

Документ подписан простой электронной подписью

Информация о владельце:

ФИО: Порохня Андрей Александрович

Должность: И.о. директора института перспективной инженерии

Дата подписания: 19.06.2026 16:15:01

Уникальный программный ключ:

990bea3aecc48c23bd8699e48288f8d19d0c800

МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ
Федеральное государственное автономное образовательное учреждение
высшего образования
«СЕВЕРО-КАВКАЗСКИЙ ФЕДЕРАЛЬНЫЙ УНИВЕРСИТЕТ»

УТВЕРЖДАЮ

И.о. директора института
перспективной инженерии,
кандидат тех. наук, доцент

А.А. Порохня

Ф.И.О.

«__» _____ 20__ г.

ФОНД ОЦЕНОЧНЫХ СРЕДСТВ

для проведения текущего контроля успеваемости и промежуточной аттестации по

Искусственный интеллект в игровых приложениях

название дисциплины (модуля)

Направление подготовки

09.03.02 Информационные системы и технологии

Направленность (профиль)

Прикладное программирование и интернет-технологии

Форма обучения

очная

Год начала обучения

2026

Реализуется в 5 семестре

Введение

1. Назначение: Оценочные материалы (оценочные средства) предназначены для проведения текущего контроля успеваемости и промежуточной аттестации обучающихся, и представляют собой совокупность контрольно-измерительных материалов и методов их использования для измерения уровня достижения обучающимся установленных результатов обучения.

2. ФОС является приложением к программе дисциплины «Искусственный интеллект в игровых приложениях».

3. Разработчик: Березина В.А., ст. преподаватель департамента цифровых, робототехнических систем и электроники.

4. Проведена экспертиза ФОС.

Члены экспертной группы:

Председатель: Азаров И. В., и.о. директора департамента цифровых, робототехнических систем и электроники

(ФИО, должность)

Члены комиссии:

Шагрова Г. В., профессор департамента цифровых, робототехнических систем и электроники

(ФИО, должность)

Винокурский Д. Л., доцент департамента цифровых, робототехнических систем и электроники

(ФИО, должность)

Представитель _____ организации-работодателя:

(ФИО, должность)

Экспертное заключение: Оценочные материалы (оценочные средства) по дисциплине «Искусственный интеллект в игровых приложениях» составлены в соответствии с требованиями самостоятельно устанавливаемого Федерального государственного образовательного стандарта высшего образования по направлению 09.03.02 Информационные системы и технологии и позволяют оценить уровень сформированности компетенции ПК-2, ПК-4.

5. Срок действия ФОС определяется сроком реализации образовательной программы.

1. Описание показателей и критериев оценивания на различных этапах их формирования, описание шкал оценивания

Уровни сформированности компетенци(ий), индикатора (ов)	Дескрипторы			
	Минимальный уровень не достигнут (Неудовлетворительно) 2 балла	Минимальный уровень (удовлетворительно) 3 балла	Средний уровень (хорошо) 4 балла	Высокий уровень (отлично) 5 баллов
Компетенция: ПК-2 Способен использовать современные инструментальные средства и технологии программирования при разработке прикладного программного обеспечения вычислительных средств и систем различного функционального назначения				
Результаты обучения по дисциплине (модулю): проводит классификацию и осуществляет выбор современных инструментальных средств разработки прикладного программного обеспечения вычислительных средств и систем различного функционального назначения <i>Индикатор:</i> ПК-2ид-2.1	Не знает классификации IDE и фреймворков. Выбор инструментов случаен и не обоснован задачами проекта.	Знает основные среды разработки, но затрудняется сопоставить их функции с конкретными требованиями системы.	Уверенно классифицирует ПО. Обосновывает выбор стека технологий, опираясь на производительность и совместимость.	Проводит глубокий сравнительный анализ альтернативных средств. Выбирает оптимальный стек с учетом масштабируемости.
Результаты обучения по дисциплине (модулю): демонстрирует знания технологий программирования, разработки прикладного программного обеспечения вычислительных средств и систем различного функционального назначения <i>Индикатор:</i> ПК-2ид-2.2	Отсутствует понимание основ ООП или других парадигм. Не знает этапов жизненного цикла разработки ПО.	Демонстрирует фрагментарные знания технологий. Понимает общие принципы, но допускает ошибки в терминологии.	Владеет современным и технологиями разработки. Понимает принципы модульности и повторного использования кода.	Обладает системными знаниями. Способен обосновать выбор архитектурного паттерна и следовать стандартам кодирования.
Результаты обучения по дисциплине (модулю): использует современные инструментальные средства разработки прикладного программного обеспечения вычислительных средств и систем различного функционального назначения <i>Индикатор:</i> ПК-2ид-2.3	Не умеет пользоваться отладчиком и системами контроля версий. Допускает критические ошибки при настройке IDE.	Использует базовый функционал инструментов разработки. Требуется помощь при решении конфликтов в Git или сложной отладке.	Самостоятельно и эффективно применяет весь спектр инструментов (IDE, Git, линтеры) для создания прикладного ПО.	Владеет продвинутыми техниками: автоматизация сборки, профилирование кода, использование плагинов для оптимизации работы.
Результаты обучения по дисциплине (модулю):	Написанный код	Реализует простейшие	Разрабатывает ПО,	Создает высококачественные

