УК-1 (ИУК-1.1, ИУК-1.2) УК-2 (ИУК-2.1, ИУК-2.2, ИУК-2.3) ОПК-3 (ИОПК-3.1, ИОПК-3.2.)	Знать: - внутреннюю организацию программных продуктов Уметь: - использовать встроенные функции Владеть: - структурой пакета прикладных программ	Сообщение	«Зачтено» «Не зачтено»
10	Тема 10. Управление программной инженерией	УК-1 (ИУК-1.1, ИУК-1.2) УК-2 (ИУК-2.1, ИУК-2.2, ИУК-2.3) ОПК-3 (ИОПК-3.1, ИОПК-3.2.)	Знать: - основные сведения о языке UML Уметь: - построить концептуальную модель предметной области Владеть: - работой в среде CASE- средств	Доклад	«Зачтено» «Не зачтено»
11	Тема 11. Процесс программной инженерии	УК-1 (ИУК-1.1, ИУК-1.2) УК-2 (ИУК-2.1, ИУК-2.2, ИУК-2.3) ОПК-3 (ИОПК-3.1, ИОПК-3.2.)	Знать: - категории специалистов, занятых разработкой и эксплуатацией программ; Уметь: - применять принципы и методы коллективной разработки программных продуктов; Владеть: - обязанностями членов бригады;	Опрос	«Зачтено» «Не зачтено»
12	Тема 12. Инструменты и методы программной инженерии	УК-1 (ИУК-1.1, ИУК-1.2) УК-2 (ИУК-2.1, ИУК-2.2, ИУК-2.3) ОПК-3 (ИОПК-3.1, ИОПК-3.2.)	Знать: - инструментальные системы технологии программирования и их основные черты Уметь: - выделять основные компоненты инструментальных систем технологии программирования Владеть: - CASE-средствами, их назначением и применением	Опрос	«Зачтено» «Не зачтено»
13	Тема 13. Качество	УК-1 (ИУК-1.1,	Знать:	Доклад	«Зачтено» «Не зачтено»

	программного обеспечения	ИУК-1.2) УК-2 (ИУК-2.1, ИУК-2.2, ИУК-2.3) ОПК-3 (ИОПК-3.1, ИОПК-3.2.)	- принципы обеспечения показателей качества программного продукта; Уметь: - обеспечить мобильность Владеть: - методами проверки корректности		
14	Тема 14. Документирование программного обеспечения	УК-1 (ИУК-1.1, ИУК-1.2) УК-2 (ИУК-2.1, ИУК-2.2, ИУК-2.3) ОПК-3 (ИОПК-3.1, ИОПК-3.2.)	Знать: - стадии разработки программ и программной документации Уметь: - соответствовать основным требованиям к содержанию документации Владеть: - понятием о ЕСПД	Сообщение, тест	«Зачтено» «Не зачтено»
15	Тема 15. Технико-экономическое обоснование проектов программных средств	УК-1 (ИУК-1.1, ИУК-1.2) УК-2 (ИУК-2.1, ИУК-2.2, ИУК-2.3) ОПК-3 (ИОПК-3.1, ИОПК-3.2.)	Знать: - особенности продаж программных продуктов Уметь: - применять способы прогнозирования рынка программного обеспечения. Владеть: - методиками оценки трудоемкости разработки программного продукта	Доклад	«Зачтено» «Не зачтено»
ИТОГО			Форма контроля	Оценочные средства промежуточной аттестации	Шкала оценивания
			Зачет	Письменный ответ на билет	«Зачтено» «Не зачтено»
			Экзамен	Письменный ответ на билет	«Отлично», «Хорошо», «Удовлетворительно», «Неудовлетворительно»

Критерии оценивания результатов обучения для текущего контроля успеваемости и промежуточной аттестации по дисциплине (модулю)

1. Критерий оценивания опроса:

- зачтено – выставляется обучающемуся, если демонстрирует знание основных разделов программы изучаемого курса: его базовых понятий и фундаментальных проблем; приобрел необходимые умения и навыки; освоил вопросы практического применения полученных знаний, не допустил фактических ошибок при ответе; достаточно последовательно и логично излагает теоретический материал, допуская лишь незначительные нарушения последовательности изложения и некоторые неточности; показывает

всесторонние, систематизированные, глубокие знания учебной программы дисциплины и умение уверенно применять их;

- не зачтено – выставляется обучающемуся, если демонстрирует фрагментарные знания основных разделов программы изучаемого курса: его базовых понятий и фундаментальных проблем; слабо выражена способность к самостоятельному аналитическому мышлению, имеются затруднения в изложении материала, отсутствуют необходимые умения и навыки; допущены грубые ошибки и незнание терминологии, отказ отвечать на дополнительные вопросы, знание которых необходимо для получения положительной оценки; выставляется обучающемуся, ответ которого содержит существенные пробелы в знаниях основного содержания рабочей программы дисциплины.

2. Критерий доклада:

- зачтено – представленный доклад соответствует тематике, экономически обоснован, выводы по изученной проблеме изложены логически, соблюдены требования, при разработке доклада были использованы современные информационные технологии;

- не зачтено – доклад обучающимся не представлена; материалы доклад не обоснованы или логически не связаны, использованы устаревшие источники информации.

3. Критерий сообщения:

- зачтено – представленный сообщение актуально, экономически обоснован, выводы по изученной представленная информация изложена логически, соблюдены требования, при разработке сообщения были использованы современные информационные технологии;

- не зачтено – сообщение обучающимся не представлена; представленная информация не обоснованы или логически не связана, использованы устаревшая информация.

4. Критерий оценивания тестов:

- зачтено – выставляется обучающемуся, если:

знает методы математического анализа объектов, явлений и процессов; использует приемы оценки математических взаимосвязей экономические явления, процессы и институты на микро- и макроуровне; 50-100% правильных ответов;

- не зачтено – выставляется обучающемуся, если: демонстрирует фрагментарные знания основных разделов программы изучаемого курса: его базовых понятий и фундаментальных проблем; слабо выражена способность к самостоятельному аналитическому мышлению, имеются затруднения в изложении материала, отсутствуют необходимые умения и навыки; допущены грубые ошибки и незнание терминологии, отказ отвечать на дополнительные вопросы, знание которых необходимо для получения положительной оценки; до 50% правильных ответов.

5. Критерии оценивания письменного ответа на билет на зачете:

- зачтено – выставляется обучающемуся, если: использует приемы анализа для обработки экономических данных в соответствии с поставленной задачей; знает особенности математического инструментария для решения экономических задач;

- не зачтено – выставляется обучающемуся, если: демонстрирует фрагментарные знания основных разделов программы изучаемого курса: его базовых понятий и фундаментальных проблем; слабо выражена способность к самостоятельному аналитическому мышлению, имеются затруднения в изложении материала, отсутствуют необходимые умения и навыки; допущены грубые ошибки и незнание терминологии, отказ отвечать на дополнительные вопросы, знание которых необходимо для получения положительной оценки.

