

Документ подписан простой электронной подписью
Информация о владельце:
ФИО: Степанов Павел Иванович
Должность: Руководитель
Дата подписания: 25.02.2026 13:39:13
Уникальный программный ключ:
8c65c591e26b2d8e460927740c752622aa3b295

МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ
Федеральное государственное автономное образовательное учреждение высшего образования
«Национальный исследовательский ядерный университет «МИФИ»
Новоуральский технологический институт
филиал федерального государственного автономного образовательного учреждения высшего
образования «Национальный исследовательский ядерный университет «МИФИ»
(НТИ НИЯУ МИФИ)
Колледж НТИ

Цикловая методическая комиссия информационных технологий

ОДОБРЕНО

Учёным Советом НТИ НИЯУ МИФИ

Протокол № 2 от 05 февраля 2024 г.

ФОНД ОЦЕНОЧНЫХ СРЕДСТВ
УЧЕБНОЙ ДИСЦИПЛИНЫ ОП.04 «ОСНОВЫ
АЛГОРИТМИЗАЦИИ И ПРОГРАММИРОВАНИЯ»

для студентов колледжа НТИ НИЯУ МИФИ,
обучающихся по программе среднего профессионального образования

специальность 09.02.07

«Информационные системы и программирование»

очная форма обучения

на базе основного общего образования

квалификация

программист

Новоуральск 2024

ОДОБРЕНО:
на заседании
цикловой методической комиссии
информационных технологий
Протокол № 2 от 02.02.2024 г.
Председатель ЦМК ИТ

_____ И.И. Горницкая

Разработан на основе рабочей
программы учебной дисциплины
ОП.04 «Основы алгоритмизации и
программирования» по специальности
09.02.07 «Информационные системы
и программирование»

Фонд оценочных средств учебной дисциплине ОП.04 «Основы алгоритмизации и программирования» – Новоуральск: Изд-во колледжа НТИ НИЯУ МИФИ, 2024. –20с.

АННОТАЦИЯ

Фонд оценочных средств предназначен для текущего контроля и промежуточной аттестации обучающихся по специальности 09.02.07 Информационные системы и программирование на соответствие их персональных достижений поэтапным требованиям программы подготовки специалистов среднего звена по учебной дисциплине ОП.04 «Основы алгоритмизации и программирования». Комплектация фонда оценочных средств: паспорт, программа оценивания, оценочные средства для текущего контроля и промежуточной аттестации по учебной дисциплине, критерии оценивания. В паспорте фонда оценочных средств указаны: место дисциплины в структуре программы подготовки специалистов среднего звена, требования ФГОС СПО к результатам освоения, перечень формируемых компетенций, компоненты фонда оценочных средств.

Разработчик: Лебедева А.Н., преподаватель первой категории
ЦМК информационных технологий.

Редактор: Горницкая И.И., преподаватель высшей категории,
председатель ЦМК информационных технологий

СОДЕРЖАНИЕ

1. ПАСПОРТ ФОНДА ОЦЕНОЧНЫХ СРЕДСТВ ПО УЧЕБНОЙ ДИСЦИПЛИНЕ ОП.04 «ОСНОВЫ АЛГОРИТМИЗАЦИИ И ПРОГРАММИРОВАНИЯ»	4
2. ПРОГРАММА ОЦЕНИВАНИЯ УЧЕБНОЙ ДИСЦИПЛИНЫ ОП.04 «ОСНОВЫ АЛГОРИТМИЗАЦИИ И ПРОГРАММИРОВАНИЯ»	7
3. ОЦЕНОЧНЫЕ СРЕДСТВА ДЛЯ ТЕКУЩЕГО КОНТРОЛЯ	8
3.1 Тестовая форма контроля	8
3.2 Критерии оценки результатов выполнения тестовых заданий	10
3.3 Письменная форма контроля	10
3.4. Критерии оценки результатов выполнения письменных заданий ...	12
3.5 Устная форма контроля.....	12
3.6 Критерии оценки результатов выполнения устных заданий	15
4. ОЦЕНОЧНЫЕ СРЕДСТВА ДЛЯ ПРОВЕДЕНИЯ ПРОМЕЖУТОЧНОЙ АТТЕСТАЦИИ	16
4.1 Критерии оценивания знаний, обучающихся на семестровом зачете.	16
4.2 Критерии оценивания знаний, обучающихся на экзамене.....	16

