



Санкт-Петербургский  
государственный  
университет

## Вопросы

- 1 Что такое наследование?
- 2 Какие бывают типы наследования?
- 3 Какие особенности наследования конструкторов, операторы присваивания?
- 4 Какие особенности наследования деструктора?
- 5 Как запретить наследование класса?
- 6 Что такое полиморфизм? Какие механизмы связывания существуют в С++?
- 7 Что такое виртуальные функции и зачем они нужны?
- 8 Для чего используют виртуальный деструктор?
- 9 Что такое абстрактный класс и чистая виртуальная функция?



Санкт-Петербургский  
государственный  
университет

# Ответы

## 1 Что такое наследование?

**Наследование** — это возможность создания иерархии классов, когда потомки наследуют все свойства своих предков, могут их изменять и добавлять свои.

```
1 class Derived : [virtual] [access-specifier] Base
2 {
3     // member list
4 };
5 class Derived : [virtual] [access-specifier] Base1,
6     [virtual] [access-specifier] Base2, . . .
7 {
8     // member list
9 };
```



Санкт-Петербургский  
государственный  
университет

## Ответы

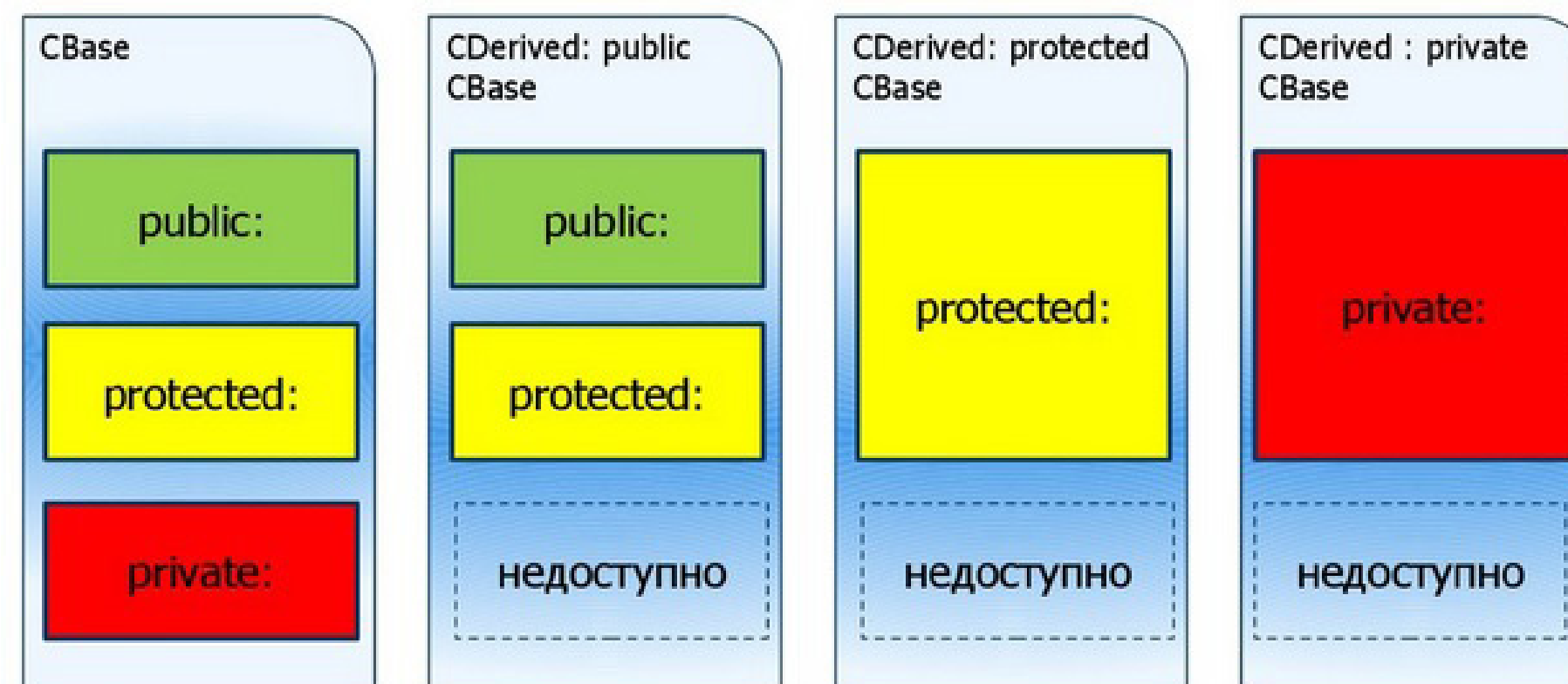
### 2 Какие бывают типы наследования?

**Наследование бывает public, protected, private.**

Наследование бывает одиночным (простым) и множественным.

Наследование бывает виртуальным / не виртуальным.

Спецификатор доступа класса контролирует разрешение производного класса на использование членов базового класса. Если спецификатор доступа опущен, то он считается private.