6. Критерии оценивания письменного ответа на билет на экзамене:

- отлично – выставляется обучающемуся, если: Даны исчерпывающие и обоснованные ответы на все поставленные вопросы, правильно и рационально (с использованием рациональных методик) решены соответствующие задачи; В ответах выделялось главное, все теоретические положения умело увязывались с требованиями руководящих документов; Ответы были четкими и краткими, а мысли излагались в логической последовательности; Показано умение самостоятельно анализировать факты, события, явления, процессы в их взаимосвязи и диалектическом развитии.

- хорошо – выставляется обучающемуся, если: Даны полные, достаточно обоснованные ответы на поставленные вопросы, правильно решены практические задания; В ответах не всегда выделялось главное, отдельные положения недостаточно увязывались с требованиями руководящих документов, при решении практических задач не всегда использовались рациональные методики расчётов; Ответы в основном были краткими, но не всегда четкими; Показано слабое умение самостоятельно анализировать факты, события, явления, процессы в их взаимосвязи и диалектическом развитии.

- удовлетворительно – выставляется обучающемуся, если: Даны в основном правильные ответы на все поставленные вопросы, но без должной глубины и обоснования, при решении практических задач обучающийся использовал прежний опыт и не применял новые методики выполнения расчётов, однако на уточняющие вопросы даны в целом правильные ответы; При ответах не выделялось главное; отдельные положения недостаточно увязывались с требованиями руководящих документов, при решении практических задач не использовались рациональные методики расчётов; Ответы были многословными, нечеткими и без должной логической последовательности, на отдельные дополнительные вопросы не даны положительные ответы; Показано неумение самостоятельно анализировать

факты, события, явления, процессы в их взаимосвязи и диалектическом развитии.

- неудовлетворительно – выставляется обучающемуся, если не выполнены требования, соответствующие оценке “удовлетворительно”.

3. Типовые контрольные задания или иные материалы, необходимые для оценки знаний, умений, навыков и (или) опыта деятельности, характеризующих этапы формирования компетенций

3.1. Вопросы для проведения опроса:

1. Основные понятия программной инженерии;
2. Программа, программное обеспечение, задачи и приложения; технологические и функциональные задачи;
3. Процесс создания программ: постановка задачи, алгоритмизация, программирование;
4. Понятие программного продукта;
5. Характеристика программного продукта и его специфика;
6. Показатели качества программного продукта: мобильность, надежность, эффективность, легкость применения, модифицируемость и коммуникативность.
7. Понятие жизненного цикла программы и его этапы;
8. Важность учета и контроля проекта.
9. Зачем нужны проверки: пассивные и активные данные.
10. Планирование учета проекта.
11. Поэтапный учет результатов.
12. Метод допустимых границ.
13. Анализ товарных запасов.
14. Учет методом S-образной кривой.
15. Метод прибавочной стоимости.
16. Отчеты о результатах проверок и организация рабочих совещаний.
17. Место дисциплины в разработки программного обеспечения.
18. Планирование процесса управления требованиями.
19. Анализ потребностей заинтересованных сторон.
20. Сбор и установление требований.
21. Организация и документирование требований.
22. Корректировка требований и управление ими.
23. Управление изменениями и внесение изменений в требования.
24. Методы проектирования программных продуктов и признаки их классификации;
25. Сущность и необходимость тестирования программного обеспечения;
26. Сопровождение программных продуктов;
27. Построение моделей программных систем с использованием структурного и объектно-ориентированного подходов;

28. Категории специалистов, занятых разработкой и эксплуатацией программ;
29. Общая характеристика инструментальных средств разработки программ;
30. Технологический процесс разработки программного обеспечения;
31. Стоимость программных средств;

3.2. Примерный перечень тем докладов и сообщений:

1. Автоматизация анализа результативности и эффективности менеджмента качества
2. Разработка автоматизированной системы гибкого управления складированием продукции
3. Автоматизация складского учета на предприятии
4. Разработка системы автоматизации финансовых операций обменного пункта
5. Разработка системы автоматизации расчета амортизационных отчислений
6. Разработка автоматизированного рабочего места бухгалтера по начислению заработной платы
7. Разработка АРМ специалиста по расчету калькуляции готовой продукции
8. Разработка автоматизированной системы расчета по лизинговым платежам
9. Разработка автоматизированной системы оптимизации складского хранения
10. Разработка автоматизированной системы резервирования процентов по видам вкладов граждан в банке
11. Разработка автоматизированной системы управления поставкой сырья
12. Разработка системы автоматизации расчета параметров денежных потоков
13. Разработка системы автоматизации календарного планирования бизнес-проектов
14. Разработка сайта «Интернет-аукцион» промышленного предприятия
15. Разработка системы автоматизации учета и торговли флористической продукции
16. Разработка автоматизированной системы распределения грузов по автотранспорту
17. Разработка автоматизированного рабочего места финансового аналитика
18. Разработка автоматизированного рабочего места медицинского работника

19. Разработка автоматизированного рабочего места сотрудника отдела динамики и мультимедиа

20. Разработка автоматизированной информационной системы торговой деятельности фирмы

21. Разработка системы учета оргтехники

22. Разработка системы автоматизации деятельности библиотеки

23. Стандартизация и метрология в разработке программного обеспечения.

24. Стандартизация информационных технологий

25. Сертификация программного обеспечения.

3.3. Вопросы для проведения зачета:

1. Прикладные программы с высокой степенью автоматизации.

2. Адаптируемость пакетов программ.

3. Проектирование программ сложной структуры.

4. Типовые приемы конструирования пакетов программ сложной структуры.

5. Организация проектирования программного обеспечения (ПО)

6. Этапы процесса проектирования.

7. Основные процессы жизненного цикла ПС.

8. Вспомогательные процессы жизненного цикла ПС.

9. Организационные процессы жизненного цикла ПС.

10. Базовый стандарт ГОСТ Р ИСО/МЭК 12207-99.

11. Модели жизненного цикла ПС.

12. Способы формального представления знаний.

13. Основы устройства и использование экспертных систем в разработке адаптируемого программного обеспечения.

14. Основные направления интеллектуализации ПО.

15. Стандартизация и метрология в разработке программного обеспечения.

16. Стандартизация информационных технологий

17. Действующие стандарты и проблемы программных интерфейсов.

18. Нормативные документы по стандартизации и виды стандартов.

19. Стандарты в области программного обеспечения.

20. ИСО/МЭК.

21. Государственный комитет РФ по стандартизации.

22. Методы и средства проектирования пользовательского и программного интерфейсов.

