Документ подписан простой электронной подписью
Информация о владельце:
ФИО: Гриб Владислав Валерьевич
Должность: Ректор
Дата подписания: 06.11.2026 16:50:38
Уникальный программный ключ:
637517d24e103c3db032acf37e839d08ec1e5bb2f5eb89c29abfcd7f43985447

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
Основы алгоритмизации и программирования

Специальность
09.02.11 Разработка и управление программным обеспечением

Квалификация выпускника:
Программист

на базе среднего общего образования

Форма обучения: очная

Москва

Рабочая программа дисциплины «Основы алгоритмизации и программирования». 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 «Основы алгоритмизации и программирования» 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 «Основы алгоритмизации и программирования» является обязательной частью общепрофессионального цикла основной профессиональной образовательной программы в соответствии с ФГОС по специальности 09.02.11 Разработка и управление программным обеспечением.

Дисциплина «Основы алгоритмизации и программирования» 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 01. Выбирать способы решения задач профессиональной деятельности применительно к различным контекстам;

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

ПК 2.1. Проектировать модули программного обеспечения;

ПК 2.2. Разрабатывать модули программного обеспечения;

ПК 2.4. Выполнять тестирование и отладку программного обеспечения;

ПК 2.5. Осуществлять документирование программных модулей программного обеспечения.

В рамках программы дисциплины обучающимися осваиваются умения и знания:

Код ПК, ОК	Умения	Знания

ОК 01. ОК 02. ПК 2.1. ПК 2.2. ПК 2.4. ПК 2.5.	- разрабатывать алгоритмы для конкретных задач; - использовать программы для графического отображения алгоритмов; - определять сложность работы алгоритмов; - настраивать среду разработки для программирования на Python; - разбираться в основных типах данных и понимать, какой тип данных использовать для хранения и обработки значений, используемых в программе; - реализовывать ветвления и циклы; - работать со строками, списками, словарями; - осуществлять чтение и запись в файл средствами Python;	- понятие алгоритмизации, свойства алгоритмов, общие принципы построения алгоритмов, основные алгоритмические конструкции; - эволюцию языков программирования, их классификацию, понятие системы программирования; -основной язык программирования Python; - способы необходимого программного обеспечения, а также настройки разработки для программирования на Python; - понятия переменных и операторов, функций; - основные типы данных в Python;
	- формализовывать задачи и реализовывать их в виде программ на Python.	- способы реализации ветвлений и циклов в программах; - способы и методы для работы со строками, списками, словарями и файлами; - синтаксис написания собственных функций; - способы формализации задач в виде программ; - основные алгоритмы сортировки, работы со строками.

2. СТРУКТУРА И СОДЕРЖАНИЕ ДИСЦИПЛИНЫ

2.1. Объем дисциплины и виды учебной работы

Вид учебной работы	**Объем часов (очная форма)**
Объем образовательной программы дисциплины	218
в том числе:	
лекции, уроки	84
практические занятия	94
курсовая работа (проект)	-
Консультации	2
Самостоятельная работа	30
Промежуточная аттестация в форме экзамена	8

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. Основы алгоритмизации.		**30**
Тема 1.1 Понятие алгоритма и его свойства	**Содержание учебного материала:** Понятие алгоритма. Свойства и виды алгоритмов. Способы описания алгоритмов: псевдокоды, блок-схема, программа. Блок-схема: основные элементы, правила составления. Стандарты графического оформления алгоритмов. Базовые алгоритмические конструкции: линейная, разветвляющаяся, циклическая. Критерии «хорошего» алгоритма.	4
	Практические занятия: Составление и оформление блок-схем простых алгоритмов	4
	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 1.2 Методы разработки алгоритмов	**Содержание учебного материала:** Основные методы и этапы проектирования алгоритмов: постановка задачи, математическое описание – математическая модель. Нисходящее, модульное и восходящее проектирование. Эффективность и сложность алгоритма, их практическая значимость. Алгоритмы поиска. Алгоритмы сортировки. Вложенные циклы. Вспомогательные алгоритмы. Различные комбинации алгоритмических конструкций. Тестовые данные. Алгоритм Евклида. Декомпозиция алгоритма.	10

