1. Принципы ООП:
   1. Инкапсуляция - скрытие внутреннего состояния и функций объекта и предоставление доступа только через открытый набор функций.
   2. Наследование - возможность создания класса на основе существующего.
   3. Полиморфизм - возможность реализации наследуемых свойств или методов, общих свойств в одном классе.
   4. Абстракция - моделирование требуемых атрибутов и взаимодействий сущностей в виде классов.
2. Все классы наследуются от System.Object.
3. Открытые: Equals (сравнения состояния), Finalize(освобождение ресурсов), GetHashCode, ToString, GetType. Можно вызвать из вне.
4. Закрытые: можно вызвать только внутри, их нет.
5. Класс – это модель для создания объектов.

public class MyClass

{

// Поле класса

private string message;

// Конструктор класса

public MyClass(string message)

{

this.message = message;

}

// Метод класса

public void DisplayMessage()

{

Console.WriteLine(message);

}

}

1. Public, private, protected, internal, protected internal, private protected. || static, partial, sealed. || class
2. Класс – это модель, шаблон для объекта. Объект – это конкретный экземпляр класса, имеющий его свойства.
3. Конструктор – это метод, вызываемый при создании экземпляра класса. Используются для инициализации объектов, задать переменные внутри класса.
4. Свойства: совпадает с именем класса, может иметь уровень доступа, ничего не возвращает, может быть статичным, вызывается при создании экземпляра класса, их может быть несколько, позволяет задавать значения по умолчанию.
5. Деструктор – это метод внутри класса, который используется для уничтожения экземпляра этого класса. Вызывается автоматически сборщиком мусора. ~Destructor() – с тильды начинается, имя как у класса.
6. This ставиться перед названием переменной в классе, в случае, когда переменная класса и метода имеют одно название.
7. Вообще ничего, т.к. сначала static main должен идти, но по идее 5 5, 7 7;
8. Public – из любой сборки и места в коде. Private – нигде, кроме кода класса. Protected – только в коде класса и наследуемом классе. Internal – только в этой сборке. Protected internal – любой код в этой сборке и из наследуемого класса другой сборки. Private protected – в наследуемых классах этой сборки.
9. Доступ разрешен в коде сборки и в наследуемом классе другой сборки.
10. Ref – для указания ссылки на переменную и взаимодействие именно с ней в функции. Out – для указания, что параметр выводится и должен быть изменен методом.
11. Параметры по умолчанию, то, что может быть опущено при вызове метода. Именованные позволяют менять порядок переменных при вызове.
12. Static – единая переменная для всего класса, а не объекта. Const = не изменяемая после инициализации. Readonly – можно задать значение при объявлении, но нельзя менять после инициализации.
13. Используется для чтения и записи приватных полей через get и set. Свойства позволяют скрывать приватные поля класса и предоставлять доступ и изменение их. Устанавливает контроль, когда и как поле может быть изменено.
14. Value.
15. Автоматическое свойство – это свойство, которое не требует явного определения полей класса и не требует наличия private поля. Оно само создает скрытое поле.
16. Индексатор – это член класса, позволяющий экземплярам класса индексироваться так же, как и массивам. Не могут быть статическими, ref и out недоступны, минимум один параметр.
17. Перегруженный метод – это метод, имеющий одно имя, но разные параметры, пример полиморфизма. Программа сама определяет, какой из двух одинаковых по имени методов применить для данного конкретного параметра.
18. Это класс, который может быть разделён на несколько частей в разных файлах и использовать все поля и функции этого же класса. Плюсы в том, что можно разделить файл класса на несколько.
19. Анонимный тип – это класс, который не имеет имени и содержит только общедоступные свойства только для чтения.
20. Это класс, который не может быть инстанцирован, т.е. не может иметь созданный объект. Они нужны для создания производных классов.
21. В том, что статическое поле едино для всех объектов класса. Экземплярное – только для этого объекта.
22. Статический конструктор используется для инициализации любых статических данных при выполнении определенного действия только один раз. Вызываются автоматически при создании первого экземпляра. Только один.
23. Поверхностное – создает новый объект и вставляет в него ссылку на оригинальный объект. Глубокое же создает новый объект и рекурсивно вставляет в него копии оригинальных объектов. Изменение оригинала не отразится на глубинной копии.
24. Равенство – полное совпадение двух объектов или выражений. Тождество – на эквивалентность или идентичность, в программировании тождество проверяется нахождением в одной области памяти. == равенство, is тождество.
25. Частичные классы позволяют разбивать класс на части для хранения в разных файлах. Частичные методы же определяются в одном классе, а реализуются в другом файле этого класса.
26. System.Int32
27. A= 2, b=1;
28. 2, конструктор не должен возвращать значение
29. От 0 до 2.
30. В пункте 4 требуется задать две переменные, а мы вводим одну. Приоритет у публичного конструктора.
31. A static и A.
32. Пункт 3, так как присвоение значение только для чтения у статического поля может быть только в статическом конструкторе.