



Основы программирования.ти(2)

- 1) Какое действие выполняет последовательность символов \n
- 2) Для префиксной формы записи операции инкремента:
- 3) Для постфиксной формы записи операции декремента:
- 4) Операция логическое И: A&&B принимает результат «ИСТИНА», если:
- 5) Операция логическое ИЛИ: A || B принимает результат «ЛОЖЬ», если:
- 6) Операция логическое И с элементами отрицания: A && !B принимает результат «ИСТИНА», если:
- 7) Операция логическое ИЛИ с элементами отрицания: A || !B принимает результат «ЛОЖЬ», если:
- 8) Операция условия Операнд-1 ? операнд-2: операнд-3; задает следующие действия:
- 9) Какие из перечисленных операций языка C++ имеют одинаковые приоритеты:
- 10) Какая из перечисленных операций языка C++ имеет самый высокий приоритет:
- 11) Какой из перечисленных указатель определяет указатель на действительное число:
- 12) Разделитель простых операторов:
- 13) Разделитель составных операторов:
- 14) К какому типу операторов относится оператор switch:
- 15) Какие операторы относятся к категории «условные операторы»:
- 16) Какие операторы относятся к категории «операторы цикла»:
- 17) Какие операторы относятся к категории «операторов перехода»:



- 18) Когда логическое выражение в операторе: `if` (логическое выражение) оператор `A`; `[else` оператор `B`;`]` принимает значение «ЛОЖЬ», то:
- 19) Оператор `2` в конструкции: `if` (условие 1) `if` (условие 2) оператор `1`; `else` оператор `2`; будет выполняться в случае, если:
- 20) Оператор `1` в конструкции: `if` (условие 1) `{ if` (условие 2) оператор `1`; `}` `else` оператор `2`; будет выполняться в случае, если:
- 21) Фрагмент программы, содержащий конструкцию `else if`: `if` (условие 1) оператор `1`; `else if` (условие 2) оператор `2`; `else if` (условие 3) оператор `3`; `else` оператор `4`; оператор `5`; выполняет следующие действия:
- 22) Оператор `switch` (выражение выбора)
- 23) Какие из перечисленных типов данных нельзя использовать в качестве значений констант в вариантах `case` оператора `switch`
- 24) Фрагмент программы, содержащий конструкцию оператора выбора `switch` (выражение выбора) `{ case` значение `1`: оператор `1`; `break`; `case` значение `2`: оператор `2`; `default`: оператор `3`; `}` оператор `4`; выполняет следующие действия:
- 25) Оператор `break` в структуре множественного выбора `switch` обеспечивает:
- 26) Оператор `while` (логическое условие) `{` рабочие операторы; `}` выполняет следующие действия
- 27) Оператор `do` `{` рабочие операторы; `}` `while` (логическое условие) выполняет следующие действия
- 28) Оператор `for` (`A=B`; `C`; `D`) рабочий оператор; выполняет следующие действия
- 29) После выполнения программы: `int sum = 2 for (int i = 2; i`
- 30) Оператор `break` предназначен для того, чтобы:
- 31) Оператор `continue` в циклических структурах `while`, `for`, `do-while` предназначен для того, чтобы
- 32) Функция - это
- 33) Какой тип значений может возвращать функция с помощью оператор `return`:





- 34) Параметры функции называются:
- 35) Тип в форме описания функции: Тип (список параметров) {операторы тела функции}
- 36) Особенности работы с функциями:
- 37) При вызове функции по значению аргумента:
- 38) При вызове функции по ссылке на аргумент:
- 39) При вызове multiply (X1=1, Y1=2) по форме описания: `int multiply (X, Y) { X +=Y++ return ++X * Y++ }` функция вернет следующее значение:
- 40) При вызове функции с использованием указателя на аргумент:
- 41) Какой из перечисленных указателей определяет указатель на действительное число:
- 42) Как объявить указатель на массив из десяти длинных целых:
- 43) Операции над указателями:
- 44) Укажите вид оператора, присваивающего адрес переменной у указателю ru:
- 45) При выполнении операции: `double *p; ++p;` значение указателя p увеличится на:
- 46) Свойства оператора разадресации:
- 47) Что такое перегрузка функций:
- 48) При проектировании независимых друг от друга функций методом функциональной или процедурной абстракции широко применяется:
- 49) При использовании модульного программирования:
- 50) Какие ключевые слова используют для объявления переменных локального типа:
- 51) Идентификатор, объявленный вне любой функции (на внешнем уровне) имеет область действия:

