

Тест 1

Основы программирования, циклы (for, while, do while), оператор множественного выбора switch

1. Какие среды программирования (IDE) предназначены для разработки программных средств?
 - MVS, NetBeans, QT Creator, RAD Studio, Dev-C++
 - MVS, Code::Blocks, QT Creator, RAD Studio, MathCAD
 - MVS, Code::Blocks, QT Creator, AutoCAD, Eclipse
2. Укажите объектно-ориентированный язык программирования
 - Все варианты ответов
 - C++
 - Java
 - Eiffel
3. Какой из ниже перечисленных операторов, не является циклом в C++?
 - repeat until
 - for
 - do while
 - while
4. Какой оператор не допускает перехода от одного константного выражения к другому?
 - break;
 - точка с запятой
 - end;
 - Stop;
5. Какой служебный знак ставится после оператора case ?
 - .
 - ;
 - :
 - -
6. Какому зарезервированному слову программа передаёт управление в случае, если значение переменной или выражения оператора switch не совпадает ни с одним константным выражением?
 - other
 - all
 - contingency
 - default
7. Укажите правильную форму записи цикла do while
 - ```
// форма записи оператора цикла do while:
do // начало цикла do while
{
/*блок операторов*/;
}
while (/*условие выполнения цикла*/) // конец цикла do while
```
  - ```
// форма записи оператора цикла do while:
do // начало цикла do while
{
/*блок операторов*/;
}
while (/*условие выполнения цикла*/); // конец цикла do while
```
 - ```
// форма записи оператора цикла do while:
do // начало цикла do while
{
/*блок операторов*/;
}
```

```
while { /*условие выполнения цикла*/ } // конец цикла do while
```

8. Простые типы данных в C++.

- целые – bool, вещественные – float или double, символьные – string
- целые – int, вещественные – float или double, символьные – char
- целые – int, вещественные – float или real, символьные – char
- целые – int, вещественные – float или double, символьные – string

9. Какая из следующих записей - правильный комментарий в C++?

- {комментарий}
- \*\* Комментарий \*\*
- /\* комментарий \*/
- \*/ Комментарий \*/

10. Какой из следующих операторов - оператор сравнения двух переменных?

- ==
- =
- :=
- equal

11. Общий формат оператора множественного выбора - switch

- ```
switch (switch_expression)
{
    case constant1, case constant2: statement1; [break;]
    case constantN: statementN; [break;]
    [default: statement N+1;]
}
```
- ```
switch (switch_expression)
{
 case constant1: statement1; [break;]
 case constant2: statement2; [break;]
 case constantN: statementN; [break;]
 [default: statement N+1;]
}
```
- ```
switch (switch_expression)
{
    case constant1: statement1; [break;]
    case constant2: statement2; [break;]
    case constantN: statementN; [break;]
    [else: statement N+1;]
}
```

12. Укажите правильное определение функции main в соответствии со спецификацией стандарта ANSI

- int main(void)
- int main()
- void main()
- void main(void)

13. Тело любого цикла выполняется до тех пор, пока его условие ...

- у цикла нет условия
- истинно
- ложно

14. Какие служебные символы используются для обозначения начала и конца блока кода?

- { }
- begin end

- < >
- ()

15. Программа, переводящая входную программу на исходном языке в эквивалентную ей выходную программу на результирующем языке, называется:

- сканер
- транслятор
- интерпретатор
- компилятор

16. Язык программирования C++ разработал

- Никлаус Вирт
- Бьерн Страуструп
- Кен Томпсон
- Дональд Кнут

17. Какую функцию должны содержать все программы на C++?

- `system()`
- `program()`
- `start()`
- `main()`

18. Цикл с предусловием?

- `for`
- `do while`
- `while`

19. Какими знаками заканчивается большинство строк кода в Си++?

- , (запятая)
- : (двоеточие)
- ; (точка с запятой)
- . (точка)

20. До каких пор будут выполняться операторы в теле цикла `while (x < 100)`?