3.5. Вопросы для проведения экзамена:

Вопросы к экзамену по дисциплине «Программная инженерия»

1. Модели и процессы жизненного цикла программного обеспечения

2. Модели оценки зрелости процессов ПО

3. Метрики процессов ПО
4. Извлечение требований
5. Методы моделирования для анализа требований
6. Функциональные и нефункциональные требования
7. Прототипирование
8. Основные понятия методов формальной спецификации.
9. Основные понятия и принципы разработки ПО
10. Архитектура ПО
11. Структурная разработка
12. Объектно-ориентированный анализ и разработка
13. Компонентно-базированная разработка
14. Разработка ПО для повторного использования
15. Планирование аттестационного тестирования
16. Основы тестирования (проектирование и генерации тестов, процесс тестирования)
17. Тестирование по методу «черного ящика» и методу «белого ящика»
18. Тестирование модулей, интеграция модулей и проверка правильности интеграции, тестирование системы
19. Объектно-ориентированное тестирование
20. Инспектирование.
21. Сопровождение ПО
22. Свойства сопровождаемого ПО
23. Реинжиниринг ПО
24. Наследуемые (legacy) системы
25. Повторное использование и переносимость ПО.
26. Управление командой проекта (процессы проекта, организация команды и принятие решений, распределение ролей и ответственности, отслеживание состояния процесса, решение проблем в команде)
27. Планирование работ
28. Методы оценки стоимости проекта и измерения характеристик качества ПО
29. Анализ рисков
30. Управление конфигурациями
31. Управление качеством
32. Средства поддержки управления проектом
33. Среда программирования
34. Средства моделирования для разработки и анализа требований ПО
35. Средства тестирования
36. Средства управления конфигурациями

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

Зачет и экзамен является заключительным этапом процесса формирования компетенций обучающегося при изучении дисциплины и имеет целью проверку и оценку знаний обучающегося по теории и применению полученных знаний, умений и навыков при решении практических задач.

Зачет и экзамен проводится по расписанию, сформированному учебно-методическим управлением, в сроки, предусмотренные календарным учебным графиком.

Зачет и экзамен принимается преподавателем, ведущим лекционные занятия.

Зачет и экзамен проводится только при предъявлении обучающимся зачетной книжки и при условии выполнения всех контрольных мероприятий, предусмотренных учебным планом и рабочей программой дисциплины.

Обучающимся на зачете и экзамене представляется право выбрать один из билетов. Время подготовки к ответу составляет 30 минут. По истечении установленного времени обучающийся должен ответить на вопросы экзаменационного билета.

Результаты зачета и экзамена заносятся в зачетно-экзаменационную ведомость и зачетную книжку. В зачетную книжку заносятся только положительные оценки. Подписанный преподавателем экземпляр ведомости сдаётся не позднее следующего дня в деканат.

В случае неявки обучающегося на экзамен в зачетно-экзаменационную ведомость делается отметка «не явка».

Обучающиеся, не прошедшие промежуточную аттестацию по дисциплине, должны ликвидировать академическую задолженность в установленном локальными нормативными актами Института порядке.

5. Материалы для компьютерного тестирования обучающихся в рамках проведения контроля наличия у обучающихся сформированных результатов обучения по дисциплине

Общие критерии оценивания

№ п/п	Процент правильных ответов	Оценка
1	86 % – 100 %	5 («отлично»)
2	70 % – 85 %	4 («хорошо»)
3	51 % – 69 %	3 («удовлетворительно»)
4	50 % и менее	2 («неудовлетворительно»)

Вариант 1

Номер вопроса и проверка сформированной компетенции

№ вопроса	Код компетенции	№ вопроса	Код компетенции
1	УК-1 (ИУК-1.1, ИУК-1.2)	11	УК-2 (ИУК-2.1, ИУК-2.2, ИУК-2.3)
2	УК-1 (ИУК-1.1, ИУК-1.2) УК-2 (ИУК-2.1, ИУК-2.2, ИУК-2.3)	12	УК-1 (ИУК-1.1, ИУК-1.2)
3	УК-1 (ИУК-1.1, ИУК-1.2)	13	УК-2 (ИУК-2.1, ИУК-2.2, ИУК-2.3)
4	УК-1 (ИУК-1.1, ИУК-1.2)	14	УК-1 (ИУК-1.1, ИУК-1.2)
5	ОПК-2 (ИУК-2.1, ИУК-2.2, ИУК-2.3)	15	УК-1 (ИУК-1.1, ИУК-1.2)
6	УК-1 (ИУК-1.1, ИУК-1.2)	16	УК-1 (ИУК-1.1, ИУК-1.2)
7	УК-1 (ИУК-1.1, ИУК-1.2)	17	УК-1 (ИУК-1.1, ИУК-1.2) УК-2 (ИУК-2.1, ИУК-2.2, ИУК-2.3)
8	УК-1 (ИУК-1.1, ИУК-1.2)	18	УК-1 (ИУК-1.1, ИУК-1.2)
9	УК-2 (ИУК-2.1, ИУК-2.2, ИУК-2.3)	19	УК-1 (ИУК-1.1, ИУК-1.2)
10	УК-1 (ИУК-1.1, ИУК-1.2)	20	УК-2 (ИУК-2.1, ИУК-2.2, ИУК-2.3)

Ключ ответов

№ вопроса	Верный ответ	№ вопроса	Верный ответ
1	1	11	4
2	2	12	2
3	2	13	2
4	1	14	4
5	2	15	1
6	2	16	4
7	4	17	1
8	1	18	1
9	4	19	1

10	1	20	4
----	---	----	---

Задание № 1.

Программа - это

Ответ:

- 1.** Упорядоченная последовательность команд (инструкций) компьютера для решения задачи
2. Совокупность программ обработки данных и необходимых для их эксплуатации документов
3. Программная реализация на компьютере решения задачи
4. Продукт программирования созданный для реализации

Задание № 2.

Программное обеспечение – это

Ответ:

1. Упорядоченная последовательность команд (инструкций) компьютера для решения задачи
- 2.** Совокупность программ обработки данных и необходимых для их эксплуатации документов
3. Программная реализация на компьютере решения задачи
4. Продукт программирования созданный для реализации

Задание № 3.

Не является обязательным свойством алгоритма:

Ответ:

1. Дискретность
- 2.** Число шагов
3. Определенность
4. Выполнимость
5. Массовость

Задание № 4.

Системный программист

Ответ:

- 1.** Занимается разработкой, эксплуатацией и сопровождением программного обеспечения, поддерживающего работоспособность компьютера
2. Осуществляет разработку и отладку программ для решения функциональных задач
3. Анализирует и проектирует комплекс взаимосвязанных программ для реализации функций предметной области
4. Нет такой категории

Задание № 5.

Прикладной программист

Ответ:

1. Занимается разработкой, эксплуатацией и сопровождением программного обеспечения, поддерживающего работоспособность компьютера
- 2.** Осуществляет разработку и отладку программ для решения функциональных задач
3. Анализирует и проектирует комплекс взаимосвязанных программ для реализации функций предметной области
4. Нет такой категории

Задание № 6.

Программные продукты shareware

Ответ:

1. Бесплатные программы, свободно распространяемые, поддерживаются самим пользователем
 - 2.** Условно-бесплатные программы
 3. Программы, поставляемые вместе с вычислительной техникой
 4. Коммерческое программное обеспечение
- 7

Задание № 7.

