1. Назовите принципы ООП. Поясните каждый из них.

►Наследование (Inheritance); процесс, благодаря которому один объект может наследовать свойства от другого объекта.

►Инкапсуляция (Encapsulation); механизм, связывающий вместе данные и код, обрабатывающий эти данные, и сохраняющий их от внешнего воздействия и ошибочного использования.

►Полиморфизм (Polymorphism) — позволяет разным сущностям выполнять одни и те же действия. Полиморфизм позволяет повысить коэффициент повторного использования кода. Программист может сделать два разных метода — один для картинки, другой для видео. А может воспользоваться полиморфизмом и создать один метод «Отправить» для обеих сущностей.

Статический – перегрузка методов и операторов.

Динамический – наследование, реализация интерфейсов, переопределение методов

►Абстракция подразумевает разделение и независимое рассмотрение интерфейса и реализации

абстракция - уровень описания/представления модели чего либо

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

По умолчанию **все классы наследуются** от базового **класса** System.Object, даже если мы явным образом не устанавливаем **наследование**.

3. Охарактеризуйте открытые методы System.Object.

***bool Equals()*** сравнивает объект с переданным объектом на тождество.(если есть переопределение – не используется)

***int GetHashCode*** возвращает хеш-код объекта.

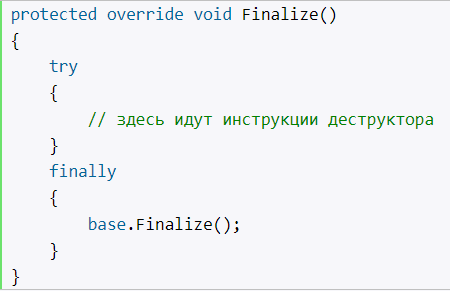
***Type GetType()*** возвращает тип текущего объекта в виде объекта **System.Type**.

***string ToString*** возвращает строковое представление объекта. (для классов – полное название класса с указанием пространства имен)

4. Охарактеризуйте закрытые методы System.Object.

1.***Finalize()*** позволяет объекту попытаться освободить ресурсы и выполнить другие операции очистки, перед тем как он будет уничтожен во время сборки мусора.

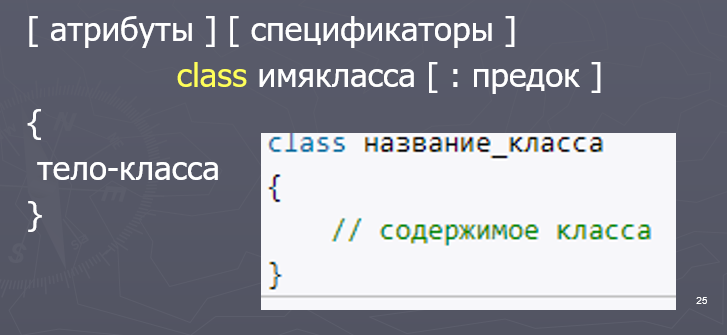
Однако на деле при очистке сборщик мусора вызывает не деструктор, а метод **Finalize**. Все потому, что компилятор C# компилирует деструктор в конструкцию, которая эквивалентна следующей:



Метод Finalize уже определен в базовом для всех типов классе Object, однако данный метод нельзя так просто переопределить. И фактическая его реализация происходит через создание деструктора.

***2.Clone()*** создает копию объекта и возвращает ссылку на эту копию (неглубокое)

5. Приведите пример определения класса.

******

6. Какие ключевые слова можно использовать при определении класса?

* **static**: определяет статические члены класса, которые принадлежат классу, а не экземплярам.
* **abstract**: не может иметь экземпляров и может содержать абстрактные методы.
* **sealed**: запрещает наследование (класс не может иметь производных).
* **partial: позволяет разбить класс на несколько частей в разных файлах. Это мб полезно в больших проектах.**

7. В чем отличие между объектом и классом?

**Класс** –шаблон, по которому определяется форма объекта

**Объект** – это физическая реализация класса.

8. Что такое конструктор? Когда вызывается конструктор?

Конструктор –специальный метод внутри класса, который выполняется при создании объекта этого класса.

9. Перечислите свойства конструктора?

1) не имеет возвращаемого значения

2) имя такое же как и имя типа (класса)

3) не наследуются

4) нельзя применять модификаторы virtual, new, override, sealed и abstract

5) для класса без явно заданных конструкторов компилятор создает конструктор по умолчанию (без параметров)

6) для статических классов (запечатанных и абстрактных) компилятор не создает конструктор по умолчанию

7) может определяться несколько конструкторов, сигнатуры и уровни доступа к конструкторам обязательно должны отличаться

8) можно явно заставлять один конструктор вызывать другой конструктор посредством зарезервированного слова this

10. Что такое деструктор (destructor) ?

Деструктор – специальный метод, который используется для освобождения ресурсов, занимаемых объектом, перед тем как объект будет уничтожен сборщиком мусора.

~Имя\_класса (){код}

Свойства:

1)класс может иметь только 1 деструктор

2)не мб унаследованы или перегружены

3)невозможно вызвать (запускаются автоматически, поэтому нельзя узнать, когда именно он вызовется)

4)не принимает модификаторы и не имеет параметров

11. Что такое this?