- Пока `x` меньше или равен стам
- Пока `x` больше ста
- Пока `x` строго меньше ста
- Пока `x` равен стам

21. В приведённом коде измените или добавьте один символ чтобы код напечатал 20 звёздочек - *.

- ```
int i, N = 20;
for(i = 0; i < N; i--)
 printf("*");
```
- ```
int i, N = 20;
for(i = 20; i < N; i--)
    printf("*");
```
- ```
int i, N = 20;
for(i = 19; i < N; i--)
 printf("*");
```
- ```
int i, N = 40;
for(i = 0; i < N; i--)
    printf("*");
```

```
int i, N = 20;
for(i = 0; i < N; N--)
    printf("*");
```

22. Название C++ предложил

- Дональд Кнут
- Бьерн Страуструп
- Кэн Томпсон
- Рик Масситти

23. Структура объявления переменных в C++

- `[:=], < идент. 2>, ...;`
- `[==]; < идент. 2>, ...;`
- `[=], < идент. 2>, ...;`
- `[=]; < идент. 2>, ...;`

24. Выберите правильный вариант объявления константной переменной в C++, где `type` - тип данных в C++ `variable` - имя переменной `value` - константное значение

- `const type variable = value;`
- `const type variable := value;`
- `const variable = value;`

25. Каков результат работы следующего фрагмента кода?

```
int x = 0;
```

```
switch(x)
{
    case 1: cout << "Один";

    case 0: cout << "Ноль";

    case 2: cout << "Привет мир";

}
```

- Один
- НольПривет мир
- Привет мир
- Ноль

26. Что будет напечатано?

```
int main()
{
    for (int i = 0; i < 4; ++i)
    {
        switch (i)
        {
            case 0 : std::cout << "0";
            case 1 : std::cout << "1"; continue;
            case 2 : std::cout << "2"; break;
            default : std::cout << "D"; break;
        }
        std::cout << ".";
    }
    return 0;
}
```

- 0112.D.
- 0.1.2.
- Ошибка компиляции в строке 10
- 01.2.D.
- 011.2.D

27. Чтобы подключить заголовочный файл в программу на C++, например `iostream` необходимо написать:

- `#include <> c iostream` внутри скобок
- `#include <>; c iostream.h` внутри скобок
- `include #iostream,h;`
- `include (iostreamh)`

28. Чему будет равна переменная `a`, после выполнения этого кода `int a; for(a = 0; a < 10; a++) {}`?

- 10
- 1
- 9

29. Какой из перечисленных типов данных не является типом данных в C++?

- `real`
- `int`
- `double`
- `float`

30. Какое значение, по умолчанию, возвращает программа операционной системе в случае успешного завершения?

- -1
- 1
- Программа не возвращает значение.
- 0

31. Цикл с постусловием?

- `while`
- `do while`
- `for`

Тест 2

Арифметические и логические операции, оператор выбора `if`, ввод/вывод, приведение типов данных.

1. Что появится на экране, после выполнения этого фрагмента кода?

```
int a = 1, b = 2;
if (a == b);
cout << a << " = " << b << endl;
```

- синтаксическая ошибка
- `1 = 2`
- вывод на экран не выполнится
- `a = b`

2. Какое ключевое слово указывает, что целая переменная не может принимать отрицательные значения?

- нет такого зарезервированного слова
- `unsigned`
- `positive`
- `long`
- другое

3. Логическая операция с большим приоритетом выполнения.

- `!`
- `&&`
- `||`

4. В каком случае лучше всего использовать приведение типов данных?

