

# Вопросы по лекции 7

Артамонов Ю.Н.

Филиал «Котельники» университета «Дубна»

3 октября 2018 г.

# Вопрос 1

Заполните пропуски в предложениях

- Вопрос: В языке C++ можно объявить несколько функций с одним и тем же именем, но различными типами и/или числом аргументов. Это называется \_\_\_\_\_ функций.

# Вопрос 1

Заполните пропуски в предложениях

- Вопрос: В языке C++ можно объявить несколько функций с одним и тем же именем, но различными типами и/или числом аргументов. Это называется \_\_\_\_\_ функций.
- Ответ: перегрузкой

# Вопрос 1

Заполните пропуски в предложениях

- Вопрос: В языке C++ можно объявить несколько функций с одним и тем же именем, но различными типами и/или числом аргументов. Это называется \_\_\_\_\_ функций.
- Ответ: перегрузкой
- Вопрос: \_\_\_\_\_ дает возможность обращаться к глобальной переменной с тем же именем, что и переменная в текущей области действия

# Вопрос 1

Заполните пропуски в предложениях

- Вопрос: В языке C++ можно объявить несколько функций с одним и тем же именем, но различными типами и/или числом аргументов. Это называется \_\_\_\_\_ функций.
- Ответ: перегрузкой
- Вопрос: \_\_\_\_\_ дает возможность обращаться к глобальной переменной с тем же именем, что и переменная в текущей области действия
- Ответ: унарная операция разрешения области действия ::

# Вопрос 1

Заполните пропуски в предложениях

- Вопрос: В языке C++ можно объявить несколько функций с одним и тем же именем, но различными типами и/или числом аргументов. Это называется \_\_\_\_\_ функций.
- Ответ: перегрузкой
- Вопрос: \_\_\_\_\_ дает возможность обращаться к глобальной переменной с тем же именем, что и переменная в текущей области действия
- Ответ: унарная операция разрешения области действия ::
- Вопрос: операция \_\_\_\_\_ динамически выделяет память для нового объекта.

# Вопрос 1

Заполните пропуски в предложениях

- Вопрос: В языке C++ можно объявить несколько функций с одним и тем же именем, но различными типами и/или числом аргументов. Это называется \_\_\_\_\_ функций.
- Ответ: перегрузкой
- Вопрос: \_\_\_\_\_ дает возможность обращаться к глобальной переменной с тем же именем, что и переменная в текущей области действия
- Ответ: унарная операция разрешения области действия ::
- Вопрос: операция \_\_\_\_\_ динамически выделяет память для нового объекта.
- Ответ: **new**

# Вопрос 1

Заполните пропуски в предложениях

- Вопрос: В языке C++ можно объявить несколько функций с одним и тем же именем, но различными типами и/или числом аргументов. Это называется \_\_\_\_\_ функций.
- Ответ: перегрузкой
- Вопрос: \_\_\_\_\_ дает возможность обращаться к глобальной переменной с тем же именем, что и переменная в текущей области действия
- Ответ: унарная операция разрешения области действия ::
- Вопрос: операция \_\_\_\_\_ динамически выделяет память для нового объекта.
- Ответ: **new**
- Вопрос: Двумя предопределенными в C++ потоковыми объектами являются \_\_\_\_\_ и \_\_\_\_\_.



# Вопрос 1

Заполните пропуски в предложениях

- Вопрос: В языке C++ можно объявить несколько функций с одним и тем же именем, но различными типами и/или числом аргументов. Это называется \_\_\_\_\_ функций.
- Ответ: перегрузкой
- Вопрос: \_\_\_\_\_ дает возможность обращаться к глобальной переменной с тем же именем, что и переменная в текущей области действия
- Ответ: унарная операция разрешения области действия ::
- Вопрос: операция \_\_\_\_\_ динамически выделяет память для нового объекта.
- Ответ: **new**
- Вопрос: Двумя предопределенными в C++ потоковыми объектами являются \_\_\_\_\_ и \_\_\_\_\_.
- Ответ: **cin, cout**