	Базовая оценка сложности O(1), O(n), O(n²).	
	Практические занятия Проектирование и оформление алгоритмов сортировки. Проектирование и оформление алгоритмов поиска.	8
	Проектирование и оформление сложных алгоритмов	
	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. Введение в программирование.		**38**
Тема 2.1. Языки программирования	**Содержание учебного материала** Развитие языков программирования. Обзор языков программирования. Области применения языков программирования. Стандарты языков программирования. Среда проектирования. Компиляторы и интерпретаторы. Жизненный цикл информационной системы. Программа. Программный продукт и его характеристики. Основные этапы решения задач на компьютере.	6
	Практические занятия IDE. IDLE. VENV. Утилита pip. Основные свойства компилятора. Командная строка и интерпретатор. Основные свойства интерпретатора. Стандарт PEP8.	12
	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 2.2. Переменные	**Содержание учебного материала** Именование переменных. Типы данных. Присваивание значения переменным. Проверка типа данных. Преобразование типов данных. Удаление переменной.	4
	Практические занятия Операции с целыми числами. Операции с вещественными числами. Операции с логическими типами данных. Преобразование переменных.	12
	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 3. Операторы языка программирования Python		**24**
Тема 3.1. Основные операторы	**Содержание учебного материала** Математические операторы. Двоичные операторы. Операторы для работы с последовательностями. Операторы присваивания. Приоритет выполнения операторов.	2
	Практические занятия Составление программ линейной структуры.	4
	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 3.2. Условные операторы и циклы	**Содержание учебного материала** Операторы сравнения. Оператор ветвления if…else. Цикл for. Функции range(), enumerate(). Цикл while. Оператор continue. Оператор break.	4

	Практические занятия Алгоритмические задачи со счётными циклами. Составление программ разветвляющейся структуры. Составление программ с оператором выбора. Составление программ циклической структуры.	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 4. Структуры данных и ООП.		**116**
Тема 4.1. **Числа.** **Строки и двоичные данные**	**Содержание учебного материала** Встроенные функции и методы для работы с числами. Математические функции. Модуль math. Модуль random. Генерация случайных чисел. Создание строки. Специальные символы. Операции над строками. Форматирование строк. Метод format(). Функции и методы для работы со строками. Настройка локали. Изменение регистра символов. Поиск и замена в строке. Вычисление выражений, заданных в виде строк. Тип данных bytes и bytearray. Преобразование объекта в последовательность байтов. Шифрование строк. Синтаксис регулярных выражений. Поиск первого совпадения с шаблоном. Поиск всех совпадений с шаблоном. Замена в	8

	строке.	
	Практические занятия Представление вещественных чисел в программе. Особенности работы с вещественными числами. Алгоритмы с заданной точностью.	8
	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 4.2. Структуризация в программировании	**Содержание учебного материала** Основы структурного программирования. Методы структурного программирования.	2
	Практические занятия Контроль структуры программы.	4
	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 4.3. Списки. Кортежи. Множества и диапазоны. Словари	**Содержание учебного материала** Создание списка. Операции над списками. Многомерные списки. Перебор элементов списка. Генераторы списков. Функции map(), zip(), filter(), reduce(). Добавление, удаление и поиск элементов списка. Выбор элементов случайным образом. Сортировка. Заполнение списка числами. Преобразование списка в строку. Кортежи. Множества. Диапазоны. Модуль itertools. Создание словаря. Операции над словарями. Перебор элементов словаря. Методы для работы со словарями. Генераторы словарей.	6
	Практические занятия Работа с функциями map(), zip(), filter(), reduce(). Работа с кортежами и множествами. Работа со словарями.	4
	Самостоятельная работа обучающихся	2

	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).	

Тема 4.4. **Работа с датой и временем.** **Пользовательские функции.** **Модули и пакеты**	**Содержание учебного материала** Получение текущих даты и времени. Форматирование даты и времени. «Засыпание» скрипта. Модули datetime, calendar. Измерение времени выполнения фрагментов кода. Определение функции и ее вызов. Анонимные функции. Функции-генераторы. Декораторы функции. Рекурсия. Глобальные и локальные переменные. Модули import, from. Пути поиска и повторная загрузка модулей. Пакеты.	6
	Практические занятия Алгоритмы, использующие рекурсию. Работа с модулями datetime, calendar. Работа с модулями import, from.	4
	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 4.5. **Работа с файлами.** **Файловая система**	**Содержание учебного материала** Функция open(). Метод close(). Чтение файла: целиком, через генератор. Чтение в список. Чтение методом read, readline и циклом for. Запись и добавление в файл. Запись методом write и writelines. Print в файл. Методы перемещения в файле. Метод tell. Метод seek. Метод truncate. Работа с каталогами. Текущий каталог. Создание каталогов. Удаление каталогов. Формирование пути. Чтение данных о каталогах. Проверка на директорию, файл и ссылку. Обход папок через os.walk().	6
	Практические занятия Работа с файлами: переименование, перемещение, копирование и удаление файлов.	4
	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 4.5. Интегрированная среда разработчика	**Содержание учебного материала** Отладка программ. Условные брейкпоинты и интерактивный режим. Предложения о слиянии.	4

	Практические занятия Работа с Git.	2
	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 4.6. Основные принципы объектно-ориентированного программирования (ООП)	**Содержание учебного материала** Определение класса. Создание экземпляра. Наследование. Полиморфизм. Перезагрузка операторов. Статистические методы и методы класса. Абстрактные методы. Свойства класса. Декораторы классов. Инкапсуляция. Классы встроенных исключений. Пользовательские исключения.	4
	Практические занятия Методы __init__ и __del__. Обработка исключений: try…except…else…finally, with…as.	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 4.7. Итераторы и генераторы. Перечисления	**Содержание учебного материала** Итераторы. Генераторы и их реализация. Аннотации типов. Перечисления.	6
	Практические занятия Реализация итераторов. Реализация генераторов.	2

	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 4.8. **Основы работы с декораторами**	**Содержание учебного материала** Функции высшего порядка. Декораторы. Некоторые особенности использования декораторов.	4
	Практические занятия Модуль functools. Декоратор functools.wraps().	2
	Самостоятельная работа обучающихся	2

	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).	
Тема 4.9. **Библиотека Tkinter.** **Основы разработки оконных приложений**	**Содержание учебного материала** Основы работы с Tkinter. Метапеременные. Управление жизненным циклом приложения. Стилизуемые и нестилизуемые компоненты.	6
	Практические занятия Работа с окнами. Обработка ошибок.	4
	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 4.10. Работа с архивами	**Содержание учебного материала** Алгоритмы сжатия и распаковки.	2
	Практические занятия Работа с архивами zip. Работа с архивами tar.	4

	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
Консультации		**2**
Промежуточная аттестация в виде экзамена		**8**
Всего:		**218**

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. Трофимов, В. В. Основы алгоритмизации и программирования: учебник для среднего профессионального образования / В. В. Трофимов, Т. А. Павловская. — 4-е изд. — Москва: Издательство Юрайт, 2025. — 108 с. — (Профессиональное образование). — ISBN 978-5-
534-20429-2. — Текст: электронный // Образовательная платформа Юрайт [сайт]. — URL: https://urait.ru/bcode/563861

2. Кудрина, Е. В. Основы алгоритмизации и программирования на языке C#: учебник для среднего профессионального образования / Е. В. Кудрина, М. В. Огнева. — Москва:
Издательство Юрайт, 2025. — 322 с. — (Профессиональное образование). — ISBN 978-5-534-10772-2. — Текст: электронный // Образовательная платформа Юрайт [сайт]. — URL: https://urait.ru/bcode/565504

3. Черпаков, И. В. Основы программирования: учебник и практикум для среднего профессионального образования / И. В. Черпаков. — 2-е изд., перераб. и доп. — Москва: Издательство Юрайт, 2026. — 196 с. — (Профессиональное образование). — ISBN 978-5-
534-18760-1. — Текст: электронный // Образовательная платформа Юрайт [сайт]. — URL: https://urait.ru/bcode/584552

4. Чернышев, С. А. Основы программирования на Python: учебник для среднего профессионального образования / С. А. Чернышев. — 2-е изд., перераб. и доп. — Москва: Издательство Юрайт, 2026. — 349 с. — (Профессиональное образование). — ISBN 978-5-534-17056-6. — Текст: электронный // Образовательная платформа Юрайт [сайт]. — URL: https://urait.ru/bcode/588671

