Документ подписан простой электронной подписью
Информация о владельце:
ФИО: Гриб Владислав Валерьевич
Должность: Ректор
Дата подписания: 06.11.2026 16:26:52
Уникальный программный ключ:
637517d24e103c3db032acf37e839d98ec1c5bb2f5eb89c29abfcd7f43985447

Образовательное частное учреждение высшего образования

«МОСКОВСКИЙ УНИВЕРСИТЕТ ИМЕНИ А.С. ГРИБОЕДОВА»

(ИМПЭ им. А.С. Грибоедова)

ФАКУЛЬТЕТ СРЕДНЕГО ПРОФЕССИОНАЛЬНОГО ОБРАЗОВАНИЯ
(МЕЖДУНАРОДНЫЙ ДИПЛОМАТИЧЕСКИЙ КОЛЛЕДЖ)

УТВЕРЖДАЮ
Директор Международного
дипломатического колледжа
_____ Н.И. Платонова
«16» февраля 2026г.

ФОНД ОЦЕНОЧНЫХ СРЕДСТВ
по учебной дисциплине

Информационные технологии в профессиональной деятельности

Специальность
09.02.11 Разработка и управление программным обеспечением

Квалификация выпускника:
Программист

на базе основного общего образования

Форма обучения: очная

Москва

Фонд оценочных средств для дисциплины «Информационные технологии в профессиональной деятельности». Специальность 09.02.11 Разработка и управление программным обеспечением /– М.: ИМПЭ им. А. С. Грибоедова.

Фонд оценочных средств является неотъемлемой частью рабочей программы дисциплины.

Организация-разработчик: Образовательное частное учреждение высшего образования «Московский университет имени А. С. Грибоедова»

Директор Международного
дипломатического колледжа_____/Н.И. Платонова

1. ФОНД ОЦЕНОЧНЫХ СРЕДСТВ ДЛЯ ТЕКУЩЕГО КОНТРОЛЯ УСПЕВАЕМОСТИ, ПРОМЕЖУТОЧНОЙ АТТЕСТАЦИИ ОБУЧАЮЩИХСЯ ПО ДИСЦИПЛИНЕ

Настоящий Фонд оценочных средств (ФОС) по дисциплине «Информационные технологии в профессиональной деятельности» является неотъемлемым приложением к рабочей программе дисциплины (РПД) «Информационные технологии в профессиональной деятельности». На данный ФОС распространяются все реквизиты утверждения, представленные в РПД по данной дисциплине.

2. ПЕРЕЧЕНЬ ОЦЕНОЧНЫХ СРЕДСТВ

Для определения качества освоения обучающимися учебного материала по дисциплине используются следующие оценочные средства:

№ п/п	Оценочное средство	Краткая характеристика оценочного средства	Представление оценочного средства в ФОС
1	Тестирование	Вид контроля, позволяющий оценить изученный теоретический материал.	Вопросы для проведения тестирования
2	Практические задания	Вид контроля, позволяющий оценить умение обучающегося применять осваиваемую компетенцию в практических ситуациях и при решении производственных задач	Задания к практическому занятию
3	Контрольная работа	Вид контроля, позволяющий определить результат освоения компетенций по дисциплине в рамках рассматриваемой темы, оцениваемый с помощью соответствующих индикаторов достижения компетенций	Задания контрольной работы
4	Самостоятельная работа	Вид контроля, позволяющий оценить проработку теоретического материала, изучение рекомендуемой литературы, выполнение практико-ориентированных заданий (заполнение таблиц, проведение сравнительного анализа, составление схем и др.), решение практических задач, создание презентаций, написание рефератов, подборку нормативного и иного материала и выполнение других заданий	Задания самостоятельной работы
5.	Зачет/Зачет с оценкой/Экзамен	Вид контроля, позволяющий выявить степень овладения знаниями, умениями и навыками, необходимыми для дальнейшего освоения образовательной программы подготовки	Вопросы для подготовки к зачету/зачету с оценкой/экзамену