применяет технологии программирования при разработке прикладного программного обеспечения вычислительных средств и систем различного функционального назначения <i>Индикатор:</i> ПК-2ид-2.4	неработоспособен или не соответствует синтаксису языка. Технологии применены неверно.	алгоритмы. Код содержит ошибки в логике или не оптимален по использованию ресурсов.	полностью выполняющее функции ТЗ. Код читаем, структурирован и содержит необходимые комментарии.	енное ПО. Применяет принципы SOLID/DRY, обеспечивает легкую расширяемость и поддержку программного кода.
Компетенция: ПК-4 Способен применять инструментальные средства и осваивать новые средства автоматизированного проектирования, разработки, тестирования и сопровождения программного обеспечения				
Результаты обучения по дисциплине (модулю): демонстрирует знания инструментальных средств для разработки приложений <i>Индикатор:</i> ПК-4ид-4.1	Не знает классификации и назначения инструментальных средств. Путает IDE, системы контроля версий и средства CASE.	Имеет фрагментарные знания. Перечисляет основные инструменты, но не может объяснить их роль в архитектуре приложения.	Уверенно описывает возможности современных средств разработки и CASE-средств, понимает их функциональные отличия.	Демонстрирует системные знания. Ориентируется в новейших инструментах и архитектурных особенностях средств разработки.
Результаты обучения по дисциплине (модулю): осуществляет выбор средств разработки, тестирования и сопровождения программного обеспечения <i>Индикатор:</i> ПК-4ид-4.2	Не способен выбрать средства разработки. Выбор не соответствует требованиям проекта или ТЗ.	Осуществляет выбор по шаблону или совету. Затрудняется аргументировать преимущества одного инструмента над другим.	Самостоятельно подбирает стек для разработки и тестирования, опираясь на базовые критерии производительности и ТЗ.	Выполняет глубокое обоснование выбора. Учитывает факторы масштабируемости, стоимости владения и поддержки в будущем.
Результаты обучения по дисциплине (модулю): применяет инструментальные средства, новые средства автоматизированного проектирования и разработки программного обеспечения <i>Индикатор:</i> ПК-4ид-4.3	Не умеет пользоваться средствами автоматизации проектирования. Работает исключительно вручную, нарушая стандарты.	Применяет инструменты разработки и CASE-средства с ошибками. Требуется постоянных консультаций и подсказок.	Эффективно использует средства автоматизированного проектирования для создания модулей ПО и схем данных.	Владеет продвинутыми техниками проектирования. Интегрирует различные средства автоматизации в единый процесс разработки.
Результаты обучения по дисциплине (модулю): осваивает новые средства автоматизированного проектирования и разработки программного обеспечения <i>Индикатор:</i> ПК-4ид-4.4	Не проявляет способности к самостоятельному изучению новых инструментов. Боится менять привычный	Осваивает новые средства медленно, только при наличии подробных инструкций	Самостоятельно изучает документацию и базовый функционал новых версий ПО и инструментов разработки.	Быстро адаптируется к новым технологиям. Способен внедрять передовые инструменты автоматизации

