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

Образовательное частное учреждение высшего образования

«МОСКОВСКИЙ УНИВЕРСИТЕТ ИМЕНИ А.С. ГРИБОЕДОВА»

(ИМПЭ им. А.С. Грибоедова)

ФАКУЛЬТЕТ СРЕДНЕГО ПРОФЕССИОНАЛЬНОГО ОБРАЗОВАНИЯ
(МЕЖДУНАРОДНЫЙ ДИПЛОМАТИЧЕСКИЙ КОЛЛЕДЖ)

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

ФОНД ОЦЕНОЧНЫХ СРЕДСТВ
по учебной дисциплине

Основы алгоритмизации и программирования

Специальность
09.02.11 Разработка и управление программным обеспечением

Квалификация выпускника:
Программист

на базе основного общего образования

Форма обучения: очная

Москва

Фонд оценочных средств для дисциплины «Основы алгоритмизации и программирования». Специальность 09.02.11 Разработка и управление программным обеспечением /– М.: ИМПЭ им. А. С. Грибоедова.

Фонд оценочных средств является неотъемлемой частью рабочей программы дисциплины.

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

Директор Международного
дипломатического колледжа_____/Н.И. Платонова

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

Настоящий Фонд оценочных средств (ФОС) по дисциплине «Основы алгоритмизации и программирования» является неотъемлемым приложением к рабочей программе дисциплины (РПД) «Основы алгоритмизации и программирования». На данный ФОС распространяются все реквизиты утверждения, представленные в РПД по данной дисциплине.

2. ПЕРЕЧЕНЬ ОЦЕНОЧНЫХ СРЕДСТВ

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

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

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

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(n²).
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