1.Архитектура .NET <http://interface.ru/home.asp?artId=3531>  
CTS – Common Type System <https://ru.wikipedia.org/wiki/Common_Type_System>  
Garbage Collector <https://ru.wikipedia.org/wiki/Сборка_мусора>  
MSIL – Microsoft Intermediate Language <https://www.bestprog.net/ru/2016/12/20/базовые-понятия-технологии-net-framework/>  
Assembly / Namespace C:\Windows\Assembly

1. Типы данных. <https://metanit.com/sharp/tutorial/2.1.php>

Преобразование типов. <https://professorweb.ru/my/csharp/charp_theory/level3/3_8.php>

Boxing & unboxing <https://heap.tech/lore/id20>  
Значимые типы: целое, вещественное, булево. <https://professorweb.ru/my/csharp/charp_theory/level3/3_6.php>  
Ссылочные типы: классы, делегаты. <https://metanit.com/sharp/tutorial/2.16.php>  
.GetHashCode()<https://metanit.com/sharp/tutorial/3.10.php> Куда отнести: структуры, списки, массивы и строки?

int Int32

Convert

int.Parse()

bool int.TryParse(… out val) // ref out

\*.ToString()

int a = (int)(6/2);

Planet p = new Planet();

// public void Process(object a, object b)

\*.Process((object)p, (object)a); // Boxing

object o = (object)p;

Planet p2 = (Planet)o; // UnBoxing

instanceof ==> is

class SpaceObject // x,y, name, Process()

class Planet : SpaceObject // R, star,

class Comet : SpaceObject

\*.Render(new Planet());

public void Render(SpaceObject so)

{

// so – Не будет R и star – потому что их нет в SpaceObject

if(so is Planet){

Planet p = (Planet)so;

p.Process();

}

.GetType(???) == “\Lab2\Planets\SpaceObject”

typeof(i) == int

1. UML. Class Diagram [https://ru.wikipedia.org/wiki/Диаграмма\_классов](https://ru.wikipedia.org/wiki/Диаграмма_классов%20%20)  + Основы ООП Инкапсуляция – содержать в себе данные и методы, но и защитить  
   Наследование  
   Полиморфизм  
   Модификаторы доступа: public+, private-, protected#  
   Делегирование, делегация (не путать с делегатом, или можно путать))))  
   Композиция, агрегация Реализация инерфейса Ассоциация (класс ассоциация) Делегирование
2. Паттерны проектирования [https://metanit.com/sharp/patterns/](https://metanit.com/sharp/patterns/%20) GOF Типы: Поведенческие, Порождающие, Корпоративных приложений, … <https://metanit.com/sharp/patterns/1.1.php>  
   Strategy / Bridge  
   Factory / Abstract Factory / Factory Method  
   Singleton – Config, MySQL\_Connector  
   Builder  
   Composite  
   Фасад  
   Декоратор  
   Front Controller  
   Page Controller  
   Observer / Subscriber / Listener
3. Теория вычислимости. [https://ru.wikipedia.org/wiki/Теория\_вычислимости](https://ru.wikipedia.org/wiki/Теория_вычислимости%20) Сложность алгоритмов <https://metanit.com/sharp/algoritm/1.1.php>  
   О(N), o(N), Omega(N)  
   O(N) = O(cN)  
   O(N\*logN) O(N^k)  
   log A B = log c B / log c A P, NP  
   Комбинаторика: сочетания, размещения  
   -- перебор всех сочетаний или размещений

for (int i=0; i<N; i++)

for (int j=i+1; j<N; j++)

for (int k=j+1; k<N; k++)

print(I, j, k);

6.Сортировки сдвигом, быстрая, пузырьковая, Хоара  
вставками

7.События, обработка событий, делегаты и шаблон Observer/Listener/Наблюдатель

<https://metanit.com/sharp/patterns/3.2.php>

1. Устар: восходящее и нисходящее программирование <https://studfiles.net/preview/4003581/page:10/> Процедурное программирование [https://habr.com/ru/post/223253/](https://habr.com/ru/post/223253/%20)

Основы функционального программирования. Замыкания <http://sergeyteplyakov.blogspot.com/2010/04/c.html>  
Fluent Interface. LINQ. <https://metanit.com/sharp/tutorial/15.1.php> Лямбда-выражения <https://metanit.com/sharp/tutorial/3.16.php>   
Когда функция возвращает функцию?

9.IEnumerable, IEnumerator, [https://metanit.com/sharp/tutorial/4.11.php](https://metanit.com/sharp/tutorial/4.11.php%20) Генератор, yield return <https://metanit.com/sharp/tutorial/4.12.php> , Lazy Loading <https://metanit.com/sharp/entityframework/3.2.php>

10.MVC, MVP, MVVM <https://habr.com/ru/post/215605/>

1. Потоки и Процессы Thread <https://metanit.com/sharp/tutorial/11.1.php> , join <https://professorweb.ru/my/csharp/thread_and_files/1/1_4.php> . Worker, BackgroundWorker <https://professorweb.ru/my/WPF/documents_WPF/level31/31_3.php>
2. Сериализация <https://metanit.com/sharp/tutorial/6.1.php> и маршалинг <https://ru.stackoverflow.com/questions/588442/Что-такое-маршалинг>   
   Binary, XML <https://metanit.com/sharp/tutorial/16.1.php>
3. Файлы Stream StreamReader / Writer / <https://metanit.com/sharp/tutorial/5.5.php> ... binary... xml Как проверить, что файл существует? <http://www.cyberforum.ru/csharp-beginners/thread358762.html> Обойти рекурсивно каталог и отобразить файлы и папки (и содержимое внутренних каталогов) <https://metanit.com/sharp/tutorial/5.2.php> Записать в файл, прочитать <https://metanit.com/sharp/tutorial/5.4.php>

Подключение DLL

Копирование/клонирование объектов. Поверхностное (shallow) и глубокое (deep) копирование

\*\* проектирование \*\*  
SOLID  
DRY

\*\* тестирование \*\*  
TDD Mock, Stub BDD (cucumber, gherkins) DDD

Функциональное тестирование (selenium)

Модульное тестирование

\*\* практическое \*\*

1. изменение размера массива
2. удаление i-го элемента в списке
3. исключить элемент из списка / вернуть список, не содержащий элемент E
4. перемножение матриц
5. обход графа вглубь
6. определение существования цикла в графе
7. реализовать алгоритм сортировки (любой)

------------ не будет ------------------ Рефлексия??? /\*\*

\*/

/// /// ЛРОЛОРЛ ///

1. Параметрическая оптимизация. Метод градиентного спуска.
2. Постановка задачи Машинного обучения – любой  
   -- либо классификация, либо регрессия  
   Воронцов (Яндекс Youtube)  
   Andrew Ng (Coursera.Org – Stanford – Machine Learning)