Документ подписан простой электронной подписью
Информация о владельце:
ФИО: Гриб Владислав Валерьевич
Должность: Ректор
Дата подписания: 06.11.2026 16:22:27
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.

РАБОЧАЯ ПРОГРАММА УЧЕБНОЙ ДИСЦИПЛИНЫ

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

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

Рабочая программа учебного предмета составлена на основании Федерального государственного образовательного стандарта 09.02.11 Разработка и управление программным обеспечением, утвержденного приказом №138 от 24.02.2025 г. «Об утверждении федерального государственного образовательного стандарта среднего профессионального образования по специальности 09.02.11 Разработка и управление программным обеспечением».

Организация-разработчик:	Образовательное частное учреждение высшего образования «Московский университет имени А. С. Грибоедова»
Ответственный рецензент:	Е.В. Михалёва, к. ф.-м. н. исполнительный директор института информационных систем и инженерно-компьютерных технологий
	(Ф.И.О., уч. степень, уч. звание, должность)

Рабочая программа дисциплины рассмотрена и одобрена на заседании Международного дипломатического колледжа «16» февраля 2026 г., протокол №6

Директор Международного дипломатического колледжа_____/Н.И. Платонова

Согласовано от Библиотеки_____/О. Е. Стёпкина

1. ОБЩАЯ ХАРАКТЕРИСТИКА РАБОЧЕЙ ПРОГРАММЫ ДИСЦИПЛИНЫ

1.1. Область применения рабочей программы дисциплины

Рабочая программа дисциплины «Информационные технологии в профессиональной деятельности» является частью программы подготовки специалистов среднего звена в соответствии с ФГОС по специальности 09.02.11 Разработка и управление программным обеспечением.

1.2. Место дисциплины в структуре программы подготовки специалистов среднего звена

Дисциплина «Информационные технологии в профессиональной деятельности» является обязательной частью социально-гуманитарного цикла основной профессиональной образовательной программы в соответствии с ФГОС по специальности 09.02.11 Разработка и управление программным обеспечением.

Дисциплина «Информационные технологии в профессиональной деятельности» обеспечивает формирование общих компетенций по видам деятельности ФГОС по специальности 09.02.11 Разработка и управление программным обеспечением.

1.3. Цель и планируемые результаты освоения дисциплины

Обучающийся должен освоить следующие общие компетенции:

ОК 02. Использовать современные средства поиска, анализа и интерпретации информации, и информационные технологии для выполнения задач профессиональной деятельности;

ОК 03. Планировать и реализовывать собственное профессиональное и личностное развитие, предпринимательскую деятельность в профессиональной сфере, использовать знания по правовой и финансовой грамотности в различных жизненных ситуациях

ПК 1.2 Разрабатывать объекты баз данных в соответствии с результатами анализа предметной области

ПК 2.2 Разрабатывать модули программного обеспечения

В рамках программы дисциплины обучающимися осваиваются умения и знания:

Код ПК, ОК	Умения	Знания

ОК 02. ОК 03. ПК 1.2. ПК 2.2.	- определять задачи для поиска информации, планировать процесс поиска, выбирать необходимые источники информации; - выделять наиболее значимое в перечне информации, структурировать получаемую информацию, оформлять результаты поиска; - оценивать практическую значимость результатов поиска; - применять средства информационных технологий для решения профессиональных задач; - использовать современное программное обеспечение в профессиональной деятельности; - использовать различные цифровые средства для решения профессиональных задач;	- актуальный профессиональный и социальный контекст, в котором приходится работать и жить; - структура плана для решения задач, алгоритмы выполнения работ в профессиональной и смежных областях; - основные источники информации и ресурсы для решения задач и/или проблем в профессиональном и/или социальном контексте; - методы работы в профессиональной и смежных сферах; - порядок оценки результатов решения задач профессиональной деятельности; - номенклатура информационных источников, применяемых в профессиональной деятельности;

	- определять актуальность нормативно-правовой документации в профессиональной деятельности; - применять современную научную профессиональную терминологию; -определять и выстраивать траектории профессионального развития и самообразования; - выявлять достоинства и недостатки коммерческой идеи; - определять инвестиционную привлекательность коммерческих идей в рамках профессиональной деятельности, выявлять источники финансирования; - презентовать идеи открытия собственного дела в профессиональной деятельности; - определять источники достоверной правовой информации; - составлять различные правовые документы; - находить интересные проектные идеи, грамотно их формулировать и документировать; - оценивать жизнеспособность проектной идеи, составлять план проекта; − разрабатывать объекты баз данных − создавать таблицы, индексы, ограничения и другие объекты базы данных − оптимизировать запросы к базе данных для повышения производительности − разрабатывать хранимые процедуры и триггеры для баз данных; − разрабатывать необходимые для различных групп пользователей представления − разрабатывать модули программного обеспечения с использованием различных языков программирования и технологий; − применять паттерны проектирования и структуры данных для создания эффективных	- приемы структурирования информации; - формат оформления результатов поиска информации; - современные средства и устройства информатизации, порядок их применения; - программное обеспечение в профессиональной деятельности, в том числе цифровые средства; − основы реляционной модели данных − язык SQL и его основные команды − принципы нормализации баз данных − принципы работы с различными СУБД − общий подход к организации представлений, таблиц, индексов и кластеров; − методы организации целостности данных; − способы контроля доступа к данным и управления привилегиями; − 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

	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 производительность модулей, выявляя и устраняя узкие места; − проводить анализ и мониторинг производительности приложений; − применять инструменты для рефакторинга и оптимизации программного кода.	данными; − кэширование данных; − управление памятью; – техники повышения производительности программного обеспечения

