

Документ подписан простой электронной подписью
Информация о владельце:
ФИО: Федорова Марина Владимировна
Должность: Директор филиала
Дата подписания: 06.08.2026 09:36:39
Уникальный программный ключ:
e766def0e2eb455f02135d659e45051ac23041da

Приложение
к ОПОП по специальности
09.02.11 Разработка и управление
программным обеспечением

ФОНД
ОЦЕНОЧНЫХ СРЕДСТВ УЧЕБНОЙ ПРАКТИКЕ
УП.02.01 УЧЕБНАЯ ПРАКТИКА
ПМ.02 РАЗРАБОТКА И ИНТЕГРАЦИЯ МОДУЛЕЙ ПРОГРАММНОГО
ОБЕСПЕЧЕНИЯ
для специальности
09.02.11 РАЗРАБОТКА И УПРАВЛЕНИЕ ПРОГРАММНЫМ
ОБЕСПЕЧЕНИЕМ
Базовая подготовка
среднего профессионального образования
(год начала подготовки 2026)

СОДЕРЖАНИЕ

1. Паспорт фонда оценочных средств.....	3
2. Результаты освоения программы учебной практики, подлежащие проверке	4
3. Типовые задания для оценки освоения программы учебной практики	13
4. Контрольно-оценочные материалы для итоговой аттестации	37

1. ПАСПОРТ ФОНДА ОЦЕНОЧНЫХ СРЕДСТВ

Фонд оценочных средств предназначен для контроля и оценки образовательных достижений обучающихся, освоивших программу учебной практики профессионального модуля ПМ.02 Разработка и интеграция модулей программного обеспечения.

Фонд оценочных средств включают контрольные материалы для проведения текущего контроля и промежуточной аттестации в форме дифференцированного зачёта.

Фонд оценочных средств разработан на основании положений:

- программы подготовки специалистов среднего звена по специальности СПО 09.02.11 Разработка и управление программным обеспечением;
- программы учебного модуля;
- учебного плана по специальности СПО 09.02.11 Разработка и управление программным обеспечением.

2. РЕЗУЛЬТАТЫ ОСВОЕНИЯ ПРОГРАММЫ УЧЕБНОЙ ПРАКТИКИ, ПОДЛЕЖАЩИЕ ПРОВЕРКЕ

2.1 Перечень умений, знаний, общих компетенций

С целью овладения указанным видом профессиональной деятельности и соответствующими профессиональными компетенциями обучающийся в ходе освоения рабочей программы учебной практики должен:

Перечень общих компетенций:

ОК 1	Выбирать способы решения задач профессиональной деятельности применительно к различным контекстам
ОК 2	Использовать современные средства поиска, анализа и интерпретации информации, и информационные технологии для выполнения задач профессиональной деятельности
ОК 3	Планировать и реализовывать собственное профессиональное и личностное развитие, предпринимательскую деятельность в профессиональной сфере, использовать знания по правовой и финансовой грамотности в различных жизненных ситуациях
ОК 4	Эффективно взаимодействовать и работать в коллективе и команде
ОК 5	Осуществлять устную и письменную коммуникацию на государственном языке Российской Федерации с учетом особенностей социального и культурного контекста
ОК 6	Проявлять гражданско-патриотическую позицию, демонстрировать осознанное поведение на основе традиционных российских духовно-нравственных ценностей, в том числе с учетом гармонизации межнациональных и межрелигиозных отношений, применять стандарты антикоррупционного поведения
ОК 7	Содействовать сохранению окружающей среды, ресурсосбережению, применять знания об изменении климата, принципы бережливого производства, эффективно действовать в чрезвычайных ситуациях
ОК 8	Использовать средства физической культуры для сохранения и укрепления здоровья в процессе профессиональной деятельности и поддержания необходимого уровня физической подготовленности
ОК 9	Пользоваться профессиональной документацией на государственном и иностранном языках

Перечень профессиональных компетенций

ПК 2.1	Проектировать модули программного обеспечения
ПК 2.2	Разрабатывать модули программного обеспечения
ПК 2.3	Выполнять интеграцию модулей и компонентов программного обеспечения
ПК 2.4	Выполнять тестирование и отладку программного обеспечения
ПК 2.5	Осуществлять документирование программных модулей программного обеспечения

В результате освоения профессионального модуля студент должен:

Код ОК, ПК	Уметь	Знать	Владеть навыками
ОК.01Выбирать способы решения задач профессиональной деятельности применительно к различным контекстам	Выбирать способы решения задач профессиональной деятельности применительно к различным контекстам	актуальный профессиональный и социальный контекст, в котором приходится работать и жить; основные источники информации и ресурсы для решения задач и проблем в профессиональном и/или социальном контексте; алгоритмы выполнения работ в профессиональной и смежных областях; методы работы в профессиональной и смежных сферах; структуру плана для решения задач; порядок оценки результатов решения задач профессиональной деятельности	-
ОК.02 Использовать современные средства поиска, анализа и интерпретации информации, и информационные технологии для выполнения задач профессиональной деятельности	Использовать современные средства поиска, анализа и интерпретации информации, и информационные технологии для выполнения задач профессиональной деятельности	номенклатура информационных источников, применяемых в профессиональной деятельности; приемы структурирования информации; формат оформления результатов поиска информации, современные средства и устройства информатизации; порядок их применения и программное обеспечение в профессиональной деятельности в том числе с использованием цифровых средств	-
ОК.03 Планировать и	Планировать и	содержание	-

реализовывать собственное профессиональное и личностное развитие, предпринимательскую деятельность в профессиональной сфере, использовать знания по правовой и финансовой грамотности в различных жизненных ситуациях	реализовывать собственное профессиональное и личностное развитие, предпринимательскую деятельность в профессиональной сфере, использовать знания по правовой и финансовой грамотности в различных жизненных ситуациях	актуальной нормативно-правовой документации; современная научная и профессиональная терминология; возможные траектории профессионального развития и самообразования; основы предпринимательской деятельности; основы финансовой грамотности; правила разработки бизнес-планов; порядок выстраивания презентации; кредитные банковские продукты	
ОК.04 Эффективно взаимодействовать и работать в коллективе и команде	Эффективно взаимодействовать и работать в коллективе и команде	психологические основы деятельности коллектива, психологические особенности личности; основы проектной деятельности	-
ОК.05 Осуществлять устную и письменную коммуникацию на государственном языке Российской Федерации с учетом особенностей социального и культурного контекста	Осуществлять устную и письменную коммуникацию на государственном языке Российской Федерации с учетом особенностей социального и культурного контекста	особенности социального и культурного контекста; правила оформления документов и построения устных сообщений	-
ОК.06 Проявлять гражданско-патриотическую позицию, демонстрировать осознанное поведение на основе традиционных российских духовно-нравственных ценностей, в том числе с учетом гармонизации межнациональных и межрелигиозных	Проявлять гражданско-патриотическую позицию, демонстрировать осознанное поведение на основе традиционных российских духовно-нравственных ценностей, в том числе с учетом гармонизации межнациональных и	сущность гражданско-патриотической позиции, общечеловеческих ценностей; значимость профессиональной деятельности по специальности; стандарты антикоррупционного поведения и последствия его нарушения	-

отношений, применять стандарты антикоррупционного поведения	межрелигиозных отношений, применять стандарты антикоррупционного поведения		
ОК.07 Содействовать сохранению окружающей среды, ресурсосбережению, применять знания об изменении климата, принципы бережливого производства, эффективно действовать в чрезвычайных ситуациях	Содействовать сохранению окружающей среды, ресурсосбережению, применять знания об изменении климата, принципы бережливого производства, эффективно действовать в чрезвычайных ситуациях	правила экологической безопасности при ведении профессиональной деятельности; основные ресурсы, задействованные в профессиональной деятельности; пути обеспечения ресурсосбережения; принципы бережливого производства; основные направления изменения климатических условий региона	-
ОК.08 Использовать средства физической культуры для сохранения и укрепления здоровья в процессе профессиональной деятельности и поддержания необходимого уровня физической подготовленности	Использовать средства физической культуры для сохранения и укрепления здоровья в процессе профессиональной деятельности и поддержания необходимого уровня физической подготовленности	роль физической культуры в общекультурном, профессиональном и социальном развитии человека; основы здорового образа жизни; условия профессиональной деятельности и зоны риска физического здоровья для специальности; средства профилактики перенапряжения	-
ОК.09 Пользоваться профессиональной документацией на государственном и иностранном языках	Пользоваться профессиональной документацией на государственном и иностранном языках	правила построения простых и сложных предложений на профессиональные темы; основные общеупотребительные глаголы (бытовая и профессиональная лексика); лексический минимум, относящийся к описанию предметов, средств и процессов	-

		профессиональной деятельности; особенности произношения; правила чтения текстов профессиональной направленности	
ПК 2.1 Проектировать модули программного обеспечения	– проектировать модули, соответствующие бизнес-задачам; – создавать архитектурные диаграммы и документацию; – определять структуру и интерфейсы модулей; – анализировать требования к модулю и определять его функциональность; – проектировать архитектуру модуля, включая выбор подходящих паттернов проектирования и структуры данных; – создавать диаграммы классов, последовательностей и прочих диаграмм для визуализации проектируемого модуля; – выбирать подходящие языки программирования и технологии для реализации модуля; – проектировать интерфейсы программного обеспечения для взаимодействия с другими модулями и системами; – учитывать требования к масштабируемости,	– основные принципы проектирования модулей программного обеспечения; – языки программирования и технологии для реализации модулей; – паттерны проектирования и структуры данных для создания эффективных и масштабируемых модулей; – методы анализа требований и способов определения функциональности модуля; – принципы создания интерфейсов для взаимодействия с другими модулями и системами; – принципы обеспечения безопасности, производительности и масштабируемости при проектировании модулей; – методы анализа и оптимизации проектируемых модулей для повышения их эффективности и качества.	– проектирования модулей ПО с учетом требований заказчика; – создания архитектурных диаграмм и спецификаций модулей; – определения интерфейсов и взаимодействия модулей в системе.

	производительности и безопасности при проектировании модуля; проводить анализ и оптимизацию проектируемого модуля для повышения его эффективности и качества		
ПК 2.2 Разрабатывать модули программного обеспечения	– разрабатывать модули программного обеспечения с использованием различных языков программирования и технологий; – применять паттерны проектирования и структуры данных для создания эффективных и масштабируемых модулей; – анализировать требования и определять функциональность модуля; – создавать интерфейсы для взаимодействия с другими модулями и системами; – обеспечивать безопасность, производительность и масштабируемость при разработке модулей; – оптимизировать проектируемые модули для повышения их эффективности и качества; – работать с системой контроля версий; – улучшать	– язык программирования, основные конструкции, синтаксис; – паттерны проектирования; – структуры данных; – принципы создания интерфейсов для взаимодействия с другими модулями и системами, таких как REST API, SOAP; – работу с инструментальным программным обеспечением; – методы оптимизации кода и алгоритмов; – эффективные алгоритмы и структуры данных для повышения производительности; – многопоточность в программных модулях; – методы оптимизации сетевых протоколов для ускорения обмена данными; – кэширование данных; – управление памятью; – техники повышения производительности программного обеспечения	– создания модулей программного обеспечения на различных языках программирования; – отладки и тестирования разработанных модулей; – применения структурного и объектно-ориентированного программирования; – оптимизации кода и алгоритмов программных модулей для увеличения производительности ; - мониторинга и анализа производительности приложений.

	производительность модулей, выявляя и устраняя узкие места; – проводить анализ и мониторинг производительности приложений; применять инструменты для рефакторинга и оптимизации программного кода.		
ПК 2.3 Выполнять интеграцию модулей и компонентов программного обеспечения	– интегрировать модули и компоненты, обеспечивая их взаимодействие; – работать с API и устанавливать соединения между компонентами; – отслеживать и устранять конфликты и ошибки интеграции; – анализировать и определять зависимости между модулями и компонентами; работать с различными форматами данных и протоколами передачи данных	– общие принципы функционирования аппаратных, программных и программно-аппаратных средств администрируемой информационно-коммуникационной системы; – международные стандарты локальных вычислительных сетей; – методы и подходы к интеграции модулей и компонентов; – принципы версионирования и управления изменениями при интеграции; – принципы безопасности при интеграции модулей и компонентов	– интеграции программных модулей и компонентов в единое программное решение; – работы с API и веб-сервисами для взаимодействия между модулями; – работы с интеграционными платформами и инструментами; обеспечения совместимости и стабильности системы
ПК 2.4 Выполнять тестирование и отладку программного обеспечения	– анализировать требования к программному обеспечению и составлять планы тестирования; – создавать тестовые сценарии и тест-кейсы для проверки функциональности и соответствия требованиям; – выполнять тестирование	– принципы и методы тестирования программного обеспечения; – основы программирования и архитектуры программного обеспечения; – основы баз данных и SQL-запросов; – инструменты для автоматизации тестирования;	– отладки программного обеспечения на уровне программных модулей; – тестирования программного обеспечения; – формирования тестовых сценариев; – подготовки тестовых платформ (установка

	программного обеспечения вручную и автоматизировать процесс тестирования; – анализировать результаты тестирования и документировать найденные ошибки; – разрабатывать стратегии отладки и исправлять ошибки в программном обеспечении; – выполнять модульные тесты с использованием инструментов тестирования, в том числе автоматизированного тестирования; – использовать системы контроля дефектов ПО; составлять отчет о выполнении тестирования ПО	– основы разработки и отладки программного обеспечения на разных языках программирования; – понятие дефекта программного обеспечения; – критерии качества ПО; – виды и типы тестирования ПО; – техники ручного тестирования; – техники автоматизированного тестирования; – жизненный цикл дефекта ПО; – принципы работы в системе контроля дефектов; основные понятия о качестве ПО	операционной системы, дополнительного ПО и другого по необходимости); – оценки объема тестирования ПО с целью определения необходимых ресурсов для его выполнения; – настройки тестовой среды и аппаратных средств для выполнения тестирования ПО в соответствии с заданием на тестирование в пределах своей компетенции; – формирования и представления отчетности о подготовке к выполнению задания на тестирование ПО в соответствии с установленными регламентами; выполнения тестовых процедур на тестовых данных
ПК 2.5 Осуществлять документирование программных модулей программного обеспечения	– описывать функциональность модулей в документации; – создавать диаграммы для иллюстрации работы модулей; – программировать с использованием комментариев для документирования кода; – использовать специальные метки/теги для отметки важных частей кода в	– стандарты технической документации; – принципы документирования программного обеспечения; инструменты для создания технической документации и комментирования кода	– создания технической документации для модулей; – документирование кода, API и интерфейсов; работы со специализированным ПО по документированию программного кода

	документации; – вести журнал изменений и фиксировать обновления программных модулей; – разбивать модули на логические блоки и описывать каждый блок отдельно; – включать в документацию особенности модулей, такие как ограничения, уязвимости или оптимальные настройки; проводить регулярное обновление документации при изменении модулей или добавлении нового функционала.		
--	---	--	--

3. ТИПОВЫЕ ЗАДАНИЯ ДЛЯ ОЦЕНКИ ОСВОЕНИЯ ПРОГРАММЫ УЧЕБНОЙ ПРАКТИКИ

Работа содержит задания по разработке программного обеспечения с использованием инструментальных средств. Все документы должны быть выполнены максимально точно по представленному образцу.

Результаты выполнения задания оформляются в виде отдельных файлов соответствующих форматов и сохраняются на ПК. Для проверки и оценки результаты выполнения экзаменационного задания предоставляются в электронном виде.

В процессе выполнения задания вы можете воспользоваться методическими пособиями, предоставленной учебной литературой и информацией сети Интернет.

Тестирование

МДК.02.01 Разработка программных модулей

Раздел 1. Разработка программных модулей

1. Какой принцип ООП позволяет скрыть внутреннюю реализацию объекта и предоставить доступ к его данным только через публичные методы?

- A) Наследование
- B) Инкапсуляция
- C) Полиморфизм
- D) Абстракция

2. Какой принцип ООП подразумевает, что дочерний класс автоматически получает поля и методы родительского класса?

- A) Наследование
- B) Композиция
- C) Агрегация
- D) Полиморфизм

3. Что означает принцип полиморфизма в контексте разработки модулей?

- A) Возможность создания объектов без конструктора
- B) Способность объектов разных классов обрабатывать сообщения с одинаковым именем, реализуя его по-своему
- C) Запрет на изменение методов в дочерних классах
- D) Автоматическое приведение типов данных при компиляции

4. Как формулируется принцип единственной ответственности (SRP) из SOLID?

- A) Класс должен содержать как можно больше методов
- B) Класс должен иметь только одну причину для изменения, то есть выполнять одну конкретную задачу
- C) Все классы должны наследоваться от одного базового
- D) Модуль должен быть доступен для модификации из любого места системы

5. Что означает принцип «Открытости/Закрытости» (Open/Closed Principle)?

- A) Код должен быть полностью открыт для правок всеми разработчиками
- B) Программные сущности должны быть открыты для расширения, но закрыты для модификации
- C) Все классы должны иметь публичные конструкторы
- D) Закрытые поля должны всегда быть статическими

6. Какой механизм в ООП обеспечивает реализацию поведения, заданного интерфейсом, без жёсткого наследования от конкретного класса?

- A) Переопределение методов
- B) Реализация интерфейса
- C) Перегрузка операторов
- D) Статическая типизация

7. В чём ключевое отличие композиции от наследования при проектировании классов?

- A) Композиция требует общего базового класса
 - B) Композиция реализует отношение «часть-целое» и обеспечивает более слабую связанность, чем наследование**
 - C) Наследование всегда предпочтительнее для переиспользования кода
 - D) Композиция запрещает использование интерфейсов
- 8. Для чего в ООП используются абстрактные классы?**
- A) Для создания готовых к использованию объектов с полной реализацией
 - B) Для определения общего контракта и базовой логики для группы родственных классов, запрещая прямое создание экземпляров**
 - C) Для замены механизмов исключений в языке
 - D) Для ускорения компиляции за счёт удаления неиспользуемых методов
- 9. Какая структура данных работает по принципу FIFO (First In, First Out)?**
- A) Стек
 - B) Очередь**
 - C) Дек
 - D) Хеш-таблица
- 10. Какой алгоритм сортировки имеет среднюю временную сложность $O(n \log n)$ и относится к алгоритмам «разделяй и властвуй»?**
- A) Сортировка пузырьком
 - B) Быстрая сортировка (QuickSort)**
 - C) Сортировка вставками
 - D) Сортировка выбором
- 11. Какая структура данных обеспечивает поиск, вставку и удаление элементов в среднем за $O(1)$ при использовании качественной хеш-функции?**
- A) Бинарное дерево поиска
 - B) Хеш-таблица**
 - C) Связный список
 - D) Очередь с приоритетом
- 12. Какой алгоритм поиска применяется исключительно к отсортированному массиву и имеет сложность $O(\log n)$?**
- A) Линейный поиск
 - B) Бинарный (двоичный) поиск**
 - C) Поиск в глубину (DFS)
 - D) Интерполяционный поиск
- 13. Для решения каких задач наиболее эффективно применяется структура данных «Дерево поиска» (BST)?**
- A) Для хранения неупорядоченных наборов ключей
 - B) Для организации иерархических данных, быстрого поиска, вставки и получения отсортированной последовательности**
 - C) Для реализации циклических зависимостей
 - D) Для хранения двоичных файлов в оперативной памяти
- 14. Какая временная сложность характерна для обхода графа в ширину (BFS)?**
- A) $O(V^2)$
 - B) $O(E \log V)$
 - C) $O(V + E)$, где V — число вершин, E — число рёбер**
 - D) $O(n!)$
- 15. Какой алгоритм используется для нахождения кратчайшего пути во взвешенном графе с неотрицательными весами рёбер?**
- A) Алгоритм Флойда-Уоршелла
 - B) Алгоритм Дейкстры**
 - C) Алгоритм Краскала
 - D) Алгоритм Беллмана-Форда

16. Какая структура данных идеально подходит для реализации механизма «Отмена последнего действия» (Undo) в программном модуле?

- A) Очередь
- B) Стек (LIFO)
- C) Массив
- D) Хеш-множество

17. Что означает термин «связанность» (coupling) при проектировании программных модулей?

- A) Количество строк кода внутри модуля
- B) Степень зависимости одного модуля от внутреннего устройства другого модуля
- C) Скорость обмена данными между процессами
- D) Число публичных методов в классе

18. Какой принцип проектирования предполагает, что модуль должен быть максимально самодостаточным и решать одну узкую задачу?

- A) Низкая связанность
- B) Высокая связность (High Cohesion) внутри модуля
- C) Полиморфная агрегация
- D) Горизонтальное масштабирование

19. Что такое публичный API программного модуля?

- A) Внутренний набор утилит для отладки
- B) Формализованный интерфейс, определяющий способы безопасного взаимодействия с модулем извне
- C) Документация по установке зависимостей
- D) Список приватных полей класса

20. Какой тип диаграммы UML используется для отображения статической структуры системы, включая классы, атрибуты, методы и связи между ними?

- A) Диаграмма последовательностей (Sequence Diagram)
- B) Диаграмма классов (Class Diagram)
- C) Диаграмма состояний (State Diagram)
- D) Диаграмма развёртывания (Deployment Diagram)

21. Какой подход к проектированию предполагает разбиение сложной системы на независимые, слабо связанные компоненты с чёткими интерфейсами?

- A) Монолитная архитектура
- B) Модульная декомпозиция
- C) Скриптовая компоновка
- D) Процедурная иерархия

22. Что такое «контракт модуля» (Design by Contract)?

- A) Юридическое соглашение между заказчиком и разработчиком
- B) Набор предусловий, постусловий и инвариантов, строго определяющих корректность входных/выходных данных и состояния модуля
- C) Файл package.json или pom.xml с версиями зависимостей
- D) Протокол сетевого взаимодействия REST

23. Какой документ описывает входные параметры, выходные результаты, алгоритм работы, ограничения и зависимости программного модуля?

- A) Протокол тестирования
- B) Спецификация (техническое задание) модуля
- C) Лог выполнения транзакций
- D) Диаграмма потоков данных (DFD)

24. Какой паттерн проектирования чаще всего используется для чёткого разделения логики приложения, пользовательского интерфейса и данных?