	стек технологий.	или прямого обучения.		в рабочий процесс.
Результаты обучения по дисциплине (модулю): осуществляет сопровождение программного обеспечения <i>Индикатор:</i> ПК-4ид-4.5	Не понимает процесса сопровождения . Код невозможно поддерживать, документация отсутствует или неверна.	Выполняет простейшие операции по сопровождению (исправление мелких багов) под руководством наставника.	Осуществляет штатное сопровождение ПО: рефакторинг, обновление версий, актуализацию документации .	Профессионально управляет жизненным циклом ПО. Разрабатывает регламенты обновления и поддержки сложных систем.
Результаты обучения по дисциплине (модулю): разрабатывает и использует методики тестирования <i>Индикатор:</i> ПК-4ид-4.6	Не умеет составлять тест-кейсы. Код сдается без проверки. Понятие «методика тестирования» отсутствует.	Проводит ручное тестирование по простым сценариям. Не умеет документировать результаты и баг-репорты.	Разрабатывает методики тестирования (модульное, интеграционное). Использует автоматизированные средства проверки.	Создает комплексные стратегии тестирования. Владеет инструментами нагрузочного и автоматизированного тестирования.

ОЦЕНОЧНЫЕ СРЕДСТВА ДЛЯ ПРОВЕРКИ УРОВНЯ СФОРМИРОВАННОСТИ КОМПЕТЕНЦИЙ

Номер задания	Правильный ответ	Содержание вопроса	Компетенция
		Вопросы для собеседования:	
1		1. Архитектура игрового ИИ. Основные уровни: восприятие, принятие решений и движение. 2. Векторная математика в Pygame. Применение класса Vector2 для расчета направления и скорости агента. 3. Кинематическое движение против рулевого (Steering). В чем разница между мгновенным изменением скорости и приложением сил. 4. Алгоритм Seek (прибытие). Математическая формула вычисления вектора желаемой скорости (Desired Velocity). 5. Алгоритм Arrive. Как реализовать плавное замедление агента при приближении к целевой точке. 6. Алгоритм Wander. Реализация случайного блуждания с помощью проецируемой окружности и смещения вектора. 7. Модель Бойдов (Boids). Назначение и логика работы правила Separation (разделение). 8. Модель Бойдов (Boids). Как правило Alignment (выравнивание) влияет на групповое поведение. 9. Модель Бойдов (Boids). Реализация правила Cohesion (сплочение) через центр масс соседей. 10. Представление игрового пространства. Плюсы и минусы использования регулярных сеток (Grids) по сравнению с графами. 11. Алгоритм Дейкстры. Принцип работы и условия, при которых он находит гарантированно кратчайший путь. 12. Алгоритм A*. Роль эвристической функции $h(n)$ и её влияние на скорость и точность поиска. 13. Допустимость эвристики в A*. Что произойдет, если эвристика будет завышать реальное расстояние до цели. 14. Оптимизация поиска пути. Метод иерархического поиска (HPA*) и использование пулов путей. 15. Конечные автоматы (FSM). Определение состояний, переходов и событий в контексте	ПК-2, ПК-4

