|  |  |
| --- | --- |
| **Singleton** | |
| **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** | **using** System;  **namespace** Singleton{  **class** Program{  **public sealed class** Singleton{  Singleton() { }// Private Constructor  // Private object instantiated with private constructor  **static** Singleton instance = **new** Singleton();  // Public static property to get the object  **public static** Singleton UniqueInstance {  **get** {  **return** instance;  }  }  **public void** SingletonOperation() {  //The method goes here...  }  **public int** SingletonData;  }  **static void** Main2(){  Singleton s1 = Singleton.UniqueInstance;  Singleton s2 = Singleton.UniqueInstance;  **if**(Singleton.ReferenceEquals(s1,s2))  Console.WriteLine("Objects are the same instance");  s1.SingletonOperation();  //s2.SingletonData = 100;  Console.ReadKey();  }  }  } |

|  |  |
| --- | --- |
| **SingletonLazy** | |
| **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** | **using** System;  **namespace** SingletonLazy{  **class** Program{  **public class** Singleton{  Singleton() { } // Private constructor  **static** Singleton uniqueInstance = **null**;  // Public static property to get the object  **public static** Singleton UniqueInstance{  **get**{  **if** (uniqueInstance==**null**) uniqueInstance = **new** Singleton();  **return** uniqueInstance;  }  }  **public void** SingletonOperation() {  //The method goes here...  }  **public int** SingletonData;  }  **static void** Main(){  Singleton s1 = Singleton.UniqueInstance;  Singleton s2 = Singleton.UniqueInstance;  **if** (Singleton.ReferenceEquals(s1,s2))  Console.WriteLine("Objects are the same instance");  }  }  } |