- A) Singleton
- B) MVC (Model-View-Controller)

C) Factory Method

D) Proxy

25. Что такое валидация пользовательского ввода на уровне клиентского модуля?

A) Шифрование паролей перед сохранением в БД

B) Проверка формата, типа и допустимых диапазонов введённых данных до их передачи на сервер

C) Аутентификация пользователя по токену

D) Логирование действий администратора

26. Какой механизм позволяет обновлять часть пользовательского интерфейса без полной перезагрузки страницы в веб-приложениях?

A) Server-Side Rendering (SSR)

B) Асинхронные запросы (AJAX/Fetch API)

C) Кеширование на уровне CDN

D) WebSocket для двусторонней связи

27. Какой принцип юзабилити требует, чтобы UI-модуль немедленно информировал пользователя о результате его действий (загрузка, успех, ошибка)?

A) Гибкость использования

B) Видимость состояния системы и обратная связь

C) Эстетический минимализм

D) Предотвращение ошибок на уровне БД

28. Какую роль выполняет паттерн «Наблюдатель» (Observer) в модулях пользовательского интерфейса?

A) Шифрует данные формы перед отправкой

B) Подписывает интерфейсные элементы на изменения модели, автоматически обновляя отображение при смене данных

C) Маршрутизирует HTTP-запросы к разным контроллерам

D) Валидирует JSON-ответы от API

29. Какой формат обмена данными является стандартом де-факто для взаимодействия современных UI-модулей с серверной частью?

A) XML

B) JSON

C) CSV

D) YAML

30. Какой подход гарантирует доступность UI-модуля для пользователей с ограниченными возможностями (скринридеры, навигация клавиатурой)?

A) Использование только графических иконок без текста

B) Соблюдение стандартов WCAG, семантическая вёрстка и поддержка ARIA-атрибутов

C) Отключение анимаций и переходов

D) Жёсткая фиксация размеров шрифтов в пикселях

Тестирование

МДК.02.02 Осуществление интеграции программных модулей

Раздел 2. Осуществление интеграции программных модулей

1. Что является основной целью интеграции программных модулей?

A) Увеличение объёма исходного кода проекта

B) Обеспечение совместной работы независимых компонентов для выполнения общей бизнес-задачи

C) Полная замена устаревших систем новыми

D) Упрощение процесса компиляции приложений

2. Какой стиль архитектурной интеграции предполагает обмен сообщениями через единую центральную шину?

A) Точка-точка (Point-to-Point)

B) Корпоративная сервисная шина (ESB — Enterprise Service Bus)

C) Монолитная компоновка

D) Прямой доступ к базе данных

3. Какой протокол чаще всего используется для интеграции веб-сервисов на основе принципов REST?

A) FTP

B) HTTP/HTTPS

C) SMTP

D) SNMP

4. Что такое API (Application Programming Interface) в контексте интеграции?

A) Графический интерфейс для конечного пользователя

B) Набор правил и спецификаций, позволяющий одному приложению взаимодействовать с другим

C) Система управления версиями исходного кода

D) Инструмент для автоматического тестирования модулей

5. Какой формат данных является наиболее распространённым для обмена информацией в современных REST API?

A) XML

B) JSON

C) CSV

D) Protocol Buffers

6. Что означает термин «синхронная интеграция»?

A) Вызывающая система ожидает немедленного ответа от целевого модуля перед продолжением работы

B) Обмен данными происходит в фоновом режиме без ожидания

C) Интеграция выполняется только в ночное время

D) Данные передаются пакетами раз в сутки

7. Какой паттерн интеграции используется для преобразования формата сообщения одной системы в формат, понятный другой системе?

A) Маршрутизатор (Router)

B) Трансформатор сообщений (Message Translator)

C) Повторитель (Idempotent Receiver)

D) Агрегатор (Aggregator)

8. Что такое «контракт сервиса» (Service Contract) при интеграции модулей?

A) Юридический договор между заказчиком и разработчиком

B) Формальное описание интерфейса: методы, параметры, форматы запросов/ответов и коды ошибок

C) Журнал аудита всех вызовов сервиса

D) Политика резервного копирования данных сервиса

9. Какой инструмент позволяет собирать, хранить и визуализировать метрики производительности распределённой системы в реальном времени?

A) Git

B) Prometheus + Grafana

C) Docker Compose

D) Postman

10. Что такое «распределённая трассировка» (Distributed Tracing) и для чего она применяется?

A) Для шифрования трафика между микросервисами

B) Для отслеживания пути одного запроса через несколько сервисов с целью диагностики задержек и сбоев

C) Для автоматического масштабирования контейнеров

D) Для резервного копирования конфигураций

- 11. Какой тип логов наиболее важен для оперативного выявления причин падения интегрированной системы?**
- A) Логи ошибок и исключений (Error/Exception logs) с указанием стека вызовов
 - B) Логи доступа к статическим файлам
 - C) Логи изменения цветов в пользовательском интерфейсе
 - D) Логи компиляции исходного кода
- 12. Что означает метрика «Availability» (доступность) в соглашении об уровне обслуживания (SLA)?**
- A) Среднее время обработки одного запроса
 - B) Процент времени, в течение которого система работоспособна и готова обрабатывать запросы
 - C) Максимальное число одновременных пользователей
 - D) Объём данных, переданных за сутки
- 13. Какой механизм позволяет автоматически перезапускать упавшие сервисы в контейнеризированной среде?**
- A) Сгруппированные задачи
 - B) Оркестратор (например, Kubernetes с политиками restartPolicy)
 - C) Статический балансировщик нагрузки
 - D) Система контроля версий
- 14. Для чего используется «Health Check» (проверка работоспособности) эндпоинт в микросервисе?**
- A) Для предоставления информации о текущем состоянии сервиса оркестратору или балансировщику
 - B) Для аутентификации администраторов системы
 - C) Для шифрования служебного трафика
 - D) Для агрегации бизнес-отчётов
- 15. Что такое «алертинг» (alerting) в системе мониторинга?**
- A) Процесс архивации старых логов
 - B) Автоматическая отправка уведомлений ответственным лицам при превышении пороговых значений метрик
 - C) Ежедневное формирование отчёта о загрузке процессора
 - D) Ручная проверка конфигурационных файлов
- 16. Какой стандарт используется для безопасной передачи токенов аутентификации между сервисами в распределённых системах?**
- A) SOAP
 - B) JWT (JSON Web Token)
 - C) CSV
 - D) SNMP
- 17. Что означает принцип «наименьших привилегий» при настройке доступа между интегрированными модулями?**
- A) Все сервисы должны иметь права администратора для бесперебойной работы
 - B) Каждый компонент получает только те права, которые минимально необходимы для выполнения его функций
 - C) Доступ к базам данных должен быть открыт для всех внутренних сервисов
 - D) Пароли должны храниться в конфигурационных файлах в открытом виде
- 18. Какой механизм защищает данные при передаче между сервисами от перехвата и подмены?**
- A) Сжатие трафика алгоритмом GZIP
 - B) Использование TLS/SSL-шифрования канала связи
 - C) Логирование всех запросов в текстовый файл
 - D) Разделение трафика по разным сетевым интерфейсам
- 19. Что такое «API Gateway» и какую роль он играет в безопасности интеграции?**

- A) Физический сервер для хранения резервных копий
 - B) Единая точка входа, которая обеспечивает аутентификацию, авторизацию, ограничение частоты запросов и защиту от атак**
 - C) Инструмент для автоматического рефакторинга кода
 - D) База данных для хранения пользовательских сессий
- 20. Какой тип атаки предполагает внедрение вредоносных скриптов через параметры интеграционного запроса?**
- A) DDoS-атака
 - B) Инъекция (SQL/NoSQL/Command Injection)**
 - C) Фишинг
 - D) Ман-ин-те-миддл (MitM)
- 21. Для чего используется механизм «Rate Limiting» (ограничение частоты запросов) в интеграционных решениях?**
- A) Для защиты сервисов от перегрузки и злоупотреблений путём ограничения числа запросов от одного источника**
 - B) Для ускорения обработки легитимных запросов
 - C) Для автоматического масштабирования базы данных
 - D) Для шифрования конфиденциальных полей в запросе
- 22. Что такое «секрет» (secret) в контексте безопасной конфигурации интеграции и как его следует хранить?**
- A) Пароль от учётной записи разработчика в тексте кода
 - B) Конфиденциальные данные (ключи, токены), которые должны храниться в специализированных хранилищах (Vault, Secrets Manager), а не в репозитории**
 - C) Скрытый параметр в строке подключения к БД
 - D) Зашифрованный комментарий в исходном коде
- 23. Какой протокол авторизации позволяет пользователю предоставить стороннему приложению доступ к ресурсам без передачи логина и пароля?**
- A) Basic Auth
 - B) OAuth 2.0**
 - C) FTP
 - D) LDAP
- 24. Что означает «горизонтальное масштабирование» (scaling out) интегрированной системы?**
- A) Увеличение мощности одного сервера (CPU, RAM)
 - B) Добавление новых экземпляров сервисов или узлов для распределения нагрузки**
 - C) Объединение нескольких баз данных в одну монолитную
 - D) Увеличение размера дискового пространства на существующем сервере
- 25. Какой механизм позволяет снизить нагрузку на бэкенд-сервисы за счёт временного хранения часто запрашиваемых данных?**
- A) Кэширование (например, с использованием Redis или Memcached)**
 - B) Репликация баз данных в реальном времени
 - C) Синхронная передача всех запросов напрямую в БД
 - D) Отключение логирования в продакшене
- 26. Что такое «асинхронная интеграция через очередь сообщений» и какое преимущество она даёт?**
- A) Мгновенный ответ клиенту без обработки запроса
 - B) Развязка сервисов по времени: отправитель и получатель работают независимо, что повышает отказоустойчивость и сглаживает пиковые нагрузки**
 - C) Упрощение отладки за счёт линейного выполнения кода
 - D) Гарантированное отсутствие дубликатов сообщений без дополнительных механизмов
- 27. Какой подход к проектированию запросов к базе данных помогает избежать проблемы «N+1» при интеграции модулей?**

- A) Использование **eager loading** (жадной загрузки) или оптимизация запросов с JOIN
- B) Выполнение каждого запроса в отдельной транзакции
- C) Отключение индексов для ускорения вставки данных
- D) Хранение всех данных в одной таблице без нормализации

28. Что такое «балансировка нагрузки» (Load Balancing) и зачем она применяется в интегрированных системах?

- A) Для шифрования трафика между дата-центрами
- B) Для **равномерного распределения входящих запросов между несколькими экземплярами сервиса, повышая производительность и доступность**
- C) Для автоматического создания резервных копий данных
- D) Для преобразования форматов сообщений между сервисами

29. Какой метод оптимизации позволяет сократить объём передаваемых данных между клиентом и сервером?

- A) **Пагинация результатов, выборка только необходимых полей и сжатие (gzip)**
- B) Увеличение таймаутов ожидания ответа
- C) Дублирование данных во всех интегрированных системах
- D) Отказ от использования индексов в базе данных

30. Что означает термин «идемпотентность» в контексте интеграционных операций и почему он важен для масштабируемости?