		игрового цикла. 16. Проблема масштабируемости FSM. Почему при большом количестве состояний код становится трудноподдерживаемым. 17. Стековые конечные автоматы (HFSM). Как реализовать временные состояния (например, испуг), возвращающие агента к прошлой логике. 18. Деревья поведения (Behavior Trees). Принцип работы узла-селектора (Selector) и его приоритеты. 19. Деревья поведения (Behavior Trees). Отличие узла-последовательности (Sequence) от селектора. 20. Декораторы в Behavior Trees. Примеры использования для инверсии условий или ограничения частоты проверок. 21. Игры с полной информацией. Определение дерева игровых состояний и роль терминальных узлов. 22. Алгоритм Минимакс. Как работает рекурсивный выбор хода для игрока-максимизатора и игрока-минимизатора. 23. Альфа-бета отсечение. Математический смысл переменных alpha и beta для оптимизации перебора. 24. Обучение с подкреплением (RL). Понятия среды (Environment), состояния (State) и функции награды (Reward). 25. Q-Learning. Смысл уравнения Беллмана и использование Q-таблицы для хранения опыта агента.	
		Задания закрытого типа:	
26.	2	Какой вектор в алгоритме seek (прибытие) представляет собой разность между целевой и текущей позицией агента? 1) Текущая скорость (Velocity) 2) Желаемая скорость (Desired Velocity) 3) Сила поворота (Steering Force) 4) Вектор ускорения	ПК-2, ПК-4
27.	2	В чем основное отличие алгоритма arrive от seek? 1) Arrive использует случайное направление 2) Arrive замедляет агента при входе в радиус торможения	ПК-2, ПК-4

		3) Arrive игнорирует препятствия 4) Arrive работает только в 3D пространстве	
28.	3	Какое правило флокинга (boids) отвечает за то, чтобы агенты не сталкивались друг с другом? 1) Cohesion (сплочение) 2) Alignment (выравнивание) 3) Separation (разделение) 4) Wander (блуждание)	ПК-2, ПК-4
29.	2	Что вычисляет правило alignment в модели бойдов? 1) Среднюю позицию соседей 2) Среднее направление (вектор скорости) соседей 3) Расстояние до ближайшего хищника 4) Вектор отталкивания от границ экрана	ПК-2, ПК-4
30.	2	Какая структура данных наиболее эффективна для хранения открытого списка (open list) в алгоритме a*? 1) Стэк (Stack) 2) Очередь с приоритетом (Priority Queue / Heap) 3) Связанный список 4) Двумерный массив	ПК-2, ПК-4
31.	2	Что представляет собой эвристическая функция $h(n)$ в алгоритме a*? 1) Стоимость пути от стартовой точки до текущего узла 2) Ожидаемая стоимость пути от текущего узла до цели 3) Общая стоимость пути $f(n)$ 4) Количество препятствий на карте	ПК-2, ПК-4
32.	3	Какое расстояние (эвристика) лучше всего подходит для сетки, где разрешено движение только по 4 направлениям (вверх, вниз, влево, вправо)? 1) Евклидово расстояние 2) Расстояние Чебышёва 3) Манхэттенское расстояние	ПК-2, ПК-4

		4) Квадрат евклидова расстояния	
33.	1	Что произойдет в алгоритме a^*, если эвристика $h(n)$ всегда будет равна 0? 1) Алгоритм станет эквивалентен алгоритму Дейкстры 2) Алгоритм перестанет находить путь 3) Алгоритм найдет путь мгновенно 4) Агент начнет двигаться по спирали	ПК-2, ПК-4
34.	2	Какое состояние в конечном автомате (fsm) обычно является активным по умолчанию при старте? 1) Финальное состояние 2) Начальное состояние (Initial State) 3) Состояние перехода 4) Пустое состояние	ПК-2, ПК-4
35.	2	В чем главный недостаток простых конечных автоматов (fsm) при усложнении логики ии? 1) Низкая скорость выполнения 2) Комбинаторный взрыв связей и состояний ('спагетти-код') 3) Невозможность работы с векторами 4) Ограничение на количество кадров в секунду	ПК-2, ПК-4
36.	2	Как называется узел в дереве поведения (behavior tree), который возвращает success, если хотя бы один из его потомков вернул success? 1) Sequence (Последовательность) 2) Selector (Селектор) 3) Decorator (Декоратор) 4) Action (Действие)	ПК-2, ПК-4
37.	3	В каком порядке узел sequence в behavior tree обходит свои дочерние узлы? 1) Случайно 2) Справа налево 3) Слева направо до первого Failure	ПК-2, ПК-4