- чтобы разрешить программе использовать только целые числа
- во всех выше указанных случаях
- при делении двух целых чисел, для того, чтобы вернуть результат с плавающей точкой
- чтобы изменить тип возвращаемого значения функции

5. Чему равен результат выполнения следующего выражения: $1000 / 100 \% 7 * 2$?
- 10
 - 250
 - 1000
 - 6
6. В каком случае можно не использовать фигурные скобочки в операторе выбора `if`?
- если в теле оператора `if` два и более операторов
 - если в теле оператора `if` нет ни одного оператора
 - если в теле оператора `if` всего один оператор
 - нет правильного ответа
7. Результат выполнения следующего фрагмента кода: `cout << 22 / 5 * 3;`
- 12
 - 1.47
 - другое
 - 1
 - 13.2
- 8.
- Ввод данных в C++
- `cin >> <выражение1> >> <выражение2> >> endl >>...;`
 - `cin >> <выражение1>, <выражение2>, ...;`
 - `cin >> <выражение1> >> <выражение2>...;`
9. Какие преобразования типов данных не возможны без потери данных?
- все перечисленные преобразования не возможны
 - `char` to `float`
 - `float` to `int`
 - `int` to `float`
10. Укажите операцию, приоритет выполнения которой ниже остальных.
- `<<`
 - `&&`
 - `?:`
 - `^`
 - `||`
 - `>>`
 - `&`
 - `|`
11. Тело оператора выбора `if`, будет выполняться, если его условие:
- ложно (`false`)
 - истинно (`true`)
12. Какое из следующих значений эквивалентно зарезервированному слову `true`?
- 66
 - -1
 - Все варианты ответов
 - 0.1
 - 1
13. Если условие оператора выбора ложное, то:
- программа завершает работу
 - выполняется тело оператора выбора
 - выполняется следующий оператор, сразу после оператора `if`
14. Преобразование целочисленной переменной `value` в ASCII эквивалент
- `char (value)`

15. Какое значение будет напечатано, в результате выполнения программы?

- ничего не напечатается, программа вообще не будет работать
- 1
- 3
- 2
- 0

16. Почему приведение типов данных может быть не безопасно.

- Некоторые преобразования не определены компилятором, такие как - преобразование символа в целое.
- нет никаких опасностей.
- Вы можете временно потерять часть данных - таких, как отсечение десятичной части чисел с плавающей точкой.
- Вы можете навсегда изменить значение переменной.

17. Результат выполнения следующего фрагмента кода: `!( (1 || 0) && 0)`

- 1
- 0
- результат не может быть заранее определен

18. Что будет напечатано, после выполнения этого кода: `cout << (5 << 3);` ?

- 53
- 40
- 35

19. Укажите неправильно записанную операцию отношения

- все операторы записаны правильно
- $=!$
- $>=$
- $<=$

20. Переменная x может быть доступна в другом блоке программы?

```
int main(int argc, char** argv)
{
    if ( argc > 2 )
    {
        int x = 5;
    }
}
```

```

    }
    else
    {

    }

    return 0;
}

```

- нет
- да

21. Какой из ниже перечисленных вариантов ответа, показывает правильно записанный оператор выбора if ?

- условное выражение if
- if (условное выражение)
- if { условное выражение }
- if условное выражение

22. Оператор if else позволяет определить действие ...

- только для ложного условия
- только для истинного условия
- для истинного и ложного условий

23. Какой из следующих логических операторов - логический оператор И?

- |
- &&
- &
- |&

24. Какое значение будет напечатано?

```
#include
```

```

int main(int argc, char** argv)
{
    int x = 0;
    int y = 0;

    if (x++ && y++)
    {
        y += 2;
    }

    std::cout << x + y << std::endl;

```

```

return 0;
}

```

- 3
- 2
- 4
- 1

25. Какое значение будет содержать переменная x?

```
#include
```

```
int x;
```

```

int main()
{

```

```

int y;
std::cout << x << std::endl;
std::cout << y << std::endl;
return 0;
}

```

- неопределённое
- 0

26. Какой заголовочный файл следует подключить, чтобы можно было пользоваться приведением типов данных?

- cmath
- Никакого
- cctype

27. Каков будет результат выражения `!(1 && !(0 || 1))`?

- False
- неоднозначность
- True

28. Укажите правильное приведение типа данных!