1. ПАСПОРТ ФОНДА ОЦЕНОЧНЫХ СРЕДСТВ ПО УЧЕБНОЙ ДИСЦИПЛИНЕ ОП.04 «ОСНОВЫ АЛГОРИТМИЗАЦИИ И ПРОГРАММИРОВАНИЯ»

Фонд оценочных средств является составной частью учебно-методических документов, обеспечивающих реализацию основной профессиональной образовательной программы СПО по специальности 09.02.07 Информационные системы и программирование.

Фонд оценочных средств предназначен для проверки результатов освоения учебной дисциплины ОП.04 «Основы алгоритмизации и программирования».

Место учебной дисциплины в структуре основной профессиональной образовательной программы: дисциплина ОП.04 «Основы алгоритмизации и программирования» принадлежит к общепрофессиональному циклу.

Цель и планируемые результаты освоения учебной дисциплины:

В результате изучения учебной дисциплины студент должен:

Уметь:

- Разрабатывать алгоритмы для конкретных задач;
- Использовать программы для графического отображения алгоритмов;
- Определять сложность работы алгоритмов;
- Работать в среде программирования;
- Реализовывать построенные алгоритмы в виде программ на конкретном языке программирования;
- Оформлять код программы в соответствии со стандартом кодирования;
- Выполнять проверку, отладку кода программы.

Знать:

- Понятие алгоритмизации, свойства алгоритмов, общие принципы построения алгоритмов, основные алгоритмические конструкции;

- Эволюцию языков программирования, их классификацию, понятие системы программирования;
- Основные элементы языка, структуру программы, операторы и операции, управляющие структуры, структуры данных, файлы, классы памяти;
- Подпрограммы, составление библиотек подпрограмм;
- Объектно-ориентированную модель программирования;
- основные принципы объектно-ориентированного программирования на примере алгоритмического языка: понятие классов и объектов, их свойств и методов, инкапсуляция и полиморфизма, наследования и переопределения.

В результате изучения дисциплины студент должен освоить профессиональные и общие компетенции:

Перечень формируемых компетенций в соответствии с требованиями ФГОС СПО:

Общие компетенции.

Код	Наименование общих компетенций
ОК 01	Выбирать способы решения задач профессиональной деятельности, применительно к различным контекстам.
ОК 02	Использовать современные средства поиска, анализа и интерпретации информации и информационные технологии для выполнения задач профессиональной деятельности.
ОК 04	Эффективно взаимодействовать и работать в коллективе и команде.
ОК 05	Осуществлять устную и письменную коммуникацию на государственном языке Российской Федерации с учетом особенностей социального и культурного контекста.

Профессиональные компетенции

Код	Наименование профессиональных компетенций
ПК 1.1	Формировать алгоритмы разработки программных модулей в соответствии с техническим заданием
ПК 1.2	Разрабатывать программные модули в соответствии с техническим заданием
ПК 1.3	Выполнять отладку программных модулей с использованием специализированных программных средств
ПК 1.4	Выполнять тестирование программных модулей
ПК 1.5	Осуществлять рефакторинг и оптимизацию программного кода
ПК 2.4	Осуществлять разработку тестовых наборов и тестовых сценариев для программного обеспечения
ПК 2.5	Производить инспектирование компонент программного обеспечения на предмет соответствия стандартам кодирования

2. ПРОГРАММА ОЦЕНИВАНИЯ
УЧЕБНОЙ ДИСЦИПЛИНЫ ОП.04 «ОСНОВЫ
АЛГОРИТМИЗАЦИИ И ПРОГРАММИРОВАНИЯ»