Санкт-Петербургский  
государственный  
университет

## Ответы

### 3 Какие особенности наследования конструкторов, операторы присваивания?

Конструкторы и операторы присваивания не наследуются.

При этом не стоит забывать, что компилятор может автоматически создавать КпоУ, КК, КП, ОПК, ОПП. При вызове одного у производного класса (ПК) будет вызван соответствующий конструктор и у базового класса (БК).

Если БК определяет КсПа, то производный не будет наследовать этот конструктор. Его надо определять.

Порядок вызова конструкторов:

1. Если в КПК явный вызов КБК отсутствует, то автоматически вызывается КБКпоУ.
2. Для иерархии, состоящей из нескольких уровней, К-ы БК-ов вызываются начиная с самого верхнего уровня (с самого базового).
3. В случае нескольких БК-ов их конструкторы вызываются в порядке объявления.

### 4 Какие особенности наследования деструктора?

Деструктор не наследуется.

При вызове деструктора производного класса (ДПК) автоматически будут вызваны все Д-ы БК-ов. Но в отличие от конструкторов, они будут вызваны в обратном порядке: сначала вызывается ДПК, затем деструкторы элементов класса, а потом ДБК.



Санкт-Петербургский  
государственный  
университет

## Ответы

### 5 Как запретить наследование класса?

Запретить наследование класса можно с помощью ключевого слова **final**.

```
1 class MyClass final {  
2     ...  
3 }
```

### 6 Что такое полиморфизм? Какие механизмы связывания существуют в C++?

**Полиморфизм** — это возможность использовать в различных классах иерархии одно имя для обозначения сходных по смыслу действий и гибко выбирать требуемое действие во время выполнения программы.

Указателю на БК можно присвоить значение адреса объекта любого ПК. Вызов методов объекта происходит в соответствии с типом указателя, а не фактическим типом объекта.

**Механизм раннего связывания** — это механизм, при котором ссылки на методы объектов разрешаются во время компиляции. И этот механизм не относится к полиморфизму.

**Механизм позднего связывания** — это механизм, когда разрешение ссылок на метод происходит во время выполнения программы в зависимости от конкретного типа объекта, вызвавшего метод. Этот механизм реализован с помощью виртуальных методов и является полиморфным.

Эти два механизма ещё называют статическим и динамическим связыванием.





Санкт-Петербургский  
государственный  
университет

## Ответы

### 7 Что такое виртуальные функции и зачем они нужны?

**Виртуальная функция** — это функция-член, которую предполагается переопределить в производных классах.

Виртуальные функции обеспечивают вызов соответствующей функции для объекта. Они обеспечивают такой принцип ООП как полиморфизм. Позднее (динамическое) связывание виртуальной функции возможно только через указатель или ссылку, а не имя объекта.

Виртуальные функции в БК должны быть определены.

```
1 // Использование имени, а не указателя или ссылки
2 // Раннее (статическое) связывание
3 DerivedA dAs;
4 Base bs = dAs;
5 bs.print(); // Base::print()
6 bs.vm(); // Base::vm()
7
8 // Невиртуальный переопределённый метод print()
9 Base *dAb = new DerivedA();
10 dAb->print(); // Base::print()
11
12 // Виртуальный переопределённый метод vm()
13 // Позднее (динамическое) связывание. Полиморфизм
14 dAb->vm(); // DerivedA::vm()
```



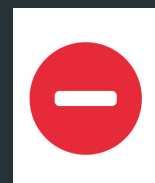
Санкт-Петербургский  
государственный  
университет

## Ответы

### 8 Для чего используют виртуальный деструктор?

Деструктор объекта вызывается по типу указателя, а не согласно фактическому типу объекта. Для этого деструктор БК должен быть виртуальным, а переопределённый деструктор в ПК должен быть описан с ключевым словом `override` (появилось в C++11).

```
1 Base dAb = DerivedA(); // ~Base()
2 DerivedA dA = DerivedA(); // ~DerivedA() -> ~Base()
3
4 DerivedA *dA_ptr = new DerivedA();
5 delete dA_ptr; // ~DerivedA() -> ~Base()
6
7 Base *dAb_ptr = new DerivedA();
8 delete dAb_ptr; // ~Base()
```



```
1 Base dAb = DerivedA(); // ~Base()
2 DerivedA dA = DerivedA(); // ~DerivedA() -> ~Base()
3
4 DerivedA *dA_ptr = new DerivedA();
5 delete dA_ptr; // ~DerivedA() -> ~Base()
6
7 Base *dAb_ptr = new DerivedA();
8 delete dAb_ptr; // OK ~DerivedA() -> ~Base()
```



### 9 Что такое абстрактный класс и чистая виртуальная функция?

Абстрактный класс (АК) предназначен для представления общих понятий, которые предполагается конкретизировать в производных классах. АК может использоваться только в качестве БК для других классов. Объекты АК создавать нельзя.

С точки зрения C++ АК — это класс, который содержит хотя бы один чисто виртуальный метод.

Чистая виртуальная функция — это виртуальная функция, не имеющая реализации. Её синтаксис объявления:

```
virtual void func() = 0;
```