

**ОБРАЗОВАТЕЛЬНОЕ ЧАСТНОЕ УЧРЕЖДЕНИЕ
ДОПОЛНИТЕЛЬНОГО ПРОФЕССИОНАЛЬНОГО
ОБРАЗОВАНИЯ "ЦЕНТР ОБУЧЕНИЯ "СПЕЦИАЛИСТ" УНЦ ПРИ
МГТУ ИМ. Н.Э. БАУМАНА
(ОЧУ ДПО «СПЕЦИАЛИСТ»)**

123317, г. Москва, Пресненская набережная, д 8, стр. 1, этаж 48, помещение 484с, комната 3,
ИНН 7701168244, ОГРН 1127799002990

Утверждаю:
Директор ОЧУ ДПО «Специалист»



_____/Е.В. Добрыднева/
« 1 » февраля ____ 2018 года

**Дополнительная профессиональная программа
повышения квалификации
М10967: Основы инфраструктуры Windows Server
2012**

город Москва

Программа разработана в соответствии с приказом Министерства образования и науки Российской Федерации от 1 июля 2013 г. N 499 "Об утверждении Порядка организации и осуществления образовательной деятельности по дополнительным профессиональным программам".

Повышение квалификации слушателей, осуществляемое в соответствии с программой, проводится с использованием модульного принципа построения учебного плана с применением различных образовательных технологий, в том числе дистанционных образовательных технологий и электронного обучения в соответствии с законодательством об образовании.

Дополнительная профессиональная программа повышения квалификации, разработана образовательной организацией в соответствии с законодательством Российской Федерации, включает все модули, указанные в учебном плане.

Содержание оценочных и методических материалов определяется образовательной организацией самостоятельно с учетом положений законодательства об образовании Российской Федерации.

Структура дополнительной профессиональной программы соответствует требованиям Порядка организации и осуществления образовательной деятельности по дополнительным профессиональным программам, утвержденного приказом Минобрнауки России от 1 июля 2013 г. N 499.

Объем дополнительной профессиональной программы вне зависимости от применяемых образовательных технологий, должен быть не менее 16 академических часов. Сроки ее освоения определяются образовательной организацией самостоятельно.

Формы обучения слушателей (очная, очно-заочная, заочная) определяются образовательной организацией самостоятельно.

К освоению дополнительных профессиональных программ допускаются:

- лица, имеющие среднее профессиональное и (или) высшее образование;
- лица, получающие среднее профессиональное и (или) высшее образование.

Для определения структуры дополнительной профессиональной программы и трудоемкости ее освоения может применяться система зачетных единиц. Количество зачетных единиц по дополнительной профессиональной программе устанавливается организацией.

Образовательная деятельность слушателей предусматривает следующие виды учебных занятий и учебных работ: лекции, практические и семинарские занятия, лабораторные работы, круглые столы, мастер-классы, мастерские, деловые игры, ролевые игры, тренинги, семинары по обмену опытом, выездные занятия, консультации, выполнение аттестационной, дипломной, проектной работы и другие виды учебных занятий и учебных работ, определенные учебным планом.

Аннотация. В ходе обучения слушателя получают знания и навыки, необходимые для понимания инфраструктуры Windows Server 2012, сетевой архитектуры и топологии, концепций безопасности и передовой практики администрирования серверных систем. Рассматриваются темы установки, настройки, обслуживания и управления производительностью Windows Server 2012, работы с Active Directory Domain Services (AD DS), Domain Name Services (DNS), хранилищами и многие другие.

1. Цель программы:

В результате прохождения обучения слушатель должен приобрести все необходимые знания и навыки, необходимые для понимания инфраструктуры Windows Server 2012, сетевой архитектуры и топологии, концепций безопасности и передовой практики администрирования серверных систем.

