|  |
| --- |
| **Builder** |
| **Line** | **Code** |
| **1****2****3****4****5****6****7****8****9****10****11****12****13****14****15****16****17****18****19****20****21****22****23****24****25****26****27****28****29****30****31****32****33****34****35****36****37****38****39****40****41****42****43****44****45****46****47****48****49****50****51****52****53****54****55****56****57****58****59****60****61****62****63****64****65****66****67****68****69****70****71****72****73****74****75****76****77****78****79****80****81****82****83****84** | // Simple theory code with one director and two builders**using** System;**using** System.Collections.Generic;**namespace** Builder{ **interface** IBuilder{ **void** BuildPartA(); **void** BuildPartB(); Product GetResult(); } **class** Builder1:IBuilder{ **private** Product product = **new** Product(); **public void** BuildPartA(){ //Building the Part A here.... product.Add("PartA "); } **public void** BuildPartB(){ //Building the Part B here.... product.Add("PartB "); } **public** Product GetResult(){ **return** product; } } **class** Builder2:IBuilder{ **private** Product product = **new** Product(); **public void** BuildPartA(){ //Building the Part A here.... product.Add("PartX "); } **public void** BuildPartB(){ //Building the Part B here.... product.Add("PartY "); } **public** Product GetResult(){ **return** product; } } **class** Director{ //Builder uses a complex series of steps **public void** Construct(IBuilder builder){ //Standard process that can be reuse builder.BuildPartA(); builder.BuildPartB(); builder.BuildPartB(); } } **class** Product{ List<**string**> parts = **new** List<**string**>(); **public void** Add(**string** part){ parts.Add(part); } **public void** Display(){ Console.WriteLine("\nProduct Parts -------"); **foreach** (**string** part **in** parts) Console.Write(part); Console.WriteLine(); } } **class** Program{ **static void** Main(**string**[] args){ // Create one director and two builders Director director = **new** Director(); Builder1 b1 = **new** Builder1(); Builder2 b2 = **new** Builder2(); // Construct two products director.Construct(b1); Product p1 = b1.GetResult(); p1.Display(); director.Construct(b2); Product p2 = b2.GetResult(); p2.Display(); Console.ReadKey(); } }} |