Ответы на вопросы лабораторная 5  
1. Для чего используют статические классы?

Статический класс может использоваться как обычный контейнер для наборов методов, работающих на входных параметрах, и не должен возвращать или устанавливать каких-либо внутренних полей экземпляра. Например, в библиотеке классов .NET статический класс System.Math содержит методы, выполняющие математические операции, без требования сохранять или извлекать данные, уникальные для конкретного экземпляра класса Math. Это значит, что члены класса применяются путем задания имени класса и имени метода.

Для явной реаоизации(explicit);

2. Что может содержать статический класс?

Статические методы могут быть перегружены, но не переопределены, поскольку они относятся к классу, а не к экземпляру класса.

Статический класс может содержать только статические поля, статические методы и статический конструктор.

Объекты статического класса запрещены!

3. Что такое производный и базовый классы?

Базовый класс содержит методы и элементы старого класса и передает их в производный. (Наследование).

Производному классу доступны

public, internal, protected и protected internal

члены базового класса (private – недоступны)

тип доступа к производному классу

должен быть таким же, как и у базового

класса или более строгим.

Ссылке на объект базового класса можно

присвоить объект производного класса (но

вызываются для него только методы и

свойства, определенные в базовом классе.)

наследуются все свойства, методы, поля и

т.д., которые есть в базовом классе

не наследуются конструкторы базового

класса

4. Как используют ключевое слово base?Когда в производном классе указывается ключевое слово base, вызывается конструктор из его непосредственного базового класса. Следовательно, ключевое слово base всегда обращается к базовому классу, стоящему в иерархии непосредственно над вызывающим классом. Это справедливо даже для многоуровневой иерархии классов.

Base Ключевое слово используется для доступа членов базового класса внутри производного класса:

Вызов метода базового класса, который был переопределен другим методом.

Укажите, какой конструктор базового класса следует вызывать при создании экземпляров производного класса.

Доступ к базовому классу разрешен только в конструкторе, методе экземпляра или методе доступа к свойству экземпляра.

Использование base ключевого слова из статического метода является ошибкой .

Базовый класс, к которому осуществляется доступ, - это базовый класс, указанный в объявлении класса. Например, если вы укажете class ClassB : ClassA, доступ к членам ClassA будет осуществляться из ClassB, независимо от базового класса ClassA.

5. В чем заключена основная задача наследования?

Наследование - это механизм получения

нового класса на основе уже существующего.

Роль наследования

►формирует иерархию

►поощряет повторное использование кода

Правила наследования:

1) В C# наследование всегда

подразумевается открытым

2) Запрещено множественное наследование

классов (но не интерфейсов)

3) наследуются все свойства, методы, поля и

т.д., которые есть в базовом классе

4) Производному классу доступны

public, internal, protected и protected internal

члены базового класса (private – недоступны)

class Student : Person

5) не наследуются конструкторы базового

класса

6) тип доступа к производному классу

должен быть таким же, как и у базового

класса или более строгим

7)Ссылке на объект базового класса можно

присвоить объект производного класса (но

вызываются для него только методы и

свойства, определенные в базовом классе.)

6. 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

Может при условиях (Когда в производном классе указывается ключевое слово base, вызывается конструктор из его непосредственного базового класса).

7. Напишите объявление конструктора без аргументов для производного класса B,

который будет вызывать конструктор без аргументов базового класса A.

using System.Collections.Specialized;

using System.Drawing;

using System.Runtime.ConstrainedExecution;

using System.Xml.Serialization;

namespace ЛР\_\_5

{

class Program

{ class A

{

public A() { }

public A(int a) { }

}

class B : A

{

public B(int a) { }

public B() : base(111) { }

}

static void Main(string[] args)

{

B b1 = new B();

B b2 = new B(111);//вызов абзового констурктора без парметров

}

}

}

8. Что такое полиморфизм? Приведите пример

Полиморфизм

►ключевой аспект объектноориентированного программирования

►способность к изменению функций,

унаследованных от базового класса.

