

**ФОНД
ОЦЕНОЧНЫХ СРЕДСТВ**

по учебной дисциплине

ОП.04 Основы алгоритмизации и программирования

для специальности

09.02.07 Информационные системы и программирование

Москва
2024

Фонд оценочных средств рассмотрен на заседании предметной (цикловой) комиссией общепрофессионального цикла

Фонд оценочных средств рассмотрен на основе федерального государственного образовательного стандарта среднего профессионального образования (далее – ФГОС СПО) по специальности 09.02.07 Информационные системы и программирование, утвержденного приказом Министерства образования и науки Российской Федерации от 09 декабря 2016 г. № 1547 (зарегистрирован Министерством юстиции РФ 26 декабря 2016 г. 44936) и Примерной основной образовательной программы по специальности 09.02.07 Информационные системы и программирование.

Внутренняя экспертиза:
Заведующая УМУ Заметта Д.Н.

**ПАСПОРТ
ФОНДА ОЦЕНОЧНЫХ СРЕДСТВ**

по дисциплине ОП.04 Основы алгоритмизации и программирования

В рамках программы учебной дисциплины обучающимися осваиваются ОК 01, ОК 02, ОК 04, ОК 05, ОК 09, ПК 1.1, ПК 1.2, ПК 1.3, ПК 1.4, ПК 1.5, ПК 2.4, ПК 2.5.

Результаты обучения	Критерии оценки	Методы оценки
<p>Знать: -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 объектно-ориентированного программирования на примере алгоритмического языка: понятие классов и объектов, их свойств и методов, инкапсуляция и полиморфизма, наследования и переопределения</p>	<p>«Отлично» - теоретическое содержание курса освоено полностью, без пробелов, умения сформированы, все предусмотренные программой учебные задания выполнены, качество их выполнения оценено высоко. «Хорошо» - теоретическое содержание курса освоено полностью, без пробелов, некоторые умения сформированы недостаточно, все предусмотренные программой учебные задания выполнены, некоторые виды заданий выполнены с ошибками. «Удовлетворительно» - теоретическое содержание курса освоено частично, но пробелы не носят существенного характера, необходимые умения работы с освоенным материалом в основном сформированы, большинство предусмотренных программой обучения учебных заданий выполнено, некоторые из выполненных заданий содержат ошибки. «Неудовлетворительно» - теоретическое содержание курса не освоено, необходимые умения не сформированы, выполненные учебные задания содержат грубые ошибки.</p>	<p>Оценка результатов выполнения практических работ. Оценка результатов устного и письменного опроса. Оценка результатов тестирования. Оценка результатов самостоятельной работы. Оценка результатов выполнения домашних заданий. Оценка результатов проведённого экзамена</p>
<p>Уметь: - Разрабатывать алгоритмы для конкретных задач.</p>		

<p>-Использовать программы для графического отображения алгоритмов.</p> <p>-Определять сложность работы алгоритмов.</p> <p>-Работать в среде программирования.</p> <p>-Реализовывать построенные алгоритмы в виде программ на конкретном языке программирования. - Оформлять код программы в соответствии со стандартом кодирования.</p> <p>-Выполнять проверку, отладку кода программы.</p>		
--	--	--

Типовые вопросы:

1. Развитие языков программирования. Обзор языков программирования.
2. Области применения языков программирования.
3. Стандарты языков программирования. Среда проектирования.
4. Компиляторы и интерпретаторы.
5. Жизненный цикл программы.
6. Программа. Программный продукт и его характеристики
7. Типы данных.
8. Простые типы данных.
9. Производные типы данных.
10. Структурированные типы данных.
11. Операции и выражения.
12. Правила формирования и вычисления выражений.
13. Структура программы. Ввод и вывод данных.
14. Оператор присваивания.
15. Составной оператор.
16. Условный оператор.
17. Оператор выбора.