Основной характеристикой программы является:

Ответ:

1. Дискретность
2. Определенность
3. Выполнимость
- 4.** Полнота и системность функций обработки

Задание № 8.

Программные продукты имеют многообразие показателей качества, которые отражают следующие аспекты:

Ответ:

- 1.** Насколько просто, надежно, эффективно можно использовать программный продукт
2. Насколько программный продукт популярен
3. Какую стоимость программный продукт имеет
4. Как давно программный продукт произведен

Задание № 9.

В условиях существования рынка программных продуктов характеристикой не является:

Ответ:

1. Стоимость
2. Количество продаж
3. Время нахождения на рынке
- 4. Размер**
5. Известность фирмы-разработчика и программы
6. Наличие программных продуктов аналогичного назначения

Задание № 10.

Проблемно-ориентированные ППП

Ответ:

- 1.** Данный класс включает программные продукты, обеспечивающие решение функциональных задач определенной предметной области
2. Данный класс включает программные продукты, обеспечивающие независимо от предметной области математические, статистические и другие методы решения задач
3. Данный класс включает программные продукты, обеспечивающие решение технологических задач
4. Данный класс программных продуктов охватывает программы, обеспечивающие организационное управление деятельностью офиса

Задание № 11.

ППП общего назначения

Ответ:

1. Данный класс включает программные продукты, обеспечивающие решение функциональных задач определенной предметной области
2. Данный класс включает программные продукты, обеспечивающие независимо от предметной области математические, статистические и другие методы решения задач
3. Данный класс включает программные продукты, обеспечивающие решение технологических задач
- 4.** Данный класс программных продуктов охватывает программы, обеспечивающие организационное управление деятельностью офиса

Задание № 12.

Методо-ориентированные ППП

Ответ:

1. Данный класс включает программные продукты, обеспечивающие решение функциональных задач определенной предметной области

2. Данный класс включает программные продукты, обеспечивающие независимо от предметной области математические, статистические и другие методы решения задач
3. Данный класс включает программные продукты, обеспечивающие решение технологических задач
4. Данный класс программных продуктов охватывает программы, обеспечивающие организационное управление деятельностью офиса

Задание № 13.

Метод нисходящего проектирования

Ответ:

1. Логически взаимосвязанной совокупности функциональных элементов
2. Предполагает последовательное разложение общей функции обработки данных на простые функциональные элементы
3. Основано на модульной структуре программного продукта и типовых управляющих структурах алгоритмов обработки данных различных программных модулей
4. Такого метод проектирования нет

Задание № 14.

При разработке функциональной структуры алгоритма нет этапа:

Ответ:

1. Определения целей автоматизации предметной области и их иерархия
2. Устанавливается состав задач обработки
3. Определяются необходимые для решения задач функции обработки данных
4. Определяются ответственные лица

Задание № 15.

Функции ввода-вывода информации делятся на:

Ответ:

1. Однократно выполняемые и повторяющиеся
2. Вычислительные и логические
3. Промежуточные и конечные
4. Исходные и конечные

Задание № 16.

Модуль не характеризует:

Ответ:

1. Один вход и один выход
2. Функциональная завершенность
3. Логическая независимость
4. Технология проектирования

Задание № 17.

Состав и вид программных модулей, их назначение и характер использования в программе в значительной степени определяются

Ответ:

1. Инструментальными средствами
2. Функциональная завершенность
3. Логическая независимость
4. Слабые информационные связи с другими программными модулями

Задание № 18.

Структурное программирование основано

Ответ:

1. На модульной структуре программного продукта
2. На использовании визуальных сред разработки
3. На языках низкого уровня
4. Такого программирования нет

Задание № 19.

Метод объектно-ориентированного проектирования основывается на:

Ответ:

1. Модели построения системы как совокупности объектов абстрактного типа
2. На использовании визуальных сред разработки
3. На языках низкого уровня
4. Такого метод проектирования нет

Задание № 20.

Объектно-ориентированный подход не использует понятие:

Ответ:

1. Объект
2. Свойство
3. Метод
4. Процедура

Задания открытого типа (типовые задания, ситуационные задачи)

Номер вопроса и проверка сформированной компетенции

№ вопроса	Код компетенции	№ вопроса	Код компетенции
1	УК-1 (ИУК-1.1, ИУК-1.2)	11	УК-2 (ИУК-2.1, ИУК-2.2, ИУК-2.3)
2	УК-1 (ИУК-1.1, ИУК-1.2) УК-2 (ИУК-2.1, ИУК-2.2, ИУК-2.3)	12	УК-1 (ИУК-1.1, ИУК-1.2)
3	УК-1 (ИУК-1.1, ИУК-1.2)	13	УК-2 (ИУК-2.1, ИУК-2.2, ИУК-2.3)
4	УК-1 (ИУК-1.1, ИУК-1.2)	14	УК-1 (ИУК-1.1, ИУК-1.2)

5	ОПК-2 (ИУК-2.1, ИУК-2.2, ИУК-2.3)	15	УК-1 (ИУК-1.1, ИУК-1.2)
6	УК-1 (ИУК-1.1, ИУК-1.2)	16	УК-1 (ИУК-1.1, ИУК-1.2)
7	УК-1 (ИУК-1.1, ИУК-1.2) ОПК-3 (ИОПК-3.1, ИОПК-3.2.)	17	УК-1 (ИУК-1.1, ИУК-1.2) УК-2 (ИУК-2.1, ИУК-2.2, ИУК-2.3)
8	УК-1 (ИУК-1.1, ИУК-1.2)	18	УК-1 (ИУК-1.1, ИУК-1.2)
9	УК-2 (ИУК-2.1, ИУК-2.2, ИУК-2.3)	19	УК-1 (ИУК-1.1, ИУК-1.2)
10	УК-1 (ИУК-1.1, ИУК-1.2)	20	УК-2 (ИУК-2.1, ИУК-2.2, ИУК-2.3)