Перегрузка методов обеспечивает

статический полиморфизм, а

виртуальный метод – динамический.

Полиморфизм (при неявной реализации impicit) может быть (virtual), что позволяет переписывать эту реализацию в классах потомках.

Полиморфизм (при явной (explicit) реализации) всегда статичен. Не может быть быть переопределен (override) или перекрыта (new) в классах потомках.

9. Определите назначение виртуальных функций.

полиморфный интерфейс в базовом классе -

набор членов класса, которые могут быть

переопределены в классе-наследнике

virtual public void A\_method() { }

переопределение виртуального метода в

производном классе:

override public void A\_method() { }

Правила переопределения

►1) Переопределенный виртуальный метод

должен обладать таким же набором

параметров, как и одноименный метод

базового класса.

►2) не может быть static или abstract

►3) вызывается ближайший вариант,

обнаруживаемый вверх по иерархии

(многоуровневая)

Виртуальные функции предоставляют

механизм позднего (отложенного) или

динамического связывания.

10. Кому доступны переменные с модификатором protected?

Protected доступны: interface, abstract class, class, sealed class.

Не доступны СТРУКТУРАМ!

11. Наследуются ли переменные с модификатором private?Производному классу доступны

public, internal, protected и protected internal

члены базового класса (private – недоступны)

12. As, is – что это, как применяется? В чем между ними отличие ?

Операция as

позволяет преобразовывать тип в

определенный ссылочный тип с

применением следующего синтаксиса:

операнд as <тип>

Выполняется

►Если <операнд> имеет тип, заданный в

<тип>.

► Если <операнд>, может быть неявно

преобразован в <тип>.

►Если операнд <операнд>, может быть

упакован в <тип>.

операнд as <тип>

Операции is

Возвращает булевское значение,

говорящее о том, можете ли вы

преобразовать данное выражение в

указанный тип

Оператор is никогда не генерирует

исключение.

13. Поддерживает ли C# множественное наследование?

Запрещено множественное наследование

классов (но не интерфейсов)

14. Можно ли запретить наследование от класса?

Да через модификатор sealed.

28. В какой строке приведенного ниже фрагмента листинга не содержится ошибки?

class A

{

public virtual abstract void m() { } //1

public virtual void g() { }//2

public virtual new new void f() { }//3

public static virtual void h() { }//4

}

29. Что будет выведено на консоль в результате выполнения следующего фрагмента?

class A

{

public int x = 1;

}

class B : A

{

public new int x = 2;

public void m(int a, int b)

{

x = a;

base.x = b;

Console.Write(x + " " + base.x);

}

}

class Test

{

static void Main(string[] args)

{

A a = new A();

B b = new B();

b.m(3, 4);

}

}Ответы 3 4;

30. Что будет выведено на консоль в результате выполнения следующего фрагмента кода?

class A

{

public class B : A

{

public override void mA()

{

Console.WriteLine("B ");

}

}

public virtual void mA()

{

Console.WriteLine("A ");

}

}

class Prоgrаm

{

static void Main(string[] args)

{

A a = new A();

A.B b = new A.B();

a.mA();

b.mA();

}Ответ: A B;

31. Чем может быть M4 если дано следующее определение:

public class C1 : M1, M2 { }

public struct S1 : M3, M4 { };

Варианты ответа:

1) M4 - только интерфейс

2) M4 - интерфейс или класс

3) M4 - только класс

4) M4 - только структура

5) M4 – делегат

32. Выберите верное присваивание для объектов, определенных в листинге.

class A { }

class B : A { }

class C : B { }

class D { }

class Test

{

static void Main(string[] args)

{

A a = new A();

B b = new B();

C c = new C();

D d = new D();

}

}

Варианты ответа:

1) b = a;

2) a = b;

3) c = a;//не уверена

4) d = a;

5) с = b;

33. Что будет выведено на консоль в результате выполнения следующего фрагмента, если

раскомментировать строчку 1?

public abstract class A

{

public virtual void method()

{ Console.Write("A "); }

}

public class B : A

{

public override void method()

{ // base.method(); // 1

// this.method(); // 2

Console.Write("B ");

}

}

class Program2

{

static void Main(string[] args)

{

A my = new B();

my.method();

}

}

Варианты ответа:

1) B

2) A

3) A B

4) B A

5) 0

15.Можно ли разрешить наследование класса, но запретить перекрытие метода?