3.ТИПОВЫЕ КОНТРОЛЬНЫЕ ЗАДАНИЯ ИЛИ ИНЫЕ МАТЕРИАЛЫ, НЕОБХОДИМЫЕ ДЛЯ ОЦЕНКИ РЕЗУЛЬТАТОВ ОБУЧЕНИЯ ПО УЧЕБНОЙ ДИСЦИПЛИНЕ

ФОНД ОЦЕНОЧНЫХ СРЕДСТВ

Темы для подготовки к промежуточной аттестации

1.Стратегии эффективного поиска технической информации. Формирование поисковых запросов.
2. Иерархия надежности информационных источников в IT (документация, Q&A, блоги, видео).
3. Критический анализ и верификация найденных решений технических проблем.
4. Основы работы с системами контроля версий. Философия и преимущества Git.
5. Ключевые понятия Git: репозиторий, коммит, ветка, история, рабочая копия, индекс.
6. Модели ветвления (Git Flow, GitHub Flow). Создание, слияние веток, разрешение конфликтов.
7. Работа с удаленными репозиториями (GitHub, GitLab). Команды push, pull, fetch.
8. Процесс code review. Pull Request / Merge Request как инструмент командной работы и контроля качества.
9. Понятие предметной области. Методы выявления сущностей, атрибутов и связей.
10. Концептуальное моделирование данных. Модель «Сущность-Связь» (ER-модель).
11. Нотации ER-диаграмм: классическая (Чена), «воронья лапка» (IE notation).
12. Основы реляционной модели данных: отношение, атрибут, кортеж, домен, схема.
13. Типы ключей в реляционной модели: первичный, потенциальный, внешний.
14. Понятие целостности данных: доменная, целостность сущности, ссылочная целостность.
15. Нормализация баз данных: цели, аномалии (вставки, обновления, удаления).
16. Нормальные формы: первая (1НФ), вторая (2НФ), третья (3НФ). Правила приведения.
17. Денормализация: цели, риски, примеры оправданного применения.
18. Язык SQL: назначение, подмножества (DDL, DML, DCL, TCL).
19. Типы данных в СУБД: числовые, строковые, дата/время, логические, специализированные.
20. Команды DDL: CREATE, ALTER, DROP для объектов БД (таблицы, индексы).
21. Ограничения целостности (CONSTRAINTS): PRIMARY KEY, FOREIGN KEY, UNIQUE, NOT NULL, CHECK, DEFAULT.
22. Опции каскадного обновления и удаления для внешних ключей (ON UPDATE, ON DELETE).
23. Индексы в базах данных: назначение, структура (B-дерево), влияние на производительность (SELECT vs. INSERT/UPDATE).
24. Транзакции. Свойства ACID (атомарность, согласованность, изоляция, долговечность).
25. Команды управления транзакциями: BEGIN, COMMIT, ROLLBACK.

26. Угрозы информационной безопасности БД. SQL-инъекция: механизм, примеры, последствия.
27. Основные методы защиты от SQL-инъекций: параметризованные запросы, валидация входных данных, использование ORM.
28. Понятие модуля в программировании. Принципы сильной связности (high cohesion) и слабого зацепления (low coupling).
29. Принципы SOLID: единственной ответственности, открытости/закрытости, подстановки Лисков, разделения интерфейса, инверсии зависимостей.
30. Паттерны проектирования: понятие, назначение, классификация (порождающие, структурные, поведенческие).
31. Паттерны уровня доступа к данным: Active Record, Data Mapper, Repository, Unit of Work.
32. Базовые структуры данных: массивы, списки, стеки, очереди, словари (хэштаблицы), множества.
33. Оценка сложности алгоритмов. Нотация «О» большое (Big O notation) для описания временной сложности.
34. Принципы тестирования программного обеспечения. Пирамида тестирования.
35. Модульное тестирование (Unit Testing): цели, принципы (FIRST), структура теста (Arrange, Act, Assert).
36. Интегрированные среды разработки (IDE): назначение, основные возможности (редактор, отладчик, рефакторинг).
37. Системы сборки и управления зависимостями: назначение, примеры (Maven для Java, pip для Python, NuGet для .NET).
38. Концепция непрерывной интеграции и непрерывного развертывания (CI/CD). Преимущества.
39. Компоненты CI/CD-пайплайна: триггеры, этапы (stages), задания (jobs), артефакты.
40. Контейнеризация. Понятие контейнера (Docker). Преимущества использования в разработке и развертывании.
41. Мониторинг и логирование в ПО. Назначение, типы логов, основные принципы.
42. Профилирование кода: цели и основные методы выявления «узких мест» (bottlenecks).
43. Оптимизация работы с базами данных: анализ плана выполнения запроса (EXPLAIN), пересмотр индексов.
44. Кэширование данных: принципы, уровни кэширования (клиент, сервер приложений, БД), стратегии инвалидации.
45. Многопоточность и параллелизм. Понятия процесса и потока. Проблемы параллельного доступа (состояние гонки, взаимная блокировка).
46. Примитивы синхронизации: мьютекс, семафор.
47. Управление памятью. Модель управления в языках со сборкой мусора (Garbage Collection) и с ручным управлением.
48. Основы сетевых протоколов применительно к разработке ПО (HTTP/HTTPS, REST). Факторы, влияющие на скорость обмена данными.
49. Принципы обеспечения безопасности в разрабатываемом ПО: аутентификация, авторизация, шифрование конфиденциальных данных
50. Роль информационных технологий в современном жизненном цикле программного обеспечения.