1.1. Планируемый результат обучения:

Лица, успешно освоившие программу, должны овладеть следующими компетенциями:

Совершенствуемые компетенции

№	Компетенция	Направление подготовки ФГОС ВО ПО НАПРАВЛЕНИЮ ПОДГОТОВКИ 09.03.02 «ИНФОРМАЦИОННЫЕ СИСТЕМЫ И ТЕХНОЛОГИИ» (УРОВЕНЬ БАКАЛАВРИАТА)
		Код компетенции
1	способностью участвовать в работах по доводке и освоению информационных технологий в ходе внедрения и эксплуатации информационных систем	ПК-15
2	способностью к инсталляции, отладке программных и настройке технических средств для ввода информационных систем в опытную и промышленную эксплуатацию	ПК-28
3	способностью поддерживать работоспособность информационных систем и технологий в заданных функциональных характеристиках и соответствии критериям качества	ПК-30
4	способностью обеспечивать безопасность и целостность данных информационных систем и технологий	ПК-31
5	способностью адаптировать приложения к изменяющимся условиям функционирования	ПК-32
6	способностью выбирать и оценивать способ реализации информационных систем и устройств (программно-, аппаратно- или программно-аппаратно-) для решения поставленной задачи	ПК-37

Совершенствуемые компетенции в соответствии с трудовыми функциями профессионального стандарта «РУКОВОДИТЕЛЬ ПРОЕКТОВ В ОБЛАСТИ ИНФОРМАЦИОННЫХ ТЕХНОЛОГИЙ», утвержденного приказом Минтруда и социальной защиты РФ от 18 ноября 2014 г. N 893н

№	Компетенция	Направление подготовки
		ПРОФЕССИОНАЛЬНЫЙ СТАНДАРТ «Руководитель проектов в области информационных технологий» Утвержден приказом Минтруда России от 18.11.2014

		<p>№ 893н» (Зарегистрировано в Минюсте России 09.12.2014 № 35117)</p> <p>Наименование вида ПД: Менеджмент проектов в области информационных технологий (ИТ)</p>
		Трудовые функции (код)
В	Управление проектами в области ИТ малого и среднего уровня сложности в условиях неопределенностей, порождаемых запросами на изменения, с применением формальных инструментов управления рисками и проблемами проекта	В/01.7 Планирование конфигурационного управления в проектах малого и среднего уровня сложности в области ИТ
		В/02.7 Идентификация конфигурации ИС
		В/03.7 Ведение отчетности по статусу конфигурации ИС
		В/04.7 Аудит конфигураций ИС в проектах малого и среднего уровня сложности в области ИТ
		В/05.7 Организация репозитория проекта в области ИТ
		В/07.7 Планирование управления изменениями в проектах малого и среднего уровня сложности в области ИТ
		В/08.7 Анализ запросов на изменение в проектах малого и среднего уровня сложности в области ИТ
		В/10.7 Согласование запросов на изменение в проектах малого и среднего уровня сложности в области ИТ
		В/09.7 Проверка реализации запросов на изменение (верификация)
		В/16.7 Организационное и методологическое обеспечение регистрации запросов заказчика в проектах малого и среднего уровня сложности в области ИТ
		В/17.7 Обработка запросов заказчика в проектах малого и среднего уровня сложности в области ИТ
		В/18.7 Закрытие запросов заказчика
		В/27.7 Подготовка предложений по новым инструментам и методам управления проектами
		В/28.7 Подготовка предложений по методам повышения эффективности системы управления проектами
		В/30.7 Сбор информации для инициации проекта в проектах малого и среднего уровня сложности в области ИТ
В/31.7 Планирование в проектах малого и среднего уровня сложности в области ИТ		
В/32.7 Организация исполнения работ проекта в проектах малого и среднего уровня сложности в области ИТ		
В/33.7 Мониторинг и управление работами проекта в проектах малого и среднего уровня сложности в области ИТ		
В/34.7 Общее управление изменениями в проектах малого и среднего уровня сложности в области ИТ		