This обеспечивает доступ к текущему экземпляру класса. В любой нестатический метод автоматически передается скрытый параметр this.

12. Что будет выведено в результате выполнения

5 5

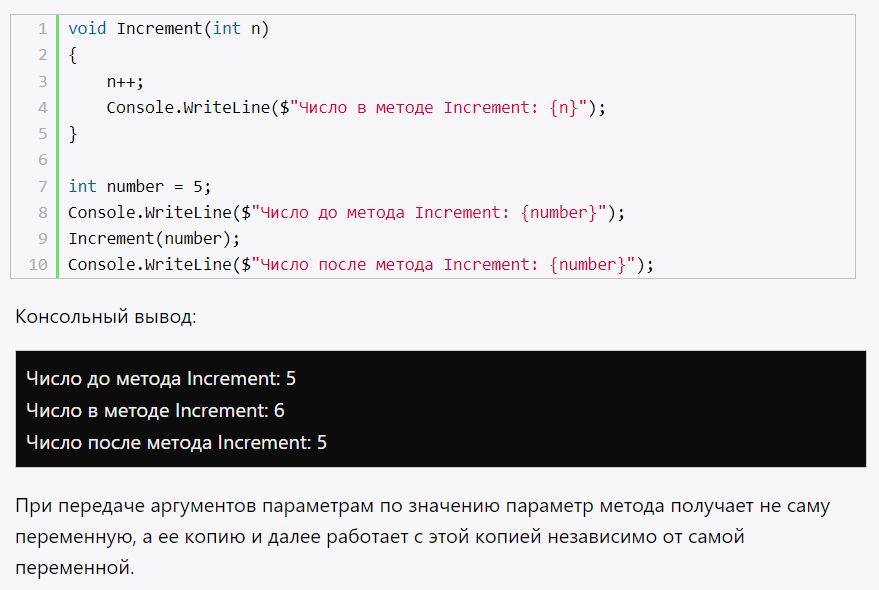
7 7

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



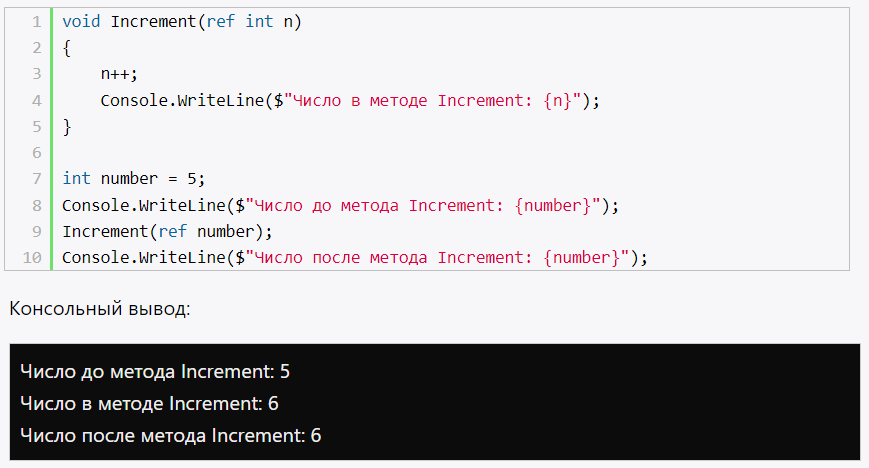
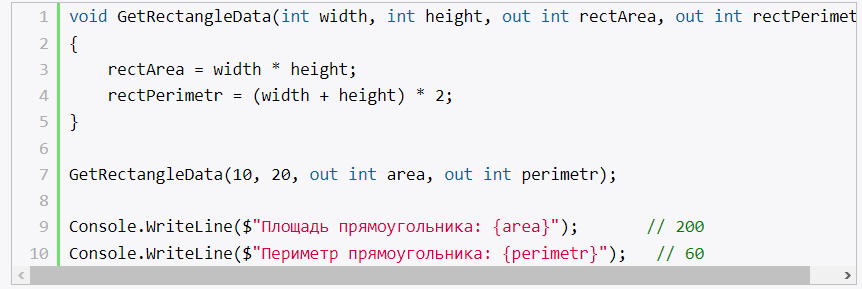
* **private**: текущий класс.
* **private protected**: текущий класс или производный из текущей сборки
* **protected**: текущий класс или производный из текущей или другой сборки.
* **internal**: из любого места кода в текущей сборке, но недоступен в других сборках.
* protected internal: из любого места кода в текущей сборке и из производных классов, которые могут располагаться в других сборках.
* **public**: из любой части кода.
* **file**: текущий файл кода