		4) Одновременно	
38.	1	Что делает узел-декоратор inverter в дереве поведения? 1) Меняет Success на Failure и наоборот 2) Зацикливает выполнение узла 3) Останавливает работу дерева 4) Инвертирует координаты агента	ПК-2, ПК-4
39.	3	Для каких игр предназначен алгоритм минимакс? 1) Шутеры от первого лица 2) Азартные игры на удачу 3) Антагонистические игры с полной информацией (шахматы, крестики-нолики) 4) Многопользовательские песочницы	ПК-2, ПК-4
40.	2	Какую цель преследует игрок 'минимизатор' в алгоритме минимакс? 1) Увеличить значение функции оценки 2) Уменьшить значение функции оценки 3) Затянуть время игры 4) Сделать случайный ход	ПК-2, ПК-4
41.	2	Что позволяет сделать альфа-бета отсечение? 1) Всегда находить более лучший ход, чем Минимакс 2) Сократить количество перебираемых узлов в дереве игры 3) Увеличить глубину дерева до бесконечности 4) Добавить элемент случайности в игру	ПК-2, ПК-4
42.	2	В обучении с подкреплением (rl), что такое 'reward' (награда)? 1) Список доступных действий 2) Числовой сигнал обратной связи от среды 3) Текущие координаты агента 4) Имя игрока	ПК-2, ПК-4
43.	2	За что отвечает параметр 'learning rate' (скорость обучения) в формуле q-learning?	ПК-2, ПК-4

		1) За скорость передвижения агента в Pygame 2) За степень обновления старых значений Q новыми данными 3) За частоту кадров в секунду 4) За количество итераций цикла поиска пути	
44.	1	Что такое 'state' (состояние) в контексте задачи навигации агента в лабиринте? 1) Текущее положение агента (например, номер клетки) 2) Клавиша, нажатая пользователем 3) Цвет фона в окне Pygame 4) Версия Python	ПК-2, ПК-4
45.	2	Какой метод pygame используется для получения времени, прошедшего с предыдущего кадра? 1) <code>pygame.display.update()</code> 2) <code>pygame.time.Clock.tick()</code> 3) <code>pygame.event.get()</code> 4) <code>pygame.init()</code>	ПК-2, ПК-4
46.	2	Какая библиотека python чаще всего используется вместе с pygame для ускорения матричных операций ии? 1) Django 2) NumPy 3) Requests 4) BeautifulSoup	ПК-2, ПК-4
47.	2	Что описывает 'политика' (policy) в обучении с подкреплением? 1) Правила отрисовки спрайтов 2) Стратегию выбора действия в зависимости от состояния 3) Лицензионное соглашение программы 4) Способ хранения игровых ассетов	ПК-2, ПК-4
48.	3	В алгоритме a^* , что произойдет, если узел уже находится в закрытом списке (closed list)?	ПК-2, ПК-4

		1) Он будет обработан заново 2) Он будет удален из памяти 3) Он будет проигнорирован, так как кратчайший путь к нему уже найден 4) Алгоритм выдаст ошибку	
49.	3	Какой тип графа используется в pygame для реализации ии на квадратной сетке? 1) Несвязный граф 2) Направленный ациклический граф (DAG) 3) Взвешенный граф (сетка как граф смежности) 4) Дерево поиска	ПК-2, ПК-4
50.	2	Что делает метод <code>vector2.normalize()</code> в pygame? 1) Обнуляет вектор 2) Приводит длину вектора к 1, сохраняя направление 3) Поворачивает вектор на 90 градусов 4) Увеличивает длину вектора в два раза	ПК-2, ПК-4
		Задания открытого типа:	
51.	pygame	Как называется библиотека python, используемая в данном курсе для визуализации и обработки игровых событий?	ПК-2, ПК-4
52.	вектор	Как называется математический объект, имеющий длину и направление, используемый для представления скорости агента?	ПК-2, ПК-4
53.	seek	Как называется алгоритм крейга рейнольдса, отвечающий за преследование цели агентом?	ПК-2, ПК-4
54.	arrive	Как называется алгоритм рулевого управления, заставляющий агента замедляться при приближении к цели?	ПК-2, ПК-4
55.	separation	Как называется поведение в модели бойдов, отвечающее за сохранение дистанции между соседями во избежание столкновений?	ПК-2, ПК-4

56.	alignment	Как называется поведение в модели бойдов, заставляющее агентов двигаться со средней скоростью группы?	ПК-2, ПК-4
57.	cohesion	Как называется поведение в модели бойдов, направляющее агента к центру масс видимых соседей?	ПК-2, ПК-4
58.	агент	Как называется универсальный термин для обозначения автономного интеллектуального существа в симуляции?	ПК-2, ПК-4
59.	взвешенный	Как называется граф, в котором ребра имеют числовое значение стоимости перехода?	ПК-2, ПК-4
60.	дейкстры	Как называется алгоритм поиска кратчайшего пути на графе, использующий только накопленную стоимость $g(n)$?	ПК-2, ПК-4
61.	эвристика	Как называется оценочная функция расстояния до цели в алгоритме a^* ?	ПК-2, ПК-4
62.	гексагональная	Как называется тип сетки, где ячейка имеет шесть соседей?	ПК-2, ПК-4
63.	автомат	Как называется модель управления, состоящая из набора дискретных состояний и переходов между ними?	ПК-2, ПК-4
64.	sequence	Как называется узел дерева поведения, который прекращает выполнение после первой же неудачи (failure) своего потомка?	ПК-2, ПК-4
65.	selector	Как называется узел дерева поведения, который ищет первого успешного (success) потомка?	ПК-2, ПК-4
66.	декоратор	Как называется тип узла в дереве поведения, который изменяет результат работы своего единственного дочернего узла?	ПК-2, ПК-4
67.	минимакс	Как называется алгоритм поиска лучшего хода в играх с полной информацией и двумя игроками?	ПК-2, ПК-4

68.	отсечение	Как называется метод оптимизации алгоритма минимакс, отсекающий заведомо плохие ветви дерева?	ПК-2, ПК-4
69.	альфа	Как называется переменная в альфа-бета отсечении, представляющая наилучшее значение для игрока-максимизатора?	ПК-2, ПК-4
70.	награда	Как называется компонент обучения с подкреплением, который выдает агенту положительный или отрицательный сигнал?	ПК-2, ПК-4
71.	q-таблица	Как называется таблица в обучении с подкреплением, хранящая ценности действий для каждого состояния?	ПК-2, ПК-4
72.	нормализация	Как называется процесс приведения длины вектора к единице?	ПК-2, ПК-4
73.	wander	Как называется алгоритм рулевого управления для имитации случайного, но плавного блуждания?	ПК-2, ПК-4
74.	гамма	Как называется параметр в обучении с подкреплением, определяющий важность будущих наград (коэффициент дисконтирования)?	ПК-2, ПК-4
75.	закрытый	Как называется список в алгоритме a^* , содержащий узлы, которые уже были посещены и обработаны?	ПК-2, ПК-4

2. Описание шкалы оценивания

В рамках рейтинговой системы успеваемость студентов по каждой дисциплине оценивается в ходе текущего контроля и промежуточной аттестации. Рейтинговая система оценки знаний студентов основана на использовании совокупности контрольных мероприятий по проверке пройденного материала (контрольных точек), оптимально расположенных на всем временном интервале изучения дисциплины. Принципы рейтинговой системы оценки знаний студентов основываются на требованиях, описанных в Положении об организации образовательного процесса на основе рейтинговой системы оценки знаний студентов в ФГАОУ ВО «СКФУ».

3. Критерии оценивания компетенций заданий открытого типа

Оценка **«отлично»** выставляется студенту, если являются верными результаты классификации и выбора современных инструментальных средств разработки прикладного программного обеспечения, демонстрации знаний технологий программирования и разработки прикладного программного обеспечения вычислительных систем, использования современных инструментальных средств разработки прикладного программного обеспечения, применения технологий программирования при разработке прикладного программного обеспечения, демонстрации знаний инструментальных средств для разработки приложений, выбора средств разработки, тестирования и сопровождения программного обеспечения, применения инструментальных средств и новых средств автоматизированного проектирования, освоения новых средств автоматизированного проектирования и разработки программного обеспечения, сопровождения программного обеспечения в процессе его эксплуатации, разработки и использования методик тестирования программных продуктов; представленный документ не содержит ошибок и решены все задачи. Компетенции ПК-2, ПК-4 освоены на высоком уровне.