Дополнительная учебная литература

1. Голицына, О. Л. Основы алгоритмизации и программирования: учебное пособие / О.Л. Голицына, И.И. Попов. — 4-е изд., испр. и доп. — Москва: ИНФРА-М, 2026. — 431 с.
— (Среднее профессиональное образование). - ISBN 978-5-16-021353-8. - Текст: электронный. - URL: https://znanium.ru/catalog/product/2222607

2. Якимов, С. П. Алгоритмизация и программирование: учебник для среднего профессионального образования / С. П. Якимов. — Москва: Издательство Юрайт, 2026. — 342 с. — (Профессиональное образование). — ISBN 978-5-534-19661-0. — Текст: электронный // Образовательная платформа Юрайт [сайт]. — URL: https://urait.ru/bcode/589849

3. Черпаков, И. В. Алгоритмизация и программирование в Python: учебник для среднего профессионального образования / И. В. Черпаков. — Москва: Издательство Юрайт, 2026. — 159 с. — (Профессиональное образование). — ISBN 978-5-534-21911-1. — Текст: электронный // Образовательная платформа Юрайт [сайт]. — URL: https://urait.ru/bcode/582413

4. Кудрявцева, И. А. Программирование: теория типов: учебник для среднего профессионального образования / И. А. Кудрявцева, М. В. Швецкий. — 2-е изд., перераб. и доп. — Москва: Издательство Юрайт, 2026. — 652 с. — (Профессиональное образование). — ISBN 978-5-534-15382-8. — Текст: электронный // Образовательная платформа Юрайт [сайт]. — URL: https://urait.ru/bcode/587345

5. Федоров, Д. Ю. Программирование на Python: учебник для среднего профессионального образования / Д. Ю. Федоров. — 6-е изд., перераб. и доп. — Москва: Издательство Юрайт, 2026. — 187 с. — (Профессиональное образование). — ISBN 978-5-534-19654-2. — Текст: электронный // Образовательная платформа Юрайт [сайт]. — URL: https://urait.ru/bcode/585807

Базы данных и/или Интернет-ресурсы

1. http://algolist.ru - Алгоритмы, методы, исходники
2. https://firststeps.ru/ - первые шаги: уроки программирования
3. https://coderbook.ru/ - Интерактивное учебное пособие по алгоритмизации и программированию
4. https://code-basics.com/ru – Школа программирования для детей и взрослых

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		
- понятие алгоритмизации, свойства алгоритмов, общие принципы построения алгоритмов, основные алгоритмические конструкции; - эволюцию языков программирования, их классификацию, понятие системы программирования; - основные элементы языка, структуру программы, операторы и операции, управляющие структуры, структуры данных, файлы, классы памяти; - подпрограммы, составление библиотек подпрограмм; - объектно-ориентированную модель программирования, основные принципы объектноориентированного программирования на примере алгоритмического языка: понятие классов и объектов, их свойств и методов, инкапсуляция и полиморфизма, наследования и переопределения.	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. по Тестирование темам дисциплины. Практические занятия. Практические по занятия заполнению документов.
Перечень умений, осваиваемых в рамках дисциплины		

- разрабатывать алгоритмы для конкретных задач; - использовать программы для графического отображения алгоритмов; - определять сложность работы алгоритмов; - работать в среде программирования; - реализовывать построенные алгоритмы в виде программ на конкретном языке программирования; - оформлять код программы в соответствии со стандартом кодирования; - выполнять проверку, отладку кода программы.	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 чек-листу качества. Защита практической работы и ответы на уточняющие вопросы. Проверка самостоятельной работы и мини-отчётов по источникам.