2. СТРУКТУРА И СОДЕРЖАНИЕ ДИСЦИПЛИНЫ

2.1. Объем дисциплины и виды учебной работы

Вид учебной работы	**Объем часов (очная форма)**
Объем образовательной программы дисциплины	124
в том числе:	
лекции, уроки	42
практические занятия	66
курсовая работа (проект)	-
консультации	2
Самостоятельная работа	10
Промежуточная аттестация в форме дифференцированного зачета	4

2.2. Тематический план и содержание дисциплины

Наименование разделов и тем	Содержание учебного материала, лабораторные и практические работы, самостоятельная работа обучающихся, курсовая работа (проект)	Объем часов очной формы
1	2	3
Модуль 1. Современные информационные технологии и инструментарий разработчика		**48**
Тема 1.1. Профессиональная среда и инструменты разработчика	**Содержание учебного материала** Роль ИТ в ЖЦ ПО. Интеллектуальная собственность и лицензии ПО (Open Source, коммерческие). Портфолио и карьерный рост в IT.Стратегии поиска. Критическая оценка источников (официальная документация, Stack Overflow, профессиональные сообщества). Git: основы, ветвление, слияние. Работа с удаленными репозиториями (GitHub/GitLab). Основы проектного менеджмента в IT (Kanban, Issues, Pull Requests).	6
	Практические занятия Сравнительный анализ лицензий (MIT, GPL, Apache). Создание карты профессионального развития (Roadmap) и оформление профиля на GitHub.Решение типовой ошибки через целенаправленный поиск. Анализ и сравнение найденных решений. Работа с технической документацией API. Создание репозитория, ведение истории коммитов. Организация командной работы через Fork & Pull Request workflow. Ведение проекта через Issues.	6
	Самостоятельная работа обучающихся Работа над учебным материалом, ответы на контрольные вопросы. Систематическая проработка конспектов занятий, учебной и специальной литературы (по вопросам к параграфам, разделам учебных пособий, составленным преподавателем).	0
Тема 1.2. Проектирование и реализация структуры данных	**Содержание учебного материала** Выявление сущностей, атрибутов и связей. Построение ER-диаграмм (нотация «воронья лапка»). Преобразование концептуальной модели в логическую. Основы реляционной алгебры. Нормальные формы (1НФ, 2НФ, 3НФ). Цели и практический смысл нормализации/денормализации. Язык определения данных (DDL): CREATE, ALTER, DROP. Типы данных, ограничения целостности (PRIMARY KEY, FOREIGN KEY, CHECK). Создание индексов.	8

	Практические занятия Разработка ER-модели для заданного кейса (например, «Онлайн-библиотека»). Защита проекта модели. Преобразование ER-диаграммы в набор таблиц. Проведение нормализации заданной ненормализованной таблицы до 3НФ. Написание SQL-скрипта	10

	для создания всех таблиц схемы. Модификация структуры БД по новому требованию. Создание индексов для оптимизации.	
	Самостоятельная работа обучающихся Работа над учебным материалом, ответы на контрольные вопросы. Систематическая проработка конспектов занятий, учебной и специальной литературы (по вопросам к параграфам, разделам учебных пособий, составленным преподавателем).	0
Тема 1.3. Автоматизация и основы интеграции	**Содержание учебного материала** Принципы сильной связности и слабого зацепления. Модульное тестирование (Unit Testing): философия, структура AAA (Arrange-Act-Assert), библиотеки (pytest/JUnit). Концепции CI/CD. Понятие пайплайна. Инструменты (GitLab CI/CD, GitHub Actions). Автоматизация сборки и тестирования. Паттерны доступа к данным (DAO/Repository). Безопасность: параметризованные запросы против SQL-инъекций. Обработка ошибок соединения.	6
	Практические занятия Разработка изолированного модуля (например, валидатора) и написание для него полного набора модульных тестов. Настройка простого CI-пайплайна для проекта: этапы сборки (build) и запуска модульных тестов. Анализ отчетов. Разработка модуля «Доступ к данным» для одной из сущностей БД (например, «Клиент»). Реализация методов выборки и добавления записей. Написание интеграционного теста.	10
	Самостоятельная работа обучающихся Работа над учебным материалом, ответы на контрольные вопросы. Систематическая проработка конспектов занятий, учебной и специальной литературы (по вопросам к параграфам, разделам учебных пособий, составленным преподавателем).	2
Модуль 2. Разработка и размещение модулей программного обеспечения		**70**

Тема 2.1. Проектирование и разработка программных модулей	**Содержание учебного материала** Методы формализации требований. Создание технического задания (ТЗ) на модуль. Проектирование интерфейсов (API) модуля: входные/выходные данные, контракты. Архитектурные паттерны для бизнес-логики (Сервисный слой). Принцип единой ответственности (SRP). Работа с исключениями и валидация входных данных. Виды документации: кодовая (docstrings), техническая (API docs), пользовательская. Инструменты генерации (Sphinx, Doxygen). Основы code review.	8
	Практические занятия Разработка подробного ТЗ для модуля на основе пользовательской истории (User Story). Проектирование и описание его публичных методов. Разработка ядра модуля согласно	16
	ТЗ. Внедрение логирования ключевых событий. Оформление документации для разработанного модуля. Проведение попарного code review кода сокурсника по чеклисту.	
	Самостоятельная работа обучающихся Работа над учебным материалом, ответы на контрольные вопросы. Систематическая проработка конспектов занятий, учебной и специальной литературы (по вопросам к параграфам, разделам учебных пособий, составленным преподавателем).	0
Тема 2.2. Сборка, тестирование и отладка	**Содержание учебного материала** Понятие зависимости (dependency). Инструменты: Maven/Gradle (Java), pip + venv/virtualenv (Python), NuGet (.NET). Конфигурационные файлы. Пирамида тестирования. Интеграционные тесты: цели и отличия от модульных. Тестирование с использованием тестовой БД. Инструменты отладки (debugger) в IDE. Методика поиска ошибок: логирование, анализ стека вызовов (stack trace), контрольные точки. Основы профилирования.	10
	Практические занятия Настройка файла сборки для проекта (pom.xml, requirements.txt). Сборка исполняемого JAR-файла / виртуального окружения Python. Написание интеграционного теста для полного сценария работы модуля (например, «регистрация пользователя»). Использование фикстур для подготовки данных. Поиск и исправление преднамеренно внесенной в код сложной ошибки (логической или связанной с данными) с использованием отладчика и	14