- A) Способность системы работать без базы данных
- B) **Гарантия того, что повторное выполнение одного и того же запроса не изменит результат после первого успешного выполнения, что критично при повторных отправках в распределённых системах**
- C) Возможность выполнения запроса на любом языке программирования
- D) Автоматическое кэширование всех ответов сервиса

Тестирование

МДК.02.03 Поддержка и тестирование программных модулей

Раздел 3. Поддержка и тестирование программных модулей

1. Что является основным показателем качества программного обеспечения с точки зрения конечного пользователя?

- A) Количество строк исходного кода
- B) **Соответствие функциональным требованиям, надёжность и удобство использования**
- C) Сложность внутренней архитектуры системы
- D) Частота обновления сторонних библиотек

2. Какой международный стандарт описывает модель оценки качества ПО, включая функциональность, надёжность, производительность, безопасность и сопровождаемость?

- A) ISO 9001
- B) **ISO/IEC 25010**
- C) IEEE 830
- D) ITIL v4

3. Что подразумевает термин «сопровождаемость» (maintainability) в контексте качества кода?

- A) Скорость первоначальной разработки модуля
- B) **Лёгкость внесения изменений, исправления дефектов и адаптации продукта после его передачи в эксплуатацию**
- C) Возможность работы приложения на мобильных устройствах
- D) Автоматическое резервное копирование данных

4. Какая метрика чаще всего используется для оценки структурной сложности кода и прогнозирования его надёжности?

- A) Количество комментариев в файле
- B) **Цикломатическая сложность (число МакКейба)**

- C) Размер скомпилированного бинарного файла
- D) Время загрузки оперативной памяти
- 5. Что означает понятие «технический долг» и как он влияет на качество ПО?**
 - A) Обязательство оплатить лицензии на используемые фреймворки
 - B) Осознанный компромисс в виде упрощённого или неоптимального кода, ускоряющий разработку сейчас, но увеличивающий затраты на поддержку в будущем**
 - C) Накопление устаревшей документации в репозитории
 - D) Долгосрочные обязательства перед заказчиком по гарантийной поддержке
- 6. Какой процесс направлен на выявление отклонений фактических характеристик продукта от запланированных стандартов качества?**
 - A) Рефакторинг
 - B) Верификация и валидация (V&V)**
 - C) Интеграция
 - D) Деплой
- 7. Что характерно для «чистого кода» (Clean Code)?**
 - A) Максимально короткие имена переменных для экономии места
 - B) Читаемость, понятная логика, минимальная связанность и следование общепринятым соглашениям**
 - C) Отсутствие комментариев, так как код должен говорить сам за себя
 - D) Использование всех доступных возможностей языка в одной функции
- 8. Какой инструмент автоматизирует статический анализ кода для выявления потенциальных дефектов, уязвимостей и нарушений стандартов?**
 - A) Docker
 - B) SonarQube / Pylint / ESLint**
 - C) Git
 - D) Postman
- 9. Какова главная цель отладки (debugging) программного модуля?**
 - A) Оптимизация скорости выполнения кода
 - B) Поиск, локализация и устранение причин дефектов в исходном коде**
 - C) Генерация документации по API
 - D) Проверка соответствия дизайн-макетам
- 10. Какой механизм отладки позволяет приостановить выполнение программы в заданной строке для анализа состояния переменных?**
 - A) Логирование в консоль
 - B) Точка останова (Breakpoint)**
 - C) Условный компилятор
 - D) Сборщик мусора (Garbage Collector)
- 11. Что такое пошаговое выполнение «Step Into» в отладчике?**
 - A) Переход к следующей строке без входа во внутренние функции
 - B) Вход внутрь вызываемой функции или метода для отладки её кода построчно**
 - C) Пропуск всего цикла до его завершения
 - D) Автоматическое исправление найденной ошибки
- 12. Для чего используется «дамп памяти» (memory dump) при отладке упавших модулей?**
 - A) Для сжатия логов перед архивированием
 - B) Для сохранения снимка состояния оперативной памяти процесса в момент сбоя с целью последующего анализа**
 - C) Для принудительного освобождения ресурсов
 - D) Для экспорта данных в CSV-формат
- 13. Почему «состояние гонки» (race condition) считается одним из самых сложных дефектов для отладки?**
 - A) Оно всегда сопровождается сообщением об ошибке в консоли
 - B) Оно проявляется недетерминированно, зависит от порядка выполнения потоков и часто**

исчезает при подключении отладчика

- C) Оно возникает только на этапе компиляции
- D) Его можно обнаружить исключительно статическим анализом

14. Какой инструмент позволяет наблюдать за изменением значений переменных или выражений в реальном времени без остановки программы?

- A) Блок try-catch
- B) Панель Watches / Expression Evaluator в IDE**
- C) Файл конфигурации
- D) Менеджер зависимостей

15. Что такое «регрессионная отладка»?

- A) Отладка кода перед первым релизом
- B) Поиск причины ошибки, которая неожиданно появилась после внесения изменений в ранее стабильную версию модуля**
- C) Отладка только пользовательского интерфейса
- D) Автоматическая генерация тестовых данных

16. Какова основная цель механизма обработки исключений (try-catch-finally)?

- A) Ускорение выполнения арифметических операций
- B) Перехват и безопасная обработка ошибок времени выполнения без аварийного завершения программы**
- C) Автоматическое резервное копирование файлов
- D) Шифрование конфиденциальных данных в памяти

17. В каком блоке гарантированно выполняется код очистки ресурсов (заккрытие соединений, файлов) независимо от возникновения ошибки?

- A) В блоке try
- B) В блоке catch
- C) В блоке finally (или при использовании try-with-resources / using)**
- D) Вне конструкции обработки исключений

18. Чем принципиально отличается выброс исключения (throw) от возврата кода ошибки через return?

- A) Исключение работает быстрее
- B) Исключение прерывает обычный поток выполнения и передаёт управление ближайшему подходящему обработчику, отделяя обработку ошибок от бизнес-логики**
- C) Коды ошибок не требуют обработки
- D) Исключения автоматически логируются в БД

19. Что такое «непроверенное исключение» (unchecked exception / RuntimeException)?

- A) Ошибка, которую обязательно нужно ловить в каждом методе
- B) Ошибка времени выполнения, которую компилятор не требует явно обрабатывать или объявлять в сигнатуре метода**
- C) Исключение, возникающее только при компиляции
- D) Синтаксическая ошибка в исходном коде

20. Почему опасно использовать «пустой» блок catch, просто игнорирующий перехваченное исключение?

- A) Это замедляет работу процессора
- B) Это скрывает проблему, усложняет диагностику и может привести к некорректному состоянию системы в дальнейшем**
- C) Компилятор автоматически завершит программу
- D) Это нарушает лицензионное соглашение IDE

21. Какой принцип обработки исключений гласит, что обрабатывать ошибку следует на том уровне абстракции, где есть достаточный контекст для её устранения?

- A) Принцип единой ответственности
- B) Принцип делегирования обработки на подходящий уровень (не «глотать» исключения)**
- C) Принцип открытости/закрытости

D) Принцип минимальных привилегий

22. Что произойдёт в языке с проверкой исключений (например, Java), если метод выбрасывает checked-исключение, но не объявляет его в сигнатуре?

A) Программа скомпилируется, но упадёт в runtime

B) Возникнет ошибка компиляции

C) Исключение будет автоматически преобразовано в RuntimeException

D) Компилятор добавит блок catch автоматически

23. Для чего создаются пользовательские (кастомные) классы исключений?

A) Чтобы заменить стандартные системные ошибки

B) Чтобы передать специфическую информацию об ошибке доменной области и упростить фильтрацию в блоках catch

C) Чтобы ускорить выполнение кода

D) Чтобы отключить механизм логирования

24. Какой уровень тестирования фокусируется на проверке работоспособности отдельной функции, метода или класса в максимальной изоляции?

A) Системное тестирование

B) Модульное (юнит) тестирование (Unit Testing)

C) Приёмочное тестирование

D) Нагрузочное тестирование

25. Что означает принцип FIRST в контексте качественных юнит-тестов?

A) Functional, Integrated, Reliable, Secure, Timely

B) Fast (быстрые), Independent (независимые), Repeatable (повторяемые), Self-validating (самопроверяемые), Timely (своевременные)

C) Fixed, Isolated, Reviewed, Scalable, Tested

D) Flexible, Iterative, Robust, Standardized, Trusted

26. Какая практика разработки предполагает написание тестов до реализации рабочего кода модуля?

A) BDD (Behavior-Driven Development)

B) TDD (Test-Driven Development)

C) DDD (Domain-Driven Design)

D) CI/CD (Continuous Integration/Delivery)

27. Что такое «мокирование» (mocking) при тестировании модулей?

A) Копирование базы данных для тестов

B) Замена реальных внешних зависимостей (БД, API, файловая система) на имитационные объекты с заданным поведением

C) Автоматическая генерация тестовых данных

D) Запуск тестов в изолированном контейнере

28. Какой вид тестирования проверяет корректность взаимодействия двух и более ранее протестированных модулей?

A) Модульное тестирование

B) Интеграционное тестирование

C) Регрессионное тестирование

D) Альфа-тестирование

29. Что измеряет метрика «покрытие кода тестами» (code coverage)?

A) Количество найденных багов

B) Процент строк, ветвей или методов исходного кода, выполненных при запуске набора тестов

C) Время прохождения всех автотестов

D) Число разработчиков, участвовавших в написании тестов

30. Какой тип тестирования нацелен на проверку поведения модуля при передаче некорректных, граничных или недопустимых входных данных?

A) Позитивное тестирование

- B) Негативное тестирование (включая тестирование граничных значений)**
- C) Исследовательское тестирование
- D) A/B-тестирование

Тестирование

МДК.02.04 Математическое моделирование

Раздел 4. Математическое моделирование

- 1. Что является основным этапом математического моделирования, на котором реальная система описывается с помощью математических символов, уравнений и неравенств?**
 - A) Верификация модели
 - B) Формализация (построение математической модели)**
 - C) Валидация модели
 - D) Анализ чувствительности
- 2. Как называется процесс проверки соответствия результатов математической модели реальной системе или экспериментальным данным?**
 - A) Калибровка параметров
 - B) Валидация модели**
 - C) Идентификация структуры
 - D) Оптимизация алгоритма
- 3. Какой тип моделей описывает поведение системы, зависящее от случайных факторов и вероятностных закономерностей?**
 - A) Детерминированные модели
 - B) Стохастические модели**
 - C) Статические модели
 - D) Аналитические модели
- 4. Что подразумевается под «адекватностью» математической модели?**
 - A) Полное совпадение модели с оригиналом во всех физических деталях
 - B) Способность модели отражать существенные свойства объекта с требуемой для решения задачи точностью**
 - C) Минимальное количество переменных в уравнениях системы
 - D) Возможность решения модели исключительно аналитическим методом
- 5. Какая задача является классическим примером применения линейного программирования?**
 - A) Поиск кратчайшего пути в ориентированном графе
 - B) Задача оптимального распределения ограниченных ресурсов для максимизации прибыли**
 - C) Прогнозирование нестационарных временных рядов
 - D) Кластеризация многомерных данных
- 6. Что представляет собой целевая функция в задаче линейного программирования?**
 - A) Набор ограничений в виде системы линейных неравенств
 - B) Вектор искомых переменных решения
 - C) Линейная функция, максимум или минимум которой необходимо найти**
 - D) Матрица коэффициентов технологических ограничений
- 7. Какой метод является основным алгоритмическим решением канонических задач линейного программирования?**
 - A) Метод Ньютона-Рафсона
 - B) Метод градиентного спуска
 - C) Симплекс-метод**
 - D) Метод Монте-Карло
- 8. Что такое «допустимое решение» в линейном программировании?**
 - A) Набор значений переменных, удовлетворяющий всем ограничениям задачи (включая условия неотрицательности)**
 - B) Решение, дающее глобальный оптимум целевой функции