ФОНД ОЦЕНОЧНЫХ СРЕДСТВ

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

Темы для подготовки к промежуточной аттестации

1. Свойства и виды алгоритмов.
2. Способы описания алгоритмов: псевдокоды, блок-схема, программа.
3. Стандарты графического оформления алгоритмов.
4. Базовые алгоритмические конструкции: линейная, разветвляющаяся, циклическая.
5. Основные методы и этапы проектирования алгоритмов.
6. Нисходящее, модульное и восходящее проектирование.
7. Эффективность и сложность алгоритма, их практическая значимость.
8. Алгоритмы поиска. Алгоритмы сортировки.
9. Вложенные циклы. Вспомогательные алгоритмы.
10. Различные комбинации алгоритмических конструкций.
11. Декомпозиция алгоритма.
12. Базовая оценка сложности O(1), O(n), O(n^2).
13. Программный продукт и его характеристики.
14. Именование переменных. Типы данных.
15. Присваивание значения переменным. Проверка типа данных.
16. Преобразование типов данных. Удаление переменной.
17. Математические операторы. Двоичные операторы.
18. Операторы для работы с последовательностями.

19. Операторы присваивания. Приоритет выполнения операторов.
20. Операторы сравнения.
21. Оператор ветвления if…else.
22. Цикл for.
23. Функции range(), enumerate().
24. Цикл while.
25. Оператор continue. Оператор break.
26. Встроенные функции и методы для работы с числами.
27. Математические функции. Модуль math.
28. Модуль random. Генерация случайных чисел.
29. Создание строки. Специальные символы.
30. Операции над строками.
31. Форматирование строк. Метод format().
32. Функции и методы для работы со строками.
33. Изменение регистра символов. Поиск и замена в строке.
34. Вычисление выражений, заданных в виде строк.
35. Тип данных bytes и bytearray. Преобразование объекта в последовательность байтов.
36. Шифрование строк. Синтаксис регулярных выражений.
37. Поиск первого совпадения с шаблоном. Поиск всех совпадений с шаблоном.
38. Замена в строке.
39. Создание списка. Операции над списками.
40. Многомерные списки. Перебор элементов списка.
41. Функции map(), zip(), filter(), reduce().
42. Добавление, удаление и поиск элементов списка.
43. Выбор элементов случайным образом. Сортировка.
44. Заполнение списка числами. Преобразование списка в строку.
45. Кортежи.
46. Множества.
47. Диапазоны.
48. Модуль itertools.
49. Создание словаря.
50. Операции над словарями.
51. Перебор элементов словаря.
52. Методы для работы со словарями.
53. Генераторы словарей.
54. Получение текущих даты и времени.
55. Форматирование даты и времени.
56. «Засыпание» скрипта.
57. Модули datetime, calendar.
58. Измерение времени выполнения фрагментов кода.
59. Определение функции и ее вызов.
60. Анонимные функции. Функции-генераторы.
61. Рекурсия.
62. Глобальные и локальные переменные.
63. Модули import, from.
64. Пути поиска и повторная загрузка модулей.

65. Пакеты.
66. Функция open(). Метод close().
67. Чтение файла: целиком, через генератор.
68. Чтение в список. Чтение методом read, readline и циклом for.
69. Запись и добавление в файл. Запись методом write и writelines.
70. Метод tell. Метод seek. Метод truncate.
71. Работа с каталогами. Текущий каталог.
72. Создание каталогов. Удаление каталогов.
73. Формирование пути. Чтение данных о каталогах.
74. Проверка на директорию, файл и ссылку. Обход папок через os.walk().
75. Определение класса.
76. Создание экземпляра.
77. Наследование.
78. Полиморфизм.
79. Перезагрузка операторов.
80. Статистические методы и методы класса.
81. Абстрактные методы.
82. Свойства класса.
83. Декораторы классов.
84. Инкапсуляция.
85. Классы встроенных исключений.
86. Пользовательские исключения.
87. Итераторы.
88. Генераторы и их реализация.
89. Аннотации типов.
90. Перечисления.
91. Функции высшего порядка.
92. Декораторы.
93. Некоторые особенности использования декораторов.
94. Основы работы с Tkinter.
95. Метапеременные.
96. Управление жизненным циклом приложения.
97. Стилизуемые и нестилизуемые компоненты.
98. Алгоритмы сжатия и распаковки.

Темы для самостоятельной работы обучающихся

1. Развитие языков программирования.
2. Обзор языков программирования.
3. Области применения языков программирования.
4. Стандарты языков программирования.
5. Среда проектирования.
6. Компиляторы и интерпретаторы.
7. Жизненный цикл программы.
8. Программа.
9. Программный продукт и его характеристики.

