

Наследование: повторение

29 июня 2017 г.

Что будет выведено при вызове f()?

```
class Base {  
public:  
    Base() { std::cout << "Class Base - constr" << std::endl; }  
    ~Base() { std::cout << "Class Base - destr" << std::endl; }  
};
```

```
class Derived : public Base {  
public:  
    Derived( ) { std::cout << "Class Derived - constr" << std::endl; }  
    ~Derived( ) { std::cout << "Class Derived - destr" << std::endl; }  
};
```

```
void f( ){  
    Base bs;  
    Derived dr;  
}
```

Что будет выведено при вызове f()?

```
class Base {  
public:  
    Base() { std::cout << "Class Base - constr" << std::endl; }  
    ~Base() { std::cout << "Class Base - destr" << std::endl; }  
};
```

```
class Derived : public Base {  
public:  
    Derived( ) { std::cout << "Class Derived - constr" << std::endl; }  
    ~Derived( ) { std::cout << "Class Derived - destr" << std::endl; }  
};
```

```
void f( ){  
    Base* bs = new Base;  
    bs->~Base();  
}
```

Что будет выведено при вызове f()?

```
class Base {  
public:  
    Base() { std::cout << "Class Base - constr" << std::endl; }  
    ~Base() { std::cout << "Class Base - destr" << std::endl; }  
};
```

```
class Derived : public Base {  
public:  
    Derived( ) { std::cout << "Class Derived - constr" << std::endl; }  
    ~Derived( ) { std::cout << "Class Derived - destr" << std::endl; }  
};
```

```
void f( ){  
    Base* bs = new Derived;  
    delete bs;  
}
```

Что будет выведено в консоль?

```
class Base {  
    int x_;  
public:  
    void setX(int x) { x_ = x; }  
    int getX() const { return x_; }  
    ...  
};  
  
class Derived : public Base {  
public:  
    int getX() const { return 0; }  
};
```

```
void f() {  
    Base* b = new Base;  
    b->setX(10);  
    Derived* der = new Derived;  
    der->setX(4);  
    std::cout << b->getX() <<  
        " " << der->getX() <<  
        std::endl;  
}
```

Что будет выведено в консоль?

```
class Base {  
    int x_;  
public:  
    void setX(int x) { x_ = x; }  
    int getX() const { return x_; }  
    ...  
};  
  
class Derived : public Base {  
public:  
    int getX() const { return 0; }  
};
```

```
void f() {  
    Base* b = new Base;  
    b->setX(10);  
    Base* der = new Derived;  
    der->setX(4);  
    std::cout << b->getX() <<  
        " " << der->getX() <<  
        std::endl;  
}
```

Что будет выведено в консоль?

```
class Base {  
    int x_;  
public:  
    void setX(int x) { x_ = x; }  
    virtual int getX() const  
        { return x_; }  
    ...  
};  
  
class Derived : public Base {  
public:  
    int getX() const { return 0; }  
};
```

```
void f( ){  
    Base* b = new Base;  
    b->setX(10);  
    Base* der = new Derived;  
    der->setX(4);  
    std::cout << b->getX() <<  
        " " << der->getX() <<  
        std::endl;  
}
```

Что будет выведено в консоль?

```
class Base {  
    int x_;  
public:  
    void setX(int x) { x_ = x; }  
    virtual int getX() const  
        { return x_; }  
    ...  
};  
  
class Derived : public Base {  
public:  
    int getX() const { return x_ * 5; }  
};
```

```
void f( ){  
    Base* der = new Derived;  
    der->setX(4);  
    std::cout << der->getX() <<  
        std::endl;  
}
```


Что будет выведено в консоль?

```
class Base {  
    int x_;  
public:  
    void setX(int x) { x_ = x; }  
    virtual int getX() const  
        { return x_; }  
    ...  
};  
  
class Derived : public Base {  
public:  
    int getX() const  
        { return Base::getX()*5; }  
};
```

```
void f( ){  
    Base* der = new Derived;  
    der->setX(4);  
    std::cout << der->getX() <<  
        std::endl;  
}
```

Что будет выведено в консоль?

```
class Base {  
protected:  
    int x_;  
public:  
    void setX(int x) { x_ = x; }  
    virtual int getX() const  
        { return x_; }  
    ...  
};  
  
class Derived : public Base {  
public:  
    int getX() const { return x_ * 5; }  
};
```

```
void f( ){  
    Base* der = new Derived;  
    der->setX(4);  
    std::cout << der->getX() <<  
        std::endl;  
}
```

Вопросы?