- С) Любое решение, полученное численным методом аппроксимации
 D) Решение, при котором все переменные равны нулю
- 9. Чем принципиально отличается задача нелинейного программирования от линейной?**
 A) В ней полностью отсутствуют ограничения на переменные
 B) Переменные могут принимать только целочисленные значения
 C) Целевая функция или хотя бы одно ограничение являются нелинейными функциями переменных
 D) Она решается исключительно графическим методом в двумерном пространстве
- 10. Какой метод чаще всего применяется для решения задач безусловной нелинейной оптимизации?**
 A) Симплекс-метод
 B) Метод потенциалов
 C) Метод градиентного спуска (или метод Ньютона)
 D) Метод ветвей и границ
- 11. Что такое «локальный оптимум» в нелинейном программировании?**
 A) Точка, где функция достигает абсолютного минимума на всей области определения
 B) Точка, значение функции в которой лучше, чем в любой другой точке некоторой окрестности, но не обязательно глобально
 C) Решение, нарушающее одно из ограничений системы
 D) Точка пересечения графиков всех ограничений
- 12. На каком принципе базируется метод динамического программирования?**
 A) Принцип наибольшего правдоподобия
 B) Принцип оптимальности Беллмана: оптимальная стратегия обладает тем свойством, что какое бы ни было начальное состояние, дальнейшее поведение должно быть оптимальным относительно этого состояния
 C) Принцип неопределённости Гейзенберга
 D) Принцип суперпозиции решений
- 13. Какая характерная черта задач решается методом динамического программирования?**
 A) Наличие многошагового процесса принятия решений с перекрывающимися подзадачами
 B) Строгая линейная зависимость целевой функции от переменных
 C) Полное отсутствие ограничений на расходуемые ресурсы
 D) Случайный характер входных данных на каждом шаге
- 14. Что означает техника «мемоизация» в контексте динамического программирования?**
 A) Удаление повторяющихся вычислений с помощью случайных выборок
 B) Сохранение результатов решения подзадач в таблице для их повторного использования без пересчёта
 C) Превращение рекурсивного алгоритма в итеративный без использования дополнительной памяти
 D) Автоматическая генерация программного кода на основе математических формул
- 15. Какой из перечисленных алгоритмов является классическим примером применения динамического программирования?**
 A) Алгоритм Дейкстры поиска кратчайшего пути
 B) Решение задачи о рюкзаке методом оптимального заполнения
 C) Метод наименьших квадратов для регрессии
 D) Быстрая сортировка массивов (QuickSort)
- 16. Что представляет собой сетевая модель в методах планирования (PERT/CPM)?**
 A) Таблицу распределения финансовых ресурсов по отделам
 B) Ориентированный граф, отображающий последовательность работ, их длительность и логические связи проекта
 C) Математическую модель линейной регрессии зависимых процессов
 D) Схему организационной иерархии компании
- 17. Как называется критический путь в сетевом графике проекта?**

- A) Самый короткий по времени маршрут выполнения всех работ
 - B) Самый длинный по времени маршрут, определяющий минимальную общую продолжительность проекта**
 - C) Путь с наименьшим потреблением трудовых ресурсов
 - D) Маршрут, проходящий через все без исключения события сети
- 18. Что показывает резерв времени работы (float/slack) в сетевом планировании?**
- A) Максимальное время, на которое можно задержать начало работы без увеличения общей продолжительности проекта**
 - B) Время, необходимое для устранения выявленных рисков
 - C) Период планового простоя оборудования
 - D) Запас финансовых ресурсов на непредвиденные расходы
- 19. Для чего используется метод PERT (Program Evaluation and Review Technique)?**
- A) Для точного расчёта себестоимости материалов
 - B) Для анализа и оценки сроков выполнения проекта в условиях неопределённости длительностей отдельных работ**
 - C) Для составления штатного расписания персонала
 - D) Для автоматической генерации отчётов по завершённым этапам
- 20. Какой параметр характеризует интенсивность входящего потока заявок в теории массового обслуживания?**
- A) Вероятность получения отказа
 - B) Среднее время обслуживания одной заявки
 - C) Параметр λ (лямбда) пуассоновского потока заявок**
 - D) Количество параллельных каналов обслуживания
- 21. Что означает формула Литтла ($L = \lambda W$) в контексте СМО?**
- A) Связь между вероятностью отказа и числом обслуживающих приборов
 - B) Фундаментальное соотношение: среднее число заявок в системе равно произведению интенсивности входящего потока на среднее время пребывания заявки в системе**
 - C) Уравнение для расчёта дисперсии времени ожидания в очереди
 - D) Формула для определения оптимального числа операторов call-центра
- 22. Как классифицируется СМО с ограниченным числом мест в очереди?**
- A) СМО с неограниченной очередью
 - B) СМО с отказами или ограниченной очередью (например, система типа M/M/1/N)**
 - C) СМО с бесконечным числом каналов обслуживания
 - D) Замкнутая сеть массового обслуживания
- 23. Что такое «равновесие Нэша» в теории игр?**
- A) Состояние, при котором все игроки получают максимально возможный суммарный выигрыш
 - B) Ситуация, в которой ни один игрок не может увеличить свой выигрыш, односторонне меняя свою стратегию при неизменных стратегиях других игроков**
 - C) Стратегия, гарантирующая минимальный проигрыш независимо от действий противника
 - D) Итог кооперативной игры с ненулевой суммой выигрышей
- 24. Какой тип игр предполагает, что выигрыш одного игрока равен проигрышу другого (сумма выигрышей постоянна)?**
- A) Кооперативные игры с формированием коалиций
 - B) Игры с нулевой суммой**
 - C) Динамические игры с неполной информацией
 - D) Повторяющиеся игры с обучением
- 25. Что представляет собой «максиминная стратегия» для игрока?**
- A) Стратегия, максимизирующая средний выигрыш при случайных действиях оппонента
 - B) Стратегия, максимизирующая минимально возможный выигрыш (гарантирующий результат в наихудшем случае)**
 - C) Стратегия, требующая обязательной кооперации с противником
 - D) Стратегия, основанная на полностью случайном выборе ходов

26. В чём основное отличие кооперативных игр от некооперативных?

- A) В кооперативных играх игроки могут заключать обязывающие соглашения, формировать коалиции и распределять общий выигрыш**
- B) В кооперативных играх полностью отсутствует конфликт интересов
- C) Кооперативные игры всегда имеют строго нулевую сумму
- D) В кооперативных играх используется исключительно чистая стратегия

27. В чём ключевое отличие имитационного моделирования от аналитического?

- A) Имитационные модели всегда дают более точный результат, чем аналитические
- B) Имитационное моделирование воспроизводит поведение системы во времени с помощью компьютерных экспериментов, когда аналитическое решение затруднено или невозможно**
- C) Аналитические модели не используют математический аппарат
- D) Имитационное моделирование применяется исключительно к детерминированным системам

28. Какой метод является основой для генерации случайных величин и проведения статистических экспериментов в имитационных моделях?

- A) Метод Монте-Карло**
- B) Симплекс-метод оптимизации
- C) Метод наименьших квадратов
- D) Метод градиентного спуска

29. Что такое «верификация» имитационной модели?

- A) Сравнение выходных результатов модели с реальными статистическими данными
- B) Проверка правильности программной реализации алгоритма и отсутствия логических ошибок (соответствие компьютерной модели концептуальной)**
- C) Оптимизация параметров модели для ускорения вычислительных экспериментов
- D) Документирование этапов жизненного цикла разработки ПО

30. Для чего применяется анализ чувствительности в имитационном моделировании?

- A) Для визуализации 3D-моделей физических объектов
- B) Для оценки влияния изменения входных параметров модели на стабильность и значения выходных показателей**
- C) Для автоматической трансляции модели на другой язык программирования
- D) Для сокращения объёма оперативной памяти, используемой симулятором

Тестирование
МДК.02.05 Численные методы
Раздел 5. Численные методы

1. Что такое абсолютная погрешность приближённого числа?

- A) Разность между точным и приближённым значением, взятая по модулю
- B) Модуль разности между точным значением величины и её приближённым значением**
- C) Отношение погрешности к истинному значению, выраженное в процентах
- D) Максимально возможное отклонение результата вычислений

2. Как определяется относительная погрешность измерения?

- A) Как разность абсолютных погрешностей двух последовательных измерений
- B) Как отношение абсолютной погрешности к модулю приближённого (или точного) значения величины**
- C) Как произведение абсолютной погрешности на число значащих цифр
- D) Как квадрат абсолютной погрешности, делённый на истинное значение

3. Что означает запись числа в нормализованной форме с плавающей запятой?

- A) Все цифры после запятой являются значащими
- B) Представление числа в виде мантиссы (в диапазоне $[0.1; 1)$ или $[1; 10)$) и порядка: $a = m \times 10^p$**
- C) Округление числа до ближайшего целого
- D) Хранение числа исключительно в двоичной системе счисления

4. Какое правило применяется для оценки предельной погрешности суммы приближённых

чисел?

A) Предельная абсолютная погрешность суммы равна сумме предельных абсолютных погрешностей слагаемых

B) Предельная относительная погрешность суммы равна сумме относительных погрешностей

C) Погрешность суммы всегда меньше погрешности любого слагаемого

D) Погрешность суммы вычисляется как геометрическое среднее погрешностей

5. Какой метод относится к итерационным методам решения нелинейных уравнений и использует касательные для приближения к корню?

A) Метод половинного деления

B) Метод Ньютона (метод касательных)

C) Метод простой итерации

D) Метод секущих

6. Какое условие является достаточным для сходимости метода простой итерации $x_{n+1} = \varphi(x_n)$ на отрезке $[a; b]$?

A) $\varphi(x)$ должна быть чётной функцией

B) $|\varphi'(x)| \leq q < 1$ для всех $x \in [a; b]$ (принцип сжимающих отображений)

C) $\varphi(x)$ должна иметь не более одного экстремума на отрезке

D) Производная $\varphi'(x)$ должна обращаться в ноль в точке корня

7. В чём основное преимущество метода половинного деления (бисекции)?

A) Наибольшая скорость сходимости среди всех методов

B) Гарантированная сходимость при выполнении условия смены знака функции на концах отрезка

C) Отсутствие необходимости вычислять значения функции

D) Возможность нахождения всех корней уравнения одновременно

8. Какой порядок сходимости имеет метод Ньютона при выполнении стандартных условий (простой корень, гладкая функция)?

A) Линейная сходимость

B) Квадратичная сходимость

C) Кубическая сходимость

D) Логарифмическая сходимость

9. Что такое «трансцендентное уравнение»?

A) Уравнение, содержащее только многочлены

B) Уравнение, содержащее трансцендентные функции (показательные, логарифмические, тригонометрические и т.п.), не сводимое к алгебраическому

C) Система линейных уравнений с комплексными коэффициентами

D) Дифференциальное уравнение в частных производных

10. Какой метод относится к прямым (точным) методам решения СЛАУ?