10. Основные этапы решения задач на компьютере.
11. Типы данных.
12. Простые типы данных.
13. Производные типы данных.
14. Структурированные типы данных.
15. Операции и выражения.
16. Правила формирования и вычисления выражений.
17. Структура программы.
18. Ввод и вывод данных.
19. Оператор присваивания.
20. Составной оператор.
21. Условный оператор.
22. Оператор выбора.
23. Цикл с постусловием.
24. Цикл с предусловием.
25. Цикл с параметром.
26. Вложенные циклы.
27. Массивы.
28. Двумерные массивы.
29. Стандартные процедуры и функции для работы со строками.
30. Структурированный тип данных - множество.
31. Методы.
32. Рекурсивные функции.
32. Комментарии в программах.
33. Классы.
34. Конструкторы в классах.
35. Наследование.
36. Рекурсия.
37. Область видимости и время жизни переменной.
38. Определение и вызов подпрограмм.
39. Инкапсуляция.
40. Полиморфизм.

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

Тестовое задание	**Оцениваемые компетенции**
ВОПРОСЫ ОТКРЫТОГО ТИПА	
(ВПИСАТЬ ПРАВИЛЬНЫЙ ОТВЕТ) Конечная последовательность шагов для решения задачи - это _____	ОК 01.
+: алгоритм	

(ВПИСАТЬ ПРАВИЛЬНЫЙ ОТВЕТ) Свойство алгоритма означающее, что он должен завершаться за конечное число шагов, называется _____ +: результативностью +: результативность	ОК 01.
(ВПИСАТЬ ПРАВИЛЬНЫЙ ОТВЕТ) Графическое представление алгоритма – это _____ +: блок-схема	ОК 01.
(ВПИСАТЬ ПРАВИЛЬНЫЙ ОТВЕТ) Структура "Если условие, то действие 1, иначе действие 2" представляет тип алгоритма _____. +: разветвляющийся	ОК 02.
(ВПИСАТЬ ПРАВИЛЬНЫЙ ОТВЕТ) Алгоритм, который вызывает сам себя, называется _____ +: рекурсивный	ОК 02.
(ВПИСАТЬ ПРАВИЛЬНЫЙ ОТВЕТ) Алгоритм сортировки, имеющий сложность $O(n^2)$ в худшем случае, называется _____ +: пузырьковая сортировка +: пузырьковой сортировкой	ОК 02.
(ВПИСАТЬ ПРАВИЛЬНЫЙ ОТВЕТ) Тип данных, используемый для хранения целых чисел в python: _____ +: int	ОК 02.
(ВПИСАТЬ ПРАВИЛЬНЫЙ ОТВЕТ) Алгоритм, который всегда дает оптимальное решение, называется _____. +: точный +: точным	ОК 02.
(ВПИСАТЬ ПРАВИЛЬНЫЙ ОТВЕТ) Структура данных "первым пришел - первым вышел" называется _____ +: очередь +: очередью	ОК 02.
(ВПИСАТЬ ПРАВИЛЬНЫЙ ОТВЕТ) Для итерации по последовательности используется цикл _____ +: for	ОК 02.
(ВПИСАТЬ ПРАВИЛЬНЫЙ ОТВЕТ) Выйти из цикла досрочно можно с помощью оператора _____ +: break	ПК 2.1.
(ВПИСАТЬ ПРАВИЛЬНЫЙ ОТВЕТ) Пропустить текущую итерацию цикла можно с помощью оператора _____. +: continue	ПК 2.1.

(ВПИСАТЬ ПРАВИЛЬНЫЙ ОТВЕТ) Метод, который вызывается при создании объекта - это _____. +: init() +: __init__	ПК 2.1.
(ВПИСАТЬ ПРАВИЛЬНЫЙ ОТВЕТ)	ПК 2.1.