	логов.	
	Самостоятельная работа обучающихся Работа над учебным материалом, ответы на контрольные вопросы. Систематическая проработка конспектов занятий, учебной и специальной литературы (по вопросам к параграфам, разделам учебных пособий, составленным преподавателем).	2
Тема 2.3 Размещение и анализ результатов	**Содержание учебного материала** Этап Deployment в CI/CD. Стратегии развертывания. Контейнеризация (Docker) как способ обеспечения переносимости. Облачные среды исполнения. Принципы презентации IT-проекта. Структура демонстрации: проблема, решение, архитектура, работающий результат. Метрики качества ПО. Принципы рефакторинга кода. Планирование итераций разработки.	4
	Практические занятия Интеграция всех компонентов (модуль бизнес-логики, модуль данных, БД) в единое консольное приложение. Подготовка скрипта запуска и демонстрационных данных.	10
	нализ кода итогового проекта. Планирование рефакторинга и новых функций (бэклог).	
	Самостоятельная работа обучающихся Работа над учебным материалом, ответы на контрольные вопросы. Систематическая проработка конспектов занятий, учебной и специальной литературы (по вопросам к параграфам, разделам учебных пособий, составленным преподавателем).	6
Промежуточная аттестация в виде дифференцированного зачета		**4**
Всего:		**122**

3. УСЛОВИЯ РЕАЛИЗАЦИИ РАБОЧЕЙ ПРОГРАММЫ ДИСЦИПЛИНЫ

3.1. Требования к минимальному материально-техническому обеспечению

Реализация дисциплины производится в учебном кабинете, предназначенном для проведения теоретических и практических занятий.

Оборудование учебного кабинета: рабочие места обучающихся; рабочее место преподавателя; технические средства обучения: мультимедийный комплекс (проектор, экран), компьютер, колонки; оборудование: камера видеонаблюдения; доска меловая. Помещение для самостоятельной и воспитательной работы: столы, стулья, ПК обучающихся, программное обеспечение: 1С Предприятие, OnlyOffice (AGPLv3), GIMP (GNU GPL 3), Krita (GNU GPL 3), Inkscape (GNU GPL 3),
SciLab (CeCILL), Kaspersky Endpoint Security, Агент администрирования Kaspersky
Security Center, Mozilla Firefox (MPL 2), Apache NetBeans IDE (Apache License 2), Notepadqq (GNU GPL 3), VSCodium (MIT), Mono Development IDE (MIT), PyCharm Community Edition (Apache License 2), Android Studio (Apache License 2), Python (Python Software Foundation License), Java (GNU GPL), Node.js (MIT), Git (GNU GPL 2), GitHub
Desktop (No Copyright), UnityHub (Unity Companion License), FreeCAD (LGPL-2.0-or-later), Sweet Home 3D (GNU GPL 2+), draw.io (Apache License 2), DBeaver (Apache License 2), PostgreSQL (License (free and open-source)), MariaDB (GNU GPL), PSPP (GNU GPL 3),
Wireshark (GNU GPL 2+), Imunes (Creative Commons Attribution 4.0 International Public License), VirtualBox (GNU GPL 2), Apache (Apache License 2), Консультант+
(Коммерческая лицензия), OBS (GNU GPL).

3.2. Информационное обеспечение реализации программы

Для реализации программы библиотечный фонд образовательной организации должен иметь электронные образовательные и информационные ресурсы, рекомендуемых для использования в образовательном процессе.

Основная учебная литература

1. Шестернинов, Е. Е. Индивидуальный проект. Шаг в профессию. Базовый уровень. Практикум: учебное пособие, разработанное в комплекте с учебником для образовательных организаций, реализующих образовательные программы среднего профессионального образования / Е. Е. Шестернинов. – 2-е изд., стер. - Москва: Просвещение, 2025. - 82 с. – (Учебник СПО). - ISBN 978-5-09-124912-5. - Текст: электронный. - URL: https://znanium.ru/catalog/product/2202038

2. Чернышев, С. А. Принципы, паттерны и методологии разработки программного обеспечения: учебник для среднего профессионального образования / С. А. Чернышев. — Москва: Издательство Юрайт, 2026. — 176 с. — (Профессиональное образование). —
ISBN 978-5-534-18705-2. — Текст: электронный // Образовательная платформа Юрайт [сайт]. — URL: https://urait.ru/bcode/589664

3. Стружкин, Н. П. Базы данных: проектирование: учебник для среднего профессионального образования / Н. П. Стружкин, В. В. Годин. — Москва: Издательство Юрайт, 2026. — 477 с. — (Профессиональное образование). — ISBN 978-5-534-11635-9. — Текст: электронный // Образовательная платформа Юрайт [сайт]. — URL: https://urait.ru/bcode/587735