	V/41.7 Планирование качества в проектах малого и среднего уровня сложности в области ИТ
	V/42.7 Обеспечение качества в проектах малого и среднего уровня сложности в области ИТ
	V/43.7 Контроль качества в проектах малого и среднего уровня сложности в области ИТ
	V/44.7 Организация приемо-сдаточных испытаний (валидация) в проектах малого и среднего уровня сложности в области ИТ
	V/45.7 Планирование управления требованиями в проектах малого и среднего уровня сложности в области ИТ
	V/46.7 Управление работами по выявлению требований в проектах малого и среднего уровня сложности в области ИТ
	V/47.7 Управление работами по анализу требований в проектах малого и среднего уровня сложности в области ИТ
	V/55.7 Планирование коммуникаций в проектах малого и среднего уровня сложности в области ИТ
	V/56.7 Идентификация заинтересованных сторон в проектах малого и среднего уровня сложности в области ИТ

Лица, успешно освоившие программу, должны овладеть следующими компетенциями: установка, настройка, обслуживание и управление производительностью Windows Server 2012, работа с Active Directory Domain Services (AD DS), Domain Name Services (DNS), хранилищами и многие другие.

После окончания обучения Слушатель будет знать, как:

- Настраивать сеть, выбрав для конкретной ситуации аппаратные компоненты и технологии, определив соответствующее сетевое оборудование и необходимые компоненты
- Описывать протоколы и службы в рамках стека протоколов Transmission Control Protocol/Internet Protocol (TCP/IP) и реализовать IPv4 в среде Windows Server
- Описывать роли сервера
- Реализовывать и настраивать лес Active Directory Domain Service (AD DS)

После окончания обучения Слушатель будет уметь:

- Выполнять установку с локального медиа-носителя Windows Server 2012.
- Выбирать соответствующую технологию хранения и настроить хранилище на Windows Server.
- Описывать основные компоненты сети и терминологию
- Описывать концепцию эшелонированной защиты и определить реализацию этого подхода в Windows Server
- Определять функции безопасности в ОС Windows Server, позволяющие реализовать эшелонированную защиту
- Определять сетевые особенности безопасности в Windows Server для предотвращения угроз безопасности вашей сети
- Определять и внедрять дополнительные программные компоненты для повышения безопасности вашей организации

- Проводить мониторинг сервера для определения уровня производительности
- Определять средства Windows Server для обслуживания и устранения неисправностей

Категория слушателей: специалисты, ориентированные на повышение квалификации по специальности специалист по техническому обслуживанию и ремонту ПК.

Требования к предварительной подготовке: успешное окончание курса «Основы сетей, сетевые операционные системы и практикум Wi-Fi», или эквивалентная подготовка.

2. Учебный план:

Срок обучения: 60 академических часов, в том числе аудиторных – 40 час.

Самостоятельные занятия: предусмотрены (20 час).

Форма обучения: очная, очно-заочная, заочная. По желанию слушателя форма обучения может быть изменена и/или дополнена.

Режим занятий: дневной, вечерний, группы выходного дня.

№ п/п	Наименование модулей по программе	Общая трудоемкость (акад. часов)	В том числе аудиторные			СРС
			Всего	Лекций	Практических занятий	
1	Установка и настройка Windows Server 2012	5	3	2	1	2
2	Реализация хранения в Windows Server	4	3	2	1	1
3	Понимание сетевой инфраструктуры	4	3	2	1	1
4	Подключение сетевых компонентов	5	3	2	1	2
5	Реализация TCP/IP	4	3	2	1	1
6	Настройка ролей Windows Server	5	4	3	1	1
7	Реализация Active Directory Domain Services	5	3	2	1	2
8	Реализация уровней ИТ-безопасности	4	3	2	1	1
9	Реализация безопасности Windows Server	4	3	2	1	1
10	Реализация сетевой безопасности	5	3	2	1	2
11	Обеспечение безопасности программного обеспечения	4	3	2	1	1

12	Мониторинг производительности сервера	4	3	2	1	1
13	Поддержка Windows Server	7	3	2	1	4
	Итого:	60	40	27	13	20
	Итоговая аттестация	Тестирование/выполнение практического задания				

Для всех видов аудиторных занятий академический час устанавливается продолжительностью 45 минут.

