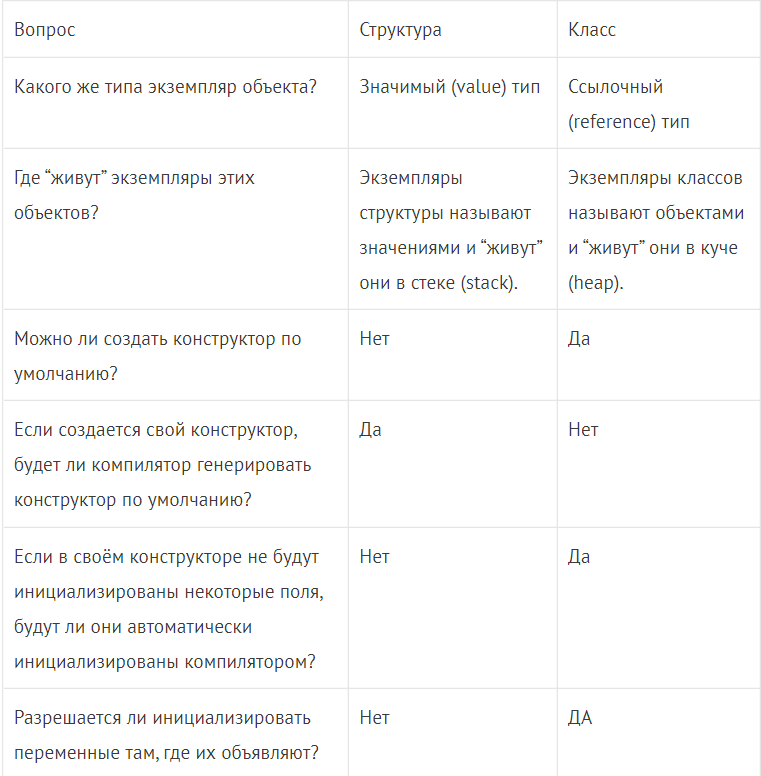
ВОПРОСЫ К ЛАБОРАТОРНОЙ РАБОТЕ №6:

1. Чем класс отличается от структуры?

СТРУКТУРА:

* нельзя инициализировать поля при объявлении
* нет авто- инициализаций полей компилятором
* нет наследования



2) Что может и чего не может быть в структуре?

может иметь конструктор с параметрами  
нельзя конструктор по умолчанию (без парам.) ибо авто-

3) Что такое перечисление?

**Перечисление (или «перечисляемый тип»)** — это тип данных, где любое значение (или «перечислитель») определяется как символьная константа, набор логически связанных констант.

Объявить перечисление можно с помощью ключевого слова **enum**.

*enum Colors*

*{*

*YELLOW,*

*BLACK, // BLACK находится в глобальном пространстве имен*

*PINK*

*};*

*enum Oper :short  
 Plus = 4,  
 Minus, //5  
 Umn = 9, //9  
 Div //10*

*class Math  
 Oper op = Oper.Umn*

4) Перечислите и поясните стандартные интерфейсы .Net?

|  |  |  |  |
| --- | --- | --- | --- |
| Интерфейс | Методы | Назначение | Применение |
| **IClonable** | object Clone() | клонирование объектов (поверх. и глуб.) |  |
| **IEnumerable** | (IEnumerator) GetEnumerator(); | перебор эл-тов необобщенной коллекции | основной для большинства коллекций |
| **IEnumerator** | Current bool MoveNext() void Reset() | перебор по необобщенной коллекции | можем перебирать объекты в цикле foreach |
| **IComparable** | int CompareTo (object obj) | для срав. объектов для выяснение порядка |  |
| **IComparer** | int Compare (object o1, object o2) | сравн. объектов | будут иметь больший приоритет |
| **IDisposable** | void Dispose() | механизм для освобождения |  |
| **IEquatable<T>** | bool Equas (T other) | срав. объектов на рав-во |  |
| **IStructuralEqu-atable** | bool Equals (object, IEqualityComparer ); int GetHashCode(…) | проверка на равенство по значению |  |

7) Что такое полиморфизм? Перечислите его формы. Приведите примеры.

**Полиморфи́зм** — возможность объектов с одинаковой спецификацией иметь различную реализацию, переопределение членов баз. класса.

Полиморфизм позволяет писать более абстрактные программы и повысить коэффициент повторного использования кода. Общие свойства объектов объединяются в систему, которую могут называть по-разному — интерфейс, класс.

Пример: *самый обычный пример специализированного полиморфизма —* ***перегрузка (overloading)****, когда один и тот же символ функции соответствует различным реализациям; компилятор выбирает подходящую реализацию для каждого случая применения функции, исходя из типов ее аргументов.*

Полиморфизм можно применять для следующих элементов:

1. *методов;*
2. *свойств;*
3. *индексаторов;*
4. *событий.*

**virtual** public void A\_method() {}  
переопределение вирт.метода в произв. классе:  
**override** public void A\_method() {}

вызов метода по типу ссылки *//просто*  
вызов метода по типу объекта *//virtual + override*

8) Зачем в классе определяют виртуальные методы?

Виртуальный метод отличается тем, что он может быть переопределен в одном или нескольких производных классах.

**override** – изменение абстрактной или виртуальной реализации  
**virtual** – разрешает переопределение (↑)

9) Как сделать запрет переопределения методов?

С помощью ключевого слова **sealed**.

