

# new (C++)

Материал из Википедии — свободной энциклопедии

**new** — оператор языка программирования C++, обеспечивающий выделение динамической памяти в куче. За исключением формы, называемой «размещающей формой new», new пытается выделить достаточно памяти в куче для размещения новых данных и, в случае успеха, возвращает адрес выделенного участка памяти. Однако, если new не может выделить память в куче, то он передаст (throw) исключение типа `std::bad_alloc`. Это устраняет необходимость явной проверки результата выделения. После встречи компилятором ключевого слова new им генерируется вызов конструктора класса<sup>[1]</sup>.

## Содержание

- 1 Синтаксис
- 2 Placement new
- 3 Реализация
- 4 Освобождение динамически выделенной памяти
- 5 Повторное выделение памяти, выделенной при помощи new[]
- 6 См. также

- 7 Ссылки
- 8 Примечания

## Синтаксис

Синтаксис `new` выглядит следующим образом:

```
p_var = new typename;
```

где `p_var` — ранее объявленный указатель типа `typename`. `typename` может подразумевать собой любой фундаментальный тип данных или объект, определенный пользователем (включая, `enum`, `class` и `struct`). Если `typename` — это тип класса или структуры, то он должен иметь доступный конструктор по умолчанию, который будет вызван для создания объекта.

Для инициализации новой переменной, созданной при помощи `new` нужно использовать следующий синтаксис:

```
p_var = new type(initializer);
```

где `initializer` — первоначальное значение, присвоенное новой переменной, а если `type` — тип класса, то `initializer` — аргумент(ы) конструктора.

new **МОЖЕТ** также создавать массив:

```
p_var = new type [size];
```

В данном случае, `size` указывает размерность (длину) создаваемого одномерного массива. Адрес первого элемента возвращается и помещается в `p_var`, поэтому

```
p_var[n]
```

означает значение `n`-го элемента (считая от нулевой позиции)

Память, выделенная при помощи `new`, должна быть освобождена при помощи `delete`, дабы избежать утечки памяти. Массивы, выделенные (созданные) при помощи `new[]`, должны освобождаться (уничтожаться) при помощи `delete[]`.

```
int *p_scalar = new int(5);  
int *p_array = new int[5];
```

Инициализаторы не могут быть указаны для массивов, созданных при помощи `new`. Все элементы массива инициализируются при помощи конструктора по умолчанию для данного типа. Если тип не имеет конструктора по умолчанию, выделенная область памяти не будет проинициализирована.

## Placement new

Существует особая форма оператора `new`, называемая `Placement new`. Данный оператор не выделяет память, а получает своим аргументом адрес на уже выделенную каким-либо образом память (например, на стеке или через `malloc()`). Происходит размещение (инициализация) объекта путём вызова конструктора, и объект создается в памяти по указанному адресу. Часто такой метод применяют, когда у класса нет конструктора по умолчанию и при этом нужно создать массив объектов. Пример вызова выглядит следующим образом:

```
#include <new> // Необходим для использования placement new

class A
{
public:
    A(int x){}
    ~A(){}
};

const int n = 50;
A* placementMemory = static_cast<A*>(operator new[] (n * sizeof(A)));
for (int i = 0; i < n; i++)
{
    new (placementMemory + i) A(rand()); //здесь память для объекта не выделяется, но инициализируется
}
//!!деинициализация памяти
for (int i = 0; i < n; i++)
{
    placementMemory[i].~A();
}
operator delete[] (placementMemory);
```

Поскольку при выделении памяти тип создаваемого объекта(ов) не был указан, компилятор не будет вызывать деструктор для каждого объекта массива, поэтому

это нужно сделать вручную, перед освобождением блока памяти.

## Реализация

В компиляторах, придерживающихся стандарта ISO C++, в случае если недостаточно памяти для выделения, то генерируется исключение типа `std::bad_alloc`. Выполнение всего последующего кода прекращается, пока ошибка не будет обработана в блоке `try-catch` или произойдет экстренное завершение программы. Программа не нуждается в проверке значения указателя; если не было сгенерировано исключение, то выделение прошло успешно. Реализованные операции определяются в заголовке `<new>`. В большинстве реализаций C++ оператор `new` также может быть перегружен для определения особого поведения.

## Освобождение динамически выделенной памяти

Любая динамическая память, выделенная при помощи `new`, должна освобождаться при помощи оператора `delete`. Существует два варианта: один для массивов, другой — для единичных объектов.

```
int *p_var = new int;  
int *p_array = new int[50];  
  
delete p_var;  
delete[] p_array;
```

Необходимо отметить, что стандарт не требует от компилятора создания диагностического сообщения при некорректном использовании `delete`; он в общем случае не может знать, когда указатель указывает на одиночный элемент, а когда — на массив элементов. Более того, использование несоответствующего освобождения является неопределённым поведением.

## Повторное выделение памяти, выделенной при помощи `new[]`

В отличие от функции `realloc` в языке Си, при помощи оператора `new[]` невозможно напрямую перераспределить уже выделенную память. Для увеличения или уменьшения размера блока памяти нужно выделить новый блок нужного размера, скопировать данные из старой памяти и удалить старый блок. Стандартная библиотека языка C++ предусматривает поддержку динамического массива, который может быть увеличен или уменьшен в своем шаблонном классе `std::vector`.

## См. также

- Распределители памяти
- `Delete (C++)`
- Обработка исключений
- `malloc`
- Пул памяти
- Синтаксис размещения

- Указатель
- Умный указатель

## Ссылки

- Документация от IBM, описывающая оператор new языка C++ (<http://publib.boulder.ibm.com/infocenter/macxhelp/v6v81/index.jsp?topic=/com.ibm.vacpp6m.doc/language/ref/clrc05cplr199.htm>)  (англ.)
- Описание оператора new применительно к Microsoft Visual Studio ([http://msdn2.microsoft.com/en-us/library/kewsb8ba\(VS.71\).aspx](http://msdn2.microsoft.com/en-us/library/kewsb8ba(VS.71).aspx))  (англ.)

## Примечания

- ↑ Всё ли мы знаем об операторах new и delete? (<http://habrahabr.ru/post/185662/>). Проверено 9 июля 2013. Архивировано (<http://www.webcitation.org/6I5qxHpCK>) из первоисточника на WebCite 14 июля 2013 года.

Источник — «[https://ru.wikipedia.org/w/index.php?title=New\\_\(C%2B%2B\)&oldid=82190351](https://ru.wikipedia.org/w/index.php?title=New_(C%2B%2B)&oldid=82190351)»

- 
- Эта страница последний раз была отредактирована 29 ноября 2016 в 07:01.
  - Текст доступен по лицензии Creative Commons Attribution-ShareAlike; в

отдельных случаях могут действовать дополнительные условия.  
Wikipedia® — зарегистрированный товарный знак некоммерческой  
организации Wikimedia Foundation, Inc.