

Методика решения, ВМК-20

Рассматривается язык C++ (стандарт C++17), если не указано иное; компилятор GCC 7.0 (MinGW).

Все задания предполагают выбор одного или нескольких правильных ответов.

1 Вопрос

Правильные утверждения о языке C++

- Память под все переменные выделяется статически
- Память под все переменные выделяется динамически
- Есть сборщик мусора
- Нет сборщика мусора
- нет правильных вариантов

2 Вопрос

Правильные утверждения о языке C++

- Язык с динамической типизацией (по умолчанию)
- Язык со статической типизацией (по умолчанию)
- Динамическая типизация возможна благодаря ключевому слову `auto`
- нет правильных вариантов

3 Вопрос

Правильные утверждения о языке C++

- это скриптовый язык
- это интерпретируемый язык
- это компилируемый язык
- нет правильных вариантов

4 Вопрос

Выберите правильные варианты объявления/определения переменной:

- `int x;`
- `x = 42;`
- `int x = 42;`
- `x : integer;`
- `var x: integer;`
- нет правильных вариантов

5 Вопрос

Правильно объявлены указатели ...

- `int *x;`
- `int* x;`
- `int x*;`
- `*int x;`

Правильно объявлены указатели ...

- `void *x;`
- `void* x;`
- `*void x;`
- нет правильных вариантов

6 Вопрос

Определена переменная:

`int x = 1729;`

Укажите правильные способы определения переменной-ссылки на целое число

- `int &y = x;`
- `int y = x;`
- `int& y = x;`
- `int& y = 42;`
- `int& y = &42;`
- `int &y = 42;`
- `int &y = &42;`
- `int y = &42;`
- нет правильных вариантов

7 Вопрос

Чему будет равно значение переменной `x`

`int x = 1729;`

`int &y = x;`

`y = y + 1`

- код не компилируется
- 1729
- 1730
- -32535
- -32536

8 Вопрос

Правильно определены функции

- `void foo() {}`
- `void foo() {return ;}`
- `foo() {}`
- `foo() {return ;}`
- `foo() {return 0;}`
- `foo() {return 0;}`
- нет правильных вариантов

9 Вопрос

Правильно определены функции

- `int foo() {return "it's correct answer";}`
- `int foo() {return "42";}`
- `int foo() {return ;}`
- `int foo() {return 0;}`
- нет правильных вариантов

10 Вопрос

Правильно определены функции

- `int foo(int x) {return 0;}`
- `int foo(int x) {return ;}`
- `int* foo(int x) {return 0x5f3759df;}`
- `int* foo(int x) {return ;}`
- `int* foo(int x) {return 0;}`
- `int foo(int x) {return 0;}`
- нет правильных вариантов

11 Вопрос

Определена функция

`void foo(int x){ }`

Правильный вызов функции:

- `void foo(int x);`
- `void foo(int x = 42);`
- `foo(int x);`
- `foo;`
- `foo(int);`
- `void foo(int x = 42);`
- `int x = foo();`
- `int a = foo();`
- нет правильных вариантов
- функция неправильно определена

12 Вопрос

Как изменится значение переменных

`void bar(int x, int &y) { x = 42; y = 1729; }`

`int main() {`

`int a = 0, b = 0;`

`bar(a, b);`

- Не компилируется: ошибка в main
- Не компилируется: ошибка в bar

• `a = 0, b = 0;` если `у меня 8`

• `a = 42, b = 1729;`

• `a = 0, b = 1729;`

• `a = 42, b = 0;`

13 Вопрос

`int x = rand() % 3;`

`if (x > 0) {`

`int y = 11;`

`int z = 22;`

`} else {`

`y = 33;`

`z = 44;`

`cout << y << endl;`

`cout << z;`

Наиболее вероятный вывод программы:

- 11 и 22
- 33 и 44
- варианты равновероятны
- нет правильных вариантов
- программа не компилируется
- значения переменных `y` и `z` не определены

14 Вопрос

Правильно определены массивы:

- `int * x;` *указатель*
- `int * x[128];`
- `int x * [128];`
- `int x[128];`
- `int x = int[128];`
- `int * x = int[128];`
- `int * x = new int[128];`
- `int x* = new int[128];`
- `int * x = new * int[128];`
- `int * x = new int * [128];`
- `int * x = new int * [128];`
- нет правильных вариантов

15 Вопрос.

Где объявлены динамические массивы?

- `int * x;`
- `int * x[128];`
- `int x * [128];`
- `int x[128];`
- `int x = int[128];`
- `int * x = int[128];`
- ☒ `int * x = new int[128];`
- `int * x = new + int[128];`
- `int * x = new int * [128];`
- `int * x = new int * [128];`
- нет правильных вариантов

16 Вопрос.