18. Цикл с постусловием.
19. Цикл с предусловием.
20. Цикл с параметром.
21. Вложенные циклы.
22. Массивы. Двумерные массивы.
23. Строки. Стандартные процедуры и функции для работы со строками
24. Структурированный тип данных – множество.
25. Операции над множествами.
26. Комбинированный тип данных – запись.
27. Файлы последовательного доступа. Файлы прямого доступа
28. Общие сведения о подпрограммах. Определение и вызов подпрограмм.
29. Область видимости и время жизни переменной.
30. Механизм передачи параметров.
31. Организация функций.
32. Рекурсия. Программирование рекурсивных алгоритмов
33. Основы структурного программирования. Методы структурного программирования
34. Модульное программирование. Понятие модуля.
35. Структура модуля. Компиляция и компоновка программы. Стандартные модули
36. Указатели. Описание указателей.
37. Основные понятия и применение динамически распределяемой памяти.
38. Создание и удаление динамических переменных.
39. Структуры данных на основе указателей. Задача о стеке.
40. История развития объектно-ориентированного программирования (ООП).
41. Базовые понятия ООП: объект, его свойства и методы, класс, интерфейс.
42. Основные принципы ООП: инкапсуляция, наследование, полиморфизм.
43. Классы объектов. Компоненты и их свойства.
44. Событийно-управляемая модель программирования. Компонентно-ориентированный подход
45. Требования к аппаратным и программным средствам интегрированной среды разработчика.
46. Интерфейс среды разработчика: характеристика, основные окна, инструменты, объекты.
47. Форма и размещение на ней управляющих элементов. Панель компонентов и их свойства. Окно кода проекта.
48. Состав и характеристика проекта. Выполнение проекта.
49. Настройка среды и параметров проекта. Панель компонентов и их свойства.
50. Окно кода проекта. Состав и характеристика проекта.

51. Выполнение проекта. Настройка среды и параметров проекта.
52. Основные компоненты (элементы управления) интегрированной среды разработки, их состав и назначение. Дополнительные элементы управления.
53. Свойства компонентов. Виды свойств.
54. Синтаксис определения свойств. Назначения свойств и их влияние на результат.
55. Управление объектом через свойства. События компонентов (элементов управления), их сущность и назначение.
56. Создание процедур на основе событий.
57. Разработка функционального интерфейса приложения.
58. Создание интерфейса приложения. Разработка функциональной схемы работы приложения. Разработка игрового приложения
59. Разработка приложения. Проектирование объектно-ориентированного приложения. Создание интерфейса пользователя.
60. Тестирование, отладка приложения
61. Классы ООП: виды, назначение, свойства, методы, события.
62. Перегрузка методов. Тестирование и отладка приложения.

Устный опрос – это средство контроля, организованное как специальная беседа преподавателя с обучающимся на темы, связанные с изучаемой дисциплиной, и рассчитанное на выяснение объема знаний, обучающегося по определенному разделу, теме, проблеме и т. п.

Критерии оценки устного опроса студентов:

Оценка «отлично»:

- глубокое и прочное усвоение материала темы или раздела;
- полные, последовательные, грамотные, логически излагаемые аргументированные ответы;
- демонстрация обучающимся знаний в объеме пройденной программы и дополнительно рекомендованной литературы;
- воспроизведение учебного материала с требуемой степенью точности.

Оценка «хорошо»:

- наличие несущественных ошибок, не достаточно аргументированные ответы на вопросы;
- демонстрация обучающимся знаний в объеме пройденной программы;
- четкое изложение учебного материала.

Оценка «удовлетворительно»:

- наличие несущественных ошибок в ответе, отсутствие аргументации, но достаточно грамотное и логичное изложение;
- демонстрация обучающимся недостаточно полных знаний по пройденной программе, отсутствие аргументации;
- не структурированное, не грамотное и не логичное изложение учебного материала при ответе.

Оценка «неудовлетворительно»:

- незнание материала темы или раздела;
- серьезные ошибки при ответе.

Тестирование

1. По линии прямой связи передаются
 - 1) команды управления и информация об объекте управления
 - 2) информация о состоянии объекта управления
 - 3) информация о состоянии управляющей системы
 - 4) команды управления
 - 5) команды управления и информация об управляющей системе
2. Какой из объектов может являться исполнителем алгоритмов?
 - 1) карта
 - 2) словарь
 - 3) стиральная машина
 - 4) тетрадь
 - 5) краска
3. Понятность алгоритма означает, что он должен быть записан с помощью:
 - 1) команд, понятных создателю алгоритма
 - 2) команд из системы команд исполнителя
 - 3) команд, понятых пользователю алгоритма
 - 4) команд, понятных для компьютера
 - 5) операторов языка программирования
4. Конечность алгоритма означает, что:
 - 1) в нем должен присутствовать оператор вывода результата
 - 2) он должен решать задачу вычислительного характера
 - 3) в нем должно присутствовать ключевое слово, означающее конец алгоритма
 - 4) он должен быть применим для решения всех задач заданного типа
 - 5) результат должен быть получен за конечное число шагов
5. Как называется свойство алгоритма, соответствующее определению: «Алгоритм должен быть записан из команд, понятных исполнителю, каждая команда должна определять однозначное действие исполнителя»?
 - 1) массовость
 - 2) точность
 - 3) конечность
 - 4) понятность
 - 5) дискретность
6. Суть такого свойства алгоритма как результативность заключается в том, что:
 - 1) алгоритм должен иметь дискретную структуру (должен быть разбит на последовательность отдельных шагов);
 - 2) записывая алгоритм для конкретного исполнителя, можно использовать лишь те команды, что входят в систему его команд;