A) Метод простой итерации

B) Метод Гаусса (исключение неизвестных)

C) Метод Зейделя

D) Метод релаксации

11. В чём суть метода прогонки для решения СЛАУ с трёхдиагональной матрицей?

A) Прямое вычисление определителя матрицы

B) Специализированная модификация метода Гаусса, использующая рекуррентные формулы для эффективного решения систем с ленточной структурой

C) Итерационное уточнение решения с помощью метода Ньютона

D) Преобразование системы к диагональному виду методом вращений

12. Какое условие обеспечивает сходимость метода Зейделя для решения СЛАУ?

A) Симметричность матрицы системы

B) Строгое диагональное преобладание матрицы коэффициентов (или положительная определённость для симметричных матриц)

C) Наличие нулей на главной диагонали

D) Целочисленность всех элементов матрицы

13. Что означает «обусловленность» системы линейных уравнений?

A) Мера чувствительности решения системы к малым возмущениям входных данных (коэффициентов и свободных членов)

B) Количество ненулевых элементов в матрице системы

C) Возможность решения системы аналитическим методом

D) Скорость сходимости итерационного процесса

14. Для чего применяется выбор главного элемента в методе Гаусса?

A) Для уменьшения количества арифметических операций

B) Для повышения численной устойчивости алгоритма и снижения накопления вычислительной погрешности

C) Для автоматического определения ранга матрицы

D) Для преобразования системы к блочно-диагональному виду

15. В чём состоит основная задача интерполяции функции?

A) Построение функции (обычно многочлена), принимающей в заданных узлах те же значения, что и исходная функция

B) Нахождение производной функции в произвольной точке

C) Определение асимптотического поведения функции на бесконечности

D) Приближение функции с минимизацией среднеквадратичного отклонения на всём отрезке

16. Какой многочлен используется для интерполяции при равноотстоящих узлах и имеет компактную запись через конечные разности?

A) Многочлен Лагранжа

B) Многочлен Ньютона для интерполяции вперёд/назад

C) Многочлен Чебышёва

D) Тригонометрический многочлен Фурье

17. Что такое «явление Рунге» при полиномиальной интерполяции?

A) Рост погрешности интерполяции на концах отрезка при увеличении степени многочлена и использовании равноотстоящих узлов

B) Невозможность интерполяции функций с разрывами первого рода

C) Потеря значащих цифр при вычислении разностей близких чисел

D) Осцилляции, возникающие только при интерполяции периодических функций

18. Чем кубические сплайны предпочтительнее интерполяции одним многочленом высокой степени?

A) Они требуют меньше вычислительной памяти

B) Они обеспечивают гладкость (непрерывность второй производной) и отсутствие сильных осцилляций при большом числе узлов

C) Они всегда дают точное совпадение с исходной функцией

D) Они не требуют знания значений функции в узлах

19. На какой идее основана формула прямоугольников для приближённого вычисления определённого интеграла?

A) Замена подынтегральной функции постоянной величиной (значением в середине или на конце отрезка) на каждом частичном интервале

B) Аппроксимация функции линейной функцией на каждом отрезке разбиения

C) Использование параболы для аппроксимации функции на паре соседних интервалов

D) Разложение функции в ряд Тейлора и почленное интегрирование

20. Какая формула численного интегрирования имеет наивысший алгебраический порядок точности среди формул Ньютона-Котеса с тремя узлами?

A) Формула трапеций

B) Формула Симпсона (парабол)

C) Формула средних прямоугольников

D) Формула $3/8$

21. Что означает термин «составная квадратурная формула»?

- А) Формула, использующая комплексные узлы интегрирования
- В) Применение базовой квадратурной формулы на каждом отрезке разбиения с последующим суммированием результатов**
- С) Формула, учитывающая вторую и высшие производные подынтегральной функции
- Д) Метод, комбинирующий аналитическое и численное интегрирование
- 22. Как оценивается погрешность составной формулы трапеций?**
- А) Пропорциональна шагу разбиения h
- В) Пропорциональна квадрату шага разбиения h^2 и максимуму модуля второй производной на отрезке**
- С) Пропорциональна кубу шага h^3
- Д) Не зависит от шага разбиения при гладкой подынтегральной функции
- 23. Какой метод является простейшим одношаговым методом численного решения задачи Коши для ОДУ первого порядка?**
- А) Метод Эйлера (явная схема)**
- В) Метод Рунге-Кутты 4-го порядка
- С) Метод Адамса-Башфорта
- Д) Метод конечных разностей для краевых задач
- 24. В чём основное преимущество методов Рунге-Кутты перед методом Эйлера?**
- А) Меньший объём требуемой памяти
- В) Более высокий порядок точности при том же шаге интегрирования за счёт многократного вычисления правой части уравнения**
- С) Возможность решения только жёстких систем ОДУ
- Д) Отсутствие необходимости задания начальных условий
- 25. Что такое «жёсткая система» обыкновенных дифференциальных уравнений?**
- А) Система с нелинейными правыми частями
- В) Система, в которой собственные числа матрицы коэффициентов сильно различаются по модулю, что требует использования неявных схем для устойчивости**
- С) Система, не имеющая аналитического решения
- Д) Система с разрывными коэффициентами
- 26. Для чего применяется правило Рунге при численном решении ОДУ?**
- А) Для апостериорной оценки погрешности метода и адаптивного выбора шага интегрирования**
- В) Для преобразования уравнения высшего порядка к системе уравнений первого порядка
- С) Для стабилизации неустойчивых разностных схем
- Д) Для аналитического нахождения частного решения
- 27. Какой метод относится к градиентным методам безусловной оптимизации?**
- А) Метод золотого сечения
- В) Метод наискорейшего спуска (градиентный спуск)**
- С) Метод дихотомии
- Д) Метод Парето-оптимальности
- 28. Что характеризует условие Куна-Таккера в задачах математического программирования?**
- А) Достаточное условие глобального минимума для выпуклых функций
- В) Необходимое условие локального оптимума в задачах с ограничениями типа неравенств (обобщение метода множителей Лагранжа)**
- С) Критерий сходимости итерационных методов
- Д) Условие единственности решения системы уравнений
- 29. В чём суть метода покоординатного спуска (метод Гаусса-Зейделя для оптимизации)?**
- А) Последовательная минимизация целевой функции по одной переменной при фиксированных значениях остальных с циклическим перебором координат**
- В) Одновременное изменение всех переменных в направлении антиградиента
- С) Поиск оптимума только по границе допустимой области

D) Использование случайных возмущений для выхода из локальных минимумов

30. Какой подход применяется для решения задач условной оптимизации с помощью сведения к последовательности задач безусловной оптимизации?

A) Методы штрафных функций и барьерных функций

B) Метод Монте-Карло

C) Метод простых итераций для неподвижных точек

D) Метод конечных разностей для аппроксимации градиента

Тестирование

МДК.02.06 Безопасность программного обеспечения

Раздел 6. Безопасность программного обеспечения

1. Что является основной целью информационной безопасности программного обеспечения в рамках модели CIA?

A) Ускорение выполнения кода и оптимизация памяти

B) Обеспечение конфиденциальности, целостности и доступности данных и сервисов

C) Полное исключение возможности несанкционированного доступа к исходному коду

D) Автоматическое резервное копирование всех артефактов разработки

2. Какой принцип безопасности гласит, что система должна быть спроектирована так, чтобы по умолчанию доступ был запрещён, а разрешения выдавались явно?

A) Принцип «запрещено по умолчанию» (Fail-Safe Defaults / Deny by Default)

B) Принцип минимальных привилегий

C) Принцип полной медиации

D) Принцип открытого дизайна

3. Что подразумевается под «принципом наименьших привилегий» (Principle of Least Privilege)?

A) Все пользователи должны иметь права администратора для удобства работы

B) Каждому субъекту (пользователю, процессу, модулю) предоставляются только те права, которые минимально необходимы для выполнения его задач

C) Пароли должны меняться не реже одного раза в 90 дней

D) Критические функции системы должны выполняться в изолированном окружении

4. Какой тип уязвимости возникает, когда программа не проверяет границы буфера при копировании данных?

A) SQL-инъекция

B) Переполнение буфера (Buffer Overflow)

C) Межсайтовый скриптинг (XSS)

D) Подделка межсайтовых запросов (CSRF)

5. Что такое «инъекция» (Injection) в классификации уязвимостей OWASP Top 10?

A) Внедрение вредоносных данных в интерпретируемый код или запрос, что приводит к выполнению непредусмотренных команд

B) Несанкционированное копирование данных из защищённого хранилища

C) Перехват сетевого трафика с помощью сниффера

D) Подбор учётных данных методом грубой силы

6. Какой механизм позволяет защититься от атак типа «межсайтовый скриптинг» (XSS) при выводе пользовательских данных в веб-страницу?

A) Шифрование базы данных

B) Экранирование (escaping) или кодирование выводимых данных в соответствии с контекстом (HTML, JS, URL)

C) Использование HTTPS для всех соединений

D) Ограничение длины вводимых строк до 255 символов

7. Что означает термин «аутентификация» в контексте безопасности ПО?

A) Проверка прав пользователя на выполнение определённого действия

B) Подтверждение подлинности субъекта (пользователя, системы) на основе представленных

учётных данных

- С) Шифрование канала связи между клиентом и сервером
- Д) Регистрация всех действий пользователя в журнале аудита

8. Чем авторизация отличается от аутентификации?

- А) Это синонимы, обозначающие один и тот же процесс
- В) Авторизация — это проверка прав доступа к ресурсу после успешной аутентификации**
- С) Авторизация выполняется только для администраторов системы
- Д) Авторизация не требует наличия учётной записи пользователя

9. Какой метод хранения паролей считается безопасным стандартом в современной разработке?

- А) Хранение хеша пароля с использованием адаптивной криптографической функции (bcrypt, Argon2, scrypt) и уникальной «соли» для каждого пользователя**
- В) Шифрование пароля симметричным алгоритмом AES-256
- С) Хранение хеша пароля без соли с использованием MD5 или SHA-1
- Д) Хранение паролей в открытом виде с ограничением прав доступа к таблице БД

10. Что такое «сессия» и почему её защита критически важна?

- А) Временный кэш статических ресурсов на стороне клиента
- В) Механизм поддержания состояния взаимодействия пользователя с приложением; компрометация идентификатора сессии позволяет злоумышленнику выдать себя за пользователя**
- С) Протокол рукопожатия при установке TLS-соединения
- Д) Период времени, в течение которого пароль считается действительным

11. Какой тип атаки направлен на исчерпание ресурсов системы (процессор, память, канал связи) с целью отказа в обслуживании?

- А) DoS/DDoS-атака (Denial of Service)**
- В) Атака «человек посередине» (MitM)
- С) Фишинговая атака
- Д) Атака перебором паролей (Brute-force)

12. Что означает «безопасная разработка» (Secure SDLC)?

- А) Использование только проверенных библиотек с открытым исходным кодом
- В) Интеграция практик безопасности на всех этапах жизненного цикла ПО: от требований и проектирования до тестирования, развёртывания и сопровождения**
- С) Проведение пентеста только перед релизом продукта
- Д) Автоматическое исправление всех уязвимостей статическим анализатором

13. Какой документ описывает потенциальные угрозы, уязвимости и меры противодействия для конкретной архитектуры приложения?