№ п/п	Контролируемые элементы учебной дисциплины	Контролируемые компетенции	Форма аттестации
1	2	3	4
1	Темы 1-7	ОК 1, ОК 2, ОК 4, ОК 5, ПК 1.1, ПК 1.2, ПК 1.3, ПК 1.4, ПК 1.5, ПК 2.4, ПК 2.5	III семестр семестровый зачет
2	Темы 8-14	ОК 1, ОК 2, ОК 4, ОК 5, ПК 1.1, ПК 1.2, ПК 1.3, ПК 1.4, ПК 1.5, ПК 2.4, ПК 2.5	IV семестр экзамен

3. ОЦЕНОЧНЫЕ СРЕДСТВА ДЛЯ ТЕКУЩЕГО КОНТРОЛЯ

Текущий контроль знаний, умений и навыков осуществляется в ходе образовательного процесса по всем изучаемым междисциплинарным курсам в формах, определяемых преподавателем в рабочей программе учебной дисциплины. Он может проводиться в ходе всех видов занятий в форме, избранной преподавателем.

Текущий контроль успеваемости может проводиться в следующих формах:

- устная (устный опрос, защита письменной работы, доклад или реферат по результатам самостоятельной работы и т. д.);
- письменная (письменный опрос, выполнение контрольных и других проверочных работ и т. д.);
- тестовая (устное, письменное, компьютерное тестирование).

Возможны и другие виды текущего контроля успеваемости, которые определяются преподавателями.

ФОРМА КОНТРОЛЯ	тестовая	письменная	устная
ТЕМЫ УЧЕБНОЙ ДИСЦИПЛИНЫ	тема 1-4, 9-12	темы 1-14	темы 1-14

Система текущего контроля успеваемости предусматривает разнообразные по форме и содержанию контрольные мероприятия, учитывающие все виды аудиторной и самостоятельной учебной деятельности обучающегося.

Данные текущего контроля используются преподавателями для анализа (мониторинга) качества освоения обучающимися основной образовательной программы среднего образования, для организации консультаций.

3.1 Тестовая форма контроля

Тестовая форма контроля осуществляется посредством использования электронной образовательной платформы Юрайт – Академия.

Федоров, Д. Ю. Программирование на языке высокого уровня Python : учебное пособие для среднего профессионального образования / Д. Ю. Федоров. — 5-е изд., перераб. и доп. — Москва : Издательство Юрайт, 2023. — 227 с. — (Профессиональное образование). — ISBN 978-5-534-17319-2. — Текст : электронный // Образовательная платформа Юрайт [сайт]. с. 174 — URL: <https://urait.ru/bcode/539652/p.174>.

Тема 1. Введение в программирование

<https://urait.ru/quiz/run-test/BADAC700-F311-40AD-A385-FA2075522733/B044CE82-28E6-450A-932D-3166BD589348>

Тема 2. Язык программирования Python

<https://urait.ru/quiz/run-test/9CE29268-ED4A-4B52-B6DB-F4549FACD2B1/CA6024D4-ECE0-4B1F-929C-AC6DFC1E4DA3>

Тема 3. Циклические операторы языка программирования

<https://urait.ru/quiz/run-test/8CFE1E30-9C59-4911-B732-CB9C700DB112/4924975E-535E-44B9-ADDE-55E1BE013D47>

Тема 4. Основные приемы работы с данными на языке программирования

<https://urait.ru/quiz/run-test/C9C4DB51-B9C2-463D-988C-1FDB576B95E5/8792A9E2-4194-49DD-87DF-87F8354710BE>
<https://urait.ru/quiz/run-test/F544155E-B537-4BE1-B27D-D550F7A9AB03/467E9A98-9BEE-4703-A549-7968106F9C9F>

Тема 9. Основные принципы объектно-ориентированного программирования (ООП)

<https://urait.ru/quiz/run-test/A1C48662-B22F-4B93-B641-2B7ABD63954F/ED760669-5483-4829-B94B-F0C55FE9BAC5>

Тема 10. Интегрированная среда разработчика

<https://urait.ru/quiz/run-test/0168C8B8-1630-4439-A86D-3F700C57C839/25E8E2C1-6A66-43F8-B197-DF888584364E>

Тема 11. Визуальное событийно –управляемое программирование

<https://urait.ru/quiz/run-test/DD5623BC-01C9-404D-91D1->

Тема 12. Разработка оконного приложения

<https://urait.ru/quiz/run-final-test/C7CE975C-8593-4F47-965C-6B34D1C57A67>

3.2 Критерии оценки результатов выполнения тестовых заданий

В качестве критерия освоения материала учебной дисциплины берётся выполнение 50% заданий от их общего числа в тест-задании.

