Interface

Interface is basically like a class. Ekhane shudhu function er signature thake, kono body thake na.

- ✓ Function er name thakbe
- ✓ Function er parameter thakteo pare, abar nao thakte pare,
  - public void setName (string name)
    { } -> Body thakbe na

❖ **Interface er moddhe thake ->**

1. Only Function Signature.
2. It is implemented by other class.
3. Multiple Inheritance implement possible hoy na but multiple interfaces implement possible.
4. Abstract method.
5. Overriding.

Interface & Abstract Class
1. Interface ke Abstract class bola jabe na. 2. Interface like as Abstract class. 3. Abstract class er moddhe regular method o thakte pare. 4. Eksathe duiti Abstract class ke inherit kora jay jay but duiti interface ke implement kora jay. 5. Interface er moddhe sob gulo By Default abstract method. 6. Interface er throw te Higher-Level Abstraction achieved kora jay.

- Interface ke **implement korte pare** -> **Non-Abstract class or a Struct.**
- Interface er **moddhe sobgulo By default Abstract methd.**

- Jodi kono function er Default implementation ba body soho deya lage, sekhetre seti Static Function hishebe deya lage, egula Special case er jonno.

- Interface er moddhe **methods, properties, events, indexer** thake. Echara **static variable, class level variable** thakte pare.

- The name of an interface must be a valid ***C# Identifier name***.

- Interface er moddhe kokhonO object create kora hoy na but extra ja kisui thakbe ta **Static hishebe define kora hobe**.

- **Multiple inheritance** er concept ti **Interface er throw te** implement kora hoy.
- Interface ke je class implement korbe tar moddhe oi method gulor Body assign korte hobe.
- **Interface is pure abstraction with no implementation.**
- Interface er maddhome **Higher level Abstraction achieve** kora jay.

❖ **Interface er Maddhome kivabe Higer level Abstraction achieved kora jay?**

➔ Abstraction basically showing only the necessary details and hiding all the unnecessary details.

✓ Abstract method ke override kora mandatory

✓ **Interface er moddhe sob method e abstract. Orthat,joto gulo Function/Method thakbe sob gulo method implement kora mandatory. Jodi implement kora na hoy sekhetre, je class ti interface ke implement korbe sei class a jodi oi interface er bodygulo notun vabe implement ba define kora na hoy tokhon oi class object create kora jabe na. Oi class k abar Inherit korte hobe. Tarpor abar implement korte hobe.**

❖ **Interface keno ebong kothay use kora hoy?**

➔ **To achieve security / Hide certain details** and only show the implement details of an object (Interface).

➔ Inherited class interface er sob member k obosshoi implement korbe otherwise segulo Abstract hoye jabe. Abar abstract hoye gele sei class er object create kora jabe na.

➔ Interface er moddhe je method gulo declare kora hoy segulo By default Abstract / By default Public.

❖ **Interface er moddhey Multiple Inheritance er problem ti solve kora jay Explicit interface Implement er maddhome.**

✓ **Explicit Implementation:** Interface er name declare kore deya.

❖ Interface er moddhe **Static Constructor** thakte pare.

1. Control kora jay na
2. Initialize kora jay na
3. Same Interface er moddhe multiple time call kora jay na
4. Value initialize kora jay na

❖ An interface **can't contain instance fields, instance constructors or finalizers.**

❖ **Interface members are abstract and public by default.**

- To implement an **interface member**, the corresponding member of the **implementing class must be public, non-static and have the same name and signature as the interface member.**
- Kono class er moddhe **Abstract method** thakle seti onno class a implement korte
  - **Base Class:** Abstract / virtual
  - **Child Class:** Override

Egula Interface er somoy use korar proyojon hoy na.

❖ Difference between Interface and Class

Interface	Class
1. Ekti interface onno ek ba ekadhik interface k inherit kore.	1. Ekti class onno ekti class k inherit kore
2. Ekti interface onno kono class ke inherit korte pare na.	2. Ekti class onno ekti interface ke implement kore.

- User Interface dekhbe
- Interface er moddhe usually normal kono abstract allow na / kono constructor thake na,
- Abstract class er moddhe constructor allowed.

Indexer

Indexer holo ekta special type of property, jeta diye object(class er instance) er behavior array er moto kora jay but array hobe na.

- Ekti class er moddhe normal variable neya jay temoni array variable O neya jay. Array variable er jonno getter/setter function neya quite difficult. Ejonno loop use kora hoy. Ar loops use korar fole bujhte oshubidha hote pare ar complexity create hote pare.
- Indexer ke kokhonO **static member hishebe declare kora hoy na**. Indexer hote hoy **Instance member**. Indexer er value peremeter **"ref/out"** hishebe pass kora jabe na.
- Indexer ke overload kora jay.
- Indexer accessor value return korbe.

✓ Ekti array er moddhe function diye kivabe assign kora hoy?

➔ Loop use kore.

✓ Property likhte – Access Modifier, Data type, Name lage.

❖ **Indexer vs Property:**

Indexer	Property
1. Indexer moddhe **"This "** keyword use kora hoy jar sathe Array Notation "[]" use kora hoy.	1. Normal property te "this" keyword use kora hoy na & third brackets "[]" orthat array notation thake na.
2. Indexer static hote pare na	2. Property static hote pare

3. Public string [] arr1 = new string [3];
Public string this [int i]
{
get {return arr1[i];}
set{arr1[i]=value;}
}
	3. Class person
	{ private string name; //field
	public string Name // property
	{
	get {return name;}
	set{name = value;}
	} }

✓ **Private date kivabe access kora jay?**

✓ Set kore constructor er throw te access kora jay

✓ Same class er public method diyeo access kora jay.

✓ Static property keno use kora hoy?

➔ Ekti class er moddhe je static variable gulo thake segulo get/set korar jonno static property use kora hoy.

➔ Static property er object create er proyojon hoy na. Class er name a property call korte hoy.

Delegates

Delegates onek gulo method er reference ke contains kore & shubidha moto call kora jay.

✓ Delegates er same signature hote hobe

Public delegate void same (int a, int b)

- ✓ Delegate use kore same category er function call kora jay. Ekhane kono limit thake na [joto gulo proyojon]
- ✓ **Delegate use korar fole bar bar function call / parameter a value pass kora lage na**
- ✓ Delegate declare outside of the class.
- ✓ Event handling er jonno delegates kaje lage.
- ✓ **Multicast delegate:** Jokhon eksathe eker odhik function call kora hoy tokhon Multicast delegate use kora hoy.

❖ **Benefits of delegates:**

- ✓ Delegate holo type safe and onek secure
- ✓ Delegate er maddhome **Event handling** simple and easy
- ✓ Delegate use kore parameter a value pass kora hoy
- ✓ Eta **call-back method** er jonno use hoy
  - Function / Constructor a amra **Variable er value/object/array variable** pass kori