4. Советов, Б. Я. Информационные технологии: учебник для среднего профессионального образования / Б. Я. Советов, В. В. Цехановский. — 8-е изд., перераб. и доп. — Москва: Издательство Юрайт, 2026. — 414 с. — (Профессиональное образование). — ISBN 978-5-534-20053-9. — Текст: электронный // Образовательная платформа Юрайт [сайт]. — URL: https://urait.ru/bcode/583524

Дополнительная учебная литература

1. Гниденко, И. Г. Технология разработки программного обеспечения: учебник для среднего профессионального образования / И. Г. Гниденко, Ф. Ф. Павлов, Д. Ю. Федоров. — 2-е изд., перераб. и доп. — Москва: Издательство Юрайт, 2026. — 248 с. — (Профессиональное образование). — ISBN 978-5-534-18131-9. — Текст: электронный // Образовательная платформа Юрайт [сайт]. — URL: https://urait.ru/bcode/585518

2. Илюшечкин, В. М. Основы использования и проектирования баз данных: учебник для среднего профессионального образования / В. М. Илюшечкин. — Москва: Издательство Юрайт, 2026. — 213 с. — (Профессиональное образование). — ISBN 978-5-534-01283-5. — Текст : электронный // Образовательная платформа Юрайт [сайт]. — URL: https://urait.ru/bcode/585059

3. Маркин, А. В. Базы данных. PostgreSQL: учебник для среднего профессионального образования / А. В. Маркин. — Москва: Издательство Юрайт, 2026. — 828 с. — (Профессиональное образование). — ISBN 978-5-534-21780-3. — Текст: электронный // Образовательная платформа Юрайт [сайт]. — URL: https://urait.ru/bcode/590497

Базы данных и/или Интернет-ресурсы

1. https://postgrespro.ru/docs/postgresql - Полный и актуальный перевод официальной документации PostgreSQL
2. https://ruvds.com/ - Образовательный проект от облачного провайдера.
3. https://www.garant.ru/ - Информационно-правовая система Гарант
4. https://www.consultant.ru/ - Справочная правовая система «КонсультантПлюс»
5. https://www.iso.org/standards.html - Официальный сайт ISO, каталог и поиск международных стандартов
6. https://www.iec.ch/publications/international-standards - Официальный сайт IEC, международные стандарты IEC

Официальные, справочно-библиографические и периодические издания

1. Информационные технологии и вычислительные системы: научный журнал / гл.
ред. Попков Ю.С.; учред. ФГУ «Федеральный исследовательский центр «Информатика и управление» РАН». - М., 2026
2. Экономика и математические методы: журнал / гл. ред. В.Л. Макаров / Российская Академия наук, Центральный экономико-математический институт, Институт проблем рынка - Российская академия наук, Москва, 2026
3. Экономический журнал Высшей школы экономики / Higher School of Economics Economic Journal: журнал / гл. ред. Е.Е. Гавриленков / Национальный исследовательский университет «Высшая школа экономики» - Издательский дом Высшей школы экономики, Москва, 2026

4. КОНТРОЛЬ И ОЦЕНКА РЕЗУЛЬТАТОВ ОСВОЕНИЯ ДИСЦИПЛИНЫ

Результаты обучения	Критерии оценки	Методы оценки
Перечень знаний, осваиваемых в рамках дисциплины		
- тренды и стандарты современной IT-индустрии. - модели жизненного цикла программного обеспечения - правовые аспекты: интеллектуальная собственность, авторское право, виды лицензий на ПО - роли в IT-команде - карьерные треки и грейды - основы предпринимательской деятельности и финансовой грамотности в IT; - принципы декомпозиции сложных задач на подзадачи. - использование систем управления задачами для планирования и трекинга прогресса. - рабочие процессы в разработке - алгоритм решения технической проблемы: от воспроизведения и изоляции до поиска решения и верификации. - иерархия доверия к источникам информации; - специализированные платформы: Stack Overflow, GitHub, профессиональные форумы и сообщества; - стратегии эффективного поиска: формирование запросов на английском языке, использование операторов	Ответы на вопросы на знание и понимание: 85 - 100% правильных ответов – «отлично» 69-84% правильных ответов – «хорошо» 51-68% правильных ответов – «удовлетворительно» 50% и менее – «неудовлетворительно»	Устный опрос. Тестирование по темам дисциплины. Практические занятия. Практические занятия по заполнению документов.

расширенного поиска; принципы командной разработки и code review; - основы проектного менеджмента в IT; - методологии разработки ПО и их применение на практике; - критерии качества кода; - использование модульного тестирования для верификации функциональности; - анализ соответствия результата исходным техническим требованиям; - создание ментальных карт и блок-схем для проектирования. - разработка технического задания (ТЗ) и пользовательских историй; - ведение личной базы знаний с использованием Markdown, Notion, Obsidian; - назначение и принципы работы CI/CD-серверов; - основы контейнеризации и облачных платформ; - работа с удаленными серверами и виртуальными машинами; - комплексный подход: от оптимизации запросов к БД и настройки индексов до рефакторинга бизнес-логики и введения многоуровневого кэширования; - принципы производительности баз данных; - базовые техники нагрузочного тестирования и анализа результатов; - понимание модели OSI/TCP-IP;		

- принципы работы HTTP/HTTPS: методы, заголовки, кэширование; - влияние сетевых задержек и оптимизация размера передаваемых данных; - анализ алгоритмической сложности; - выбор оптимальной структуры данных для конкретной задачи; - практическое применение алгоритмов поиска и сортировки; - отладчики в IDE, работа с точками останова (breakpoints), пошаговое выполнение;		

