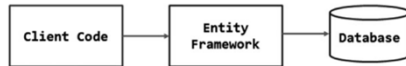
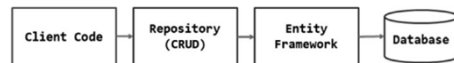


Repository Pattern (一)

■Data access without a repository



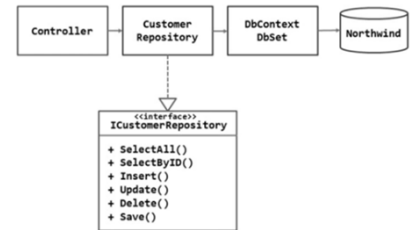
■The repository pattern mediates between the data-access layer and the rest of the system.



29

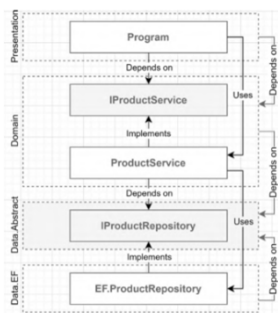
Repository Pattern (二)

■The repository pattern provides collection-like access to the underlying data.



30

Repository Pattern (三)



31

Project: Add a repository interface

Models

- c# Employee.cs
- c# IEmployeeRepository.cs

```

using SampleApplication.Models;

public interface IEmployeeRepository
{
    // 罗列所有员工
    List<Employee> SelectAll();
}
  
```

32

Project: Implement the interface

```

public class EmployeeRepository:IEmployeeRepository
{
    //将员工数据保存在一个列表中
    private List<Employee> data = new List<Employee>();
    //添加初始数据
    public EmployeeRepository()
    {
        data.Add(new Employee() { EmployeeId = 1, EmployeeName = "Adam", Title = "manager",
        Country = "USA" });
        data.Add(new Employee() { EmployeeId = 2, EmployeeName = "Bob", Title = "president",
        Country = "UK" });
        data.Add(new Employee() { EmployeeId = 3, EmployeeName = "Cathy", Title = "sale",
        Country = "USA" });
    }
    public List<Employee> SelectAll()
    {
        return data;
    }
}
  
```

33

Project: Implement the List action method in controller

```

//导入名称空间
using SampleApplication.Models;

public class EmployeeManagerController : Controller
{
    private IEmployeeRepository repo;
    public EmployeeManagerController(IEmployeeRepository r)
    {
        //获取接口实例
        repo = r;
    }
    //罗列所有员工
    public IActionResult List()
    {
        return View(repo.SelectAll());
    }
}
  
```

34

5. View

Conventions for Views

- Views are stored in the Views folder, organized into subfolders.
- Views that are associated with the Home controller, for example, are stored in a folder called Views/Home.
- Views that are not specific to a single controller are stored in a folder called Views/Shared.

36

Razor View Engine

- View engines provide a specific syntax to work with server-side elements and process the view to render the HTML in the browser.
- Razor is a view engine, and the expressions in views are known as Razor expressions.

37

Invoking a View (一)

- When an action method invokes the View method, it creates a ViewResult that tells the MVC Framework to use the default convention to locate a view.
- The Razor view engine looks for a view with the same name as the action method, with the addition of the cshtml file extension.

```
public IActionResult List(){return View(repo.SelectAll());}
```

38

Invoking a View (二)

- Action methods can select a view by providing a name as an argument to the View method.

```
return View("Watersports", prod);
```

39

Understanding the Razor Syntax (一)

- The Razor compiler separates the static fragments of HTML from the C# expressions.
- Directives are expressions that give instructions to the Razor view engine.
 - @model
 - @using
 - @addTagHelper
 -

40

Understanding the Razor Syntax (二)

- Razor content expressions produce content that is included in the output generated by a view.
- The key transition character in Razor is the “at” sign (@).
- Blocks of code require curly braces to delimit the block of code.

```
<h1>Listing @items.Length items.</h1>
@{
    string s = "One line of code.";
    ViewBag.Title "Another line of code";
}
```

41

Understanding the Razor Syntax (三)

```
@*Razor*
<div>This page rendered at @DateTime.Now</div>
```

```
@if (DateTime.Today.Year == 2013) {
    <span> Current year </span>
}
else { <span> Past year </span> }
```

```
<ul>
@foreach( var auction in auctions) {
    <li><a
        href="@auction.Href">@auction.Title</a></li>
}
</ul>
```

42