Да. Указываем класс как public, а метод как sealed.

16. Что такое абстрактный класс?

Это класс, объект которого не может быть создан. Такой класс должен иметь класс-наследник с реализацией абстрактных методов. Абстрактный класс – это фактически чертёж нормального класса без реализации.

17. В каком случае вы обязаны объявить класс абстрактным?

1. В том случае, если класс является наследником абстрактного класса, но не все методы базового класса перекрыты и имеют реализацию.

2. В том случае, если хотя бы один метод класса является абстрактным.

18. В чем разница между абстрактными и виртуальными классами? Между

виртуальными и абстрактными методами?

Абстрактный класс это класс, содержащий хотя бы один метод (abstract).. Виртуальный метод имеет реализацию и м.б. переопределен в производном классе. Абстрактный метод не имеет реализацию, только описание метода, который д.б. реализован в производных классах.

19. Какие компоненты класса могут быть виртуальными?

Виртуальными могут быть:

· Методы

· Свойства

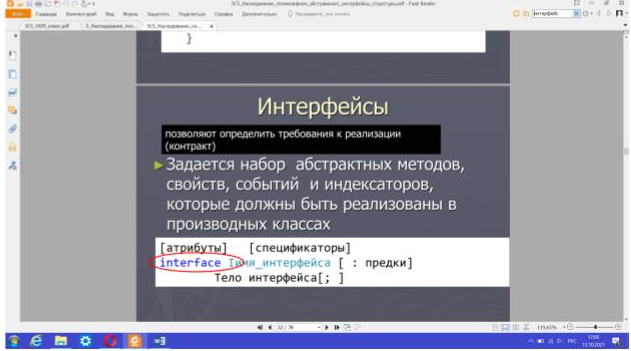
· Индексаторы

· События

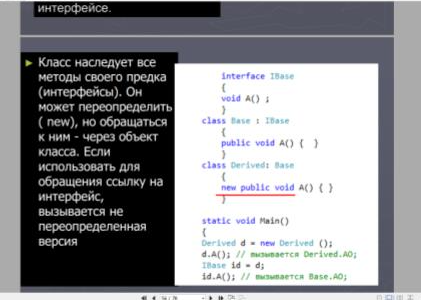
20. Что такое интерфейс?

Интерфейсы, как и классы, определяют набор свойств, методов и событий. Но, в отличие от классов, они не содержат их реализации. Интерфейсы реализуются классами и определяются как самостоятельные сущности.

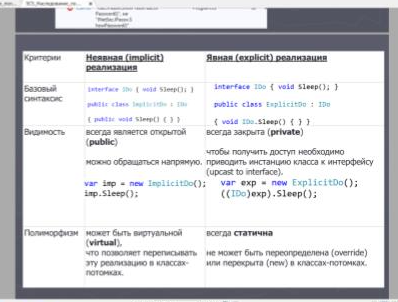
21. Что может содержать интерфейс?



22. Как работать с объектом через унаследованный интерфейс?



23. Приведите пример явной реализации интерфейса.



24. Почему нельзя указать модификатор видимости для методов интерфейса?

Потому что все они должны иметь модификатор public, который и установлен по умолчанию.

25. Можно ли наследовать от нескольких интерфейсов?

