**Конструкторы не наследуются**  
тип доступа к производному классу дб таким же или более строгим

Person anna = new Person();  
Person uman = new Student():  
~~uman.study()~~ (определен в произв. классе)  
\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Person per = new Person();  
Student st = new Student();  
~~st = per;~~  
per = st;  
­­\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  
\* обыч. наследование  
\* переопределение членов баз.класса в классе-наследнике (полиморфизм)  
\* сокрытие членов баз. класса в классе-наследнике

**Сокрытие членов баз. класса в классе-наследнике:**

public class A  
 public int x = 10;  
 public string ToString()  
 { return “x = “ + x; }  
public class B : A  
 public int x = 78; *//маскирует (скрывает)*  
 **new** public string ToString() *//ЯВНО скрываем метод баз.класса*  
 { return “B – “ + x + ***base***.ToString(); } *//обращ. к скрытому члену*

1) Что такое производный и базовый классы?

насл. все св-ва, методы, поля…

2) В чем заключена основная задача наследования?

Получить новый класс на основе сущ-щего  
формирует иерархию  
поощряет повт. исп. кода

3) Пусть базовый класс содержит метод basefunc(), а производный класс не имеет метода с таким именем. Может ли объект производного класса иметь доступ к методу basefunc()? Если да, то при каких условиях?

да, при обычном наследовании  
если basefunc() не private

4) Напишите объявление конструктора без аргументов для производного класса В, который будет вызывать конструктор без аргументов базового класса А.

class X  
 public X() {} *//к-ры не наслед.*  
 public X(int key) {}  
class Y : X  
 public Y(int key) {}  
 public Y() : ***base***(125) {}  
void main()  
 Y y0 = new Y(); *//авто- вызов к-р базового класса без парам*  
 Y y1 = new Y(125);

5) Что такое полиморфизм? Приведите пример.

переопределение членов баз. класса  
м. измен. ф-ции

**virtual** public void A\_method() {}  
переопределение вирт.метода в произв. классе:  
**override** public void A\_method() {}

вызов метода по типу ссылки *//просто*  
вызов метода по типу объекта *//virtual + override*

6) Определите назначение виртуальных функций.

**override** – изменение аб. или вирт. реализации  
**virtual** – разрешает переопределение (↑)

переопр.:  
\* те же парам.  
\* не static, не abstract  
\* вызов ближ. вариант (вверх по иерархии)

7) Кому доступны переменные с модификатором protected?

базовому и производным классам

8) Наследуются ли переменные с модификатором private?

нет

9) As, is – что это, как применяется?

**Is** – можно преобразовать данное выр. в указ. тип? (t/f)  
int j = 12  
object boxed = j  
object obj = new Object()  
bool a = boxed is int //true  
bool b = obj is int //false

**As** – позв. преобр. тип в опр. ссылочный тип:  
операнд as <тип> //если оп-д имеет <тип>, мб неявно-, мб упакован

10) Поддерживает ли С# множественное наследование?

Для классов запрещено  
Для интерфейсов можно

11) Можно ли запретить наследование от класса?

**sealed** – запрещает другим классам наследовать от данного  
*sealed class B : A {} – класс В наследует от класса А, но никакие не могут насл. от В*при применении к методу/свойству, исп. OVERRIDE

12) Можно ли разрешить наследование класса, но запретить перекрытие метода?

class A  
 public virtual int Sum()  
 { return 2 + 3; }  
class B : A  
 public **sealed override** int Sum() *//метод в незапеч. классе явл. запеч.* { return 2 + 5; }

13) Что такое абстрактный класс?

\* предост. базовый функционал для насл.  
\* служит только для порождения потомков  
\* м. сод. абстр. методы  
\* м. опред. методы, пер., к-ры, св-ва  
\* нельзя созд. объекты абстр. класса

abstract class Shapes  
 abstract public int Area();  
  
class Square : Shapes {  *//д. обесп. реализацию Area* public override int Area()  
 { return side\*side; }14) В каком случае вы обязаны объявить класс абстрактным?

если он сод. абстр. метод

15) В чем разница между абстрактными и виртуальными классами? Между виртуальными и абстрактными методами?

аб. метод – виртуальный  
вирт. метод разрешает переопределение  
абстр. метод требует реализацию в насл.

16) Какие компоненты класса мб виртуальными?

методы, свойства, индексаторы

17) Что такое интерфейс?

позвол. опр. требования к реализации (контракт)  
задает абс. методы, св-ва, события, индексаторы,  
кот. дб реализ. в производных классах

Interface IDo  
 void Go()  
 int Jumn (int a)  
 void Sleep()  
 int Energy { get; }

20) Почему нельзя указать модификатор видимости для методов интерфейса?

public по умолч.

21) Можно ли наследовать от нескольких интерфейсов?

да, через запятую

22) Назовите отличия между интерфейсом и абстрактным классом.



23) Для чего используются стандартные интерфейсы ICloneable, IComparable, IComparer, IEnumerable?

ICloneable – клонир. объектов (пов. и глуб.)  
ICompareable – сравн. объектов для выяснения порядка  
IComparer – сравн. объектов  
IEnumerable – перебор эл-тов необобщенной коллекции