Темы для самостоятельной работы обучающихся

1. Эволюция парадигм программирования: сравнительный анализ императивного, декларативного, функционального и объектно-ориентированного подходов. Их влияние на современную разработку ПО.
2. Правовые кейсы в IT: анализ известных судебных разбирательств, связанных с нарушением лицензий GPL или авторских прав на код. Выводы для разработчика.
3. Этика искусственного интеллекта и машинного обучения: исследование проблем bias (смещения) в алгоритмах, ответственности за решения, принимаемые ИИ.
4. Модели монетизации программного обеспечения: сравнительный анализ продажи лицензий, подписочной модели (SaaS), freemium, open-core. Плюсы и минусы для разработчика и пользователя.
5. Сравнительный анализ IT-рынка труда в России, США и ЕС: востребованные специальности, уровень зарплат, требования к квалификации, тенденции.
6. Роль soft skills в карьере IT-специалиста: исследование важности коммуникации, управления временем, работы в команде, эмоционального интеллекта на разных карьерных этапах.
7. Финансовая грамотность для IT-фрилансера: изучение систем налогообложения (НПД, ИП), составление коммерческого предложения, калькуляция стоимости проекта.
8. История возникновения и философия движения Open Source: от истоков (Free Software Foundation) до современных экосистем. Вклад в индустрию.
9. Анализ профессиональных стандартов в IT (например, Skills Framework for the Information Age - SFIA): сопоставление с собственным уровнем навыков и построение карьерного плана.
10. Культура "хакерспейсов" и DIY в IT: их роль в образовании, нетворкинге и инновациях вне корпоративной среды.
11. Сравнительный анализ методологий DevOps, DevSecOps и DataOps: цели, принципы, инструменты, области применения.
12. Концепция "Serverless"-архитектуры (FaaS): принципы работы, сценарии использования, экономические и технические преимущества/ограничения по сравнению с традиционными подходами.
13. Принципы Domain-Driven Design (DDD): изучение основных понятий (ограниченный контекст, ubiquitous language, сущности, объекты-значения, агрегаты) на примере.
14. Паттерны интеграции корпоративных приложений (Enterprise Integration Patterns): исследование паттернов "шины данных" (Enterprise Service Bus) и "гибридной интеграции".
15. Угрозы информационной безопасности на уровне приложения
16. Принципы криптографии для разработчика: симметричное и асимметричное шифрование, хэширование, цифровые подписи, SSL/TLS. Область применения в типовых задачах.
17. Концепция "Памяти передачи данных" (Data Transfer Object - DTO) и "Модели предметной области" (Domain Model): сравнение целей, способов использования и антипаттернов.

18. Теория CAP-теоремы (Согласованность, Доступность, Устойчивость к разделению) в распределенных системах: последствия выбора той или иной комбинации свойств для архитектуры ПО.
19. Анализ моделей зрелости процессов разработки ПО (CMMI, ISO/IEC 15504): уровни зрелости и их характеристика.
20. Концепция "Долговременной поддержки" (LTS) в версиях ПО и экосистемах (ОС, фреймворки): важность для корпоративной разработки и планирования жизненного цикла проекта.
21. Сравнительный анализ реляционных (SQL) и нереляционных (NoSQL) баз данных: типы NoSQL (документные, ключ-значение, колоночные, графовые), принципы ACID vs. BASE, критерии выбора.
22. Теоретические основы транзакций и уровни изоляции в SQL (Read Uncommitted, Read Committed, Repeatable Read, Serializable): проблемы, которые решает каждый уровень (грязное чтение, неповторяющееся чтение, фантомное чтение).
23. Оконные функции (Window Functions) в SQL: теоретическое изучение синтаксиса, понятий "окно" (OVER), "партиция" (PARTITION BY), "упорядочение" (ORDER BY), "фрейм". Примеры использования.
24. Принципы построения и оптимизации звездообразной и снежинкообразной схем в хранилищах данных (Data Warehouse). Роль фактов и измерений.
25. Основы теории графов и ее применение в IT: понятия вершины, ребра, пути, циклов. Использование в графовых базах данных, анализе социальных сетей, системах рекомендаций.
26. Алгоритмы сортировки: "быстрая сортировка" (Quicksort) и "сортировка слиянием" (Mergesort). Сравнение алгоритмической сложности, устойчивости, принципа "разделяй и властвуй".
27. Структуры данных "дерево" и "граф": теоретическое исследование свойств, обходов (in-order, pre-order, post-order для дерева), сценариев применения (бинарное дерево поиска, красно-черное дерево).
28. Концепция "Big Data": определение 3V (Volume, Velocity, Variety). Обзор экосистемы и парадигм обработки (MapReduce, Apache Spark) на концептуальном уровне.
29. Принципы репликации и шардирования (сегментирования) баз данных: цели (масштабируемость, отказоустойчивость), основные стратегии, возникающие проблемы (согласованность в конечном счете).
30. Теоретические основы полнотекстового поиска: индексация, лемматизация, стемминг, работа с стоп-словами. Принципы работы поисковых движков (на примере Elasticsearch/Solr).