- char:a;
- to(char, a);
- (char)a;
- a(char);

29. Укажите операцию, приоритет выполнения которой больше остальных

- ()
- +
- /
- ++
- *

30. Вывод данных в C++

- ```
cout << <переменная >, < "< строка выводится на экран>" , <выражение > , endl;
```
- ```
cout << <переменная > << "< строка выводится на экран>" << <выражение > << endl;
```
- ```
cout << <переменная >, < "< строка выводится на экран>" , <выражение > , endl;
```

31. Укажите блок кода, в котором переменная `y` доступна

```

int main(int argc, char** argv)
{

```

```

 if (argc > 10)
 {
 }
 else if (int y = argc - 1)
 {

 }
 else
 {

 }

```

```

return 0;
}

```

- строки 8 -15
- строки 4 -17

- строки 8 -17
  - строки 8 -11
  - строки 4 -15
32. Результат выполнения следующего фрагмента кода: `54 << 3`?
- 556
  - 440
  - 432
  - 623
  - нет правильного ответа
33. Оператор вывода `cout` может печатать несколько значений или переменных в одной команде, используя следующий синтаксис:
- `cout << "Привет", name, "n";`
  - `cout << "Привет" << name << "n";`
  - `cout << "Привет" + name + "n";`
  - `cout << ("Привет" & name & "n");`
34. Это значение `5.9875e17` может быть сохранено в переменной, типа:
- float
  - bool
  - int
  - long
  - short

### Тест 3

#### Строки, ссылки, указатели, массивы.

1. Укажите зарезервированное ключевое слово для высвобождения выделенной памяти!
- free
  - delete
  - remove
  - clear
2. В какой из следующих записей используется операция взятия адреса?
- `address(a);`
  - `*a;`
  - `a ;`
  - `&a;`
3. Какой заголовочный файл необходимо подключить, чтобы вызвать функцию `isalpha()`?
- cctype
  - conio.h
  - cstring
  - ifstream.h
4. Строка `Привет Мир` будет показана на экране или нет?
- ```
int main(int argc, char** argv)
{
    int array[33];

    if ( &array[4] < &array[23] )
    {
        std::cout << "Привет мир" << std::endl;
    }
}
```

```
return 0;
}
```

- синтаксическая ошибка
- да
- нет

5. Какой стандартный код используется для Символьных данных типа `char` в C++?

- Код ASCII
- Код UTF-8
- Код ASCII
- Код cp-1251

6. Объявлена переменная `char a`; Какое из следующих выражений не верно?

- `a = "3";`
- `a = 3;`
- `a = '3';`

7. Будет ли напечатано сообщение не равны?

```
struct Foo
{
};
```

```
struct Bar
{
};
```

```
int main(int argc, char** argv)
{
    Foo* f = new Foo;
    Bar* b = new Bar;

    if ( f == b )
        std::cout << "равны" << std::endl;
    else
        std::cout << "не равны" << std::endl;

    return 0;
}
```

- да
- нет

8. Программа напечатает строку Я программист или нет?

```
struct Foo
{
    int x;
    int y;
};
```

```
int main(int argc, char** argv)
{
    Foo f;

    if ( &f.x < &f.y )
    {
        std::cout << "Я программист" << std::endl;
    }

    return 0;
}
```

}

- некорректное определение
- да
- нет

9. Каким символом завершается Си-строка?

- 'n'
- ''
- '.'
- ''

10. Что такое ссылка?

- ссылка является псевдонимом для объекта
- оператор
- используется для переименования объектов
- нет правильного ответа

11. Какая из следующих функций считывает 100 символов из входного потока в строку x?

- `cin.getline(x, 100, 'n');`
- `read(x);`
- `cin.getline(100, x, 'n');`
- `readline(x, 100, 'n');`

12. Какой порядковый номер последнего элемента массива, размер массива 19?

- порядковый номер определяется программистом
- 18
- 19

13. Укажите правильное объявление указателя в C++

- `int &x;`
- `int x;`
- `ptr x;`
- `int *x;`

14. Код, указанный ниже объявляет массив ссылок. Правда это или нет?

```
int main()
{
    int& x[50];

    return 0;
}
```

- да
- нет

15. Укажите зарезервированное ключевое слово для динамического выделения памяти!

- value
- malloc
- create
- new

16. Какая из следующих записей возвращает значение переменной a, хранящееся в памяти по адресу на который указывает указатель?