, Jira, GitHub Issues) для планирования и трекинга прогресса; - рабочие процессы (workflow) в разработке: Git Flow, GitHub Flow; - алгоритм решения технической проблемы: от воспроизведения и изоляции до поиска решения и верификации. - инструменты статического анализа кода; - системы управления зависимостями; - концепция API (Application Programming Interface); - основы архитектурного стиля REST (ресурсы, HTTPметоды, статус-коды); - создание модулей с четко определенным программным интерфейсом; - интегрированные среды разработки (IDE) и их расширенные возможности. - системы контроля версий (Git), включая графические клиенты и CLI; - инструменты для проектирования БД (pgAdmin, DBeaver, draw.io для ER-диаграмм); - программное обеспечение для автоматизации (CI/CD), тестирования, мониторинга; - создание ментальных карт и блок-схем для проектирования; - разработка технического задания (ТЗ) и пользовательских историй (User Stories);		

- ведение личной базы знаний (Knowledge Base) с использованием Markdown, Notion, Obsidian; - принципы декомпозиции сложных задач на подзадачи; - использование систем управления задачами (Trello,		

Перечень умений, осваиваемых в рамках дисциплины

-анализировать правовые последствия использования различных типов лицензионного ПО в проектах. -составлять личный план профессионального развития на среднесрочную перспективу. -рассчитывать экономическую эффективность решения для простых проектов.; -декомпозировать сложную техническую задачу на последовательность простых, реализуемых этапов; -формализовывать требования к программному модулю в виде технического задания (ТЗ) или пользовательских историй; -планировать свою работу с использованием систем управления задачами (создавать, оценивать, назначать, отслеживать задачи); -формулировать точные поисковые запросы на русском и английском языках для решения технических проблем; -критически оценивать найденные решения по критериям актуальности, авторитетности источника и релевантности; -эффективно использовать официальную документацию по технологиям; -работать в команде над общим кодом, используя систему контроля версий Git ; -проводить и участвовать в процедуре code review, используя установленные критерии качества кода; -применять базовые практики гибкой методологии	Ответы на вопросы на знание и понимание: выполнено полностью, без критических ошибок, обоснования корректны, оформление соответствует требованиям– «отлично» выполнено в целом верно, есть 1–2 некритичные ошибки или недочёта в оформлении – «хорошо» выполнено частично, есть ошибки в логике или оформлении, требуется доработка по замечаниям – «удовлетворительно» задание не выполнено или содержит существенные ошибки, отсутствуют ключевые элементы – «неудовлетворительно»	Проверка выполнения практических работ по темам модуля. Оценка оформленных документов по чеклисту качества. Защита практической работы и ответы на уточняющие вопросы. Проверка самостоятельной работы и мини-отчётов по источникам.

(Agile/Scrum) в учебном проекте (спринты, стендапы, backlog);		

-проводить самооценку выполненной работы на соответствие исходным техническим требованиям; -анализировать отчеты о тестовом покрытии (code coverage) для оценки качества разработанного кода; - формулировать аргументированные выводы по итогам выполнения проекта; -выбирать адекватный тип информационного источника для решения конкретной учебной или профессиональной задачи; -создавать ER-диаграммы предметной области в нотации «воронья лапка»; -составлять структурированную документацию к проекту с использованием языка разметки Markdown; - систематизировать полученные знания в личной базе (конспекты, сниппеты кода); -оформлять результаты поиска и анализа в виде краткого отчета с выводами и ссылками на источники; -вести содержательную историю коммитов в Git, следуя соглашению Conventional Commits; -настраивать и выполнять простой CI/CD пайплайн для автоматической сборки и тестирования проекта; - создавать базовый Dockerобраз для развертывания учебного приложения.		

ФОНД ОЦЕНОЧНЫХ СРЕДСТВ

Типовые контрольные задания или иные материалы, необходимые для оценки результатов обучения по учебной дисциплине

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

ЛИСТ АКТУАЛИЗАЦИИ РАБОЧЕЙ ПРОГРАММЫ ДИСЦИПЛИНЫ

МЕТОДИЧЕСКИЕ УКАЗАНИЯ ПО ОСВОЕНИЮ РАБОЧЕЙ ПРОГРАММЫ ДИСЦИПЛИНЫ

Методические рекомендации при подготовке к теоретическим занятиям

Основными видами аудиторной работы обучающихся являются лекции.

В ходе теоретических занятий преподаватель излагает и разъясняет основные, наиболее сложные понятия темы, а также связанные с ней теоретические и практические проблемы, дает рекомендации и указания на самостоятельную работу.

В ходе теоретических занятий необходимо вести конспектирование учебного материала. Обращать внимание на категории, формулировки, раскрывающие содержание тех или иных явлений и процессов, научные выводы и практические рекомендации. Желательно оставить в рабочих конспектах поля, на которых делать пометки из рекомендованной литературы, дополняющие материал прослушанной лекции, а также подчеркивающие особую важность тех или иных теоретических положений. Задавать преподавателю уточняющие вопросы с целью уяснения теоретических положений, разрешения спорных ситуаций.

Контроль ведения конспекта лекций

1. Конспект лекций обучающегося должен вестись в отдельной тетради с количеством страниц, рекомендованном ведущим лектором. На обороте титульного листа тетрадь с конспектом лекций должна содержать подписи:

- фамилия и инициалы обучающегося;
- номер группы;
- ФИО ведущих учебный предмет преподавателей.

2. В начале каждой лекции обучающийся должен записать тему и перечень рассматриваемых вопросов, а также перечень рекомендуемой литературы (при условии, что они оговариваются лектором).

3. При записи содержания лекции следует применять минимум сокращений, желательно общеупотребимых. Обучающимся может быть разработана своя система сокращений, с выносом ее в конце конспекта для пояснения.

4. Табличный, графический, цифровой материал должен быть точно перенесен в конспект с доски (проектора) без искажений динамики зависимостей и обозначений формул, а также значений результатов расчетов.

5. Тетрадь с конспектом должна содержать полный объем прочитанного материала лекций по их количеству и полноте изложения, включая материал для самостоятельного изучения и конспектирования.

Записи должны быть сделаны аккуратным почерком, разборчиво.

Методические рекомендации при проведении практических занятий

Практические занятия необходимы для закрепления изученного теоретического материала, контроля преподавателем степени подготовленности обучающихся по изучаемым учебным предметом, дисциплине или профессиональному модулю.

Перед практическим занятием следует изучить конспект лекции и рекомендованную преподавателем литературу, обращая внимание на практическое применение теории и на методику решения типовых задач. Логическая связь теоретических и практических занятий заключается в том, что информация, полученная на теоретических занятиях, в процессе

самостоятельной работы на практическом занятии осмысливается и перерабатывается, при помощи преподавателя анализируется, после чего усваивается.

Методические рекомендации обучающимся для самостоятельной работы

Самостоятельная работа - это планируемая работа обучающихся, выполняемая по заданию и при методическом руководстве преподавателя, но без его непосредственного участия.

Самостоятельная работа выполняет ряд функций, среди которых особенно выделяются:

- ✓ развивающая (повышение культуры умственного труда, приобщение к творческим видам деятельности, обогащение интеллектуальных способностей, обучающихся);
- ✓ ориентирующая и стимулирующая (процессу обучения придается ускорение и мотивация);
- ✓ воспитательная (формируются и развиваются профессиональные качества специалиста);
- ✓ исследовательская (новый уровень профессионально-творческого мышления);
- ✓ информационно-обучающая (учебная деятельность обучающихся на аудиторных занятиях).

Целью самостоятельных занятий является самостоятельное более глубокое изучение обучающимися отдельных вопросов курса с использованием рекомендуемой дополнительной литературы и других информационных источников.

Задачами самостоятельной работы обучающихся являются:

- ✓ систематизация и закрепление полученных теоретических знаний и практических умений, обучающихся;
- ✓ углубление и расширение теоретических знаний;
- ✓ формирование умения использовать справочную литературу;
- ✓ развитие познавательных способностей и активности обучающихся:

творческой инициативы, самостоятельности, ответственности и организованности;

Внеаудиторная самостоятельная работа включает такие формы работы, как:

Индивидуальные занятия (домашние занятия):

- ✓ изучение программного материала дисциплины (работа с учебником и конспектом лекции);
- ✓ изучение рекомендуемых литературных источников;
- ✓ конспектирование источников;
- ✓ выполнение контрольных работ;
- ✓ работа со словарями и справочниками;
- ✓ использование аудио- и видеозаписи;
- ✓ работа с электронными информационными ресурсами и ресурсами Internet;
- ✓ составление плана и тезисов ответа на лекции;
- ✓ составление схем, таблиц, для систематизации учебного материала;
- ✓ выполнение тестовых заданий;
- ✓ решение задач;

✓ подготовка презентаций;

✓ ответы на контрольные вопросы;

✓ аннотирование, реферирование, рецензирование текста;

✓ написание эссе, тезисов, докладов, рефератов;

✓ составление глоссария, кроссворда по темам дисциплины; ✓ работа с компьютерными программами;

✓ подготовка к дифференцированному зачету.

Групповая самостоятельная работа обучающихся:

✓ подготовка к занятиям, проводимым с использованием активных форм обучения (круглые столы, деловые игры);

✓ анализ деловых ситуаций (мини-кейсов) и др; ✓ участие в Интернет - конференциях.

Получение консультаций для разъяснений по вопросам изучаемого учебного предмета.

Но наиболее часто при изучении учебного предмета используются следующие виды самостоятельной работы обучающихся:

✓ поиск (подбор) литературы (в том числе электронных источников информации) по заданной теме; ✓

подготовка рефератов; ✓

подготовка докладов.

Реферат - форма письменной работы. Как правило, реферат является самостоятельным библиографическим исследованием обучающегося, носящим описательно-аналитический характер. Объем реферата может достигать 10-15 страниц. Подготовка реферата подразумевает самостоятельное изучение обучающимся нескольких литературных источников (монографий, научных статей и т.д.) по теме, не рассматриваемой подробно на лекции, систематизацию материала и краткое его изложение. Цель написания реферата - привитие навыков краткого и лаконичного представления собранных материалов и фактов в соответствии с требованиями, предъявляемыми к научным отчетам, обзорам и статьям.

Работа обучающегося над рефератом включает следующие операции:

✓ уяснение сути темы (по её названию);

✓ выявление литературных источников по данной теме;

✓ ознакомление с содержанием источников, направленным на осмысление его внешней и внутренней структуры;

✓ работа над выделением главных смысловых компонентов текста (ключевых слов и предложений);

✓ отбор наиболее важных сведений из выделенных фрагментов;

✓ составление «связок» из отобранного материала в соответствии с логикой изложения фактов;

✓ написание реферата.