Ключ ответов к заданиям открытого типа

№ вопроса	Верный ответ
1	Классификация: POST /predict/class возвращает JSON с {"class": "A", "probabilities": {...}}. Регрессия: POST /predict/value возвращает {"value": 42.5, "confidence_interval": {...}}. Разные схемы валидации входных данных.
2	Extract: gety logic при сбое подключения к источнику. Clean: валидация диапазонов значений, логирование отброшенных записей. Transform: мониторинг статистик (среднее, дисперсия) после преобразований. Load: checksum для проверки целостности загруженных данных.
3	Компоненты: Prediction Service (gRPC/REST), Feature Cache (Redis), Model Update Listener (Kafka consumer), Model Storage (S3). Сообщения в Kafka: {"model_id": "knn_v1", "action": "update", "data_path": "s3://..."}.
4	UML Components: Data Collector → Feature Store (стрелка к Model Training) → Model Registry → Prediction API ← A/B Testing Framework. Все компоненты общаются через API Gateway.
5	FR: Загрузка документов (PDF, DOCX), обучение классификатора (поддержка incremental learning), предсказание категории. NFR: Latency <100ms, throughput >1000 doc/sec, availability 99.9% SLA. Data: минимальный размер обучающей выборки 10К документов, 10+ категорий.
6	Unit tests: функции нормализации цен. Integration: end-to-end пайплайн от сырых данных до предсказания. Data drift: еженедельная проверка распределения площади квартир (Kolmogorov-Smirnov test). Load test: 1000 RPS к prediction endpoint.
7	Архитектура: In-memory хранилище (например, FAISS) для быстрого поиска соседей. Online update: dual-write strategy - новая версия данных загружается в параллельный индекс, затем атомарный switch по версиям.
8	Масштабирование: горизонтальное scaling Feature Store и Prediction API. Стратегия: переход на cloud (AWS SageMaker, GCP Vertex AI) для elastic scaling. Обучение распределяется на несколько инстансов (Ray, Dask).
9	Версионирование: DVC для датасетов (хранит метаданные в Git, данные в S3), MLflow для моделей (артефакты + параметры + метрики). Воспроизводимость: любой эксперимент можно повторить командой dvc repro и mlflow run.
10	CI/CD: 1) pytest, 2) great_expectations для валидации данных, 3) train model (MLflow), 4) validate model (F1 > threshold), 5) deploy to staging (smoke tests), 6) canary deployment to production.
11	Детектирование: мониторинг PSI (Population Stability Index) для числовых признаков. Реакция: автоматический запуск пайплайна переобучения при PSI > 0.2, уведомление data scientists.
12	Размер выборки: power analysis с expected lift 2% и power 80%. Метрики: технические - AUC, бизнес-метрики - conversion rate, average order value. Статистическая значимость проверяется t-test.
13	API: POST /cluster {"user_features": [...]} → {"cluster_id": 3, "centroid_distance": 0.12}. Хранение: PostgreSQL таблица user_clusters. Переобучение: Airflow DAG запускает retraining еженедельно.
14	Баточная обработка: Apache Kafka → Spark Streaming (микробатчи по 1 минуте). Идемпотентность: дедупликация по message id, checkpointing в Kafka. Отказоустойчивость: репликация состояний в RocksDB.
15	Архитектура: Spark MLlib для distributed hierarchical clustering. Хранение дендрограмм: Apache Parquet (каждый уровень кластеризации - отдельный файл). Визуализация: отдельный сервис для построения дендрограмм из чанков.
16	Код-ревью: проверка unit tests для split criteria, обработка категориальных признаков (one-hot vs label encoding), сериализация через pickle или ONNX для production. Security: проверка на возможность model stealing через API.
17	Монолит: быстрее разработка для стартапа, проще дебаг. Микросервисы: лучше для enterprise с отдельными командами по компонентам. Сложности: distributed tracing (Jaeger), мониторинг

№ вопроса	Верный ответ
	(Prometheus + Grafana).
18	Фреймворк: REST API /explain возвращает SHAP values. Интеграция с UI: React компонент для визуализации feature importance. Хранение: кэширование объяснений в Redis на 24 часа.
19	Оптимизация: предвычисление kernel matrix для частых запросов. GPU: использование CUDA kernel для RBF. Пул воркеров: Celery с GPU-нодами, балансировка через Redis queue.
20	Архитектура: KernelFactory с registry доступных ядер. Плагины: загрузка .so библиотек с новыми ядрами через dynamic linking. Конфигурация: YAML файл с выбором ядра и параметрами.

1. Объясните различия между задачами классификации и регрессии в контексте проектирования API для ML-микросервиса. Какие разные контракты (endpoints, request/response схемы) потребуются для каждого типа задач?

2. При разработке ETL-пайплайна для подготовки данных к обучению ML-модели выделите ключевые этапы (извлечение, очистка, трансформация, загрузка) и для каждого этапа предложите методы обработки ошибок (error handling) и мониторинга качества данных.

3. Опишите процесс интеграции алгоритма k-NN в существующую корпоративную систему. Какие компоненты потребуются разработать (сервис предсказаний, кеширование результатов, обновление модели) и как организовать их взаимодействие через message queue?

4. Создайте UML-диаграмму компонентов для системы рекомендаций на основе ML. Включите компоненты: Data Collector, Feature Store, Model Training Service, Model Registry, Prediction API, A/B Testing Framework.

5. Напишите техническое задание (software requirements specification) на разработку системы классификации документов. Включите функциональные требования (загрузка, обучение, предсказание), нефункциональные требования (производительность: <100 мс на документ, доступность: 99.9%) и требования к данным.

6. Разработайте стратегию тестирования ML-модели регрессии для прогнозирования стоимости недвижимости. Включите: unit-тесты для функций предобработки, интеграционные тесты пайплайна, тесты на смещение данных (data drift) и нагрузочное тестирование API предсказаний.

7. Спроектируйте сервис для обслуживания (serving) модели k-NN в режиме реального времени. Какую архитектуру хранения обучающих данных вы выберете (in-memory база, векторная БД) и почему? Предложите механизм online обновления данных без остановки сервиса.

8. Проанализируйте, как изменится архитектура системы при увеличении объема данных в 100 раз (с 1 ГБ до 100 ГБ) для обучения k-NN. Какие

компоненты потребуется масштабировать горизонтально? Как изменится стратегия развертывания (on-premise vs cloud)?

9. Почему при развертывании ML-пайплайна в production необходима версионизация не только кода, но и данных, и моделей? Опишите подходы к версионированию (например, DVC для данных, MLflow для моделей) и как это влияет на воспроизводимость экспериментов.

10. Разработайте CI/CD пайплайн для ML-проекта, включающий этапы: тестирование кода, тестирование данных (проверка схемы, распределений), обучение модели, валидация модели (порог метрики), развертывание в staging и production.

11. Объясните концепцию "дрейфа данных" (data drift) применительно к ML-системам в production. Какую метрику можно использовать для его детектирования (например, KL-дивергенция распределений) и как автоматически реагировать на дрейф (retraining pipeline)?

12. Опишите процесс A/B тестирования новой ML-модели против старой в production. Как определить размер выборки для теста? Какие метрики бизнеса (конверсия, средний чек) нужно мониторить в дополнение к техническим метрикам модели (accuracy, MSE)?

13. Реализуйте сервис кластеризации пользователей (k-means) как отдельный микросервис. Опишите его API (REST/gRPC), схему хранения состояний кластеров, механизм периодического переобучения модели по расписанию (cron job).

14. Спроектируйте отказоустойчивую систему для алгоритма DBSCAN, работающую с потоковыми данными. Как организовать обработку данных батчами? Как обеспечить идемпотентность обработки при возможных сбоях?

15. Разработайте архитектуру системы для агломеративной кластеризации больших данных (>1 ТБ). Предложите использование распределенных вычислений (Apache Spark) и специализированных хранилищ для дендрограмм.