Механизм объектно-ориентированного программирования, который позволяет создавать новый класс (подкласс) на основе существующего класса (суперкласса) - это _____. +: наследование	
(ВПИСАТЬ ПРАВИЛЬНЫЙ ОТВЕТ) Модуль, используемый для работы с регулярными выражениями - это _____. +: re	ПК 2.1.
(ВПИСАТЬ ПРАВИЛЬНЫЙ ОТВЕТ) Метод, используемый для чтения всего файла - это _____ +: read()	ПК 2.1.
(ВПИСАТЬ ПРАВИЛЬНЫЙ ОТВЕТ) Модуль, используемый для работы с датой и временем - это _____. +: datetime	ПК 2.1.
(ВПИСАТЬ ПРАВИЛЬНЫЙ ОТВЕТ) Параметр, который надо изменить, чтобы встроенная функция print() вывела переданные ей аргументы в столбик - это _____ +: sep	ПК 2.2.
(ВПИСАТЬ ПРАВИЛЬНЫЙ ОТВЕТ) На экран в результате выполнения команды print(4 + 3.0) будет выведено _____ +: 7.0	ПК 2.2.
(ВПИСАТЬ ПРАВИЛЬНЫЙ ОТВЕТ) Для получения остатка от деления в Python используется оператор _____. +: %	ПК 2.2.
(ВПИСАТЬ ПРАВИЛЬНЫЙ ОТВЕТ) На экран в результате выполнения команды print(5**(5 - 2)) будет выведено _____. +: 125	ПК 2.2.
(ВПИСАТЬ ПРАВИЛЬНЫЙ ОТВЕТ) Встроенная функция Python, которая может быть использована для нахождения модуля числа - это_____. +: abs	ПК 2.2.
(ВПИСАТЬ ПРАВИЛЬНЫЙ ОТВЕТ) Оператор, используемый для повторения строки в Python - это _____ +: *	ПК 2.4.

(ВПИСАТЬ ПРАВИЛЬНЫЙ ОТВЕТ) На экран в результате выполнения команды print("123456789"[5]) будет выведено _____ +: 6	ПК 2.4.
(ВПИСАТЬ ПРАВИЛЬНЫЙ ОТВЕТ) Метод, используемый для преобразования строки в верхний регистр - это _____. +: upper()	ПК 2.4.
(ВПИСАТЬ ПРАВИЛЬНЫЙ ОТВЕТ) В результате выполнения команды print("123456789"[:5]) на экран будет выведено _____. +: 12345	ПК 2.4.
(ВПИСАТЬ ПРАВИЛЬНЫЙ ОТВЕТ)	ПК 2.4.

Метод, используемый для соединения подстрок в одну строку по заданному разделителю - это _____. +: join()	
(ВПИСАТЬ ПРАВИЛЬНЫЙ ОТВЕТ) Метод, используемый для удаления пробельных символов с начала и конца строки - это _____. +: strip()	ПК 2.4.
(ВПИСАТЬ ПРАВИЛЬНЫЙ ОТВЕТ) В результате выполнения команды print([1, 2, 3, 4, 5, 6][-4]) на экран будет выведено _____. +: 3	ПК 2.4.
(ВПИСАТЬ ПРАВИЛЬНЫЙ ОТВЕТ) В результате выполнения команды print(['a', 'b', 'cd', 'e', 'f'][3]) на экран будет выведено _____. +: e	ПК 2.4.
(ВПИСАТЬ ПРАВИЛЬНЫЙ ОТВЕТ) Тип данных в Python, который представляет кортежи - это _____. +: tuple	ПК 2.4.
(ВПИСАТЬ ПРАВИЛЬНЫЙ ОТВЕТ) Тип данных в Python, который представляет словари - это _____. +: dict	ПК 2.5.
(ВПИСАТЬ ПРАВИЛЬНЫЙ ОТВЕТ) Тип данных в Python, который представляет множества - это _____. +: set	ПК 2.5.
(ВПИСАТЬ ПРАВИЛЬНЫЙ ОТВЕТ) Тип данных в Python, который представляет логические значения - это _____. +: bool	ПК 2.5.
(ВПИСАТЬ ПРАВИЛЬНЫЙ ОТВЕТ) Результат преобразования типа int(-0.319e2) будет _____. +: -31	ПК 2.5.