- 3) алгоритм должен обеспечивать решение не одной конкретной задачи, а некоторого класса задач данного типа;
 - 4) при точном исполнении всех команд алгоритма процесс должен прекратиться за конечное число шагов, приведя к определенному результату;
 - 5) исполнитель алгоритма не должен принимать решения, не предусмотренные составителем алгоритма
7. Свойство алгоритма как понятность заключается в том, что:
- 1) алгоритм должен иметь дискретную структуру (должен быть разбит на последовательность отдельных шагов);
 - 2) записывая алгоритм для конкретного исполнителя, можно использовать лишь те команды, что входят в систему его команд;
 - 3) алгоритм должен обеспечивать решение не одной конкретной задачи, а некоторого класса задач данного типа;
 - 4) исполнитель алгоритма не должен принимать решения, не предусмотренные составителем алгоритма
8. Суть такого свойства алгоритма как массовость заключается в том, что:
- 1) алгоритм должен иметь дискретную структуру (должен быть разбит на последовательность отдельных шагов);
 - 2) записывая алгоритм для конкретного исполнителя, можно использовать лишь те команды, что входят в систему его команд;
 - 3) алгоритм должен обеспечивать решение не одной конкретной задачи, а некоторого класса задач данного типа;
 - 4) при точном исполнении всех команд алгоритма процесс должен прекратиться за конечное число шагов, приведя к определенному результату;
 - 5) исполнитель алгоритма не должен принимать решения, не предусмотренные составителем алгоритма
9. Языки программирования условно разделены на классы: машинные, машинно-ориентированные, алгоритмические, процедурно-ориентированные, проблемно-ориентированные. По какому признаку классифицированы языки?
- 1) синтаксис образования конструкций языков программирования
 - 2) принадлежность к одному из оформившихся к настоящему времени стилей программирования
 - 3) уровень языка программирования, характеризующий степень его близости к машинному языку
10. Языки программирования условно разделены на классы: процедурные, функциональные, логические, объектно-ориентированные. По какому признаку классифицированы языки?
- 1) синтаксис образования конструкций языков программирования
 - 2) принадлежность к одному из оформившихся к настоящему времени стилей программирования
 - 3) уровень языка программирования, характеризующий степень его близости к машинному языку
11. Примером функциональных языков программирования может служить

- 1) LISP
- 2) PROLOG
- 3) DELPHI
- 4) PASCAL

12. Транслятор, который переводит каждую команду программы с одновременным её выполнением и, если обнаруживает ошибку, сообщает о ней и прекращает выполнение программы, называется

- 1) компилятором
- 2) редактором связей
- 3) системным администратором
- 4) интерпретатором

13. Программа, на основе которой машина преобразует вводимые в нее команды на машинный язык, называется

- 1) переводчиком
- 2) транслятором
- 3) системным администратором
- 4) редактором связей

14. Примером логических языков программирования может служить

- 1) LISP
- 2) PROLOG
- 3) DELPHI
- 4) PASCAL

15. К языкам высокого уровня относятся

- 1) Ada
- 2) язык Ассемблера
- 3) двоичный язык
- 4) C/C++

16. К языкам высокого уровня не относятся

- 1) Ada
- 2) язык Ассемблера
- 3) двоичный язык
- 4) C/C++

Критерии и шкала оценивания (тестирование)

Оценка «отлично» -	90-100% правильных ответов
Оценка «хорошо» -	70-89% правильных ответов
Оценка «удовлетворительно» -	51-69% правильных ответов
Оценка «неудовлетворительно» -	Менее 51 % правильных ответов

Практические задания

№ 1

Вопросы.