16. Опишите процесс код-ревью для реализации решающего дерева. На какие аспекты кода следует обратить особое внимание: корректность расчета критериев (Gini/Entropy), обработка missing values, возможность сериализации модели для production?

17. Сравните монолитную и микросервисную архитектуры для системы градиентного бустинга. В каких сценариях (стартап vs enterprise) каждая

архитектура предпочтительнее? Как микросервисы усложняют дебаггинг и мониторинг?

18. Для кредитного скоринга на основе решающих деревьев разработайте фреймворк для объяснения предсказаний (model interpretability). Какие библиотеки (SHAP, LIME) использовать и как интегрировать их в пользовательский интерфейс для аналитиков?

19. Спроектируйте высоконагруженный сервис для SVM классификации изображений. Как оптимизировать процесс вычисления ядерных функций? Стоит ли использовать GPU и как организовать пул воркеров для параллельной обработки?

20. Разработайте систему для динамического выбора ядерных функций в SVM на основе характеристик данных. Предложите архитектуру с плагинами (plugin architecture) для добавления новых ядер без перекомпиляции кода.

Вариант 2**Номер вопроса и проверка сформированной компетенции**

№ вопроса	Код компетенции	№ вопроса	Код компетенции
1	УК-1 (ИУК-1.1, ИУК-1.2)	11	УК-2 (ИУК-2.1, ИУК-2.2, ИУК-2.3)
2	УК-1 (ИУК-1.1, ИУК-1.2) УК-2 (ИУК-2.1, ИУК-2.2, ИУК-2.3)	12	УК-1 (ИУК-1.1, ИУК-1.2)
3	УК-1 (ИУК-1.1, ИУК-1.2)	13	УК-2 (ИУК-2.1, ИУК-2.2, ИУК-2.3)
4	УК-1 (ИУК-1.1, ИУК-1.2)	14	УК-1 (ИУК-1.1, ИУК-1.2)
5	УК-2 (ИУК-2.1, ИУК-2.2, ИУК-2.3)	15	УК-1 (ИУК-1.1, ИУК-1.2)
6	УК-1 (ИУК-1.1, ИУК-1.2)	16	УК-1 (ИУК-1.1, ИУК-1.2)
7	УК-1 (ИУК-1.1, ИУК-1.2)	17	УК-1 (ИУК-1.1, ИУК-1.2), УК-2 (ИУК-2.1, ИУК-2.2, ИУК-2.3)
8	УК-1 (ИУК-1.1, ИУК-1.2)	18	УК-1 (ИУК-1.1, ИУК-1.2)
9	УК-2 (ИУК-2.1, ИУК-2.2, ИУК-2.3)	19	УК-1 (ИУК-1.1, ИУК-1.2)
10	УК-1 (ИУК-1.1, ИУК-1.2)	20	УК-2 (ИУК-2.1, ИУК-2.2, ИУК-2.3)

Ключ ответов

№ вопроса	Верный ответ	№ вопроса	Верный ответ
1	1	11	4
2	2	12	3
3	3	13	3
4	4	14	1
5	2	15	4
6	3	16	2
7	2	17	4
8	3	18	2
9	4	19	2
10	2	20	3

Задание № 1.

Программный продукт, созданный с помощью инструментальных средств объектно-ориентированного программирования

Ответ:

1. Содержит объекты с их характерными свойствами, для которых разработан графический интерфейс пользователя
2. Не содержит объекты с их характерными свойствами
3. Не имеет графический интерфейс
4. Отличается простотой

Задание № 2.

Инкапсуляция

Ответ:

1. Способность объекта реагировать на запрос сообразно своему типу
- 2.** Означает сочетание структур данных с методами их обработки в абстрактных типах данных
3. Свойство системы использовать объекты с одинаковым интерфейсом без информации о типе и внутренней структуре объекта
4. Способ выделить набор значимых характеристик объекта, исключая из рассмотрения незначимые

Задание № 3.

В настоящее время нет подхода к созданию нейросетей:

Ответ:

1. Аппаратный
2. Программный
- 3.** Информационный
4. Гибридный

Задание № 4.

Нет такого направления применения искусственного интеллекта

Ответ:

1. Игры и творчество
2. Разработка естественно-языковых интерфейсов и машинный перевод
3. Новые архитектуры компьютеров
- 4.** Хранение информации

Задание № 5.

Знания - это

Ответ:

1. Отдельные факты, характеризующие объекты, процессы и явления в предметной области, а также их свойства
- 2.** Выявленные закономерности предметной области
3. Модели (структуры) данных в виде диаграмм, графиков, функций
4. Базы данных на машинных носителях

Задание № 6.

Не существует модели представления знаний:

Ответ:

1. Продукционные
2. Семантические сети
- 3.** Текстовые
4. Формальные логические модели

Задание № 7.

Семантическая сеть - это

Ответ:

1. Модель, основанная на правилах
2. Ориентированный граф
3. Абстрактный образ или ситуация
4. Модели, основанные на классическом исчислении предикатов

Задание № 8.

Фреймы

Ответ:

1. Модель, основанная на правилах
2. Ориентированный граф
3. Абстрактный образ или ситуация
4. Модели, основанные на классическом исчислении предикатов

Задание № 9.

Формальные логические модели

Ответ:

1. Модель, основанная на правилах
2. Ориентированный граф
3. Абстрактный образ или ситуация
4. Модели, основанные на классическом исчислении предикатов

Задание № 10.

Не бывает фреймов:

Ответ:

1. Фреймы-структуры
2. Фреймы-категории
3. Фреймы-сценарии
4. Фреймы-ситуации

Задание № 11.

В коллектив разработчиков экспертной системы не входит:

Ответ:

1. Эксперт
2. Инженер по знаниям
3. Программист
4. Специалист по базам данных

Задание № 12.

Не является инструментальным средством построения экспертных систем

Ответ:

1. Традиционные языки программирования
2. Языки искусственного интеллекта
3. Языки программирования низкого уровня

Задание № 13.

Основным направлением государственной политики в сфере информатизации является:

Ответ:

1. Обеспечение прав граждан на информацию, провозглашенных Конституцией Российской Федерации
2. Создание и поддержание необходимого для устойчивого развития общества уровня информационного потенциала
3. Создание условий для качественного и эффективного информационного обеспечения граждан
4. Создание отечественных современных информационных технологий и развитие производства средств для их реализации

Задание № 14.

Термин "критические" технологии применительно к информации означает

Ответ:

1. Именно уровень и масштабы применения технологий определяет эффективность достижения главных целей информатизации
2. Слабо исследованные направления применения информационных технологий
3. Важные для государственного управления
4. Применяемые впервые

Задание № 15.

К критическим технологиям нельзя отнести:

Ответ:

1. Многопроцессорные ЭВМ с параллельной структурой
2. Системы распознавания и синтеза речи, текста и изображений
3. Системы искусственного интеллекта и виртуальной реальности
4. Комплексы систем автоматизированного проектирования

Задание № 16.