- А) Модель угроз (Threat Model)**
- В) Техническое задание на разработку
- С) Политика резервного копирования
- Д) Реестр инцидентов информационной безопасности

14. Что такое «переполнение стека» (Stack Overflow) как уязвимость?

- А) Исчерпание оперативной памяти из-за утечек в куче
- В) Запись данных за пределы выделенного буфера в стеке, что может привести к перезаписи адреса возврата и выполнению произвольного кода**
- С) Ошибка рекурсивного вызова функции без условия выхода
- Д) Переполнение очереди сообщений между потоками

15. Какой принцип проектирования снижает риск компрометации всей системы при взломе одного компонента?

- А) Монолитная архитектура с единой базой данных
- В) Принцип разделения ответственности и изоляции компонентов (Defense in Depth, Least Privilege)**
- С) Использование одного учётного записи для всех сервисов

D) Хранение всех конфигураций в открытом репозитории

16. В чём принципиальное отличие симметричного шифрования от асимметричного?

A) Симметричное шифрование всегда медленнее

B) В симметричном шифровании один и тот же ключ используется для шифрования и расшифрования; в асимметричном — пара ключей: открытый и закрытый

C) Асимметричное шифрование применяется только для хеширования

D) Симметричное шифрование не требует управления ключами

17. Какой алгоритм является современным стандартом симметричного блочного шифрования?

A) DES

B) RC4

C) AES (Advanced Encryption Standard)

D) MD5

18. Для чего в криптографии используется «соль» (salt) при хешировании паролей?

A) Для защиты от атак с использованием предварительно вычисленных радужных таблиц и обеспечения уникальности хешей для одинаковых паролей

B) Для ускорения процесса вычисления хеш-функции

C) Для сжатия хеша до фиксированной длины

D) Для преобразования хеша в читаемый формат Base64

19. Что обеспечивает цифровая подпись на основе асимметричной криптографии?

A) Целостность данных, аутентификацию отправителя и неотказуемость (невозможность для отправителя отрицать факт подписания)

B) Конфиденциальность передаваемого сообщения

C) Сжатие данных перед передачей

D) Автоматическое шифрование канала связи

20. Какой режим работы блочного шифра обеспечивает наилучшую защиту при шифровании данных с повторяющимися блоками?

A) CBC (Cipher Block Chaining) или GCM (Galois/Counter Mode) с уникальным вектором инициализации (IV)

B) ECB (Electronic Codebook)

C) Простое побитовое сложение с ключом

D) Шифрование каждого блока одним и тем же ключом без вектора инициализации

21. Что такое «криптографически стойкий генератор псевдослучайных чисел» (CSPRNG) и почему он важен?

A) Генератор, выдающий числа в диапазоне от 0 до 1 с высокой точностью

B) Генератор, выходные данные которого невозможно предсказать даже при знании предыдущих значений; критичен для создания ключей, токенов, соли

C) Генератор, работающий на основе детерминированного алгоритма без энтропии

D) Генератор, оптимизированный для скорости, а не для безопасности

22. Какой протокол обеспечивает безопасную передачу данных в веб-приложениях, комбинируя асимметричное шифрование для обмена ключами и симметричное — для шифрования трафика?

A) FTP

B) HTTP

C) TLS/SSL

D) SNMP

23. Что означает термин «ключевое соглашение» (Key Agreement), например, в протоколе Диффи-Хеллмана?

A) Процедура, позволяющая двум сторонам, обмениваясь данными по открытому каналу, совместно выработать общий секретный ключ, неизвестный третьим лицам

B) Механизм централизованной выдачи сертификатов удостоверяющим центром

C) Процесс замены устаревшего ключа на новый без прерывания сессии

D) Протокол аутентификации по одноразовым паролям

24. Почему не рекомендуется разрабатывать собственные криптографические алгоритмы?

A) Криптоанализ — сложная научная дисциплина; самодельные алгоритмы, не прошедшие открытое экспертное сообщество, с высокой вероятностью содержат скрытые уязвимости

B) Это нарушает лицензионные соглашения на использование языков программирования

C) Собственные алгоритмы не поддерживаются аппаратным ускорением

D) Они всегда работают медленнее стандартных библиотек

25. Что такое «атака по сторонним каналам» (Side-Channel Attack)?

A) Взлом системы через социальную инженерию

B) Получение секретной информации путём анализа физических параметров реализации: времени выполнения, потребления энергии, электромагнитного излучения

C) Перехват данных в незащищённой сети Wi-Fi

D) Эксплуатация уязвимости в сторонней библиотеке

26. Какой стандарт описывает формат цифровых сертификатов открытого ключа (X.509)?

A) PKCS#7

B) X.509 (стандарт ITU-T)

C) RFC 2119

D) ISO 27001

27. Что означает «прямая секретность» (Perfect Forward Secrecy, PFS) в контексте протоколов обмена ключами?

A) Ключи никогда не истекают и могут использоваться бессрочно

B) Компрометация долгосрочного закрытого ключа не позволяет расшифровать ранее перехваченные сессии, так как для каждой сессии использовался уникальный временный ключ

C) Все ключи хранятся в аппаратном модуле безопасности (HSM)

D) Шифрование выполняется только на стороне сервера

28. Какой инструмент позволяет автоматически выявлять уязвимости безопасности в исходном коде на этапе разработки (SAST)?

A) Wireshark

B) SonarQube, Checkmarx, Fortify, Semgrep

C) Metasploit

D) Nmap

29. Что такое «управление секретами» (Secrets Management) в безопасной разработке?

A) Практика безопасного хранения, ротации и контроля доступа к конфиденциальным данным: ключам, токенам, паролям — с использованием специализированных хранилищ (Vault, AWS Secrets Manager)

B) Шифрование всех файлов проекта перед коммитом в Git

C) Запрет на использование внешних библиотек в проекте

D) Автоматическая замена паролей на случайные строки в конфигурационных файлах

30. Какой принцип криптографической защиты гласит, что безопасность системы не должна зависеть от секретности алгоритма, а только от секретности ключа?

A) Принцип Керкгоффса

B) Принцип минимальной достаточности

C) Принцип нулевого разглашения

D) Принцип открытого исходного кода

Критерии оценки тестовых заданий

Процент правильных ответов	Количество баллов	Оценка (по 5-балльной шкале)	Уровень освоения
90–100%	27–30	5 (отлично)	Высокий: глубокое понимание концепций, способность применять знания в нестандартных ситуациях
75–89%	23–26	4 (хорошо)	Повышенный: уверенное владение материалом, возможны единичные ошибки в деталях
60–74%	18–22	3 (удовлетворительно)	Базовый: знание основных понятий, но затруднения в применении к практическим задачам
Менее 60%	0–17	2 (неудовлетворительно)	Недостаточный: фрагментарные знания, требуется повторное изучение материала

3.1 Основные источники:

Перечень используемых учебных изданий, Интернет-ресурсов, дополнительной литературы

3.1.1. Основные печатные и/или электронные издания

1. Слабнов, В. Д. Численные методы и программирование: учебное пособие для СПО / В. Д. Слабнов. — 2-е изд., стер. — Санкт-Петербург: Лань, 2022. — 460 с. — ISBN 978-5-8114-9250-3. — Текст: электронный // Лань: электронно-библиотечная система. — URL: <https://e.lanbook.com/book/189402> — Режим доступа: для авториз. пользователей.

4.2.2. Дополнительные источники

1. ГОСТ 19.001-77. Государственный стандарт Союза ССР. Единая система программной документации. Общие положения (введен в действие Постановлением Госстандарта СССР от 20.05.1977 N 1268). - URL: <https://www.consultant.ru> - Режим доступа: Правовой сервер КонсультантПлюс. - Текст: электронный

2. ГОСТ 19.101-77. Государственный стандарт Союза ССР. Единая система программной документации. Виды программ и программных документов (введен Постановлением Госстандарта СССР от 20.05.1977 N 1268). - URL: <https://www.consultant.ru> - Режим доступа: Правовой сервер КонсультантПлюс. - Текст: электронный

3. ГОСТ 19.102-77. Государственный стандарт Союза ССР. Единая система программной документации. Стадии разработки (введен в действие Постановлением Госстандарта СССР от 20.05.1977 N 1268). - URL: <https://www.consultant.ru> - Режим доступа: Правовой сервер КонсультантПлюс. - Текст: электронный

4. ГОСТ 19.201-78. Государственный стандарт Союза ССР. Единая система программной документации. Техническое задание. Требования к содержанию и оформлению (введен в действие Постановлением Госстандарта СССР от 18.12.1978 N3351). - URL: <https://www.consultant.ru> - Режим доступа: Правовой сервер КонсультантПлюс. - Текст: электронный

5. ГОСТ 19.701-90. Единая система программной документации. Схемы алгоритмов, программ, данных и систем. Обозначения условные и правила выполнения (утв. Постановлением Госстандарта СССР от 26.12.1990 N 3294). - URL: <https://www.consultant.ru> - Режим доступа: Правовой сервер КонсультантПлюс. - Текст: электронный

6. ГОСТ Р ИСО/МЭК 25023-2021. Национальный стандарт Российской Федерации. Системная и программная инженерия. Требования и оценка качества систем и программной продукции (SQuaRE). Измерения качества системы и программной продукции (утв. и введен в действие Приказом Росстандарта от 19.11.2021 N 1524-ст). - URL: <https://www.consultant.ru> - Режим доступа: Правовой сервер КонсультантПлюс. - Текст: электронный

7. Згода Ю. Н. Проектирование программного обеспечения: учебно-методическое пособие / Ю. Н. Згода. – СПб.: Наукоемкие технологии, 2024. – 74 с. URL:<https://publishing.intelgr.com/archive/Proektirovanie-programmnogo-obespecheniya.pdf>. - Текст: электронный
8. Библиотека профессионала №1 <https://profspo.ru/>

4. КОНТРОЛЬНО-ОЦЕНОЧНЫЕ МАТЕРИАЛЫ ДЛЯ ИТОГОВОЙ АТТЕСТАЦИИ

Вопросы для итогового контроля по МДК.02.01 Разработка программных модулей

1. Модульная архитектура построения приложений. Принципы. Преимущества.

Примеры приложений

2. Архитектурные шаблоны, применяемые при разработке программных модулей (MVC, MVVM, MVP)

3. Инструменты разработки приложений с модульной архитектурой. Системы контроля версий.

4. Работа с библиотеками (применение стандартных библиотек, создание библиотек). Базовые принципы работы с массивами, коллекциями, строками. Работа с датой и временем.

5. Паттерны проектирования: отношения между классами и объектами (наследование, реализация, ассоциация, композиция, агрегация), интерфейсы, абстрактные классы, порождающие паттерны, паттерны поведения, структурные паттерны, поведенческие паттерны, паттерны объектов.

6. Система ввода-вывода, средства доступа к файлам и папкам файловой системы, чтения/записи, сжатия потоков и механизмов изолированного хранения.

7. Работа со строками, регулярными выражениями, кодирование/декодирование текста.

8. Асинхронная модель программирования. Пул потоков. Шаблон асинхронного вызова методов. Синхронизация вызывающего потока. Передача и прием специальных данных состояния.