- `a ;`
- `*a;`

- val(a);
- &a;

17. Массив - это ...

- Массив - это упорядоченные в памяти элементы одного и того же типа, имеющие имя. Доступ к отдельным элементам массива осуществляется по имени массива и индексу
- Массив - это упорядоченные в памяти элементы одного и того же типа, имеющие имя. Доступ к отдельным элементам массива осуществляется по имени массива и адресу
- Массив - это упорядоченные в памяти элементы одного и того же типа, имеющие общий адрес. Доступ к отдельным элементам массива осуществляется по адресу и индексу

18. Строковый типы данных в C++

- строки в C++ представляются как массивы элементов типа char, заканчивающиеся терминатором строки - символом с нулевым значением '0'.
- строки в C++ представляются как массивы элементов типа char, заканчивающиеся терминатором строки - символом с нулевым значением ('\0').
- строки в C++ представляются как массивы элементов типа char, заканчивающиеся терминатором строки - символом с нулевым значением ('').

19. Какая из следующих функций добавляет одну строку в конец другой?

- Append ();
- stringadd ();
- strcat ();
- stradd();

20. Допустим, у нас есть код

```
char arr[8];
cin >> arr;
```

И в массив arr мы попытались записать следующий набор символов Hello World. Что в действительности будет содержать массив arr?

- Hello Wo
- Hello W
- Другой ответ
- Hello
- Hello World

21. После выполнения ряда операций с указателем, что будет выведено на экран?

```
int main(int argc, char** argv)
{
    // предположим, int занимает 4 байта
    std::cout << sizeof(int) << std::endl;

    int *x = new int;

    // предположим адрес равен 0x60450000
    std::cout << x << std::endl;

    std::cout << x + 3 << std::endl;

    return 0;
}
```

- 0x60450003

- нельзя заранее сказать, каково будет значение адреса
- некорректное определение
- 0x60450000
- 0x6045000C

22. В каком из вариантов ответов объявлен двумерный массив?

- `int array[20, 20];`
- `char array[20];`
- `int anarray[20][20];`
- `array anarray[20][20];`

23. В какой из следующих строк выполняется обращение к седьмому элементу массива, размер массива равен 10?

- `mas[7];`
- `mas[6];`
- `mas;`
- `mas(7);`

24. Какое значение будет напечатано, в результате выполнения следующего кода?

```
#include <iostream>
```

```
int main()
{
    int sum = 0;

    int array[3][] = {{0, 1, 2}, {3, 4, 5}, {6, 7, 8}};

    for (int i = 0; i < 3 ; ++i)
    {
        for (int j = 2; j < 3 ; j++)
        {
            sum += array[i][j];
        }
    }

    std::cout << sum << std::endl;

    return 0;
}</iostream>
```

- 9
- 21
- синтаксическая ошибка
- 15

25. Укажите статическую строку!

- `char string[100];`
- `"Статическая строка"`
- `'Статическая строка'`

26. Укажите правильное объявление массива!

- `int anarray[10];`
- `anarray{10};`
- `array an array[10];`
- `int anarray;`

27. В какой из следующих записей используется операция разименования?

- `1*a;`
-

- `a ;`
- `address(a) ;`
- `&a;`

28. Корректное выделение памяти

- `int a = new sizeof(int * 20);`
- `int a = new int(20);`
- `int *a = new sizeof(int * 20);`
- `int a = new int[20];`
- `int *a = new int[20];`
- `int *a = new 20;`

29. Как правильно высвободить память, после выполнения этого кода?

`char *a; a = new char[20];`

- `delete a[];`
- `delete [] a;`
- `delete a;`

30. Словосочетание "Hello world!" может быть сохранено в символьном массиве размером n элементов. Укажите чему равно n?

- 10
- 11
- 13
- 12

31. Укажите корректное определение строковой переменной

- `string mystr[20];`
- `string mystr;`
- `char mystr[20];`
- `string[20] mystr;`

32. Какая из следующих функций сравнивает две строки?