Качество средств и систем информатизации сегодня определяется:

Ответ:

1. Стоимостью
2. Степенью экологичности
3. Производителем
4. Соответствие международному уровню

Задание № 17.

Целью сертификации средств информатизации, информационных технологий и услуг не является:

Ответ:

1. Защита пользователей средств и систем информатизации от приобретения средств и систем, которые представляют опасность для жизни, здоровья, имущества, а также для окружающей среды
2. Обеспечение разработчиков систем, а также широкого круга пользователей этих систем достоверной информацией
3. Обеспечение условий для информационного взаимодействия субъектов негосударственной принадлежности с субъектами государственной принадлежности
- 4.** Содействие улучшению условий труда персонала коммерческих предприятий

Задание № 18.

Все протоколы, на которые ссылается GOSIP, обладают общей характеристикой

Ответ:

1. Распространенность
- 2.** Доступность
3. Надежность
4. Правильность

Задание № 19.

Программой по стандартизации в сфере информатизации предусматривается сотрудничество с международными организациями по стандартизации при проведении работ

Ответ:

1. По двум приоритетным для Российской Федерации направлениям
- 2.** По трем приоритетным для Российской Федерации направлениям
3. По четырем приоритетным для Российской Федерации направлениям
4. Сотрудничество не предусмотрено

Задание № 20.

К направлению 1-го приоритета (программа по стандартизации в сфере информатизации предусматривает сотрудничество с международными организациями по стандартизации при проведении работ) относят

Ответ:

1. Сбор данных и системы идентификации
2. Пользовательский интерфейс
- 3.** Программная инженерия

Задания открытого типа (типовые задания, ситуационные задачи)

Номер вопроса и проверка сформированной компетенции

№ вопроса	Код компетенции	№ вопроса	Код компетенции
1	УК-1 (ИУК-1.1, ИУК-1.2)	11	УК-2 (ИУК-2.1, ИУК-2.2, ИУК-2.3)
2	УК-1 (ИУК-1.1, ИУК-1.2) УК-2 (ИУК-2.1, ИУК-2.2, ИУК-2.3)	12	УК-1 (ИУК-1.1, ИУК-1.2)
3	УК-1 (ИУК-1.1, ИУК-1.2)	13	УК-2 (ИУК-2.1, ИУК-2.2, ИУК-2.3)
4	УК-1 (ИУК-1.1, ИУК-1.2)	14	УК-1 (ИУК-1.1, ИУК-1.2)
5	ОПК-2 (ИУК-2.1, ИУК-2.2, ИУК-2.3)	15	УК-1 (ИУК-1.1, ИУК-1.2)
6	УК-1 (ИУК-1.1, ИУК-1.2)	16	УК-1 (ИУК-1.1, ИУК-1.2)
7	УК-1 (ИУК-1.1, ИУК-1.2) ОПК-3 (ИОПК-3.1, ИОПК-3.2.)	17	УК-1 (ИУК-1.1, ИУК-1.2) УК-2 (ИУК-2.1, ИУК-2.2, ИУК-2.3)
8	УК-1 (ИУК-1.1, ИУК-1.2)	18	УК-1 (ИУК-1.1, ИУК-1.2)
9	УК-2 (ИУК-2.1, ИУК-2.2, ИУК-2.3)	19	УК-1 (ИУК-1.1, ИУК-1.2)
10	УК-1 (ИУК-1.1, ИУК-1.2)	20	УК-2 (ИУК-2.1, ИУК-2.2, ИУК-2.3)

Ключ ответов к заданиям открытого типа

№ вопроса	Верный ответ
1	Онлайн: gRPC/HTTP2 + Protobuf для low-latency, API ключи per service. Офлайн: REST/HTTP1.1 + JSON для compatibility, OAuth2 для долгоживущих batch jobs. Механизмы: онлайн - connection pooling, офлайн - retry с exponential backoff.
2	Метрики: completeness (>95%), uniqueness (0 дублей), timeliness (данные <24ч), validity (значения в допустимых диапазонах). Оповещения: Slack/Teams webhook при нарушении, escalation после 3 нарушений подряд. Реализация: Great Expectations + Airflow data quality DAGs.
3	Решение: Pinecone для векторного поиска, обновление через Pinecone API. Схема репликации: тренировочный пайплайн → S3 → Lambda → Pinecone upsert. Консистенция: eventual consistency, метрика lag <5 минут.
4	ADR-001: Контекст - стартап с 3 разработчиками. Решение - монолит на FastAPI. Последствия - быстрый старт, но сложность масштабирования. ADR-002: Контекст - enterprise с 5 командами. Решение - микросервисы на gRPC. Последствия - higher operational overhead, но независимое scaling.
5	OpenAPI: endpoints: POST /v1/predict, GET /v1/explain/{prediction_id}, POST /v1/calibrate. Коды ошибок: 429 - too many requests, 503 - model not ready. Пример: {"customer_id": "123", "features": {...}} → {"churn_probability": 0.87, "prediction": "high risk"}.
6	Динамическое распределение: Thompson Sampling для баланса exploration-exploitation. Статистическая значимость: sequential testing с использованием mSPRT (sequential probability ratio test). Автовыбор: автоматический переход на лучшую модель при p-value <0.01.
7	Стратегия кеширования: двухуровневый кеш - L1: per-user (TTL 5 мин), L2: per-user-segment (TTL 1 час). Инвалидация: cache keys включают model_version и user_data_version. Обновление: cache warming при деплое новой модели.
8	TCO анализ: 1K RPS: \$500/мес (2 инстанса), 1M RPS: \$50K/мес (auto-scaling group). Оптимизации: spot instances для тренировки, модель quantization, кеширование предсказаний, использование резервированных инстансов. Команда: 2 MLE + 1 DevOps.
9	Версионирование фичей: Feature Store с time-travel (Apache Iceberg таблицы). Point-in-time: запрос фичей на момент as_of_time. Консистенция: использование одинаковых snapshot'ов для тренировки и инференса через shared timestamps.