# Вопрос 1

Заполните пропуски в предложениях

- Вопрос: В языке C++ можно объявить несколько функций с одним и тем же именем, но различными типами и/или числом аргументов. Это называется \_\_\_\_\_ функций.
- Ответ: перегрузкой
- Вопрос: \_\_\_\_\_ дает возможность обращаться к глобальной переменной с тем же именем, что и переменная в текущей области действия
- Ответ: унарная операция разрешения области действия ::
- Вопрос: операция \_\_\_\_\_ динамически выделяет память для нового объекта.
- Ответ: **new**
- Вопрос: Двумя предопределенными в C++ потоковыми объектами являются \_\_\_\_\_ и \_\_\_\_\_.
- Ответ: **cin**, **cout**
- Вопрос: \_\_\_\_\_ функций дают возможность определить одну функцию для выполнения некоторых действий над различными типами данных.

# Вопрос 1

Заполните пропуски в предложениях

- Вопрос: В языке C++ можно объявить несколько функций с одним и тем же именем, но различными типами и/или числом аргументов. Это называется \_\_\_\_\_ функций.
- Ответ: перегрузкой
- Вопрос: \_\_\_\_\_ дает возможность обращаться к глобальной переменной с тем же именем, что и переменная в текущей области действия
- Ответ: унарная операция разрешения области действия ::
- Вопрос: операция \_\_\_\_\_ динамически выделяет память для нового объекта.
- Ответ: **new**
- Вопрос: Двумя предопределенными в C++ потоковыми объектами являются \_\_\_\_\_ и \_\_\_\_\_.
- Ответ: **cin**, **cout**
- Вопрос: \_\_\_\_\_ функций дают возможность определить одну функцию для выполнения некоторых действий над различными типами данных.
- Ответ: шаблоны

## Вопрос 2

Являются ли следующие утверждения верными?

- Вопрос: при написании комментария в виде блока, для которого требуется много строк текста, более экономичным является использование разделителя комментария C++ `//`, чем обычных разделителей C `/*` и `*/`

## Вопрос 2

Являются ли следующие утверждения верными?

- Вопрос: при написании комментария в виде блока, для которого требуется много строк текста, более экономичным является использование разделителя комментария C++ `//`, чем обычных разделителей C `/*` и `*/`
- Ответ: неверно. Каждый из обычных разделителей в стиле C необходимо написать один раз, в то время как разделитель `//` языка C++ должен появляться в начале каждой строки.

# Вопрос 2

Являются ли следующие утверждения верными?

- Вопрос: при написании комментария в виде блока, для которого требуется много строк текста, более экономичным является использование разделителя комментария C++ `//`, чем обычных разделителей C `/*` и `*/`
- Ответ: неверно. Каждый из обычных разделителей в стиле C необходимо написать один раз, в то время как разделитель `//` языка C++ должен появляться в начале каждой строки.
- Вопрос: Все вызовы в C++ выполняются с передачей параметров по значению.

# Вопрос 2

Являются ли следующие утверждения верными?

- Вопрос: при написании комментария в виде блока, для которого требуется много строк текста, более экономичным является использование разделителя комментария C++ `//`, чем обычных разделителей C `/*` и `*/`
- Ответ: неверно. Каждый из обычных разделителей в стиле C необходимо написать один раз, в то время как разделитель `//` языка C++ должен появляться в начале каждой строки.
- Вопрос: Все вызовы в C++ выполняются с передачей параметров по значению.
- Ответ: неверно. C++ допускает настоящую передачу по ссылке посредством параметров-ссылок в дополнение к использованию для этой цели указателей.

# Вопрос 3

- Вопрос: что бы произошло, если бы в языке С вы дважды объявили одну и ту же функцию с различными типами параметров? Что бы произошло в языке С++?



# Вопрос 3

- Вопрос: что бы произошло, если бы в языке C вы дважды объявили одну и ту же функцию с различными типами параметров? Что бы произошло в языке C++?
- Ответ: В языке C произошла бы ошибка компиляции, означающая, что двум функциям было присвоено одно и то же имя. В языке C++ это является примером перегрузки функций, что допустимо, и к никакой ошибке не приводит.