Примерные варианты тестовых оценочных заданий (ТОЗ) для контрольного рубежа в рамках текущего контроля

Тестовое задание	Оцениваемые компетенции
ВОПРОСЫ ОТКРЫТОГО ТИПА	
(ВПИСАТЬ ПРАВИЛЬНЫЙ ОТВЕТ) Лицензия, которая обязывает разработчика распространять исходный код производных работ под той же лицензией, называется _____ +: Копилефт	ОК 03.
(ВПИСАТЬ ПРАВИЛЬНЫЙ ОТВЕТ) Модель данных, в которой информация организована в виде таблиц (отношений), состоящих из строк и столбцов, это _____ +: Реляционная модель	ПК 1.2.
(ВПИСАТЬ ПРАВИЛЬНЫЙ ОТВЕТ) Процесс устранения избыточности данных и аномалий вставки, обновления, удаления в БД называется_____. +: Нормализация	ПК 1.2.
(ВПИСАТЬ ПРАВИЛЬНЫЙ ОТВЕТ) Первая нормальная форма требует, чтобы все значения в столбцах были_____. +: Атомарными	ПК 1.2.
(ВПИСАТЬ ПРАВИЛЬНЫЙ ОТВЕТ) Объект базы данных, который ускоряет выполнение операций поиска и выборки данных, но замедляет вставку, это_____. +: Индекс	ПК 1.2.
(ВПИСАТЬ ПРАВИЛЬНЫЙ ОТВЕТ) Уязвимость, возникающая при внедрении злонамеренного SQL-кода через пользовательский ввод, называется_____.	ОК 02.
+: SQL-инъекция	
(ВПИСАТЬ ПРАВИЛЬНЫЙ ОТВЕТ) Команда, которая создает точку сохранения состояния проекта в системе Git, называется_____. +: Коммит	ОК 02.
(ВПИСАТЬ ПРАВИЛЬНЫЙ ОТВЕТ) Принцип SOLID, гласящий, что класс должен иметь только одну причину для изменения, это принцип_____. +: Единственной ответственности	ПК 2.2.
(ВПИСАТЬ ПРАВИЛЬНЫЙ ОТВЕТ) Структура данных, которая хранит элементы в порядке "первый вошел, первый вышел" (FIFO), называется_____. +: Очередь	ПК 2.2.

(ВПИСАТЬ ПРАВИЛЬНЫЙ ОТВЕТ) Технология "упаковки" приложения и всех его зависимостей в изолированный контейнер для единообразного запуска в любой среде, это_____. +: Контейнеризация	ОК 02.
(ВПИСАТЬ ПРАВИЛЬНЫЙ ОТВЕТ) Шаблон проектирования, который абстрагирует доступ к хранилищу данных (например, БД), представляя его как коллекцию объектов в памяти, называется_____. +: Репозиторий	ПК 1.2.
(ВПИСАТЬ ПРАВИЛЬНЫЙ ОТВЕТ) Проблема, возникающая, когда несколько потоков одновременно обращаются к общим данным и хотя бы один поток их изменяет, называется _____. +: Состояние гонки	ПК 2.2.
(ВПИСАТЬ ПРАВИЛЬНЫЙ ОТВЕТ) Примитив синхронизации, который разрешает доступ к общему ресурсу только одному потоку в данный момент времени, называется _____. +: Мьютекс	ПК 2.2
(ВПИСАТЬ ПРАВИЛЬНЫЙ ОТВЕТ) Процесс изменения внутренней структуры кода без изменения его внешнего поведения с целью улучшения читаемости и сопровождаемости называется _____. +: рефакторинг	ПК 2.2.
(ВПИСАТЬ ПРАВИЛЬНЫЙ ОТВЕТ) Вид тестирования, который проверяет взаимодействие нескольких модулей или систем, называется _____. +: Интеграционное тестирование	ПК 2.2.
(ВПИСАТЬ ПРАВИЛЬНЫЙ ОТВЕТ) Инструмент в IDE, позволяющий выполнять код пошагово, анализировать значения переменных и находить ошибки, называется _____. +: Откладчик	ОК 02.
(ВПИСАТЬ ПРАВИЛЬНЫЙ ОТВЕТ) ременное быстрое хранилище данных, предназначенное для ускорения доступа к часто запрашиваемой информации, называется _____. +: Кэш	ПК 2.2.

