Белорусский государственный университет

информатики и радиоэлектроники

Кафедра ПОИТ

Отчет по лабораторной работе №7

по дисциплине

«Надежность программного обеспечения»

«Модульное тестирование»

Выполнил:

Гладкий М.Г.

группа 851005

Проверил:

Деменковец Д.В.

Минск, 2020

Тестированию подвергнется лабораторная работа №5 по предмету Современные платформы программирования с использованием библиотеки NUnit. Цель лабораторной работы заключается в создании фабрики объектов Dependency Injection Container. При тестировании необходимо осуществить логирование с помощью библиотеки NLog.

Тестирование с использованием Mock-объектов:

[Test]  
public void CountMock()  
{  
 Mock<IService> mock = new Mock<IService>();  
 Mock<ServiceImpl1> mock1 = new Mock<ServiceImpl1>();  
 var dependencies = new DependenciesConfig();  
 dependencies.Register(mock.Object.GetType(), mock1.Object.GetType());  
 dependencies.Register(mock1.Object.GetType(), mock1.Object.GetType());  
 Assert.AreEqual(2, dependencies.Count());  
}

Тесты:

public class Tests  
{  
 [Test]  
 public void Count()  
 {  
 var dependencies = new DependenciesConfig();  
 dependencies.Register<IService, ServiceImpl1>();  
 dependencies.Register<AbstractService, Service2>();  
 dependencies.Register<ServiceImpl1, ServiceImpl1>();  
 Assert.AreEqual(3, dependencies.Count());  
 }  
   
 [Test]  
 public void ResolveBasic()  
 {  
 var dependencies = new DependenciesConfig();  
 dependencies.Register<IService, ServiceImpl1>();  
 dependencies.Register<AbstractService, Service2>();  
   
 var provider = new DependencyProvider(dependencies);  
 var service1 = provider.Resolve<IService>();  
  
 Assert.AreEqual( typeof(ServiceImpl1), service1.GetType());  
 }  
   
 [Test]  
 public void ResolveRecursive()  
 {  
 var dependencies = new DependenciesConfig();  
 dependencies.Register<IService, ServiceImpl>();  
 dependencies.Register<IRepository, RepositoryImpl>();  
   
 var provider = new DependencyProvider(dependencies);  
 var service = provider.Resolve<IService>();  
  
 Assert.AreEqual( typeof(ServiceImpl), service.GetType());  
 }  
   
   
 [Test]  
 public void ResolveSingleton()  
 {  
 var dependencies = new DependenciesConfig();  
 dependencies.Register<IService, ServiceImpl1>(isSingleton: true);  
   
 var provider = new DependencyProvider(dependencies);  
 var service1 = provider.Resolve<IService>();  
 var service2 = provider.Resolve<IService>();  
  
 Assert.AreSame( service1, service2);  
 }  
   
 [Test]  
 public void ResolveEnumerable()  
 {  
 var dependencies = new DependenciesConfig();  
 dependencies.Register<IService, ServiceImpl1>();  
 dependencies.Register<IService, ServiceImpl2>();  
   
 var provider = new DependencyProvider(dependencies);  
 var services = provider.Resolve<IEnumerable<IService>>();  
   
 Assert.AreEqual(2, (services as List<object>)?.Count);   
 }  
   
 [Test]  
 public void ResolveGenericDependency()  
 {  
 var dependencies = new DependenciesConfig();  
 dependencies.Register<IRepository, MySqlRepository>();  
 dependencies.Register<IService<IRepository>, ServiceImpl<IRepository>>();  
   
 var provider = new DependencyProvider(dependencies);  
 var service = provider.Resolve<IService<IRepository>>();  
   
 Assert.NotNull((service as ServiceImpl<IRepository>)?.Repository);   
 }  
   
 [Test]  
 public void ResolveOpenGenericsDependency()  
 {  
 var dependencies = new DependenciesConfig();  
 dependencies.Register<IRepository, MySqlRepository>();  
 dependencies.Register(typeof(IService<>), typeof(ServiceImpl<>));  
   
 var provider = new DependencyProvider(dependencies);  
 var service = provider.Resolve<IService<IRepository>>();  
   
 Assert.NotNull((service as ServiceImpl<IRepository>)?.Repository);   
 }  
}

Использование библиотеки NLog:

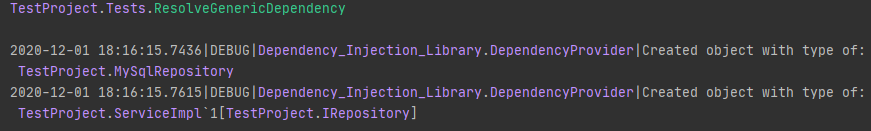
Настройка конфигурации:

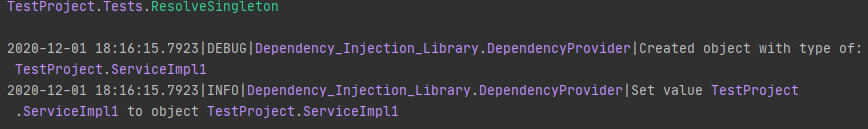
var config = new NLog.Config.LoggingConfiguration();  
  
var logfile = new NLog.Targets.FileTarget("logfile") { FileName = "file.txt" };  
var logconsole = new NLog.Targets.ConsoleTarget("logconsole");  
  
config.AddRule(LogLevel.Debug, LogLevel.Fatal, logconsole);  
config.AddRule(LogLevel.Debug, LogLevel.Fatal, logfile);  
  
NLog.LogManager.Configuration = config;

Пример логирования:

private Object Resolve(ImplementationInfo implementationInfo)  
{  
 Type type = implementationInfo.Type;  
 if (implementationInfo.IsSingleton && implementationInfo.GetValue() != null)  
 {  
 return implementationInfo.GetValue();  
 }  
  
 Object value = CreateObject(type);  
 Logger.Debug("Created object with type of: {0}", type);  
  
 if (implementationInfo.IsSingleton)  
 {  
 implementationInfo.SetValue(value);  
 Logger.Info("Set value {0} to object {1}", value, implementationInfo.GetValue());  
 }  
 return value;  
}

Вывод в консоль:





Вывод в файл:

