

Методические материалы по дисциплине:

Основы программирования

Основная профессиональная образовательная программа высшего образования – программа специалитета.

12.05.01 Электронные и оптико-электронные приборы и системы специального назначения

1. Алгоритм - это:
последовательность действий для решения задачи;
2. Блок-схема используется для:
графического представления алгоритма;
3. Переменная в программировании - это:
именованная область памяти для хранения данных;
4. Тип данных `integer` представляет:
целые числа;
5. Логический тип данных (`boolean`) может принимать значения:
`true` (истина) или `false` (ложь);
6. Оператор присваивания обозначается символом:
`=` (равно);
7. Условный оператор `if` используется для:
выполнения действий при определенном условии;
8. Цикл `while` повторяет действия:
пока условие истинно;
9. Цикл `for` используется когда:
известно количество повторений;
10. Функция в программировании - это:
именованный блок кода для выполнения конкретной задачи;
11. Массив - это структура данных для хранения:
элементов одного типа;
12. Основное свойство стека:
последний пришел - первый ушел (LIFO);
13. Основное свойство очереди:
первый пришел - первый ушел (FIFO);
14. Рекурсия - это когда:
функция вызывает саму себя;
15. Компилятор преобразует:
исходный код в машинный код;
16. Интерпретатор выполняет программу:
построчно без предварительной компиляции;
17. Синтаксис языка программирования - это:
правила написания команд;
18. Отладка (`debugging`) - это процесс:
поиска и исправления ошибок в программе;
19. Понятие "алгоритмическая сложность" характеризует:
зависимость времени выполнения от объема данных;
20. Линейный поиск имеет сложность:
 $O(n)$;
21. Бинарный поиск требует предварительной:
сортировки данных;
22. Основные принципы ООП (объектно-ориентированного программирования):
инкапсуляция, наследование, полиморфизм;
23. Инкапсуляция означает:
объединение данных и методов в одном объекте;

24. Наследование позволяет:
создавать новые классы на основе существующих;
25. Полиморфизм позволяет объектам:
использовать методы с одинаковыми именами по-разному;
26. Конструктор класса - это метод:
вызываемый при создании объекта;
27. Модификатор доступа `public` означает что:
метод или поле доступны из любого места программы;
28. Исключение (exception) - это:
ошибка возникающая при выполнении программы;
29. Блок `try-catch` используется для:
обработки исключений;
30. Понятие "интерфейс" в ООП определяет:
набор методов которые должен реализовать класс;
31. Абстрактный класс может содержать:
абстрактные методы без реализации;
32. Композиция означает что:
объект содержит другие объекты как части;
33. Жизненный цикл программы включает этапы:
написание, компиляция, выполнение;
34. Понятие "среда разработки" (IDE) включает:
редактор кода, компилятор, отладчик;
35. Система контроля версий `Git` используется для:
отслеживания изменений в коде;
36. Основная ветка в `Git` обычно называется:
`main` или `master`;
37. Команда `git commit` используется для:
сохранения изменений в репозитории;
38. Тестирование программы бывает:
модульное, интеграционное, системное;
39. Модульное тестирование проверяет:
отдельные компоненты программы;
40. Стил кодирования определяет правила:
оформления и написания кода;
41. Комментарии в коде используются для:
пояснения логики программы;
42. Основные структуры данных:
массивы, списки, деревья, хеш-таблицы;
43. Бинарное дерево имеет не более:
двух потомков у каждого узла;
44. Хеш-таблица позволяет быстро:
находить значения по ключу;
45. Рекурсивный алгоритм должен иметь:
условие завершения;
46. Основные алгоритмы сортировки:
пузырьковая, быстрая, слиянием;

- 47. Быстрая сортировка (quicksort) имеет среднюю сложность:
 $O(n \log n)$;
- 48. Динамическое программирование используется для:
решения задач путем разбиения на подзадачи;
- 49. Жадные алгоритмы на каждом шаге выбирают:
локально оптимальное решение;
- 50. Основные парадигмы программирования:
императивная, функциональная, объектно-ориентированная.