Правильно определены пространства имен.

- `namespace std;`
- ☒ `using namespace std;`
- `using namespace std();`
- `namespace std{;`
- `namespace fog;`
- ☒ `using namespace bar;`
- `using namespace baz();`
- `namespace bzz();`

17 Вопрос.

Правильные утверждения о динамическом массиве:

- размер массива должен быть известен до компиляции
- размер массива должен быть известен до запуска программы
- память для массива выделяется всегда во время запуска программы
- ☒ память для массива выделяется всегда во время работы программы
- память для массива выделяется всегда во время компиляции программы
- нет правильных вариантов

18 Вопрос.

переменная `x` находится внутри пространства имен `std`. Как правильно обратиться к переменной?

- `std.x;`
- `std:x;`
- ☒ `std::x;`
- `x`
- ☒ `using namespace std; x;`
- нет правильных вариантов

19 Вопрос.

`int x;`

привожу +9 ???
`if (x == 9) x = x + 1;`

`else x = x - 1;`

Значение переменной `x` после выполнения фрагмента кода равно ...

- -1
- 0
- 1
- 8
- 9
- 10
- код не компилируется
- значение не определено
- нет правильных вариантов

20 Вопрос.

`int foo(int e){`
`throw 0; return 1;}`

/...*/*

`try{`

`foo(2);`

`} catch (int e) {`

`cout << e; }`

на экран будет выведено:

- 1
- 0
- 2
- ☒ код не компилируется
- exception
- ☒ программа аварийно завершится
- нет правильных вариантов

21 Вопрос.

Бросать исключения стоит когда ...

- нужно вернуть значение из функции
- ☒ нужно сообщить вызывающему функцию коду об исключительной ситуации
- ☒ нужно сообщить вызываемой функции об исключительной ситуации
- не прошла проверка предусловия функции
- выполнение по дальнейшему алгоритму невозможно / а если функция не выбрасывает?
- нет правильных вариантов ответа

22 Вопрос

Выберите код, который подключает библиотеку для ввода и вывода в консоль

- using namespace;
- using namespace std;
- #include std;
- ☒ #include <iostream>
- ☒ #include <iostream>
using namespace std;
- using namespace std;
#include <iostream>
- нет правильных вариантов ответа

23 Вопрос

Отличия ссылок от указателей

- это одно и то же
- указатели нельзя менять
- ☒ адрес у ссылки нельзя менять
- ссылки используются для переменных, указатели для функций
- указатели используются для переменных, ссылки для функций
- нет правильных вариантов ответа

24 Вопрос

Отличия ссылок от указателей

- это одно и то же!
- ☒ ссылки не нужно разыменовывать
- указатели ~~не~~ ^{нужно} разыменовывать
- ссылка – это адрес указателя
- указатель – это адрес ссылки
- нет правильных вариантов ответа

25 Вопрос

Правильные утверждения о ссылках

- это указатели!!
- ☒ для объявления ссылки используется &
- для объявления ссылки используется *
- для разыменования ссылки используется *
- для разыменования ссылки используется &
- нет правильных вариантов ответа

26 Вопрос

Правильные утверждения о заголовочных файлах

- нужны только для программиста, компилятор их игнорирует
- нужно компилировать отдельно
- нужно передавать их имена компилятору
- ☒ могут содержать заголовки функций
- могут содержать определения функций
- нет правильных вариантов ответа

27 Вопрос

Правильные утверждения о пространствах имён (namespace)

- ☒ помогают логически отделить фрагменты кода друг от друга
- ☒ можно объявить несколько пространств имён с одинаковым именем – *если в файле 13*
- пространства имён с одинаковым именем объединяются в одно
- пространства имён с одинаковым именем создают конфликт имён –
- ☒ помогают избежать конфликта имён

28 Вопрос

Директива #include

- подключает .h и .cpp файлы с одинаковым именем
- вставляет код из заголовочного файла в месте своего приведения
- подключает .cpp файл
- ☒ подключает заголовочный файл в проект
- нет правильных вариантов ответа

29 Вопрос

В main.cpp используются функции объявленные в файле linalg.h и определенные в linalg.cpp. Все файлы находятся в одном каталоге.