- `stringcompare();`
- `cmp();`
- `strcmp();`
- `compare();`

33. Укажите строку, которая возвращает адрес первого элемента в массиве `arr`?

- `&arr;`
- `arr[0];`
- `arr[1];`
- `arr;`

Тест 4

Структуры и файлы.

1. Какое значение будет напечатано?

```
#include <iostream>
```

```

const int x = 12;

int main(int argc, char** argv)
{
    enum dog
    {
        x = x,
        y
    };

    std::cout << y << std::endl;

    return 0;
}

```

- 12
- неопределенное
- 13

2. Какой заголовочный файл C++ содержит инструкции файлового ввода/вывода?

- fstream
- **iostream**
- ifstream
- ofstream

3. Что означает константа `ios_base::ate`, передаваемая в конструктор, в качестве аргумента?

- Открыть файл, предварительно создав его.
- При открытии переместить указатель в конец файла.
- Открыть файл только для чтения.
- Открыть файл, не создавая его.

4. При определении структуры необходимо использовать следующее ключевое слово

- structure
- **struct**
- object
- record

5. Какой из следующих классов обрабатывает процесс записи в файл?

- ofstream
- **другое**
- input_file
- ifstream

6. Какие из перечисленных констант определяют режим открытия файлов в C++?

- ios_base::trunc
- ios_base::create
- ios_base::application
- **ios_base::noreplace**

7. Укажите правильный доступ к переменной структуры!

- b>var;
- **b->var;**
- b-var;
- b.var;

8. Правильное определение структуры в C++!

- struct a_struct {int a;}
- **struct a_struct int a;**

- `struct a_struct {int a;};`
- `struct {int a;}`

9. Как организовать запись в файл?

- `a_file << "запись";`
- `a_file.printf("запись");`
- `a_file.out("запись");`
- `a_file="запись";`

10. В каком из следующих вариантов ответов выполнен корректный доступ к переменной структуры, причём структура объявлена через указатель?

- `b->var;`
- `b>var;`
- `b.var;`
- `b-var;`

11. Какое значение будет напечатано?

```
#include <iostream>
```

```
const int SIZE = 5;
```

```
struct tester
```

```
{
    int array[::SIZE];
```

```
enum
```

```
{
    SIZE = 3
};
```

```
void size()
```

```
{
    std::cout << sizeof(array) / sizeof(int);
}
```

```
};
```

```
int main(int argc, char** argv)
```

```
{
    tester t;
    t.size();
    return 0;
}
```

- 5
- 3
- заранее не известно

12. Правильное объявление переменной, типа структуры `foo`!

- `foo;`
- `struct foo;`
- `foo var;`
- `int foo;`

Тест 5

Аргументы (параметры) функции `main`, встроенные функции, локальные и глобальные переменные, функции в C++

1. Какой тип данных имеет переменная `ARGV`?

- это не переменная
- int
- char *
- char **

2. Какие из следующих функций являются встроенными?

- inline: void foo() {}
- inline void foo() {}
- void foo() inline {}
- нет правильного ответа

3. Что будет напечатано на экране, после выполнения этого кода?

```
#include <iostream>
```

```
int foo(int x, int y)
{
    return x+y;
}
```

```
int foo(const int x, const int y)
{
    return x+y+1;
}
```

```
int main(int argc, char** argv)
{
    const int x = 3;
    const int y = 2;

    std::cout << foo(x,y) << std::endl;

    return 0;
}
```

- ошибка компиляции
- 6
- 3
- 5

4. Какая строка содержит зарезервированные слова языка программирования C++?

- defaulted, goto, return, extern, private, public, protected
- if, else, for, while do, switch, continue, break
- char, int, float, doubled, short, long, unsigned, signed
- sizeof, const, typedef, static, voided, enum, struct, union

5. Будет ли работать следующий код?

```
int x = 5;
```

```
template <typename T>
class x
{
    T member;
};
```

```
int main(int argc, char** argv)
{
    class x<int> y;

    return 0;
}
```

- да
- нет

6. Что будет напечатано на экране, после выполнения этого кода?

```
#include <ostream>

int foo(int y);
int foo(int x)
{
    return x+1;
}

int main(int argc, char** argv)
{
    int x = 3;
    int y = 6;

    std::cout << foo(x) << std::endl;

    return 0;
}
```

- 4
- 9
- 3
- ошибка компиляции

7. Какие из следующих утверждений верны?

