**Lesson02 what is Fluent Builder Design Pattern**

**Notes:-**

**1-with fluent builder you can chain multiple command on the same line instead of put on each line for each command as below**

**var builder = new HtmlBuilder("ul");**

**builder.AddChild("li", "hello");**

**builder.AddChild("li", "world");**

**//with fluent builder like below**

**var builder = new HtmlBuilder("ul");**

**builder.AddChild("li", "hello")AddChild("li", "world");**

**Lesson03 Fluent Builder Inheritance with Recursive Generic**

**Steps:-**

**1-create abstract class**

**public abstract class PersonBuilder{**

**protected Person person = new Person();**

**public Person Build(){return person;}}**

**2-create first layer that inherit from base class as below**

**//layer 01**

**//This class inherit from PersonBuilder with pass T which is the PersonInfoBuilder<T>**

**public class PersonInfoBuilder<T> : PersonBuilder where T : PersonInfoBuilder<T>{**

**public T Called(string name){**

**person.Name = name;**

**return (T)this;}}**

**3-create second layer that inherit from the first layer**

**//layer 02**

**//This class inherit from PersonInfoBuilder<PersonJobBuilder<T>>**

**//with pass T which is the PersonJobBuilder<T>**

**public class PersonJobBuilder<T>: PersonInfoBuilder<PersonJobBuilder<T>>**

**where T : PersonJobBuilder<T>{**

**public T WorksAsA(string position){**

**person.Position = position;**

**return (T)this;}}**

**4-create third layer that inherit from second layer**

**//layer 03**

**//This class inherit from PersonJobBuilder<PersonBirthDateBuilder<T>>**

**//with pass T which is the PersonBirthDateBuilder<T>**

**public class PersonBirthDateBuilder<T> : PersonJobBuilder<PersonBirthDateBuilder<T>>**

**where T : PersonBirthDateBuilder<T>{**

**public T Born(DateTime dateOfBirth){**

**person.DateOfBirth = dateOfBirth;**

**return (T)this;}}**

**5-on Program.cs we apply call the layer3 filter**

**//so we make chain call with it execute first PersonInfoBuilder.cs then PersonJobBuilder then PersonBirthDateBuilder then Build method on PersonBuilder;**

**internal class Program{**

**public static void Main(string[] args){**

**var me = Person.New**

**.Called("Dmitri")**

**.WorksAsA("Quant")**

**.Born(DateTime.UtcNow)**

**.Build();**

**WriteLine(me);}}**