Для интерпретации результатов тестирования используется переводная шкала, с помощью которой результаты теста преобразуют в оценку знаний студентов по пятибалльной системе.

Переводная шкала для интерпретации результатов выполнения тестов

% ошибочных ответов	Интерпретация результатов тестирования по 5-ой системе
до 10%	5
от 15% до 30%	4
от 35% до 50%	3
более 50%	2

3.3 Письменная форма контроля

ТЕМА	НАЗВАНИЕ ПРАКТИЧЕСКИХ И ЛАБОРАТОРНЫХ РАБОТ	САМОСТОЯТЕЛЬНАЯ РАБОТА
1	-	СР№1. Выполнение докладов на темы: Методы анализа, ориентированные на структуры данных Быстрая разработка приложений Жизненный цикл программы. Программный продукт и его характеристики. Обзор языков программирования. Области применения языков программирования. Стандарты языков программирования.
2	ПЗ№1. Математические операторы и функции в Python ПЗ№2 Создание функций в языках программирования ПЗ№3 Условный оператор в Python	СР№2. Создание эссе на темы: История языка программирования Python Разработка алгоритмов и написание кода программ для решения задач с

		использованием ветвления.
3	ПЗ№5 Циклические операторы Python. ПЗ№6 Работа со списками ПЗ№7 Массивы NumPy. Работа с матрицами	-
4	ПЗ№8 Обработка исключений в Python. ПЗ№9 Множества. Кортежи. Словари. ПЗ№10 Работа с файлами. ПЗ№11 Чтение информации из файла. ПЗ№12 Запись информации в файл.	-
5	ПЗ№13 Применение рекурсивных функций. ПЗ№14 Набор функций в Python.	-
6	ПЗ№15 Создание библиотеки подпрограмм.	-
7	ПЗ №16 Модули и пакеты в Python	-
8	ПЗ№17 Использование указателей для организации связанных списков.	-
9	ПЗ №18 Разработка объектно-ориентированного приложения в интегрированной среде разработчика.	Создание отчета по СР№3: «Изучение интегрированной среды разработчика»
10	ПЗ№19 Создание приложения «Будильник» ПЗ№20 Создание приложения «Блокнот»	Отчеты по СР№4, 5: Создание проекта с использованием компонентов для работы с текстом. Создание проекта с использованием компонентов ввода и отображения чисел, дат и времени.
11	ПЗ №21 Создание приложения для просмотра графических файлов. ПЗ №22 Создание процедур на основе событий. ПЗ №23 Создание проекта с использованием кнопочных компонентов. ПЗ №24 Создание проекта с использованием компонентов стандартных диалогов и системы меню.	-
12	ПЗ№25 Разработка функциональной схемы работы приложения. ПЗ№26 Разработка оконного приложения с несколькими формами. ПЗ№27 Разработка игрового приложения.	Выполнение отчета по СР№6: «Создание модели графического интерфейса приложения»
13	ПЗ №28 Разработка интерфейса приложения; ПЗ №29 Тестирование, отладка приложения.	СР№7. Создание доклада на тему «Общая методика отладки программного обеспечения» Создание отчета по СР№8 «Создание простого тестирующего модуля»
14	ПЗ №30 Классы ООП: виды,	СР№9. Создание диаграммы классов

	назначение, свойства, методы, события; ПЗ №31 Объявления класса; ПЗ №32 Создание наследованного класса; ПЗ №33 Программирование приложений; ПЗ №34 Перегрузка методов.	для разработки приложения
--	--	---------------------------

3.4. Критерии оценки результатов выполнения письменных заданий

Оценка «5» (отлично) – краткая, ясная и четкая описание решения задания, верно, на высоком профессиональном уровне, с правильными пояснениями. Обоснованность ответа. Необходимо мотивировать выбранный курс действий, приводящих к разрешению ситуации, и объяснить причины и рациональность его выбора.

Оценка «4» (хорошо) – задание выполнено в целом верно, но с небольшими неточностями, имеются неточности в пояснении.

Оценка «3» (удовлетворительно) – задание выполнено верно, но не доведен до конца, либо в нем имеются ошибки, которые, однако, не приводят к принципиально неверному решению.

Оценка «2» (неудовлетворительно) – задание не выполнено или выполнено не верно

3.5 Устная форма контроля

Примерный список вопросов для устного ответа по учебной дисциплине.

1. Опишите сильные и слабые стороны языка программирования Python. Ответ обоснуйте
2. Расскажите, какие цели преследовал Гвидо в процессе разработки языка Python?
3. Перечислите основные этапы разработки программ.
4. Перечислите основные области применения язык Python. Приведите примеры программ.
5. Опишите схему запуска программы на языке Python.

6. Опишите правила наименования переменных в Python.
7. Дайте развернутое описание модели памяти Python при работе с переменными.
8. Опишите процесс создания функций в Python.
9. Объясните, в чем состоят различия между выполнением команд из файла и в интерактивном режиме.
10. Раскройте смысл понятия области видимости переменных, опишите возможности создания локальных и глобальных переменных в Python, ключевые слова, определяющие область видимости переменных, .
11. Опишите схему передачи чисел в качестве аргументов функции.
12. Перечислите операции над строками, существующие в языке Python, опишите их назначение. Приведите примеры escape-последовательностей.
13. Опишите операторы отношений, существующие в Python. Перечислите правила логических операций над объектами.
14. Приведите примеры случаев, когда применяется условная инструкция if.
15. Приведите пример «оператора моржа».
16. Опишите, как формируются строки документации.
17. Объясните, что такое модуль в Python. Приведите пример.
18. Объясните, что такое пакет в Python. Приведите пример.
19. Опишите процесс создания собственных модулей в Python.
20. Объясните, что такое разработка через тестирование. Приведите пример.
21. Опишите алгоритм генерации случайных чисел в Python.
22. Опишите строковые методы, существующие в Python. Объясните отличия методов от функций.
23. Опишите процесс создания списка. Перечислите основные операции над списками в Python.
24. Приведите примеры преобразования типов в Python (списки,

строки).

25. Охарактеризуйте виды циклов, существующие в Python. Приведите примеры применения циклов for, while.

26. Опишите возможности применения вложенных списков в Python.

27. Объясните, что такое множество. Опишите операции над множествами, существующие в Python.

28. Объясните, что такое кортеж. Опишите операции над кортежами, существующие в Python.

29. Объясните, что такое словарь. Опишите операции над словарями, существующие в Python.

30. Объясните, что такое стек и как работает стековая машина языка Python.

31. Расскажите, как происходит обработка исключений в Python.

32. Объясните, как работает менеджер контекста при обработке файлов.

33. Опишите операции над байтовым типом данных в Python.

34. Объясните, что такое регулярные выражения, приведите примеры их применения.

35. Опишите преимущества использования формата JSON.

36. Опишите особенности объектно-ориентированного программирования в Python. Что такое классы, объекты?

37. Дайте определение наследования в ООП. Приведите примеры его использования на Python.

38. Объясните, что такое итераторы и приведите пример создания итератора.

39. Опишите структуру оконного приложения на примере модуля tkinter.

40. Расскажите, что такое генератор и как он связан с итератором.

41. Объясните, что такое декоратор, и приведите примеры декораторов.

42. Опишите работу шаблона «модель — вид — контроллер» (на примере модуля tkinter).

3.6 Критерии оценки результатов выполнения устных заданий

Оценка «5» (отлично) предполагает грамотное и логичное изложение ответа в устной форме, на практикоориентированные вопросы; обоснование собственного ответа с точки зрения известных теоретических положений.

Оценка «4» (хорошо) – если студент полно освоил учебный материал, владеет научнопонятийным аппаратом, ориентируется в изученном материале, осознанно применяет теоретические знания на практике, грамотно излагает ответ (в устной или письменной форме), но содержание и форма ответа имеют отдельные неточности.

