Члены класса подразделяются на поля, свойства, методы, события.  
*поле* – член-переменная, сод. нек. значение (public int x)  
*константы* – поле, объяв. с пом. const (public cons tint x=25)

1. Назовите принципы ООП. Поясните каждый из них.

**ООП** – программа предст. в виде сов. объектов, каждый – экземпляр класса, а классы образ. иерархию наследования.

**Принципы:**  
\* наследование (объект наслед. свойства другого)  
\* инкапсуляция (связ. данные+код, скрывает детали реализации от польз.)  
\* полиморфизм (вызывать метод потомка через экземпляр предка)  
\* абстракция д-х (разделения интерфейса и реализации)

2. Назовите класс .NET, от которого наследуются все классы.

System.Object

3. Охарактеризуйте открытые методы System.Object

\* ToString() – строковое представление объекта  
\* GetHashCode() – возвр. хэш-код для значения данного объекта  
\* Equals() – срав. 2 объекта на рав-во  
\* GetType() – возвр. тип объекта

4. Охарактеризуйте закрытые методы System.Object

\* Finalize() – очистка ресурсов ссылочных объектов  
\* Clone() – созд. копию объекта и возвр. ее ссылку

5. Приведите пример определения класса.

class CA  
{  
 public int x;  
 protected float m;  
 decimal sum;  
}

6. Какие ключевые слова можно использовать при определении класса?

**abstract** – класс обязан иметь производные классы  
*abstract class Shapes {  
 abstract public int Area();  
}  
class Square : Shapes { //д. обесп. реализацию Area   
 public override int Area() {  
 return side\*side;  
 }*м. сод. абстрактные методы и методы доступа  
созд. экземпляры абстр. класса нельзя  
производный аб. класса д. включать реализации всех наслед. аб. методов и методов доступа  
аб. метод – виртуальный  
объявления аб. методов только в аб. классах  
\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

**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  
структуры неявно запечатаны

7. В чем отличие между объектом и классом?

Класс опр. тип объекта, но не явл. объектом  
Объект – сущность, осн. на классе (экземпляр класса)

8. Что такое конструктор? Когда вызывается конструктор?

Если не создать констр. класса, C# создаст его по умолч. (все 0)

Конструктор – это метод класса, имеющий имя класса.  
*public class Person {  
 private string last;  
 private string first;*

*public Person (string lastName, string firstName) { //нет тд, только имя+парам.  
 last = lastName;  
 first = firstName;  
 }  
}*

авто- вызывается при компиляции при созд. экземпляра  
нельзя вызвать явно  
*class CA {  
 public int x,y,z;  
 public CA() { //по умолч. (без парам.)  
 x=3; y=2; z=x+y;  
 }  
}*

9. Перечислите свойства конструктора?

1. не м. возвр. значения даже void  
2. имя = имя класса  
3. не наследуются  
4. нельзя примен. virtual, new, override, sealed, abstract  
5. для класса без явно-зад. конструкторов компилятор созд. к.по умолч.  
6. для статических классов – 5 нет  
7. мб неск.

10. Что такое деструктор?

вызыв. перед уничтожением объекта системой «сборки мусора»

11. Что такое this?

ссылается на тек. объект  
*public classEmployee {  
 private string alias;  
 private string name;*

*public Employee (string name, string alias) {  
 this.name = name;  
 this.alias = alias;  
 }  
}*

13. Какие спецификаторы доступа для класса и методов класса существуют в С#?

\* public – общедоступный член класса  
\* private – член класса доступен только внутри данного класса (по умолч.)  
\* protected – член класса доступен только внутри данного класса и внутри классов, производных от данного  
\* internal – член класса доступен только внутри данной программы

14. Опишите модификатор protected internal.

доступ из тек. сборки или из производного класса в другой сборке

15. Зачем и как используются ref и out параметры функции?

ref – указ., что аргумент передается по ссылке (не по значению)  
(дб иниц. перед вызовом ф-и)

*void Method (ref int a) {  
 a = a+44;  
}  
int num = 1;  
Method (ref num);  
Console.WriteLine(num); //45*

Методы – обычные функции С стиля, но при передаче параметров надо указ. ref/out  
они сообщ. компилятору, что адреса параметров совп. с адресами пер-ных  
любое изменение значения пар. изменит и пер-ные

для входного пар. – **ref**  
для выходного пар. - **out**

16. Приведите пример необязательных и именованных параметров метода.