# Вопрос 3

- Вопрос: что бы произошло, если бы в языке C вы дважды объявили одну и ту же функцию с различными типами параметров? Что бы произошло в языке C++?
- Ответ: В языке C произошла бы ошибка компиляции, означающая, что двум функциям было присвоено одно и то же имя. В языке C++ это является примером перегрузки функций, что допустимо, и к никакой ошибке не приводит.
- Вопрос: запишите строку «Welcome to C++» в стандартный выходной поток **cout**.

# Вопрос 3

- Вопрос: что бы произошло, если бы в языке C вы дважды объявили одну и ту же функцию с различными типами параметров? Что бы произошло в языке C++?
- Ответ: В языке C произошла бы ошибка компиляции, означающая, что двум функциям было присвоено одно и то же имя. В языке C++ это является примером перегрузки функций, что допустимо, и к никакой ошибке не приводит.
- Вопрос: запишите строку «Welcome to C++» в стандартный выходной поток **cout**.
- Ответ:

```
cout<<' 'Welcome to C++';
```

# Вопрос 3

- Вопрос: что бы произошло, если бы в языке C вы дважды объявили одну и ту же функцию с различными типами параметров? Что бы произошло в языке C++?
- Ответ: В языке C произошла бы ошибка компиляции, означающая, что двум функциям было присвоено одно и тоже имя. В языке C++ это является примером перегрузки функций, что допустимо, и к никакой ошибке не приводит.
- Вопрос: запишите строку «Welcome to C++» в стандартный выходной поток **cout**.
- Ответ:

```
cout<<' 'Welcome to C++';
```
- Вопрос: прочитайте значение переменной **age** из стандартного входного потока **cin**.

# Вопрос 3

- Вопрос: что бы произошло, если бы в языке C вы дважды объявили одну и ту же функцию с различными типами параметров? Что бы произошло в языке C++?
- Ответ: В языке C произошла бы ошибка компиляции, означающая, что двум функциям было присвоено одно и то же имя. В языке C++ это является примером перегрузки функций, что допустимо, и к никакой ошибке не приводит.
- Вопрос: запишите строку «Welcome to C++» в стандартный выходной поток **cout**.

- Ответ:

```
cout<<' 'Welcome to C++';
```

- Вопрос: прочитайте значение переменной **age** из стандартного входного потока **cin**.

- Ответ:

```
cin>>age;
```

# Вопрос 4

Определите, имеются ли ошибки в следующем фрагменте программы. Для каждой ошибки покажите, как ее исправить.

```
main()  
{  
    cout << x;  
    int x = 7;  
    return 0;  
}
```

# Вопрос 5

Определите, имеются ли ошибки в следующем фрагменте программы. Для каждой ошибки покажите, как ее исправить.

```
template <class A>
int sum (int a, int b)
{
    return a+b;
}
```

# Вопрос 6

Определите, имеются ли ошибки в следующем фрагменте программы. Для каждой ошибки покажите, как ее исправить.

```
float &ref;  
*ref = 7;
```



# Вопрос 7

Определите, имеются ли ошибки в следующем фрагменте программы. Для каждой ошибки покажите, как ее исправить.

```
template <A>
A product (A num1, A num2)
{
    return num1*num2;
}
```

# Вопрос 8

Определите, имеются ли ошибки в следующем фрагменте программы. Для каждой ошибки покажите, как ее исправить.

```
char *s = new char[10];  
delete s;
```

# Вопрос 9

Определите, имеются ли ошибки в следующем фрагменте программы. Для каждой ошибки покажите, как ее исправить.

```
double cube(int);  
int cube(int);
```

## Вопрос 10

Напишите программу, которая использует шаблон функции с именем `max` для определения наибольшего из трех аргументов. Протестируйте программу, используя тройки элементов типа `int`, `char`, `float`.

# Вопрос 11

Реализуйте динамическую структуру двухсвязного списка, для которой предусмотрите следующие операции:

- добавление элемента в конец списка;
- добавление элемента в начало списка;
- добавление элемента в заданную позицию списка;
- удаление элемента из начала списка;
- удаление элемента из конца списка;
- поиск элемента в списке;
- сортировка элементов списка.