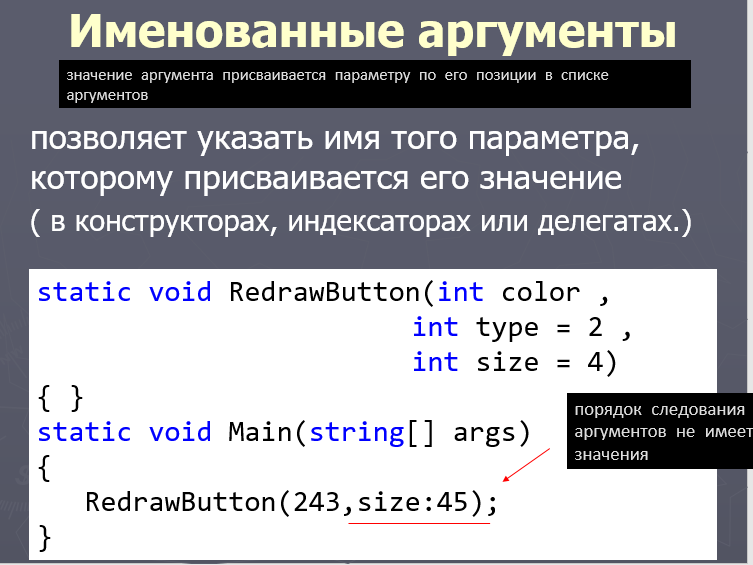
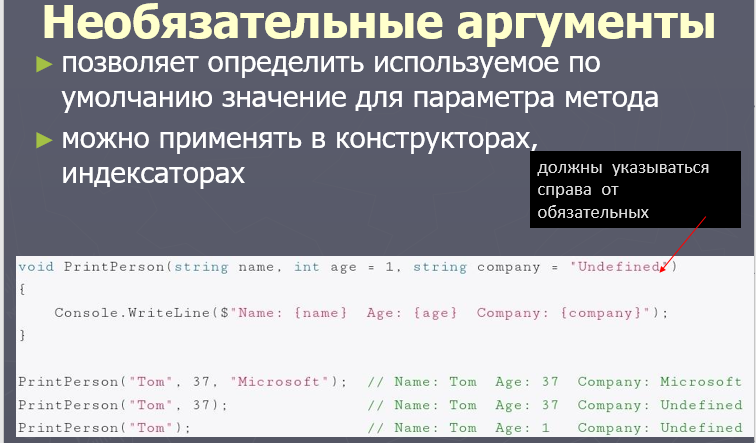
14. Опишите модификатор protected internal.

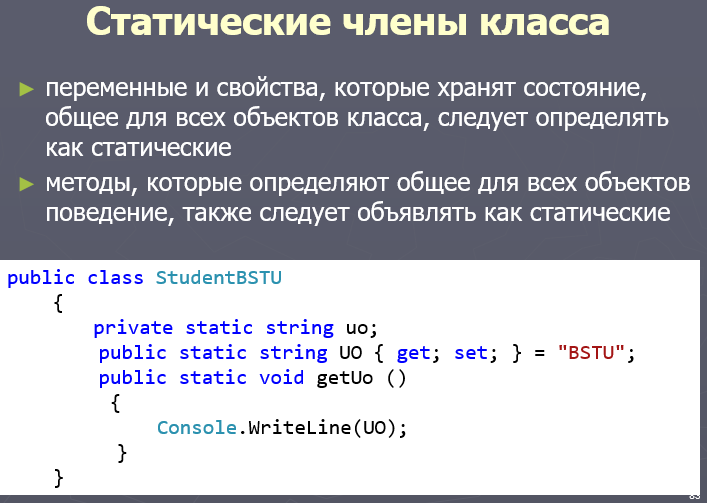
15. Зачем и как используются ref и out параметры функции? 

При передаче параметров по ссылке перед параметрами используется модификатор **ref**. При передаче значений параметрам по ссылке метод получает адрес переменной в памяти. Если в методе изменяется значение параметра, передаваемого по ссылке, то также изменяется и значение переменной, которая передается на его место. **Аргументу должно быть присвоено значение до вызова метода**.

Методы, использующие out параметры, обязательно должны присваивать им определенное значение. Мы можем вернуть из метода не одно значение, а несколько.  При этом можно определять переменные, которые передаются **out**-параметрам в непосредственно при вызове метода.

16. Приведите пример необязательных и именованных параметров метода. 