№ вопроса	Верный ответ
10	GitOps пайплайн: Git repo → ArgoCD Application → Kubernetes (Deployment, ConfigMap, Service). Стратегия: canary с Istio (10% → 50% → 100%). Rollback: git revert + ArgoCD sync автоматически откатывает к предыдущей версии.
11	Мониторинг дрейфа: KS-тест для распределения предсказанных вероятностей между sliding windows (последние 24ч vs предыдущие 7 дней). Порог: KS statistic > 0.2 запускает alert. Реакция: автоматический запуск переобучения.
12	Изоляция: orthogonal layers через hash-based assignment (например, hash(user_id) mod 100). Минимальный эффект: MDE = 2% при alpha=0.05, beta=0.2, baseline conversion=10% → sample size ~15K пользователей на вариант.
13	Бессерверная архитектура: Cloud Scheduler → Pub/Sub → Cloud Function (Python 3.9) → BigQuery (данные) → AI Platform Training. Зависимости: packaging через Docker container для training job.
14	Apache Flink пайплайн: Kafka source → Flink windowing (tumbling 1 min) → DBSCAN operator. Поздние данные: allowed lateness 10 sec + side output для слишком поздних. Exactly-once: checkpointing каждые 30 sec + Kafka transactional producer.
15	Инкрементальный алгоритм: BIRCH (Balanced Iterative Reducing and Clustering using Hierarchies). Хранение дендрограммы: Cassandra с wide rows, где каждая строка - путь в дендрограмме. Обновление: вставка новых точек в существующие кластеры с пересчётом proximity матрицы.
16	DSL пример: RULE: IF income > 50000 AND debt_ratio < 0.4 THEN risk = "low". Трансляция: ANTLR grammar → AST → Python code generation. Исполнение: динамическая загрузка сгенерированных правил как Python модуля.
17	Автоматический feature engineering: использование библиотеки featuretools для deep feature synthesis. Избегание переобучения: feature selection через permutation importance с cross-validation. Отбор: selection top-50 features по mutual information.
18	Архитектура: ML model → SHAP explainer → Natural Language Generation (template-based) → UI/API. GDPR compliance: хранение объяснений 3 года, возможность экспорта пользователем. Безопасность: audit log всех доступов к объяснениям.
19	Edge оптимизация: 8-bit quantization через TensorFlow Lite, pruning опорных векторов с threshold 0.01, аппроксимация RBF ядра через random Fourier features. Тестирование: сравнение accuracy на holdout set до/после оптимизации (допустимое падение <1%).
20	JupyterLab плагин: extension на TypeScript + React, kernel на Python для вычислений. Интерактивность: ipywidgets для слайдеров, matplotlib для визуализации. Интеграция: сохранение конфигураций в notebook metadata.

1. Разрабатывая сервис машинного обучения, вы должны предложить две разные схемы API для режимов онлайн-инференса (realtime классификация) и офлайн-обработки (пакетная регрессия). Опишите различия в протоколах (REST vs gRPC), форматах данных (JSON vs Protobuf) и механизмах аутентификации (API ключи vs OAuth2).

2. Спроектируйте систему контроля качества данных (Data Quality Framework) для ML-пайплайна. Какие метрики качества данных (completeness, uniqueness, timeliness, validity) должны отслеживаться на каждом этапе? Как реализовать автоматическое оповещение о нарушениях SLA качества данных?

3. При интеграции k-NN алгоритма в распределённую систему возникла проблема с поиском ближайших соседей в реальном времени. Предложите решение с использованием векторной базы данных (например, Pinecone, Weaviate). Опишите схему репликации данных между тренировочным пайплайном и продакшен-системой.

4. Создайте архитектурные decision records (ADRs) для выбора между монолитной и микросервисной архитектурой ML-платформы. Укажите контекст, варианты решения, последствия (производительность, стоимость разработки, операционные расходы).
5. Разработайте контракт API (OpenAPI спецификацию) для сервиса предсказания оттока клиентов. Включите endpoint'ы для: получения предсказания, объяснения предсказания (SHAP values), калибровки модели (платтинг). Добавьте примеры запросов/ответов и коды ошибок.
6. Спроектируйте систему A/B/n-тестирования ML-моделей с поддержкой многоруких бандитов (Multi-Armed Bandit). Как распределять трафик между моделями динамически? Как обеспечить статистическую значимость при автоматическом выборе победителя?
7. Для сервиса рекомендаций на основе k-NN требуется кеширование результатов предсказаний. Разработайте стратегию кеширования с учётом персонализации (user-specific features) и инвалидации кеша при обновлении модели или данных пользователя.
8. Оцените стоимость владения (Total Cost of Ownership) ML-системы при масштабировании с 1К до 1М запросов в день. Включите затраты на: инфраструктуру (CPU/GPU, память, сеть), лицензии ПО, команду DevOps/MLE. Предложите оптимизации для снижения TCO.
9. Реализуйте систему версионирования фичей (Feature Store) с поддержкой point-in-time корректности. Как организовать хранение исторических значений фичей? Как обеспечить консистентность между тренировочными и инференс данными?
10. Разработайте GitOps пайплайн для развертывания ML-моделей в Kubernetes кластере. Опишите компоненты: Git репозиторий с манифестами, CI/CD инструмент (ArgoCD/Flux), стратегию rollout (canary, blue-green), механизм rollback.
11. Для обнаружения концептуального дрейфа (concept drift) в классификации спама предложите систему мониторинга, отслеживающую изменение распределения предсказанных вероятностей. Какой статистический тест (Kolmogorov-Smirnov, Cramer-von Mises) использовать? Как определить порог срабатывания?
12. При проведении online-экспериментов с ML-моделями возникла проблема симбиоза (interference) между экспериментами. Предложите методологию изоляции экспериментов (ортогональные слои, хеширование пользователей). Как рассчитать минимальный detectable effect?

13. Разработайте бессерверную (serverless) архитектуру для периодического переобучения моделей кластеризации (k-means). Используйте AWS Lambda / Google Cloud Functions для запуска обучения по расписанию. Как управлять зависимостями (ML libraries) в serverless среде?
14. Спроектируйте отказоустойчивый пайплайн для потоковой кластеризации (DBSCAN) с использованием Apache Flink. Как обрабатывать поздние данные (late data)? Как обеспечить ровно-однократную (exactly-once) семантику обработки?
15. Для агломеративной кластеризации больших наборов данных предложите инкрементальный алгоритм с возможностью добавления новых точек без полного пересчета. Как организовать хранение и обновление дендрограммы?
16. Разработайте DSL (Domain Specific Language) для описания решающих деревьев бизнес-аналитиками. DSL должен позволять задавать правила вида:
IF income > 50000 AND credit_score > 700 THEN "approve". Как транслировать DSL в исполняемый код (Python класс)?
17. Спроектируйте систему автоматического feature engineering для градиентного бустинга. Как генерировать полиномиальные фичи, взаимодействия? Как избежать переобучения на сгенерированных фичах? Предложите механизм отбора наиболее важных фич.
18. Для кредитного скоринга на EU рынке требуется соответствие GDPR "праву на объяснение". Разработайте архитектуру системы, которая для каждого отказа автоматически генерирует объяснение на естественном языке (например: "Отказ из-за высокой кредитной нагрузки и короткой кредитной истории").
19. Оптимизируйте инференс SVM модели для edge-устройств с ограниченными ресурсами. Предложите методы: квантование весов, pruning опорных векторов, аппроксимацию ядерных функций. Как протестировать точность после оптимизаций?
20. Создайте плагин для JupyterLab, позволяющий визуально настраивать ядерные функции SVM. Плагин должен предоставлять интерфейс для выбора типа ядра, подбора гиперпараметров через interactive слайдеры и immediate визуализацию разделяющей границы.