*static int Sum (int a, int b = 5) { //b – необяз. пар.  
 return a=b;  
}*

*int sum1 = Sum (a : 3, b : 10) //именован. арг.*

17. Приведите пример полей класса – статические, константные, только для чтения.

Поля для чтения – конст. поле, нач. значение мб вычислено в процессе вып.  
public readonly int rd;

public const int x = 25;

18. Приведите пример определения свойств класса. Как свойства связаны с инкапуляцией?

**Свойства** – упр. доступом к полям класса  
сост. из поля и аксессора (get/set)  
вызов – как метод

private int m=35;  
public int y  
{  
 get { return m; } //возврат значения свойства  
 set { m=value; } //присвоение нового значения  
}

Student Марина = new Student();  
Марина.Name = “Марина”; (сраб. set перед. в value)  
String nameMa = Марина.Name; (сраб. get)

для чтения и записи (get, set)  
только для чтения (get)  
только для записи (set)

19. Назовите явное имя параметра, передаваемого в метод set свойства класса.

ключ. слово **value** исп. для определения значения, присв. с пом. set

20. Что такое автоматические свойства?

тип имя { get; set; }  
компилятор авто- реализует методы возвр. и присв. значений полю

21. Что такое индексаторы класса? Какие ограничения существуют на индексатор?

**Индексатор** – позв. индексировать экземпляры класса, как массивы  
в кач. парам – индекс массива  
в кач. объекта – массив  
в кач. имени класса – this

class Coll<T> {  
 private T[] arr = new T[100];

public T this[int i] {  
 get { return arr[i]; }  
 set {arr[i] = value; }  
 }  
}  
…  
var str = new Coll<string>();  
str[0] = “hello”;

1) значение, получ. индексатором, нельзя передать методу в кач. параметра ref/out  
2) не мб static, т.к. дб членом своего класса

22. Что такое перегруженный метод?

один и тот же метод, но с разным набором параметров  
позв. обращ. к связанным методам через общее имя  
д. отличаться кол-вом или типов параметров

23. Что такое partial класс и какие его преимущества?

раздел. класс на неск. файлов  
+ крупные проекты

24. Что такое анонимный тип в C#?

позв. создать объект с нек. набором св-в без определения класса  
var someType = new {Name = “Anna”};

25. Для чего делают статические классы?

Переменные и св-ва, кот. хранят состояние, общее для всех объектов класса, наз. **статическими.**методы, кот. опр. общее поведение объектов, наз. **статическими**

26. В чем отличие статического поля от экземплярного?

Поля экземпляра отн. только к экземпляру типа:  
 класс Т  
 поле экземпляра F  
 м. создать 2 объекта типа Т и изменить значение F в каждом  
 они не влияют друг на друга

Статическое поле принадл. к классу, явл. общим для всех экземпляров

27. Поясните работу статических конструкторов.

\*закрытые авто-  
\* не им. парам.  
\* нельзя вызвать явно

28. Какая разница между поверхностным (shallow) и глубоким (deep) копированием?

**неглубокое** – для примитивных типов  
**глубокое** – для сложных объектов

*public interface CCC {  
 object Clone();  
}  
class Company {  
 public string Name { get; set; }  
}  
class Person : CCC {  
 public string Name { get; set; }  
 public Company Work { get; set; }*

*public object Clone() {  
 return this.****MemberwiseClone****(); //неглубокое  
 Company comp = new Company {Name = this.Work.Name};  
 return new Person {  
 Name = this.Name;  
 Work = company  
 }  
}  
…  
Person p1 = new Person {Name = “Tom”, Work = new Company {Name=”Microsoft”}};  
Person p2 = (Person)p1.Clone(); //у Тома останется  
p2.Work.Name = “Google”; //у Тома станет тоже*

29. В чем разница между равенством и тождеством объектов?

равные сод. одну ссылку  
тождественные совп. по содержимому

30. Что такое частичные классы и частичные методы?

ч. методы:  
\* внутри частичного класса  
\* д. иметь тип void  
\* не м. иметь парам. out  
\* private не пишется (закрыт)