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

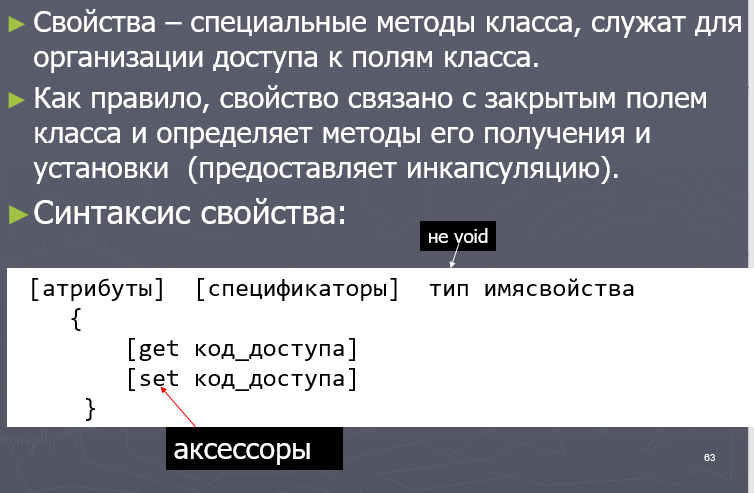
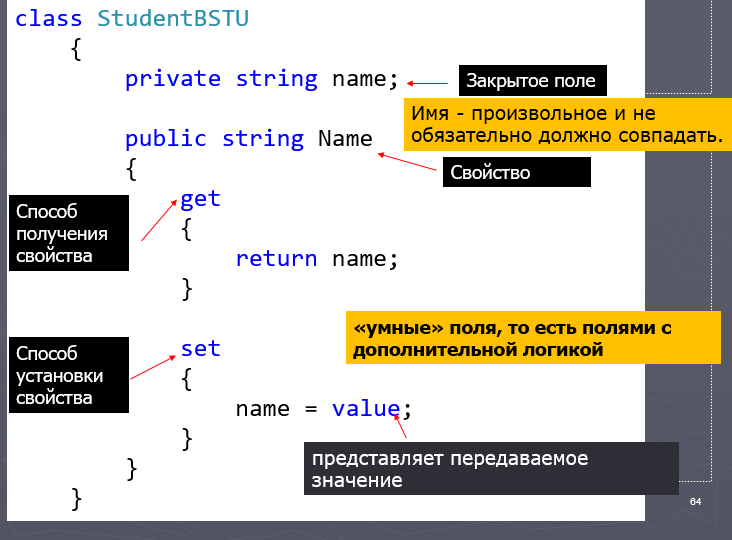
При использовании статических членов необязательно создавать экземпляр класса. Для статических полей будет создаваться участок в памяти, который будет общим для всех объектов класса.

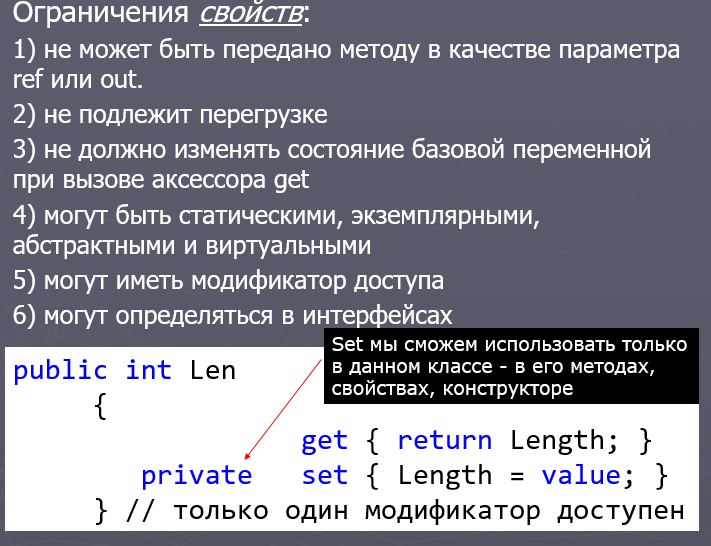
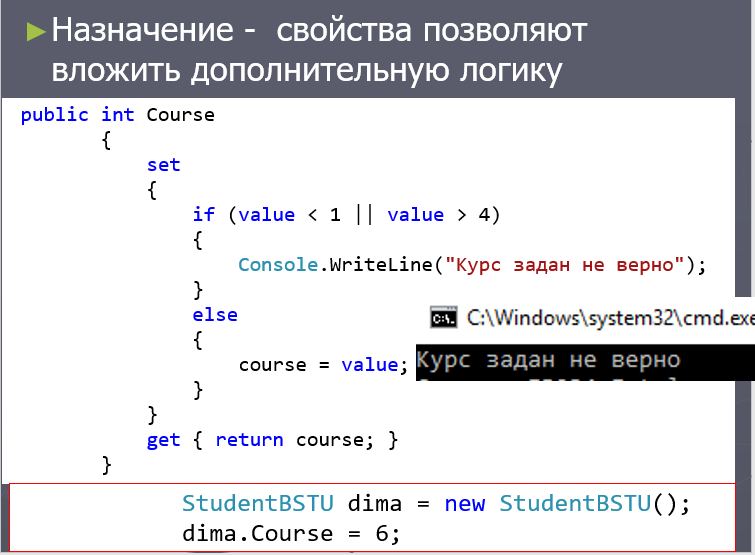
18. Приведите пример определения свойств класса. Как свойства связаны с инкапуляцией?