- компилятор может проигнорировать объявление встроенной функции.
- встроенные функции не должны содержать более 10 строк кода.
- встроенные функции должны возвращать значение.
- встроенные функции не могут возвращать значения.

8. Можно ли перегрузить функцию `main()`?

- да
- нет

9. Какое значение будет содержать локальная переменная `x`, в конце `main`?

```
int x = 5;
int main(int argc, char** argv)
{
    int x = x;
    return 0;
}
```

- неопределенное
- 0
- 5

10. Выберите правильное (полное) определение функции.

- ```
int funct();
```
- ```
void funct(int)
{
    cout << "Hello"
}
```
- ```
void funct(x)
{
 cout << "Hello"
}
```
- ```
int funct(int x)
{
```

```
    return x = x + 1;
}
```

11. Каков будет результат выполнения следующего кода?

```
int f(int a)
{
    return ++a;
}
int f(unsigned int a)
{
    return --a;
}
cout << f(5);
```

- ошибка компиляции
- 5
- 6
- 4

12. Укажите правильный вызов функции, предполагается, что функция была объявлена ранее.

- `funct();`
- `funct x, y;`
- `int funct();;`
- `funct;`

13. В каком порядке, эти два параметра, объявлены в функции `main`? Параметры: `argc` и `argv`

- Существует только один аргумент
- количество аргументов, затем массив аргументов
- они не объявлены в `main`
- массив аргументов, затем количество элементов

14. Что такое `ARGV[0]`?

- `ARGV[0]` нигде не используется
- массив
- первый аргумент, который передается в программу из командной строки

15. Что означает количество аргументов?

- количество аргументов
- количество аргументов, плюс один
- количество аргументов, минус один
- общий размер массива аргументов

16. Можно ли гарантировать, что объявленная встроенная функция действительно является встроенной?

- гарантировать не возможно, в каждом индивидуальном случае бывает по разному
- можно с уверенностью гарантировать, что объявленная вами функция как встроенная, действительно будет встроенной

17. Укажите тип возвращаемого значения следующей функции `int func(char x, float v, double t);`

- `float`
- `char`
- `double`
- `int`

18. Что будет напечатано на экране, после выполнения этого кода?

```
#include <iostream>

int foo(int x, int y)
{
    return x+y;
}
```

```
double foo(double x, double y)
{
    return x+y;
}

int main(int argc, char** argv)
{
    double (*ptr)(int, int);

    ptr = foo;

    std::cout << ptr(3,8) << std::endl;

    return 0;
}
```

- 11
- 8
- ошибка компиляции
- 3

19. Для чего используются встроенные функции?

- Чтобы уменьшить размер программы
- Для удаления ненужных функций
- Для увеличения скорости работы программы
- Для упрощения файла с исходным кодом

20. Возможна ли такая ситуация?

```
int x = 5;
```

```
class x
{
};
```

```
int main(int argc, char** argv)
{
    class x y;

    return 0;
}
```

- да
- нет

21. Что значит ключевое слово inline?

- нет правильного ответа
- все вызовы встроенных функции заменяются кодом этой функции
- все вызовы встроенных функций заменяются кодом этой функции
- Сообщает компилятору использовать функцию только в пределах одного файла с исходным кодом

22. Какая из переменных хранит количество аргументов, передаваемых в программу?

- argc
- argv
- arglen
- count

23. Какое значение будет содержать переменная y?

```
const int x = 5;
int main(int argc, char** argv)
{
    int x[x];

    int y = sizeof(x) / sizeof(int);
}
```

```
    return 0;  
}
```

- 5
- 20

24. Что из нижеперечисленного не является прототипом функции?

- `void funct();`
- `double funct(char x)`
- `int funct(char x, char y);`
- `char x();`

Тест 6

Объявление, определение и использование классов в C++.

