1. Clase

* Ce semnifica clasele?
* Ce contine clasele? Ce reprezinta fiecare? (campuri, proprietati, metode)
* Cum se creaza clasele?
* Diferenta dintre clase si structure
* Modificatorii de access(public, private, internal) {static - unic}
* Supraincarcarea metodei

1. POO

* Inheritance – Mostenirea unei clase
* Metode virtuale (Pentru proprietati si Metode) -> override – schimbarea
* Clase abstracte (abstract – continut mai general)
* Generalizări sau tipuri generice (Обобщения)

1. Tratarea exceptiilor

* Try-Catch-Finally

1. Delegate si Lambda expresii

* Ce reprezinta o delegata?
* Cu ce scop avem nevoie de delegate?
* Ce reprezinta Lambda? Cum se noteaza?
* Delegatele de baza (Action<>, Predicate<>, Func<>)

1. Interfata

* Ce reprezinta o interfata?
* La ce ne trebuie interfata?
* Realizarea interfetei?

1. Caracteristici avansata al POO

* Operators
* Metode de extindere
* Clase si metode partiale
* Tipuri anonime
* Tupluri (Tuple)
* Tipuri Records

1. List. Collections

* Ce reprezinta List?
* De ce List dar nu array[]?
* Queue, Stack, Dictionary
* Interfetele IEnumerator<type>, IEnumerable<type>
* Yeld Return (returneaza o lista cu ajutorul IEnumerator<type> )