Свойство связано с закрытым полем класса и определяет метода его получения и установки (предоставляет инкапсулацию)

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

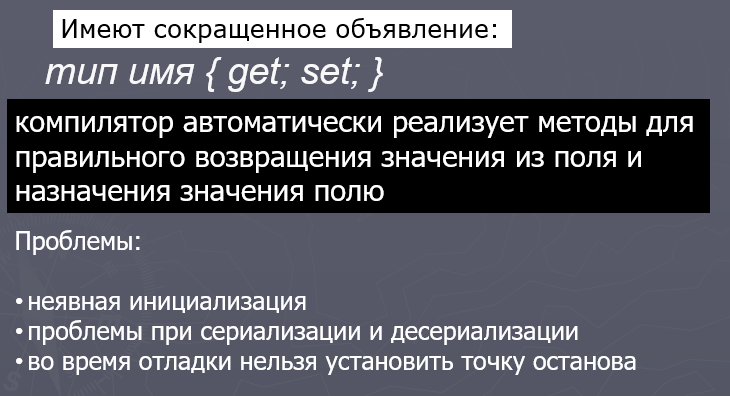
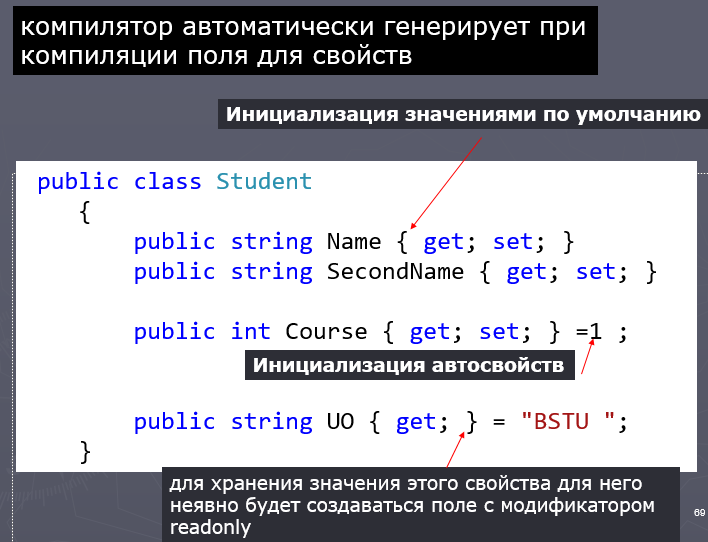
value.

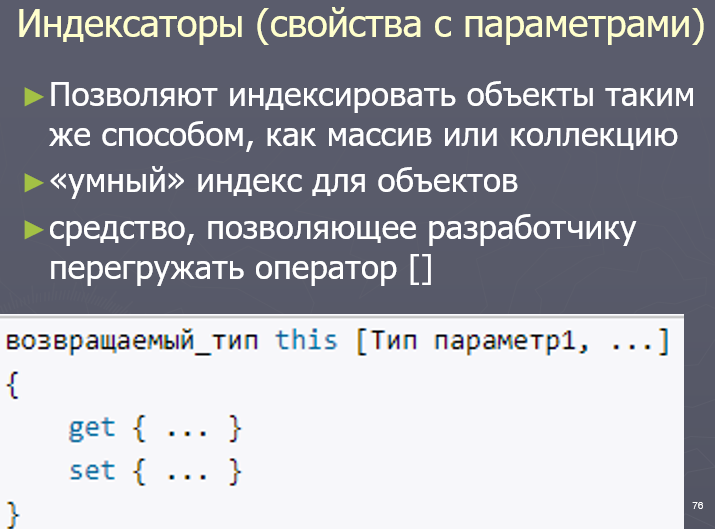
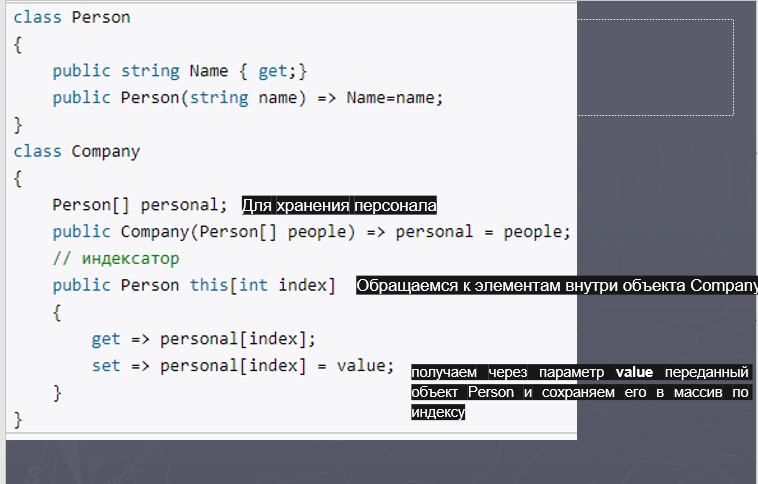


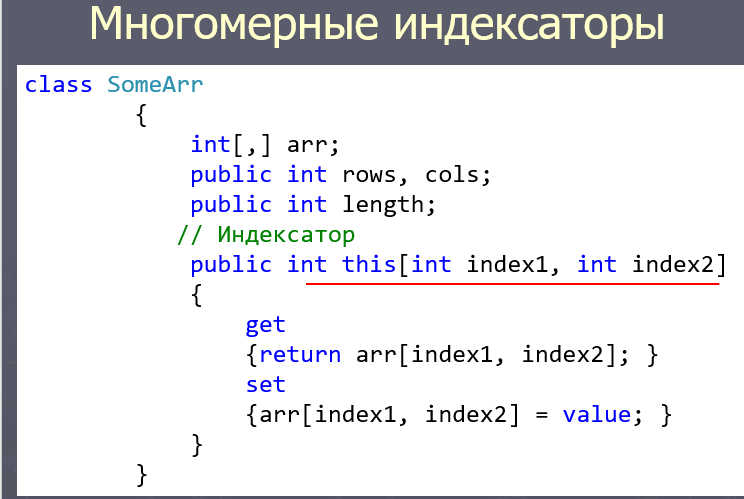
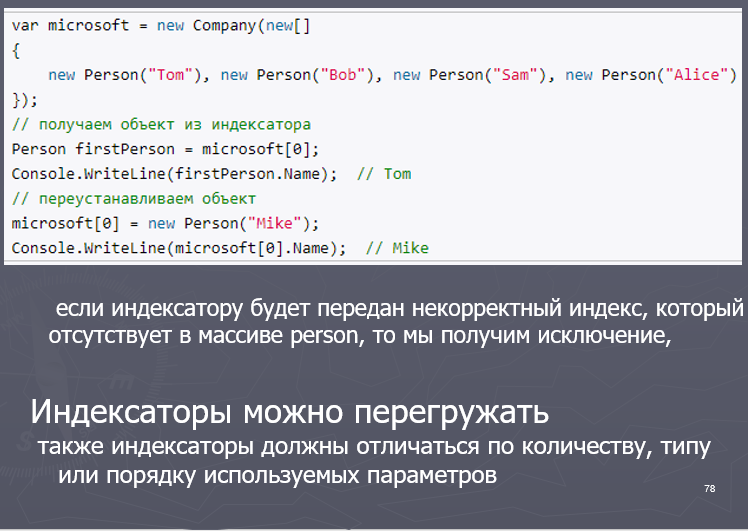
20. Что такое автоматические свойства? **{get ; set}**

Автоматические свойства позволяют определить свойство класса без явной реализации для полей. Компилятор автоматически создает скрытое поле для хранения значения свойства.

****

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

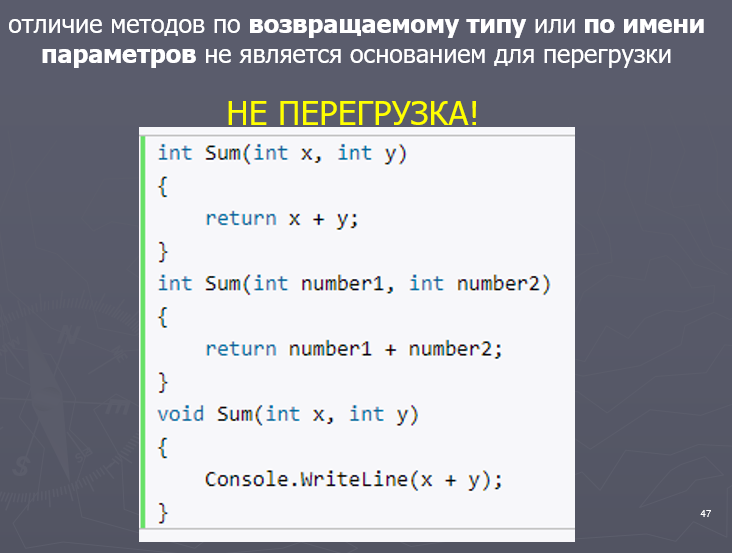
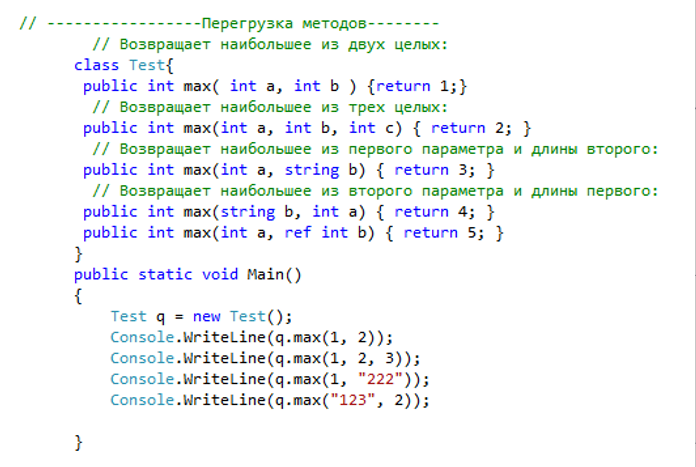
 

 Ограничения на индексаторы:

1. Значение, выдаваемое индексатором, нельзя передавать методу в качестве ref или out
2. Не мб static

22. Что такое перегруженный метод?