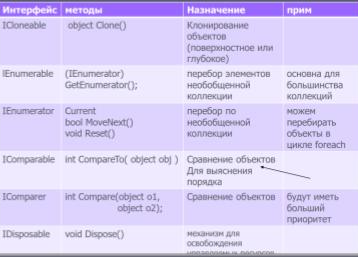
Да, перечисляются через запятую

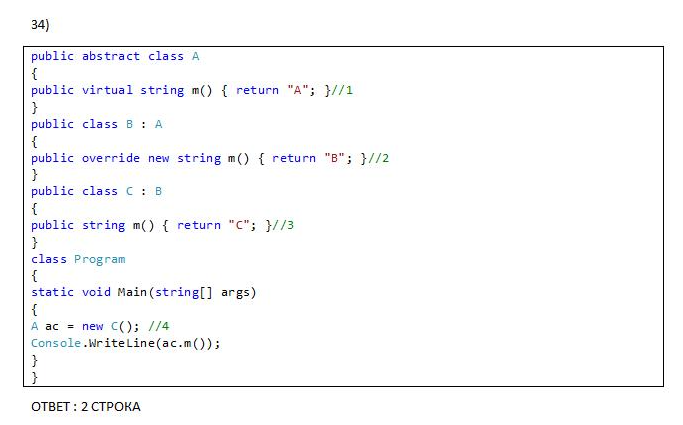
26. Назовите отличия между интерфейсом и абстрактным классом.

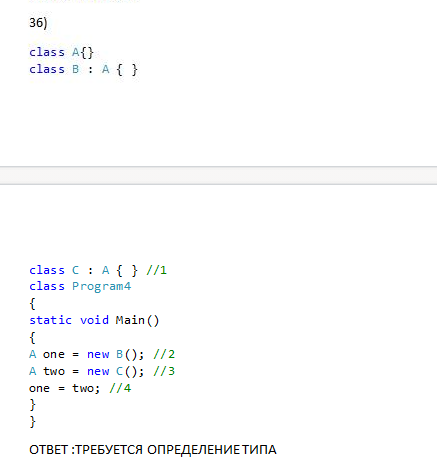
В интерфейсе все методы (свойства и т.д.) абстрактны и не имеют реализации. В абстрактном классе некоторые методы могут быть реализованы. В интерфейсе члены не могут иметь модификатора видимости (все они являются public по умолчанию), а в абстрактном классе члены могут иметь модификатор видимости.

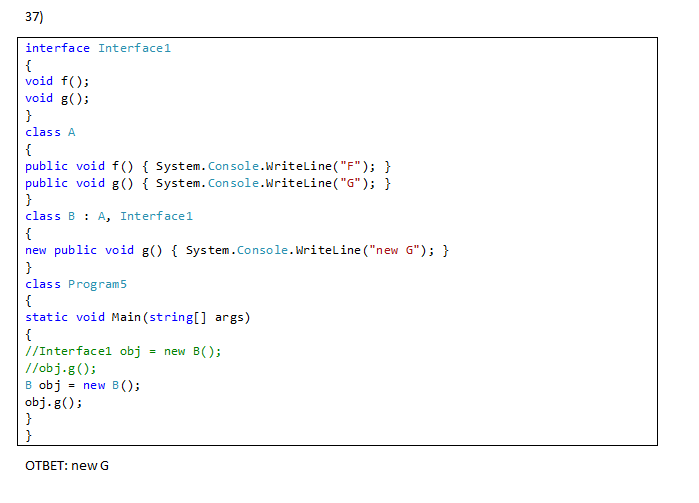
27. Для чего используются стандартные интерфейсы ICloneable, IComparable, IComparer,

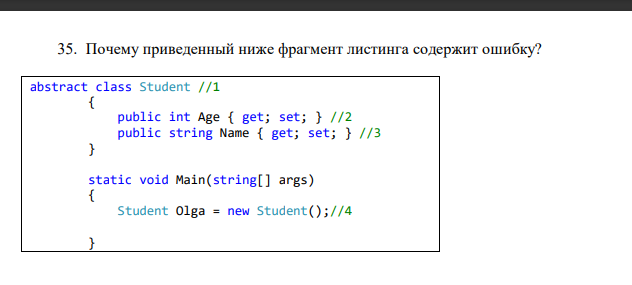
lEnumerable?











4… нельзя создавать экзмепляры абстрактных классов