Оценка «3» (удовлетворительно) – если студент обнаруживает знание и понимание основных положений учебного материала, но излагает его неполно, непоследовательно, допускает неточности в определении понятий, в применении теоретических знаний при ответе на практико-ориентированные вопросы;

Оценка «2» (неудовлетворительно) – если студент имеет разрозненные, бессистемные знания, допускает ошибки в определении базовых понятий, искажает их смысл; не может практически применять теоретические знания.

4. ОЦЕНОЧНЫЕ СРЕДСТВА ДЛЯ ПРОВЕДЕНИЯ ПРОМЕЖУТОЧНОЙ АТТЕСТАЦИИ

Промежуточная аттестация по учебной дисциплине состоит из следующих компонентов:

III семестр – семестровый зачет;

IV семестр – экзамен.

4.1 Критерии оценивания знаний, обучающихся на семестровом зачете.

До даты проведения с/зачета необходимо пройти оценивание всех работ, выполненных в течение III семестра: теоретическое обучение, практические занятия, задания для самостоятельного выполнения.

К установленной дате с/зачета обучающийся должен выполнить 100% работ, предусмотренных рабочей программой учебной дисциплины ОП.04 «Основы алгоритмизации и программирования».

4.2 Критерии оценивания знаний, обучающихся на экзамене.

В результате аттестации дисциплине комплексная проверка профессиональных и общих компетенций осуществляется в форме оценки качества выполнения заданий на экзамене:

КОД	ПОКАЗАТЕЛИ ОЦЕНКИ РЕЗУЛЬТАТА	ФОРМА ПРОВЕРКИ
ПК 1.1	-правильность разработки блок-схемы алгоритма программы -правильность внутренней логики программы -правильность определения сложности алгоритма	задание 1
ПК 1.2	- правильность использования синтаксиса языка программирования - правильность оформления кода программного продукта - правильность создания эффективного кода	задание 1
ПК 1.3	- правильность выполнения настройки процесса отладки в инструментальном средстве программирования -правильность выполнения этапов отладки - правильность создания пошагового выполнения программы - правильность мониторинга значений переменных	задание 1
ПК 1.4	- правильность создания тестового сценария - правильность определения количества тестов -правильность выбора методов тестирования	задание 1, 2

	- правильность выполнения этапов тестирования	
ПК 1.5	-правильность анализа программного кода с целью рефакторинга -правильность выполнения этапов рефакторинга - правильность определения эффективности программного кода - правильность изменения программы с целью увеличения эффективности	задание 1, 2
ПК 2.4	- правильность создания тестового сценария - правильность определения количества тестов -правильность выбора методов тестирования - правильность выполнения этапов тестирования	задание 2
ПК 2.5	-правильность анализа программы на соответствие стандартам качества -правильность определения степени документированности кода	задание 2

Пример типового экзаменационного задания¹:

Задание 1.

Постановка задачи: Оценки каждого из 12 учеников по трем предметам представлены в виде таблицы:

Ученик	Предмет		
	1	2	3
1	4	4	5
2	3	4	3
...			
12	5	4	4

Необходимо ввести в программу каждую из оценок и найти их сумму.

Задачу решить в двух вариантах:

- 1) ввод оценок осуществляется по строкам;
- 2) ввод оценок осуществляется по столбцам

Необходимо представить на проверку:

Блок-схему программы

Текст программы должен содержать комментарии.

Задание 2

¹ Использованы формулировки заданий Златопольский, Д. М. Основы программирования на языке Python / Д. М. Златопольский. — 2-ое изд., испр. и доп. — Москва : ДМК Пресс, 2018. — 396 с. — ISBN 978-5-97060-641-4. — Текст : электронный // Лань : электронно-библиотечная система. — URL: <https://e.lanbook.com/book/131683> (дата обращения: 11.03.2024). — Режим доступа: для авториз. пользователей.

Для разработанной программы создать:
сценарий и набор для тестирования;
провести анализ эффективности разработанного кода.