(ВПИСАТЬ ПРАВИЛЬНЫЙ ОТВЕТ) Результат преобразования типа list({'a': 1, 'b': 2, 'c': 3}) будет _____. +: ['a', 'b', 'c']	ПК 2.5.
(ВПИСАТЬ ПРАВИЛЬНЫЙ ОТВЕТ) Один из ключевых принципов объектно-ориентированного программирования (ООП), который позволяет скрывать внутренние данные и методы класса, предоставляя доступ к ним только через определенные интерфейсы – это _____. +: инкапсуляция	ПК 2.5.
(ВПИСАТЬ ПРАВИЛЬНЫЙ ОТВЕТ) Символ, который используют для комментирования одной строки кода в Python – это _____. +: #	ПК 2.5.
(ВПИСАТЬ ПРАВИЛЬНЫЙ ОТВЕТ) Многоугольник, которым обозначается действие, которое следует выполнить - это_____. +: прямоугольник	ПК 2.5.
(ВПИСАТЬ ПРАВИЛЬНЫЙ ОТВЕТ)	ПК 2.5.

Количество базовых структур алгоритмов равно _____. +: 3 +: три +: трем	
(ВПИСАТЬ ПРАВИЛЬНЫЙ ОТВЕТ) Свойство алгоритма, означающее, что данный алгоритм применим к решению целого класса задач, называется _____. +: массовость	ПК 2.5.
(ВПИСАТЬ ПРАВИЛЬНЫЙ ОТВЕТ) Величины, значения которых меняются в процессе исполнения алгоритма, называются _____. +: переменными	ПК 2.5.
ВОПРОСЫ ЗАКРЫТОГО ТИПА	
Какой алгоритм сортировки является устойчивым? +: Сортировка слиянием -: Быстрая сортировка -: Пирамидальная сортировка -: Сортировка Шелла	ОК 01.
Какой алгоритм используется для нахождения наибольшего общего делителя? +: Алгоритм Евклида -: Решето Эратосфена -: Метод Ньютона -: Алгоритм Флойда	ОК 01.

Какой алгоритм сортировки имеет сложность O(n log n) в среднем случае? +: сортировка слиянием -: сортировка пузырьком -: сортировка вставками -: сортировка выбором	ОК 01.
Как называется графическое представление алгоритма? +: блок-схема -: таблица -: диаграмма -: граф	ОК 02.
Какой оператор используется для возведения в степень? +: ** -: ^ -: // -: %%	ОК 02.
Как получить длину списка? +: len(list) -: length(list) -: list.length() -: list.size	ОК 02.
Как правильно написать условный оператор if? +: if x > 0: -: if (x > 0) {} -: if x > 0 then -: if: x > 0	ПК 2.1.
Какой оператор используется для обработки исключений? +: try-except -: catch	ПК 2.1.

-: error -: rescue	
Как объявить функцию в Python? +: def my_func(): -: function my_func() -: func my_func() -: void my_func()	ПК 2.1.
Как вывести текст на экран в Python? +: print("Hello") -: echo("Hello") -: console.log("Hello") -: System.out.println("Hello")	ПК 2.1.
Как объявить переменную в Python? +: x = 5 -: var x = 5 -: int x = 5 -: variable x = 5	ПК 2.2.

Какой параметр функции позволяет принимать произвольное количество аргументов? +: *args -: *param -: **kwargs -: &args	ПК 2.2.
Как импортировать модуль math? +: import math -: include math -: require math -: using math	ПК 2.2.
Что вернет функция range(5)? +: [0, 1, 2, 3, 4] -: [0, 1, 2, 3, 4, 5] -: [1, 2, 3, 4, 5] -: (0, 1, 2, 3, 4)	ПК 2.4.
Какой тип данных является изменяемым? +: list -: tuple -: str -: int	ПК 2.4.
Как создать пустое множество? +: set() -: {} -: [] -: ()	ПК 2.5.
Какой метод удаляет и возвращает последний элемент списка? +: pop() -: remove() -: delete() -: cut()	ПК 2.5.
Как создать кортеж из одного элемента? +: (1,) -: (1) -: [1] -: tuple(1)	ПК 2.5.

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** осуществляется в форме проведения итогового тестирования по дисциплине.

Промежуточная аттестация **для обучающихся с применением дистанционных образовательных технологий** осуществляется в форме письменных и/или устных зачетов и выполнения 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:**	при форме контроля по дисциплине –
• «неудовлетворительно» – 49 баллов и менее;	«не зачтено» – 49 баллов и менее;
• «удовлетворительно» – от 50 до 65	«зачтено» – от 50 до 100 баллов баллов;
• «хорошо» – от 66 до 84 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).