✓ Реферат в структурном отношении должен включать:

✓ заголовочную часть (введение);

✓ собственно, реферативную часть (изложение основных положений по плану с соответствующими названиями и нумерацией);

✓ заключительную часть (выводы);

✓ справочную часть (список использованной литературы); ✓ оглавление (содержание).

Текст реферата должен иметь связанное, цельное построение. Содержание реферата должна пронизывать главная идея, которая объединяет все информационные элементы источников, а также приводимые факты. При этом допускается сохранение отдельных структурных элементов оригинала. Однако не допустимо простое переписывание положение литературных источников. Язык должен быть кратким, ясным, доступным.

Оформление реферата должно быть выполнено машинописным способом в соответствии со следующими требованиями: **печатный шрифт** – Times New Roman, кегль (размер) 14; **листы бумаги** – формата А4, все страницы должны быть пронумерованы, с размером полей по 2 см; **междустрочный интервал** – полуторный; **абзац –** с отступом первой строки 1,25 см; **текст –** должен быть **выровнен по ширине** и **структурирован** по главам (разделам, параграфам), сопровождаться ссылками на источники при использовании прямых (в кавычках) цитат и фрагментов опубликованных текстов, т.е. на такое использование источников должно быть явным образом указано в сносках или за текстовом списке литературы в соответствии с требованиями ГОСТа.

Реферат оценивается исходя из следующих критериев:

✓ поставлена ли цель в работе;

✓ сумел ли обучающийся самостоятельно составить логический план к теме и реализовать его;

✓ каков научный уровень реферата;

✓ собран ли достаточный фактический материал;

✓ удалось ли раскрыть тему;

✓ показана ли связь рассматриваемой темы с современными проблемами науки и общества, со специальностью обучающегося;

✓ каков авторский вклад в систематизацию, структурирование материала, в составлении заключения;

✓ достигнута ли цель работы.

Тематика рефератов приводится в рамках тем лекционных занятий. Возможно написание реферата по теме, предложенной самим обучающимся (по согласованию с преподавателем). Реферат на занятиях (научно-практических конференциях) может быть представлен в виде доклада.

Реферат должен быть выполнен за один месяц до начала экзаменационной сессии. Обучающиеся, не представившие в установленный срок реферат, либо получившие оценку «неудовлетворительно», к сдаче экзамена не допускаются.

Углубленному изучению материала по отдельным разделам учебного предмета способствует подготовка обучающимися докладов. Доклад - это самостоятельная работа, анализирующая и обобщающая публикации по заданной тематике, предполагающая выработку и обоснование собственной позиции автора в отношении рассматриваемых вопросов. Подготовка доклада - достаточно кропотливый труд. Его написанию предшествует изучение широкого круга экономических первоисточников, монографий, статей, обобщение личных наблюдений. Работа над докладом способствует развитию самостоятельного, творческого мышления, учит применять экономические знания на практике при анализе актуальных социальных и правовых проблем. Рекомендуемое время доклада - 10-12 минут.

Непременным условием закрепления теоретических и практических знаний, обучающихся является выполнение ими домашних заданий. Одним из основных видов деятельности обучающегося является самостоятельная работа, которая включает в себя изучение лекционного материала, учебников и учебных пособий, первоисточников, подготовку сообщений, выступления на групповых занятиях, выполнение заданий преподавателя.

Методические рекомендации обучающимся по изучению рекомендованной литературы

Самостоятельная работа обучающихся включает в себя выполнение различного рода заданий, которые ориентированы на более глубокое усвоение материала изучаемого учебного предмета. По каждой теме учебного предмета обучающимся предлагается перечень заданий для самостоятельной работы.

К выполнению заданий для самостоятельной работы предъявляются следующие требования: задания должны исполняться самостоятельно и представляться в установленный срок, а также соответствовать установленным требованиям по оформлению.

Обучающимся следует:

- выполнять все плановые задания, выдаваемые преподавателем для

самостоятельного выполнения, и разбирать на консультациях неясные вопросы;

- при подготовке к экзамену параллельно прорабатывать соответствующие теоретические и практические разделы учебного предмета, фиксируя неясные моменты для их обсуждения на плановой консультации.

Для подготовки к занятиям, текущему контролю и промежуточной аттестации обучающиеся могут воспользоваться электронной библиотекой ВУЗа, расположенной по электронному адресу https://e.muiv.ru/local/library/index.php, где они имеют возможность получить доступ к учебно-методическим материалам как библиотеки ВУЗа, так и иных электронных библиотечных систем. В свою очередь, обучающиеся могут взять на дом необходимую литературу на абонементе в библиотеке, а также воспользоваться читальным залом.

Методические рекомендации по балльно-рейтинговой системе учета и оценки достижений обучающихся

Общие положения

- Балльно-рейтинговая система учета и оценки достижений, обучающихся (далее – БРС) в Частном образовательном учреждении высшего образования «Московский университет имени А. С. Грибоедова» (далее - Университет) предназначена для решения следующих задач:

- стимулирования обучающихся к регулярной учебной и внеучебной деятельности;

- организации непрерывного контроля освоения обучающимися элементов основных профессиональных образовательных программ среднего профессионального образования;

- комплексной оценки результативности учебной и внеучебной деятельности обучающихся в течение семестра;

- создания для преподавателей, руководства факультетов и кафедр надежного инструментария для осуществления систематического и регулярного контроля за учебной и внеучебной деятельностью обучающихся.

БРС является обязательной для всех участников образовательного процесса и реализуется в электронной информационно-образовательной среде «Электронный университет» (далее – ЭИОС) для всех форм и технологий обучения.

БРС обеспечивает рейтинговый учет и оценку достижений обучающегося за семестр и включает в себя учебный (далее – Rуд) и внеучебный (далее – Rвр) рейтинги.

