Лабораторная работа №3

Вопросы

1. Назовите принципы ООП. Поясните каждый из них.

* Наследование ( Процесс, благодаря которому один объект может наследовать свойства от другого объекта);
* Инкапсуляция (Механизм, связывающий вместе данные и код, обрабатывающий эти данные, и сохраняющий их от внешнего воздействия и ошибочного использования);
* Полиморфизм (Способность вызывать метод потомка через экземпляр предка);
* Абстракция данных (Подразумевает разделение и независимое рассмотрение интерфейса и реализации)

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

[Object](https://docs.microsoft.com/dotnet/api/System.Object)

1. Охарактеризуйте методы System.Object.

* Метод ToString() возвращает символьную строку, содержащую описание того объекта, для которого он вызывается.
* GetHashCode() возвращает хещ-код для данного объекта
* Equals() и ReferenceEquals() По умолчанию метод Equals (object) определяет, ссылается ли вызывающий объект на тот же самый объект, что и объект, указываемый в качестве аргумента этого метода, т.е. он определяет, являются ли обе ссылки одинаковыми.
* Finalize() Вызывается, когда уборщик мусора определяет, что объект является мусором, но до возвращения занятой объектом памяти в кучу
* GetType() Возвращает экземпляр объекта, производного от Type, который идентифицирует тип объекта
* Clone() Создает новый экземпляр типа и присваивает полям нового объекта значения объекта this. Возвращается ссылка на созданный экземпляр

5. Приведите пример определения класса.

Public class Student {…}

6. Какие ключевые слова можно использовать при определении класса?

Public, private, internal

1. В чем отличие между объектом и классом?

Класс – это некоторое абстрактное понятие – шаблон, по которому определяется форма объекта.

Объект – это физическая реализация класса (шаблона)

1. Что такое конструктор? Когда вызывается конструктор?

Конструктор – это специальные методы, позволяющие корректно инициализировать новый экземпляр типа.

1. Перечислите свойства конструктора?

* Не имеет возвращаемого значения
* Имя такое же как и имя типа (класса)
* Не наследуется
* Нельзя применять модификаторы virtual, new, override, sealed, abstract
* Для класса без явно заданных конструкторов компилятор создает конструктор по умолчанию (без параметров)
* Для статических классов компилятор не создает конструктор по умолчанию
* Может определяться несколько конструкторов, сигнатуры и уровни доступа к конструкторам обязательно должны отличаться
* Может явно заставлять один конструктор вызывать другой конструктор посредством зарезервированного слова this.

1. Что такое деструктор (destructor) ?

Вызывается непосредственно перед окончательным уничтожением объекта системой « сборки мусора», чтобы гарантировать четное окончание срока действия объекта.

1. Что такое this?

Обеспечивает доступ к текущему экземпляру класса.

1. Что будет выведено в результате выполнения.

Null null

7 null

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

public, private, protected и internal.

1. Опишите модификатор protected internal.

С помощью модификатора доступа protected обеспечивается создание защищенного члена класса, доступ к которому открыт в пределах иерархии классов. А модификатор internal служит в основном для сборок.

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

Out - выходные параметры должны присваиваться вызываемым методом (и, следовательно, передаваться по ссылке). Если параметрам out в вызываемом методе значения не присвоены, компилятор сообщит об ошибке

Ref- это значение первоначально присваивается вызывающим кодом и при желании может повторно присваиваться в вызываемом методе (поскольку данные также передаются по ссылке). Если параметрам ref в вызываемом методе значения не присвоены, компилятор никакой ошибки генерировать не будет

1. Приведите пример необязательных и именованных параметров метода.

static int sum(int a, int b = 5, int c = 10)

Аргументы b и с указывать при вызове необязательно

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

public readonly int polden = 12; // поле для чтения

public const int polnoch = 0; //константа

static int z; // статическое

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

public int Secynd

{

get { if (sec <= 59 & sec >= 0) return 1; return sec; }

}

Как правило, свойство связано с закрытым полем класса и определяет методы его получения и установки (предоставляет инкапсуляцию)

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

value

1. Что такое автоматические свойства?

Компилятор автоматически реализует методы для правильного возвращения значения из поля и назначения значения полю

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

Они позволяют индексировать объекты таким же способом, как массив или коллекцию. Ограничения:

* Значение, выдаваемое индексатором, нельзя передавать методу в качестве параметра ref или out
* Индексатор не может быть static

1. Что такое перегруженный метод?

В C# допускается совместное использование одного и того же имени двумя или более методами одного и того же класса, при условии, что их параметры объявляются по-разному. В этом случае говорят, что методы перегружаются, а сам процесс называется перегрузкой методов.

1. Что такое partial класс и какие его преимущества?

В языке C# возможно разбиение определения класса, структуры или интерфейса между двумя или больше исходными файлами. Каждый исходный файл содержит свою часть определения класса и все такие части собираются во время компиляции. Для этого используются partial классы.

1. Что такое анонимный тип в C#?

Позволяют создать объект с некоторым набором свойств без определения класса (тип в одном контексте или один раз)

1. Для чего делают статические классы?

Назначения:

* При создании метода расширения
* Для хранения совокупности связанных друг с другом статических методов

1. В чем отличие статического поля от экземплярного?

Для статических полей будут создаваться участок в памяти, который будет общим для всех объектов класса.

1. Поясните работу статических конструкторов.

Статический конструктор, как правило, используется для инициализации компонентов, применяемых ко всему классу, а не к отдельному экземпляру объекта этого класса. Конструктор типа static вызывается автоматически, когда класс загружается впервые, причем до конструктора экземпляра

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

При поверхностном копировании копируются значения полей класса, включая значения любых указателей или ссылок. При этом скопированные значения этих указателей и ссылок указывают на одни и те же объекты, что и в оригинальном объекте, что зачастую ведет к ошибкам. При глубоком копировании мы копируем значения полей не только на первом "уровне", но и заходим глубже, копируя все значения.

1. В чем разница между равенством и тождеством объектов?

Иногда возникает необходимость проверить равенство двух значений. В некоторых случаях проверяется равенство значений, которое также называют их эквивалентностью, что означает, что содержащиеся в двух переменных значения равны. В других случаях требуется определить, ссылаются ли две переменные на один и тот же объект в памяти. Такой тип равенства называется равенством ссылок или идентичностью. В этом разделе описаны эти два типа равенства и приведены ссылки на разделы, содержащие дополнительные сведения.

1. Что такое частичные классы и частичные методы?

Частичные классы объединяют все частичные файлы класса во время компиляции.

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

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

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

35. Почему не удается создать объект класса A?

2

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

А static A

37. Какая строка приведенного далее класса вызовет ошибку компиляции?

3