(ВПИСАТЬ ПРАВИЛЬНЫЙ ОТВЕТ) Метрика, показывающая, какая часть исходного кода программы была выполнена в ходе тестов, называется _____. +: Покрытие кода	ПК 2.2.
(ВПИСАТЬ ПРАВИЛЬНЫЙ ОТВЕТ) Механизм автоматического освобождения памяти, занятой объектами, на которые больше нет ссылок в программе, называется _____. +: Сборка мусора	ПК 2.2.

(ВПИСАТЬ ПРАВИЛЬНЫЙ ОТВЕТ) Процесс, при котором пользователям предоставляется новая версия приложения, пока часть трафика продолжает работать на старой, называется _____. +: Сине-зеленое развертывание	ОК 03.
(ВПИСАТЬ ПРАВИЛЬНЫЙ ОТВЕТ) Фундаментальный принцип, согласно которому 80% времени выполнения программы приходится на 20% её кода, известен как _____. +: Принцип Парето	ПК 2.2.
ВОПРОСЫ ЗАКРЫТОГО ТИПА	
Какая из перечисленных лицензий является пермиссивной и позволяет использовать код в коммерческих закрытых продуктах без обязательного раскрытия исходного кода производных работ? -: GNU GPL v3 +: MIT License -: Apache License 2.0 -: GNU AGPL v3	ОК 03.
При возникновении сложной ошибки (exception) в коде разработчик в первую очередь должен искать решение: -: В комментариях на YouTube -: В общих чатах в мессенджерах +: В официальной документации к используемой библиотеке или на Stack Overflow -: Методом случайного перебора вариантов в коде	ОК 02.
Какой SQL-оператор используется для изменения структуры существующей таблицы (например, добавления нового столбца)? -: UPDATE -: INSERT +: ALTER TABLE -: MODIFY	ПК 1.2.
Процесс приведения структуры базы данных к виду, исключающему избыточность данных и аномалии обновления, называется: -: Денормализация +: Нормализация -: Индексация -: Транзакция	ПК 1.2.
Для чего в реляционной базе данных primarily используется внешний ключ (FOREIGN KEY)? -: Для ускорения поиска записей -: Для обеспечения уникальности значений в столбце +: Для обеспечения ссылочной целостности и связи между таблицами -: Для автоматического шифрования данных	ПК 1.2.
Какой принцип SOLID гласит: "Программные сущности должны быть	ПК 2.2.

открыты для расширения, но закрыты для модификации"? -: Принцип единственной ответственности (Single Responsibility) +: Принцип открытости/закрытости (Open/Closed) -: Принцип подстановки Барбары Лисков (Liskov Substitution) -: Принцип разделения интерфейса (Interface Segregation)	
Паттерн проектирования Repository primarily используется для: -: Визуализации данных в пользовательском интерфейсе -: Создания сложных условий в SQL-запросах +: Абстрагирования доступа к источнику данных (например, БД) и предоставления коллекции объектов предметной области г) Кэширования HTTP-запросов	ПК 2.2.
Вопрос: Интеграционное тестирование — это тестирование, которое проверяет: -: Корректность работы одной отдельной функции или метода +: Взаимодействие и совместную работу нескольких модулей или компонентов системы -: Соответствие интерфейса системы дизайн-макетам -: Устойчивость системы под высокой нагрузкой	ПК 2.2.
то является основной целью использования параметризованных запросов (prepared statements) при работе с базой данных из кода? -: Ускорение выполнения всех типов запросов -: Упрощение синтаксиса SQL-команд +: Предотвращение атак типа SQL-инъекция за счет отделения данных от кода запроса -: Автоматическое логирование всех операций с БД	ПК 1.2

Шкала оценивания тестовых заданий

Данный вид контроля, рассчитан на выявление уровня усвоения теоретического и практического материала в рамках изучения дисциплин. Критерии оценивания теста: 1 вопрос = 1 балл.

Оценка «5»	30-27 Баллов
Оценка «4»	26-23 Баллов
Оценка «3»	22-18 Баллов
Оценка «2»	0-17 Баллов