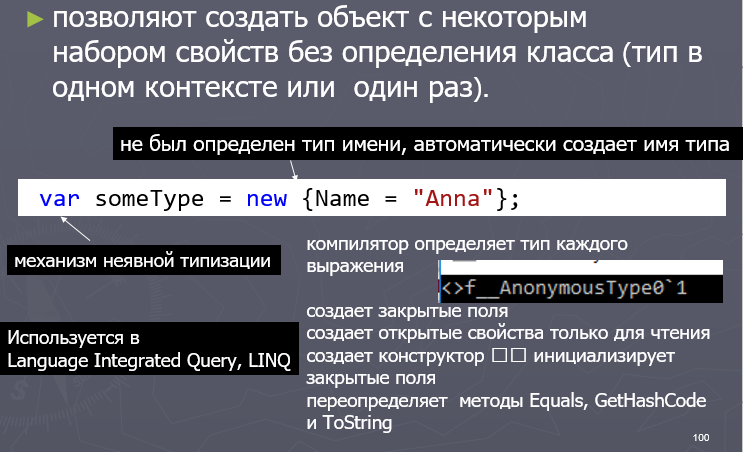
* один и тот же метод, но с разным набором параметров (количество, тип, порядок, модификаторам).



23. Что такое partial класс и какие его преимущества?

Частичный класс (partial class) - это класс, определенный в нескольких частях (файлах) и объединенный компилятором в один класс. Преимущества частичных классов включают разделение больших классов на более управляемые части и возможность генерации кода (например, из средств разработки пользовательского интерфейса) без изменения ручного кода.

24. Что такое анонимный тип в C#?



25. Для чего делают статические классы?

Статические классы объявляются с модификатором static и могут содержать только статические поля, свойства и методы.

Назначение:

1) при создании метода расширения

2) для хранения совокупности связанных друг с другом статических методов

26. В чем отличие статического поля от экземплярного?

статические поля принадлежат классу, а не конкретному объекту класса. Это означает, что у всех экземпляров класса будет общее значение статического поля, в то время как у каждого объекта будет свое значение экземплярного поля.

27. Поясните работу статических конструкторов.

Статические конструкторы - это специальные методы в статических классах, которые вызываются только один раз при первом обращении к классу или его членам. Они используются для инициализации статических данных и выполняются автоматически при самом первом создании объекта класса или при первом обращении к его статическим членам.

1. не имеют модификаторы доступа и не принимают параметров
2. нельзя использовать this для ссылки на текущий объект класса и можно обращаться только к статическим членам класса
3. нельзя вызывать вручную

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

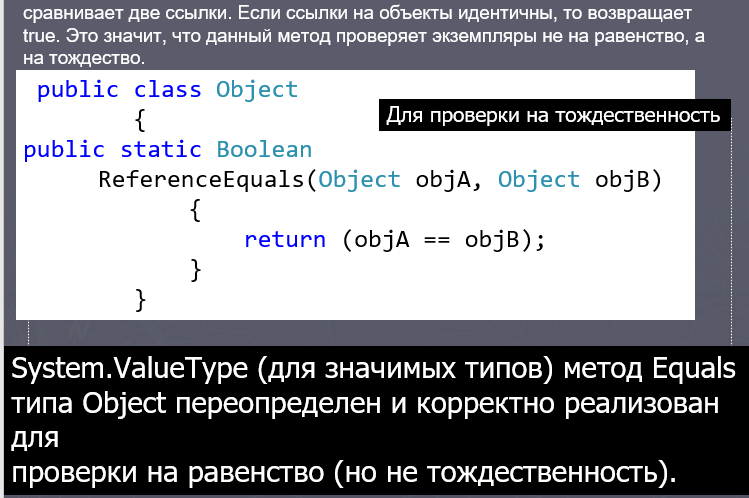
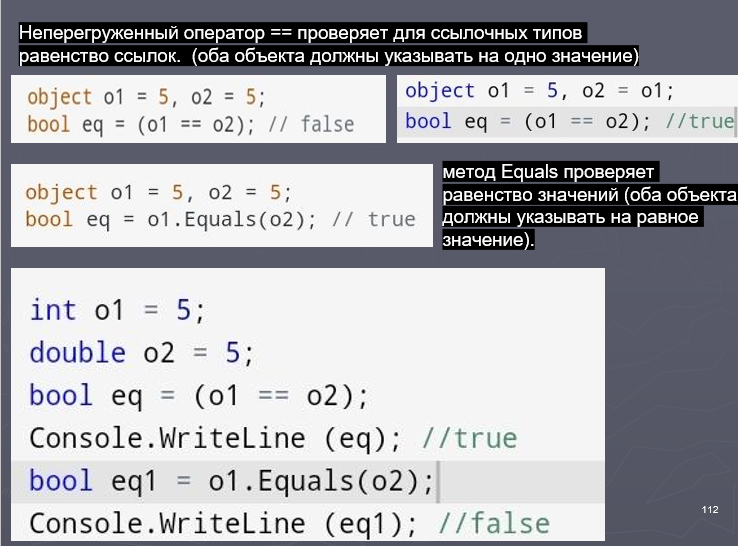
Поверхностное копирует только ссылки, а не объекты, на которые они указывают. Глубокое создает копии как самого объекта, так и всех объектов, на которые он ссылается.

При поверхностном копировании создается новый объект, но внутри него ссылки на объекты, содержащиеся в оригинале, остаются теми же. ПК копирует только ссылки на данные, но не сами данные. Если изменить данные в скопированном объекте, это также повлияет на оригинал, так как оба объекта указывают на одни и те же данные.