Варианты постановок задач для разработки программ:

1. Разработать программу для игры. «Описание игры. На экране появляется вопрос: «Чет (введите 2) или нечет (введите 1)?» Играющий должен ответить, какое число – четное или нечетное – выберет компьютер, и ввести соответственно 2 или 1. После этого компьютер случайным образом генерирует одно из них. Результат сравнения ответа играющего с числом компьютера выводится на экран.»

2. Разработать игру, моделирующую бросание игрового кубика каждым из двух участников, после чего определяется у кого, выпало больше очков.

3. Разработать игру «Описание игры. Компьютер генерирует случайное целое число, большее 0 и меньше 101. Играющий пытается отгадать это число, делая несколько попыток. В случае несовпадения «задуманного» компьютером числа и числа-ответа на экран выводится сообщение о том, какое из них больше, после чего играющий вновь вводит число, и т. д. до отгадывания»

4. Разработать игру «Игра моделирует выбор каждым из двух играющих «наугад» по одной карте из полного набора игровых карт, включающего четыре масти («пики», «трефы», «бубны» и «червы») и по 9 достоинств карт каждой масти («шестерка», «семерка», «восьмерка», «девятка», «десятка», «валет», «дама», «король», «туз»), и определение того из участников игры, у которого выбранная карта «старше». При этом условимся, что приведенный выше перечень мастей и карт одной масти дан в порядке увеличения их «старшинства» (например, любая карта масти «бубны» старше любой карты масти «пики», а «валет червей» старше «десятки червей»)»

5. Разработать игру «Описание игры. На столе выложены n

предметов (спичек, монет, ка-мешков и т. п.). Играют двое. Они поочередно забирают несколько предметов, причем заранее договорено, что число забранных предметов не превышает k ($1 < k \leq n$). Проигрывает тот, кто своим ходом вынужден забрать последний предмет4.»

Экзаменационные задания оцениваются по пятибалльной шкале следующим образом:

Оценка «Отлично» ставится, если студент строит ответ логично в соответствии с планом, показывает максимально глубокие знания профессиональных терминов, понятий, категорий, концепций и теорий. Устанавливает содержательные межпредметные связи. Развернуто аргументирует выдвигаемые положения, приводит убедительные примеры. На этом уровне освоения вида деятельности, творческое действие – самостоятельное конструирование способа деятельности, поиск новой информации. Формулирование оценочных суждений на основе имеющихся фактов и заданных критериев.

Оценка «Хорошо» ставится, если студент строит свой ответ в соответствии с планом. В ответе представлены различные подходы к проблеме, но их обоснование недостаточно полно. Устанавливает содержательные межпредметные связи. Развернуто аргументирует выдвигаемые положения, приводит необходимые примеры, однако показывает некоторую непоследовательность анализа. Выводы правильны. Речь грамотна, используется профессиональная лексика. На этом уровне освоения вида деятельности, воспроизведение, репродуктивное действие – самостоятельное воспроизведение и применение информации для выполнения данного действия. Студент способен по памяти воспроизводить ранее усвоенную информацию и применять усвоенные алгоритмы деятельности для решения типовых задач.

Оценка «Удовлетворительно» ставится, если ответ недостаточно логически выстроен, план ответа соблюдается непоследовательно. Студент обнаруживает слабость в развернутом раскрытии профессиональных

понятий. Выдвигаемые положения декларируются, но недостаточно аргументированы. Ответ носит преимущественно теоретический. На этом уровне освоения вида деятельности, применение, продуктивное действие – поиск и использование информации для самостоятельного выполнения нового действия (знания, умения, навыки). Этот уровень предполагает комбинирование студентом известных алгоритмов и приемов деятельности, применения навыков эвристического мышления

Оценка «Неудовлетворительно» ставится при условии недостаточного раскрытия профессиональных понятий, категорий, концепций, теорий. Студент проявляет стремление подменить научное обоснование проблем рассуждениями обыденноповседневного бытового характера. Ответ содержит ряд серьезных неточностей. Выводы поверхностны. Репродуктивная деятельность (узнавание объектов, свойств, процессов при повторном восприятии информации о них или действий с ними). На этом уровне студент не способен самостоятельно, без помощи извне, воспроизводить и применять полученную информацию.