Перечислите составляющие единицы алфавитов языков программирования. Опишите структуру программы на языке программирования Паскаль.

Опишите структуру программы на языке программирования С.

Приведите примеры директив препроцессора языка программирования С, поясните их назначение и порядок работы в программе.

Дайте определение понятия «тип данных».

На какие группы делятся все типы данных?

Перечислите типы данных относящиеся к группе простых типов.

Перечислите простые типы данных языка программирования Паскаль.

Дайте определение понятия «тип данных».

На какие группы делятся все типы данных?

Перечислите типы данных относящиеся к группе простых типов.

Перечислите простые типы данных языка программирования С.

Дайте определение понятия «тип данных».

Опишите правила построения символьных данных в языках программирования Паскаль и С.

Дайте определение понятия «тип данных».

Опишите порядок использования вещественных типов данных в языках программирования Паскаль и С.

Дайте определение понятия «тип данных».

Опишите использование логических данных в языках программирования Паскаль и С.

Опишите порядок использования оператора присваивания и правила построения выражений в языках Паскаль и С.

Назовите операции над данными простых типов в языках Паскаль и С.

Расскажите, как построен ввод и вывод данных в языке программирования Паскаль.

Расскажите, как построен ввод и вывод в языке программирования С.

Опишите правила реализации конструкции ветвления в языке программирования Паскаль.

Опишите правила реализации конструкция ветвление в языке программирования С.

Опишите правила реализации циклов с неизвестным числом повторов в языке программирования Паскаль.

Опишите правила реализации циклов с неизвестным числом повторов в языке программирования С.

Опишите правила реализации циклов с параметром в языке программирования Паскаль.

Опишите правила реализации циклов с параметром в языке программирования С.
Дайте определение структуры данных - массив.

Опишите правила реализации массивов в языке Паскаль.

Дайте определение структуры данных - массив.

Опишите правила реализации массивов в языке С.

Дайте определение структуры данных строка.

Опишите правила реализации строк в языке Паскаль.

Дайте определение структуры данных строка.

Опишите правила реализации строк в языке С.

Дайте определение структуры данных файл.

Опишите правила реализации файлов в языке Паскаль.

Дайте определение структуры данных файл.

Опишите правила реализации файлов в языке С.

Перечислите принципы объектно-ориентированного программирования.

Опишите возможности среда разработки Delphi.

Приведите примеры визуальных компонент среды Delphi.

Поясните порядок работы с ними.

№ 2

Выполнить тест за компьютером

Создать программу на языке программирования Турбопascal и запустить ее на исполнение:

Составить программу, которая запрашивает ввод с клавиатуры любых чисел с и d.
Затем производит их умножение и выводит на экран результат: произведение $c*d=?$

Критерии и шкала оценивания (выполнение практических заданий)

«отлично» - По решению задачи дан правильный ответ и развернутый вывод

«хорошо» - По решению задачи дан правильный ответ, но не сделан вывод

«удовлетворительно» - По решению задачи дан частичный ответ, не сделан вывод

«неудовлетворительно» - Задача не решена полностью

Творческая проблемно-ориентированная самостоятельная работа

включает следующие виды:

- поиск, анализ, структурирование информации по темам, выносимым на самостоятельное изучение;
- составление и разработка словаря (глоссария).

Самостоятельная работа с литературой ставит своей целью – закрепление, углубление, расширение и систематизация знаний, полученных в ходе аудиторных занятий, самостоятельное овладение новым учебным материалом описательного характера, развитие самостоятельного мышления.

Домашнее задание – выполняется обучающимися после изучения соответствующих тем программного материала данного курса и является формой текущего контроля знаний обучающихся по дисциплине.

Варианты задания

В массиве из 20 целых чисел найти наименьший элемент и поменять местами первым элементом.

В массиве из 15 вещественных чисел найти наибольший элемент и поменять его местами с последним элементом.

Упорядочить по неубыванию массив, содержащий 20 целых чисел.

Вычислить сумму и количество элементов массива $X(100)$ для $0 \leq X \leq 1$

Вычислить среднее арифметическое значение элемента положительного массива $A(80)$.

Вычислить минимальный элемент массива $C(40)$ и его порядковый номер.

Переписать элементы массива $X(70)$ в массив Y и подсчитать их количество, ($-1 \leq X \leq 1$).

Определить максимальный элемент массива $B(50)$ и его порядковый номер.