

Lebenszyklus von Objekten

```
class MyClass {  
    public:  
    MyClass(void);  
    MyClass(int param);  
    virtual ~MyClass(void);  
};
```

```
class MyNewClass: MyClass {  
    public:  
    MyNewClass(void);  
    MyNewClass(int p1, int p2);  
    virtual ~MyNewClass(void);  
};
```

```

// -----
// MyClass
// -----
MyClass::MyClass(void)
{
}

MyClass::MyClass(int param)
{
}

MyClass::~MyClass(void)
{
}

// -----
// MyNewClass
// -----

MyNewClass::MyNewClass(void)
{
}

MyNewClass::MyNewClass(int p1, int p2):
                                MyClass(p2)
{
}

MyNewClass::~MyNewClass(void)
{
}

```

Source

Konstruktor- und Destruktor-
Aufrufe

```

MyClass a;
MyClass b(6);
MyClass *c;

void test(void)
{
    MyNewClass d(6,7);
    c=(MyClass*)new
        MyNewClass(1,2);
}

int main(void)
{
    test();
    delete c;

    return 0;
}

```

```

d.MyClass(int)
d.MyNewClass(int, int)

```

```

c->MyClass(int)
c->MyNewClass(int, int)
d.~MyNewClass()
d.~MyClass()

```

```

a.MyClass()
b.MyClass(int)

```

```

c->~MyNewClass()
c->~MyClass()

```

```

a.~MyClass()
b.~MyClass()

```