3. Календарный учебный график

Календарный учебный график формируется при осуществлении обучения в течение всего календарного года. По мере набора групп слушателей по программе составляется календарный график, учитывающий объемы лекций, практики, самоподготовки, выезды на объекты.

Неделя обучения	1	2	3	4	5	6	7	Итого часов
	пн	вт	ср	чт	пт	сб	вс	
1 неделя	4	4	4	4	4	-	-	20
СРС	2	2	2	2	2	-	-	10
2 неделя	4	4	4	4	4ИА	-	-	20
СРС	2	2	2	2	2	-	-	10
Итого:	12	12	12	12	12	-	-	60

4. Рабочие программы учебных предметов

Модуль 1 . Настройка и устранение неполадок службы DNS (5 академ.час.)

- Настройка серверной роли DNS
- Настройка зон DNS
- Настройка передачи зоны DNS
- Управление службой DNS и устранение неполадок
- **Лабораторная работа 1.Настройка и устранение неполадок службы DNS**

Модуль 2 . Поддержка доменных служб Службы Каталога (4 академ.час.)

- Обзор AD DS
- Использование виртуализированных контроллеров домена
- Применение контроллеров домена с доступом только на чтение (RODC)
- Администрирование AD DS
- Управление базой данных AD DS
- **Лабораторная работа 1 Поддержка AD DS**

Модуль 3 . Управление пользовательскими и служебными учетными записями (4 академ.час.)

- Настройка Политики паролей и Политики блокировки учетной записи
- Настройка Управляемой служебной учетной записи

- **Лабораторная работа 1. Управление пользовательскими и служебными учетными записями**

Модуль 4 . Внедрение инфраструктуры Групповых политик (5 академ.час.)

- Обзор Групповой политики
- Внедрение и администрирование Групповых политик
- Область действия и порядок обработки Групповых политик
- Устранение неполадок применения Групповых политик
- **Лабораторная работа 1. Внедрение инфраструктуры Групповых политик**

Модуль 5 . Управление пользовательским рабочим столом через Групповую политику (4 академ.час.)

- Применение Административных шаблонов
- Настройка применения скриптов и перенаправления папок
- Настройка предпочтений в Групповой политике
- Управление программным обеспечением через Групповую политику
- **Лабораторная работа 1. Управление пользовательским рабочим столом через Групповую политику**

Модуль 6 . Установка, настройка и устранение неполадок роли Сервер Сетевой политики. (5 академ.час.)

- Установка и настройка роли Сервер Сетевой политики
- Настройка клиентов и серверов RADIUS
- Методы проверки подлинности сервера Сетевой политики
- Мониторинг и устранение неполадок роли Сервер Сетевой политики
- **Лабораторная работа 1. Установка и настройка роли Сервер Сетевой политики**

Модуль 7 . Применение защиты доступа к сети (5 академ.час.)

- Обзор защиты доступа к сети (NAP)
- Обзор процесса применения защиты доступа к сети
- Настройка NAP
- Настройка применения NAP через принудительные IPSec взаимодействия.
- Мониторинг и устранение неполадок NAP
- **Лабораторная работа 1. Применение защиты доступа к сети**

Модуль 8 . Использование удаленного доступа (4 академ.час.)

- Обзор технологии удаленного доступа
- Внедрение технологии DirectAccess с помощью мастера начальной настройки
- Внедрение и управление расширенной инфраструктурой DirectAccess
- Внедрение VPN
- Внедрение Web Application Proxy
- **Лабораторная работа 1. Внедрение технологии DirectAccess с помощью мастера начальной настройки**
- **Лабораторная работа 2 Развертывание расширенной инфраструктуры DirectAccess**
- **Лабораторная работа 3. Внедрение VPN**

- **Лабораторная работа 4. Внедрение Web Application Proxy**

Модуль 9 . Оптимизация файловых сервисов (4 академ.час.)