Учебный рейтинг обучающихся

Перечень дисциплин к изучению в семестре Вы можете увидеть в Личном кабинете в ЭИОС «Электронный Университет» (далее Личный кабинет). Чтобы попасть в Личный кабинет, Вам нужно перейти по ссылке https://impe.lms.mirapolis.ru/mira/ и ввести свой логин и пароль, выданный при поступлении в Университет.

Ваш учебный рейтинг по дисциплине будет формироваться на основе результатов выполнения следующих мероприятий:

- прохождение рубежного контроля (тестирования) по результатам изучения каждого модуля (раздела) дисциплины;

- выполнение рейтинговой работы (домашней творческой работы, расчетноаналитического задания, реферата, контрольной работы, группового проекта) при ее наличии.

Доступ к тестовым заданиям по всем модулям изучаемых дисциплин (рубежному контролю) обеспечивается с начала семестра, однако проходить тестирование необходимо только по окончании изучения каждого модуля (раздела) дисциплины.

Рейтинговая работа должна быть выполнена и передана на проверку преподавателю не позднее, чем за 2 недели до начала экзаменационной сессии через Личный кабинет. Выполнение обучающимся рейтинговой работы допускается не более одного раза в течение установленного календарным учебным графиком периода. Представленная на проверку работа исправлению не подлежит.

При возникновении вопросов по процедуре тестирования, некорректного отображения контента, иных вопросов, связанных с технической поддержкой системы тестирования, обращайтесь в Центр поддержки.

Для прохождения тестирования Вам предоставляются две попытки с ограничением по времени (45 минут). Выполнение тестирования допускается не более 2-ух раз в течение установленного календарным учебным графиком периода. Как итоговый учитывается Ваш лучший результат из двух попыток.

Допуск обучающегося к промежуточной аттестации осуществляется с любым количеством баллов, набранных в ходе выполнения Рейтинговых мероприятий по учебной дисциплине, в том числе с количеством баллов – 0.

Если по результатам выполнения всех рейтинговых мероприятий (тестирования по всем модулям дисциплины и выполнения рейтинговой работы) Вы набрали средний балл - 75 баллов и более, Вы можете претендовать на оценку «хорошо», «отлично» или «зачтено» по итогам работы в семестре без прохождения промежуточной аттестации (т.е. «автоматом»).

Средний балл рассчитывается как сумма баллов за все элементы рейтингового контроля (тестирование по каждому модулю, выполнение рейтинговой работы)

разделенная на количество элементов контроля. При этом оценка определяется в соответствии со шкалой:

75-84 балла – хорошо, 85-100 баллов – отлично, 75-100 – зачтено.

Если средний балл менее 75, или Вы претендуете на более высокую оценку, то Вы сдаете зачет/экзамен по дисциплине на общих основаниях.

Промежуточная аттестация **для обучающихся по очной, очно-заочной и заочной формам обучения с применением дистанционных образовательных технологий** осуществляется в форме проведения и/или устных зачетов и выполнения дисциплине.	Промежуточная аттестация **для обучающихся с применением заочной** осуществляется в форме письменных итогового тестирования по экзаменов.

Если студент, обучающийся с применением ЭИОС, претендует на более высокую оценку, чем «автомат», то для открытия итогового тестирования необходимо обратиться в сервис «Центр поддержки».

Для выполнения итогового тестирования предоставляются также две попытки.

Обращаем Ваше внимание на то, что если в течение семестра Вы оформляли отсрочку оплаты обучения, то за две недели до начала зачетно-экзаменационной сессии финансовая задолженность должна быть погашена. В противном случае у Вас не будет возможности получить оценки по дисциплинам «автоматом» или пройти промежуточную аттестацию.

Внеучебный рейтинг обучающихся

Внеучебный рейтинг применяется для активизации обучающихся к участию в творческой, спортивной, общественной и научно-исследовательской деятельности Университета и является составной частью портфолио обучающегося, складывается из баллов, полученных обучающимися за участие в различных внеучебных мероприятиях. Подтверждение суммы баллов за внеучебный рейтинг осуществляется деканами на основе анализа представленных Вами документов (сертификатов, грамот, свидетельств, благодарностей и иных материалов) и заполненного Вами в Личном кабинете портфолио.

Ликвидация академической задолженности

Если в ходе промежуточной аттестации по учебной дисциплине Вы получили неудовлетворительную оценку или не явились на зачет/экзамен (не прошли итоговое тестирование), то у Вас образуется академическая задолженность.

Все набранные ранее в течение семестра по результатам выполнения рейтинговых мероприятий баллы обнуляются.

Ликвидировать академическую задолженность Вы должны в течение года, следующего за семестром образования академической задолженности (но не более двух раз).

Ликвидация академической задолженности проходит в форме выполнения итогового теста. Шкала оценивания при ликвидации академической задолженности:

при форме контроля по дисциплине – **экзамен/ зачет с оценкой: зачет:**

- «неудовлетворительно» – 49 баллов и
- «удовлетворительно» – от 50 до 65
- «хорошо» – от 66 до 84 баллов;

при форме контроля по дисциплине –

«не зачтено» – 49 баллов и менее; менее;
«зачтено» – от 50 до 100 баллов баллов;
«отлично» – от 85 до 100 баллов

Пройти итоговое тестирование Вы можете не более двух раз в течение установленного календарным графиком периода. Учитывается лучший результат, полученный Вами в ходе выполнения итогового тестирования. Для выполнения одного сеанса тестирования устанавливается время – один академический час (45 минут).