Оценка **«хорошо»** выставляется студенту, если являются верными результаты классификации и выбора современных инструментальных средств разработки прикладного программного обеспечения, демонстрации знаний технологий программирования и разработки прикладного программного обеспечения вычислительных систем, использования современных инструментальных средств разработки прикладного программного обеспечения, применения технологий программирования при разработке прикладного программного обеспечения, демонстрации знаний инструментальных средств для разработки приложений, выбора средств разработки, тестирования и сопровождения программного обеспечения, применения инструментальных средств и новых средств автоматизированного проектирования, освоения новых средств автоматизированного проектирования и разработки программного обеспечения, сопровождения программного обеспечения в процессе его эксплуатации, разработки и использования методик тестирования программных продуктов; отчетный документ содержит незначительные ошибки, не влияющие на общий результат и не представлены отдельные задачи, не влияющие на общий результат. Компетенции ПК-2, ПК-4 освоены на среднем уровне.

Оценка **«удовлетворительно»** выставляется студенту, если в основном верными являются результаты классификации и выбора современных инструментальных средств разработки прикладного программного обеспечения, демонстрации знаний технологий программирования и разработки прикладного программного обеспечения вычислительных систем, использования современных инструментальных средств разработки прикладного программного обеспечения, применения технологий программирования при разработке прикладного программного обеспечения, демонстрации знаний инструментальных средств для разработки приложений, выбора средств разработки, тестирования и сопровождения программного обеспечения, применения инструментальных средств и новых средств автоматизированного проектирования, освоения новых средств автоматизированного проектирования и разработки программного обеспечения, сопровождения программного обеспечения,

ния в процессе его эксплуатации, разработки и использования методик тестирования программных продуктов; отчетный документ содержит ошибки, не позволяющие оценить качество разработки; представлены не все данные и задачи решены не в полном объеме. Компетенции ПК-2, ПК-4 освоены на минимальном уровне.

Оценка **«неудовлетворительно»** выставляется студенту, если являются неверными результаты классификации и выбора современных инструментальных средств разработки прикладного программного обеспечения, демонстрации знаний технологий программирования и разработки прикладного программного обеспечения вычислительных систем, использования современных инструментальных средств разработки прикладного программного обеспечения, применения технологий программирования при разработке прикладного программного обеспечения, демонстрации знаний инструментальных средств для разработки приложений, выбора средств разработки, тестирования и сопровождения программного обеспечения, применения инструментальных средств и новых средств автоматизированного проектирования, освоения новых средств автоматизированного проектирования и разработки программного обеспечения, сопровождения программного обеспечения в процессе его эксплуатации, разработки и использования методик тестирования программных продуктов; представленный документ содержит много грубых ошибок, не представлены ключевые данные и задачи решены недостаточно полно. Компетенции ПК-2, ПК-4 не сформированы.

4. Критерии оценивания компетенций заданий закрытого типа

Оценка «отлично» выставляется студенту, если он правильно ответил на 88-100% от общего количества заданий. Компетенции ПК-2, ПК-4 освоены на высоком уровне.

Оценка «хорошо» выставляется студенту, если он правильно ответил на 72-87% от общего количества заданий. Компетенции ПК-2, ПК-4 освоены на среднем уровне.

Оценка «удовлетворительно» выставляется студенту, если он правильно ответил на 53-71% от общего количества заданий. Компетенции ПК-2, ПК-4 освоены на минимальном уровне.

Оценка «неудовлетворительно» выставляется студенту, выставляется студенту, если он правильно ответил на менее 53% от общего количества заданий. Компетенции ПК-2, ПК-4 не сформированы.