9. Параллельное программирование. Создание задачи. Методы ожидания выполнения задачи. Лямбда-выражения в качестве задачи. Создание продолжения задачи. Возврат значений из задачи. Отмена задачи.

10. Алгоритмы и структуры данных. Оценка сложности алгоритмов. Понятие асимптотической оценки. Большие О-нотации. Временная сложность алгоритма. Пространственная сложность алгоритма. Анализ худшего, лучшего и среднего случаев.

11. Основные структуры данных (массив, связный список, стек, очередь; операции вставки, поиска и удаления; представление данных в памяти).

12. Алгоритмы сортировки и поиска. Основы рекурсии: примеры, преимущества и недостатки.

13. Хеш-таблица и хеш-функция. Коллизии и разрешение коллизий. Методы хеширования и сжатия данных. Эффективность и применение хеш-структур.

14. Деревья и графы. Представление графов и деревьев. Поиск в глубину и ширину. Минимум затратный путь (алгоритм Дейкстры). Деревья поиска и обхода.

15. Жадные алгоритмы и динамическое программирование. Основные идеи динамического программирования.

16. Алгоритмы работы с текстовыми данными. Операции над строками. Поиск подстроки (наивный алгоритм поиска, алгоритм Кнута-Морриса-Пратта, алгоритм Бойера-Мура). Проблемы на строках (Задача о рюкзаке, редакционное расстояние). Алгоритмы с использованием хеширования (хеш-функции для строк, алгоритм Рабина-Карпа). Строки и структуры данных (операции с динамическими строками, триальные деревья)

17. Кучи и очереди. Очереди с приоритетом и кучи. Куча и ее применение.

18. Основные принципы проектирования модулей программного обеспечения. Методы анализа требований и способов определения функциональности модуля. Методы анализа и оптимизации проектируемых модулей для повышения их эффективности и качества. Декомпозиция задачи на подзадачи. Создание спецификаций модуля.

19. Принципы обеспечения безопасности, производительности и масштабируемости при проектировании модулей

20. Принципы проектирования классов. Проектирование классов с учётом инкапсуляции. Использование наследования: создание иерархий классов. Полиморфизм: перегрузка методов и интерфейсов.

21. Применение диаграмм классов при проектировании требований к внутренней структуре программного модуля.

22. Применение диаграмм компонентов для визуализации организации компонентов проектируемого модуля

23. Виды пользовательского интерфейса (командная строка, графический, речевой). Основные этапы и принципы разработки графического пользовательского интерфейса.

24. Технологии и инструменты разработки графического пользовательского интерфейса.

25. Компоненты графического пользовательского интерфейса. Типы элементов управления. Компоновка элементов управления. События. Обработчики событий.

26. Работа с окнами. Основные методы работы с окнами. Создание окна: функции и классы. Открытие и закрытие окон. Взаимодействие с окнами (например, передача данных). Примеры валидации (проверка формата ввода). Сообщения об ошибках и уведомления пользователя. Использование регулярных выражений для валидации.

27. Многопоточность и асинхронная работа окон. Многопоточность в GUI-приложениях. Проблемы синхронизации потоков. Использование асинхронных вызовов для долго выполняемых операций.

28. Значение стиля в UX/UI дизайне. Основы теории цвета. Работа с цветом и шрифтами. Стилизация.

29. Работа с текстом, изображениями. Построение графиков и диаграмм. Библиотеки для построения графиков и диаграмм. Работа с мультимедиа

Вопросы для итогового контроля по МДК.02.02 Осуществление интеграции программных модулей

1. Разработка REST API. Клиент-серверное взаимодействие. Особенности передачи информации по HTTP протоколу. Структура HTTP запроса. HTTP методы: GET, POST, DELETE, PUT, PATCH. HTTP заголовки. Тело запроса.

2. Маршрутизация запросов. Группировка маршрутов. Статические ресурсы.

3. Обработка запросов пользователя. Path, Query параметры. Обработка содержимого body: raw, objects, forms, multipart. Валидация данных.

4. Формирование и отправка ответов: object, file. Параметры ответов: статус код, тип содержимого, заголовки, cookies. Перенаправления. Сериализация/десериализация объектов.

5. Создание и управление фоновыми задачами.

6. Аутентификация и авторизация. OAuth, JWT, forms. Сессии. Ролевое разграничение доступа к ресурсам.

7. Разработка WebSocket API. Взаимодействие клиента и сервера по WebSocket протоколу. Настройки соединения. Открытие и закрытие соединения. Передача сообщения серверу.

8. Разработка микросервисов. Микросервисная и монолитная архитектура.

9. Синхронное (REST, gRPC) и асинхронное (брокеры сообщений) взаимодействие между микросервисами.

10. Настройка конфигурации и сборки приложения.

11. Логирование событий. Конфигурация логирования. Уровни логирования. Логирование в файлы различного формат.

12. Мониторинг приложения: нагрузка, ошибки, сбор статистики. Внедрение сборщика метрик.

13. Инструменты контейнеризации. Контейнеризация приложения. Средства доставки и средства развертывания решения.

14. Протоколы с использованием безопасного соединения: HTTPS, WSS (WebSocket Secure).
15. Предотвращение угроз безопасности: SQL инъекции, CSRF, XSS. Хеширование чувствительных данных, применение алгоритмов хеширования паролей с солью.
16. Анализ уязвимостей. Регулярные аудиты безопасности. Применение лучших практик защиты информации.
17. Масштабирование интегрированных решений. Горизонтальное и вертикальное масштабирование.
18. Оптимизации производительности. Кэширование данных. Оптимизация запросов к базам данных.
19. Профилирование кода. Уменьшение времени отклика.

Вопросы для итогового контроля по МДК.02.03 Поддержка и тестирование программных модулей

1. Определение качества программного модуля. Метрики качества программных модулей (статические метрики: количество строк кода, цикломатическая сложность, коэффициент связности и сцепленности; динамические метрики: покрытие кода тестами, частота отказов, время отклика). Принципы проектирования качественных модулей.
2. Стандарты и модели качества программных модулей. Применение моделей качества. Инструменты для оценки качества. Практические аспекты повышения качества.
3. Понятие отладки. Понятия ошибки, дефекта, сбоя, отказа. Типы ошибок. Инструменты для отладки. Процесс пошаговой отладки (установка точек останова, шаг за шагом выполнение кода, просмотр состояния переменных, выполнение отдельных частей кода). Стратегии поиска ошибок (метод половинного деления, метод исключения, проверка граничных условий, поиск паттернов повторяющихся ошибок). Документирование процесса отладки.
4. Понятие исключения. Типы исключений. Механизм обработки исключений. Логика работы с исключениями. Методы отладки кода с использованием исключений и логирования.
5. Понятие процесса тестирования программного обеспечения. Этапы процесса тестирования программного обеспечения. техники ручного тестирования и автоматизированного тестирования
6. Модель работы с дефектами. Принципы работы в системе контроля дефектов.
7. Виды тестирования (функциональное тестирование, нефункциональное тестирование, статическое и динамическое тестирование).
8. Типы тестирования (модульное тестирование, интеграционное тестирование, системное тестирование, приемочное тестирование, нагрузочное тестирование, стресс-тестирование)
9. Тестирование по белому ящику. Метод покрытия операторов. Метод покрытия условий.
10. Тестирование по белому ящику. Метод комбинаторного покрытия условий.
11. Тестирование по черному ящику. Метод классов эквивалентности.
12. Тестирование по черному ящику. Метод граничных значений.
13. Модульные тесты. Тестирование интеграции. Методы и инструменты для тестирования интегрированных решений.

Вопросы для итогового контроля по МДК.02.04 Математическое моделирование

1. Понятие модели. Классификация моделей. Понятие математической модели. Типы математических моделей. Принципы построения математических моделей. Основные этапы математического моделирования.
2. Каноническая задача линейного программирования. Основные определения. Графический метод решения задач линейного программирования. Симплексный метод решения задач линейного программирования. Транспортная задача. Задача о назначениях. Целочисленное

программирование.

3. Основные понятия и определения нелинейного программирования. Методы решения задач нелинейного программирования.

4. Основные понятия и определения динамического программирования. Задачи, решаемые методами динамического программирования.

5. Основные понятия и определения теории графов. Нахождение кратчайшего пути. Дерево решений. Сетевые графики. Расчет временных параметров.

6. Марковский случайный процесс. Системы массового обслуживания: основные понятия, классификация. Схема гибели и размножения

7. Предмет и задачи теории игр. Основные понятия теории игр. Матричные игры. Биматричные игры. Игры в развернутой форме

8. Основные понятия имитационного моделирования. Примеры имитационных моделей. Методы имитационного моделирования. Инструментальные средства имитационного моделирования

Вопросы для итогового контроля по МДК.02.05 Численные методы

1. Способы хранения чисел в памяти компьютера. Абсолютная погрешность, относительная погрешность.

2. Верные, сомнительные, значащие цифры.

3. Погрешности арифметических действий. Оценка погрешностей значений функции. Отделение корней. Метод половинного деления. Метод простой итерации.

4. Методы Ньютона: метод хорд, касательных.

5. Методы Ньютона: комбинированный метод хорд и касательных. Сравнение методов вычислений по скорости сходимости итерационного процесса. Решение систем линейных алгебраических уравнений методом Гаусса. Применение метода Гаусса для вычисления определителей и нахождения обратной матрицы.

6. Метод простой итераций. Метод Зейделя. Сравнение методов вычислений по скорости сходимости итерационного процесса. Понятие интерполяции. Интерполяционный многочлен Лагранжа. Интерполяционные формулы Ньютона.

7. Интерполяция сплайнами.

8. Экстраполяция функций. Квадратурные формулы Ньютона-Котеса.

9. Квадратурная формула Гаусса. Сравнение методов численного интегрирования. Метод Эйлера. Уточненная схема Эйлера.

10. Метод Рунге – Кутты. Сравнение методов. Методы минимизации функции одной переменной: метод дихотомии, метод золотого сечения. Методы минимизации функции двух переменных: покоординатный спуск, наискорейший спуск

Вопросы для итогового контроля МДК.02.06 Безопасность программного обеспечения

1. Введение в кибербезопасность и уязвимости ПО.

2. Модели угроз и анализ рисков.

3. Уязвимости веб-приложений: OWASP Top 10.

4. Безопасная аутентификация и авторизация.

5. Криптография для разработчиков.

6. Принципы безопасного проектирования архитектуры.

7. Криптографические протоколы и их реализация.

8. Криптография в мобильных приложениях.

9. Криптография в веб-приложениях.

10. Криптография в облачных средах.

Критерии оценки

- оценка «отлично» выставляется студенту, если дан правильный ответ на 2

теоретических вопроса и выполнены правильно все практические задания;

- **оценка «хорошо»** если дан правильный ответ на 2 теоретических вопроса и выполнено правильно одно практическое задание или дан правильный ответ на теоретический вопрос и выполнены правильно все практические задания;

- **оценка «удовлетворительно»** если дан правильный ответ на теоретический вопрос и выполнено правильно одно практическое задание или дан правильный ответ на 2 теоретических вопроса, или выполнены правильно 2 практических задания;

- **оценка «неудовлетворительно»** если не дан правильный ответ на 2 теоретических вопроса и не выполнены правильно все практические задания.