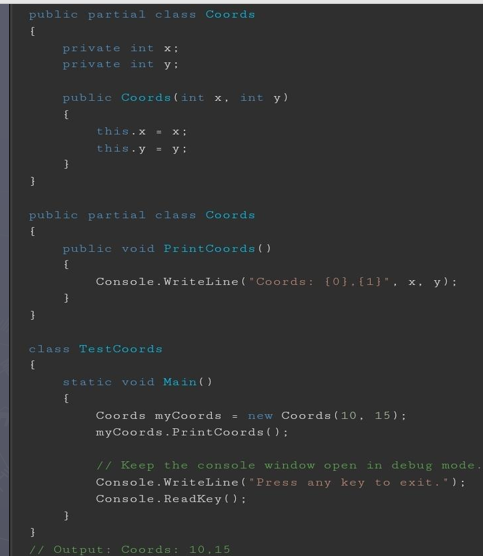
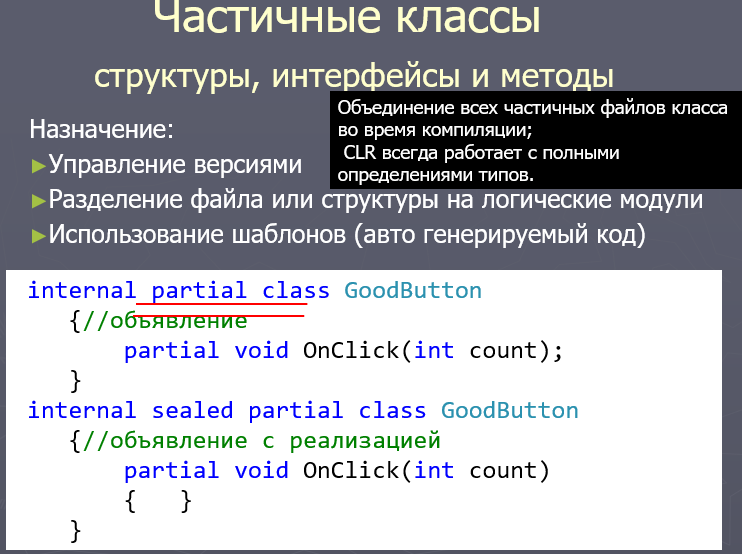
Глубокое копирование создает полностью независимую копию объекта, включая копирование всех объектов, на которые ссылается исходный объект. Это обеспечивает полную изоляцию между оригиналом и копией, и изменения в одном объекте не влияют на другой.

29. В чем разница между равенством и тождеством объектов?

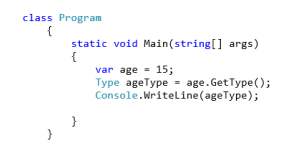
Равенство объектов (**==**) сравнивает значения объектов, в то время как тождество объектов (**ReferenceEquals()**) сравнивает ссылки на объекты. Два объекта могут иметь одинаковые значения, но быть разными объектами в памяти.



30. Что такое частичные классы и частичные методы?

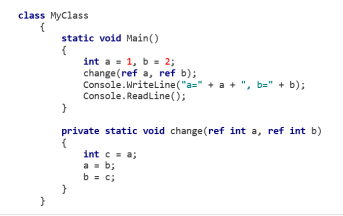


31. Что будет выведено на консоль результате выполнения следующего кода:



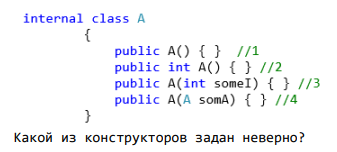
System.Int32

32. Что будет выведено на консоль результате выполнения следующего кода:



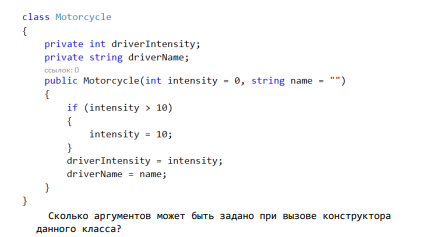
a=2; b=1;

33. Пусть задан следующий класс. Какой из конструкторов задан неверно?



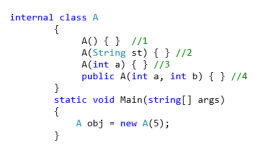
2

34. Пусть задан следующий класс. Сколько аргументов может быть задано при вызове конструктора данного класса?



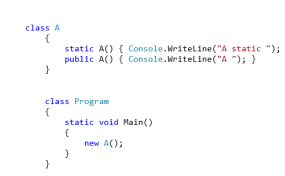
0, 1 или 2

35. Почему не удается создать объект класса A?



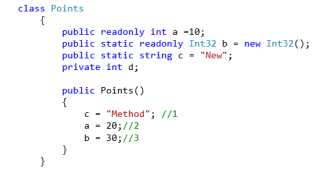
|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Серьезность | Код | Описание | Проект | Файл | Строка |
| Ошибка | CS0122 | 'A.A(int)"недоступен из-за его уровня защиты. | ConsoleApp1 |  | 16 |

36. Что будет выведено в консоль при выполнении данной программы?



A staticA

37. Какая строка приведенного далее класса вызовет ошибку компиляции?



3