Чтобы программа компилировалась нужно

- ☒ написать #include "linalg.h"
- ☒ добавить заголовочный файл в проект IDE
- добавить cpp файл в проект IDE
- нет правильных вариантов ответа

30 Вопрос

Правильные утверждения о статических (static) переменных

- не могут менять значение
- ☒ объявляются с ключевым словом static
- объявляются с ключевым словом const
- ☒ статическая переменная может быть определена в функции
- статическая переменная может быть определена только как глобальная переменная

31 Вопрос

Правильные утверждения о статических (static) переменных

- локальная статическая переменная уничтожится после завершения функции
- локальная статическая переменная НЕ уничтожится после завершения функции
- локальная статическая переменная доступна только внутри своей области видимости
- нет правильных вариантов ответа

32 Вопрос

Нужно вычислить индекс массы тела. Какая из функций лучше соответствует принципам программирования?

- ```
void bmi(){
 float h, m;
 cin >> h;
 cin >> m;
 cout << m/h/h;
```
- ```
float bmi(){
    float h, m;
    cin >> h;
    cin >> m;
    return m/h/h;
```
- ```
float bmi(float h, float m){
 cin >> h;
 cin >> m;
 return m/h/h;
```
- ```
float bmi(float &h, float &m){
    cin >> h;
    cin >> m;
    return m/h/h;
```
- ```
float bmi(float h, float m){
 return m/h/h;
```
- ```
float bmi(float h, float m){
    if (h != 0)
        return m/h/h;
    else
        throw invalid_argument("Error: h = 0");}
```

33 Вопрос

```
void foo(/* ??? */){
    cout << s1 << endl;
    for (unsigned i = 0; i < n; i++)
        cout << rand();
    s2 += itoa( rand ); }
```

Какой набор параметров подойдет лучше?

- string s1, s2, unsigned n;
- string s1, string s2, unsigned n;
- string &s1, &s2, unsigned n;
- string s1, s2, unsigned &n;
- string &s1, string &s2, unsigned &n;
- const string &s1, string &s2, unsigned n;

```
• const string s1, const string &s2,
  unsigned n;
```

```
• const string &s1, const string &s2,
  unsigned n;
```

34 Вопрос

Объявите динамический массив a. Объявите функцию, которая его принимает (имя параметра arr) Вызовите функцию.

```
#include <iostream>
void foo(int* arr)
{
    // ...
}

int main()
{
    int n = 10;
    int* arr = new int[n];
    foo(arr, n);
}
```

35 Вопрос

Объявите двумерный динамический массив m. Объявите функцию, которая его принимает (имя параметра matr. Вызовите функцию.

```
#include <iostream>
void foo(int** matr, int i, int j)
{
    // ...
}
```

```
int main()
{
    int i = 5; // строки
    int j = 4; // столбцы
    int** matr = new int*[i], // динамический
    // массив
    // 1-элемент
    foo(matr, i, j);
}
```

36. Вопрос
Для чего стоит использовать перечисляемый тип (enum)?
- Для перечисления значений статического массива
 - Для перечисления значений динамического массива
 - Для перечисления значений структуры (struct)
 - ☒ Для замены числовых условных обозначений на именованные
 - нет правильного варианта

41. Вопрос
Для чего нужны утверждения (assert)?
- заменяют if... else
 - заменяет throw
 - заменяет try{...} catch {...} {...}
 - чтобы сообщить *пользователю* об ошибке в программе
 - чтобы сообщить *пользователю* об ошибке в данных
 - чтобы сообщить *программисту* об ошибке в данных
 - ☒ чтобы сообщить *программисту* об ошибке в программе
 - нет правильных вариантов ответа

38. Вопрос
Как определить длину строки?
- ```
string s = " ... ";
```
- length( s )
  - size( s )
  - sizeof( s )
  - ☒ s.size()
  - ☒ s.length()
  - s[0]
  - s.length( s )

39. Вопрос  
Фрагмент главного файла программы:
- ```
int x = 42; глобальная
void foo(){
    int x = 1729; локальная
    cout << x; }
```

На экран будет выведено:

- 42
- ☒ 1729
- программа не компилируется, конфликт имён: x
- нет правильных вариантов

40. Вопрос
Пронумеруйте этапы компиляции программы написанной C++ в порядке их выполнения
- 4 Линковка (компоновка)
 - 1 Обработка директив препроцессора (препроцессинг)
 - 2 Перевод кода на C++ в код на языке семейства Ассемблера

41. Вопрос
Компоновка это
- ☒ объединение разрозненных файлов с машинным кодом в один файл
 - выполнение директив #include
 - подстановка кода функций в места их вызова
 - нет правильного варианта
 - подключение динамических библиотек

41. Вопрос
Что вам нравится в языке C++?

На нём можно проинициализировать всё что угодно :)

Предложите свой вопрос

- Вопрос
Какой шрифт использован в тесте

моноширинный :)

ФИО _____