1. Укажите корректное объявление класса!

- `object A { int x; };`
- `class A { int x; };`
- `class B { }`
- `public class A { }`

2. Понятие `this` в классе

- Указатель `this` является скрытым аргументом метода, существует во всех методах объекта и указывает на его (объект) адрес: `this ->< объект >`
- Указатель `this` является скрытым аргументом метода, превращает функцию в область памяти только для чтения.
- Объект `this` является аргументом метода другого класса, существует во всех методах и указывает на адрес: `this -><объект>`

3. Основные типы наследования в классах

- Открытое наследование классов позволяет выполнить образование производного класса и объекта. Закрытое наследование классов позволяет выполнить образование только производного класса. Множественное наследование - если у производного класса имеется несколько базовых классов
- Открытое наследование классов позволяет выполнить образование производного класса и объекта. Закрытое наследование классов позволяет выполнить образование только производного объекта. Множественное наследование - если у производного класса имеется несколько базовых классов
- Открытое наследование классов позволяет выполнить образование производного класса и объекта. Закрытое наследование классов позволяет выполнить образование только производного класса. Множественное наследование - если у производного класса имеется несколько закрытых классов

4. Какого спецификатора доступа в классах нет?

- `hidden`
- `private`
- `public`
- `protected`

5. Правильная конструкция определения класса в C++

- ```
Class = < имя класса >
{
```

```

 Private
 <список объявлений данных-элементов и функций-элементов,
 скрытых от доступа>
 Public
 <список прототипов функций-элементов общедоступного
 интерфейса>};

```

- ```

class < имя класса >
{
    private:
        <список объявлений данных-элементов и функций-элементов,
        скрытых от доступа>
    public:
        <список прототипов функций-элементов общедоступного интерфейса
        >
};

```
- ```

Class < имя класса >
{
 Private:
 <список объявлений данных-элементов и функций-элементов,
 скрытых от доступа>
 Public:
 <список прототипов функций-элементов общедоступного
 интерфейса>
};

```

#### 6. Что такое объект?

- Объект - это конкретное представление абстракции, которое обладает индивидуальностью, состоянием и поведением.
- Объекты это инструмент борьбы для борьбы со сложностью различных систем реальных сущностей с характеристиками – агрегация, зависимость, конкретизация.
- Объект - это конкретное представление абстракции с характеристиками – модификатором, селектором, итератором.

#### 7. Назовите принципы объектно-ориентированного представления программных систем

- абстрагирование; модульность; иерархическая организация
- абстрагирование; инкапсуляция; модульность;
- абстрагирование; инкапсуляция; модульность; иерархическая организация

#### 8. Какие функции есть у любого класса?

- нет таких
- конструктор и деструктор
- деструктор
- конструктор

#### 9. Назовите преимущества классов!

- все указанные ответы
- удобный способ моделирования объектов реального мира
- удобство повторного использования кода
- инкапсуляция данных

#### 10. Что такое деструктор?

- Деструктор - это специальная функция-элемент, которая должна отслеживать данные в экземпляре класса в процессе работы
- Деструктор - это специальная функция-элемент, которая должна уничтожать экземпляр класса после завершения его работы
- Деструктор - это функция, которая должна открывать динамическую область для экземпляра класса

#### 11. Может ли деструктор иметь аргументы?

- нет
- да

12. Что такое класс?

- Класс - это абстракция существенных характеристик системы или это описание множества систем, которые имеют свойства с одним смыслом
- Класс - это абстракция существенных характеристик объекта или это описание множества объектов, которые разделяют одинаковые свойства, операции, отношения семантику (смысл)
- Класс - это копия характеристик объекта , которые разделяют одинаковые свойства, операции, отношения семантику (смысл).

13. Какое значение должен возвращать деструктор?

- деструкторы не возвращают значение
- указатель на класс
- код состояния о правильном удалении класса
- объект класса.

14. Какая функция выполняет начальную инициализацию данных в классе?

- деструктор
- конструктор
- нет правильного ответа