- Обзор диспетчера ресурсов файлового сервера - FSRM
- Использование FSRM для управления квотами, файловым экранированием и отчетами по использованию хранилища
- Применение классификации файлов и задач по управлению файлами
- **Лабораторная работа 1. Настройка Квот и файлового экранирования в FSRM**
 - Обзор распределенной файловой системы DFS
 - Настройка именованного пространства DFS
 - Настройка и устранение неполадок репликации DFS
- **Лабораторная работа 2. Применение DFS**

Модуль 10 . Настройка шифрования и расширенного аудита (5 академ.час.)

- Шифрование дисков с использованием BitLocker
- Шифрование файлов с использованием EFS
- Настройка расширенного аудита
- **Лабораторная работа 1. Настройка шифрования и расширенного аудита**

Модуль 11 . Развертывание и поддержка серверных образов (4 академ.час.)

- Обзор службы развертывания Windows
- Управление образами
- Применение развертывания с помощью службы развертывания Windows
- Администрирование службы развертывания Windows
- **Лабораторная работа 1. Использование службы развертывания Windows для развертывания Windows Server 2012**

Модуль 12 . Внедрение управления обновлениями (4 академ.час.)

- Обзор WSUS
- Развертывание обновлений посредством WSUS
- **Лабораторная работа 1. Внедрение управления обновлениями**

Модуль 13 . Мониторинг Windows Server 2012 (7 академ.час.)

- Средства мониторинга
- Использование Монитора производительности
- Мониторинг журналов событий
- **Лабораторная работа 1. Мониторинг Windows Server 2012**

5. Организационно-педагогические условия

Соблюдение требований к кадровым условиям реализации дополнительной профессиональной программы:

а) преподавательский состав образовательной организации, обеспечивающий образовательный процесс, обладает высшим образованием и стажем преподавания по изучаемой тематике не менее 1 года и (или) практической работы в областях знаний, предусмотренных модулями программы, не менее 3 (трех) лет;

б) образовательной организацией наряду с традиционными лекционно-семинарскими занятиями применяются современные эффективные методики преподавания с

применением интерактивных форм обучения, аудиовизуальных средств, информационно-телекоммуникационных ресурсов и наглядных учебных пособий.

Соблюдение требований к материально-техническому и учебно-методическому обеспечению дополнительной профессиональной программы:

а) образовательная организация располагает необходимой материально-технической базой, включая современные аудитории, библиотеку, аудиовизуальные средства обучения, мультимедийную аппаратуру, оргтехнику, копировальные аппараты. Материальная база соответствует санитарным и техническим нормам и правилам и обеспечивает проведение всех видов практической и дисциплинарной подготовки слушателей, предусмотренных учебным планом реализуемой дополнительной профессиональной программы.

б) в случае применения электронного обучения, дистанционных образовательных технологий каждый обучающийся в течение всего периода обучения обеспечивается индивидуальным неограниченным доступом к электронной информационно-образовательной среде, содержащей все электронные образовательные ресурсы, перечисленные в модулях дополнительной профессиональной программы.

6. Формы аттестации и оценочные материалы

Образовательная организация несет ответственность за качество подготовки слушателей и реализацию дополнительной профессиональной программы в полном объеме в соответствии с учебным планом.

Оценка качества освоения дополнительной профессиональной программы слушателей включает текущий контроль успеваемости и итоговую аттестацию.

Конкретные формы и процедуры текущего контроля успеваемости, промежуточной аттестации и итоговой аттестации слушателей устанавливаются образовательной организацией самостоятельно.

Текущий контроль включает в себя посещение семинаров, выполнение практических/лабораторных заданий (если предусмотрены).

Слушателям, успешно освоившим дополнительную профессиональную программу и прошедшим итоговую аттестацию, выдается удостоверение о повышении квалификации.

Слушателям, не прошедшим итоговой аттестации или получившим на итоговой аттестации неудовлетворительные результаты, а также лицам, освоившим часть дополнительной профессиональной программы и (или) отчисленным из образовательной организации, выдается справка об обучении или о периоде обучения по образцу, самостоятельно устанавливаемому образовательной организацией.

Итоговая аттестация проводится по форме тестирования и/или выполнения задания в соответствии с учебным планом.

Результаты итоговой аттестации слушателей в соответствии с формой итоговой аттестации, установленной учебным планом, выставляются по двух бальной шкале («зачтено\не зачтено»). Результаты итоговой аттестации заносятся в соответствующие документы.

10. Оценочные материалы к итоговой аттестации

Итоговая аттестация проводится в форме теста/выполнения задания.

Результаты итоговой аттестации слушателей выставляются по двух бальной шкале («зачтено\не зачтено»). Итоговая аттестация считается пройденной («зачтено»), если слушатель выполнил все лабораторные работы и итоговое задание (не менее 60% правильных ответов).

Пример задания: «Внедрение Web Application Proxy»
Внедрение зависимостей в ASP.NET Web API

Объект *зависимостей* — это любой объект, который требует другой объект. Например, общие для определения репозитория, обрабатывает доступ к данным. Давайте рассмотрим пример. Во-первых мы определим модель домена:

C#Копировать

```
public class Product
{
    public int Id { get; set; }
    public string Name { get; set; }
    public decimal Price { get; set; }
}
```

Ниже приведен простой репозиторий класс, который хранит элементы в базе данных, использующий Entity Framework.

C#Копировать

```
public class ProductsContext : DbContext
{
    public ProductsContext()
        : base("name=ProductsContext")
    {
    }
    public DbSet<Product> Products { get; set; }
}

public class ProductRepository : IDisposable
{
    private ProductsContext db = new ProductsContext();

    public IEnumerable<Product> GetAll()
    {
        return db.Products;
    }
    public Product GetByID(int id)
    {
        return db.Products.FirstOrDefault(p => p.Id == id);
    }
    public void Add(Product product)
    {
        db.Products.Add(product);
        db.SaveChanges();
    }

    protected void Dispose(bool disposing)
    {
        if (disposing)
        {
            if (db != null)
            {

```

```

        db.Dispose();
        db = null;
    }
}

public void Dispose()
{
    Dispose(true);
    GC.SuppressFinalize(this);
}
}

```

Теперь давайте определить контроллер веб-API, который поддерживает запросы GET для `Product` сущностей. (Я пропускаю POST, а также другие методы для простоты.) Вот первой попытки.

C#Копировать

```

public class ProductsController : ApiController
{
    // This line of code is a problem!
    ProductRepository _repository = new ProductRepository();

    public IEnumerable<Product> Get()
    {
        return _repository.GetAll();
    }

    public IHttpActionResult Get(int id)
    {
        var product = _repository.GetByID(id);
        if (product == null)
        {
            return NotFound();
        }
        return Ok(product);
    }
}

```

Обратите внимание, что зависит от класса контроллера `ProductRepository`, и мы позволяя создать контроллер `ProductRepository` экземпляра. Однако это рекомендуется жестко зависимостей таким образом, по следующим причинам.

- Если вы хотите заменить `ProductRepository` с другой реализации, необходимо также изменить класс контроллера.
- Если `ProductRepository` имеет зависимости, необходимо настроить в контроллере. Для больших проектов с несколькими контроллерами код конфигурации становится в разных частях проекта.

- Трудно модульный тест, так как данный контроллер является жестко запрограммировано на запрос в базу данных.

Можно устранить эти проблемы с *вводится* репозитория в контроллер. Во-первых, рефакторинг `ProductRepository` класс в интерфейсе:

C#Копировать

```
public interface IProductRepository
{
    IEnumerable<Product> GetAll();
    Product GetById(int id);
    void Add(Product product);
}

public class ProductRepository : IProductRepository
{
    // Implementation not shown.
}
```

Затем укажите `IProductRepository` как параметр конструктора:

C#Копировать

```
public class ProductsController : ApiController
{
    private IProductRepository _repository;

    public ProductsController(IProductRepository repository)
    {
        _repository = repository;
    }

    // Other controller methods not shown.
}
```

В этом примере используется внедрение конструктора. Можно также использовать *внедрения setter*, где установить зависимость через метод или свойство.

Но теперь имеется проблема, поскольку приложение не создает контроллер напрямую. Создает веб-API контроллер, когда он направляет запрос и веб-API ничего не знает о `IProductRepository`. Это происходит, когда вступает в дело Сопоставитель зависимостей веб-API.

Сопоставитель зависимостей веб-API

Определяет веб-API `IDependencyResolver` интерфейс для разрешения зависимостей. Вот определение интерфейса:

C#Копировать

```
public interface IDependencyResolver : IDependencyScope, IDisposable
{
```

```
        IDependencyScope BeginScope();
    }

    public interface IDependencyScope : IDisposable
    {
        object GetService(Type serviceType);
        IEnumerable<object> GetServices(Type serviceType);
    }
}
```

IDependencyScope интерфейса есть два метода:

- `GetService` создает один экземпляр типа.
- Метод `GetServices` создает коллекцию объектов определенного типа.

IDependencyResolver наследует метод IDependencyScope и добавляет `BeginScope` метод. Мы поговорим об областях далее в этом учебнике.

Если веб-API создает экземпляр контроллера, он сначала вызывает `IDependencyResolver.GetService`, передав тип контроллера. Этот обработчик расширения можно использовать для создания контроллера, разрешаются любые зависимости. Если `GetService` возвращает значение `null`, веб-API ищет конструктор в классе контроллера.

Разрешение зависимостей с контейнером Unity

Несмотря на то, что можно написать полный `IDependencyResolver` действительно реализацию с нуля, интерфейс предназначен для работы в качестве моста между веб-API и существующие контейнеры IoC.

Контейнер IoC — это программный компонент, который отвечает за управление зависимостями. Зарегистрировать типы с контейнером и затем использовать для создания объектов контейнера. Контейнер автоматически решает, отношений зависимостей. Многие контейнеры IoC также позволяют управлять вещью, как время существования объектов и области.

Примечание

«IoC» означает «инверсии элемента управления», который является общий шаблон, когда платформа вызывает в код приложения. Контейнер IoC создает объекты, который «инвертирует «обычного потока управления».

В этом учебнике мы будем использовать Unity из шаблонов Microsoft & рекомендации. (Другие популярные библиотеки включают в себя Castle Windsor, Spring.Net, Autofac, Ninject, и StructureMap.) Установка Unity можно использовать диспетчер пакетов NuGet. Из средства в Visual Studio, выберите пункт меню диспетчер пакетов библиотеки, а затем выберите консоль диспетчера пакетов. В окне консоли диспетчера пакетов введите следующую команду:

```
consoleКопировать
```

```
Install-Package Unity
```

Здесь — это реализация `IDependencyResolver` -оболочки контейнера Unity.

```
C#Копировать
```

```
using Microsoft.Practices.Unity;
using System;
using System.Collections.Generic;
using System.Web.Http.Dependencies;

public class UnityResolver : IDependencyResolver
{
    protected IUnityContainer container;

    public UnityResolver(IUnityContainer container)
    {
        if (container == null)
        {
            throw new ArgumentNullException("container");
        }
        this.container = container;
    }

    public object GetService(Type serviceType)
    {
        try
        {
            return container.Resolve(serviceType);
        }
        catch (ResolutionFailedException)
        {
            return null;
        }
    }

    public IEnumerable<object> GetServices(Type serviceType)
    {
        try
        {
            return container.ResolveAll(serviceType);
        }
        catch (ResolutionFailedException)
        {
            return new List<object>();
        }
    }

    public IDependencyScope BeginScope()
    {
        var child = container.CreateChildContainer();
        return new UnityResolver(child);
    }

    public void Dispose()
    {

```

```
        Dispose(true);
    }
    protected virtual void Dispose(bool disposing)
    {
        container.Dispose();
    }
}
```

Примечание

Если `GetService` метод не может разрешить тип, он должен возвращать `null`. Если `GetServices` метод не может разрешить тип, он должен возвращать пустой объект коллекции.