

.NET Conf China

2022

.NET在教育行业的应用分享 ——以“全栈开发技术”课程为例

张引

东北大学软件学院-副教授



.NET Conf China

关于我

相比于一名“教师”，我更觉得自己是一名“开发者开发爱好者”

2002年秋

进入东北大学开始本科学习
开始学习.NET开发技术

2012年秋

留校任教
参与“可视化程序设计”课程授课，介绍Windows Form开发技术

2017年春

负责并改造“可视化程序设计”课程，采用主动学习（Active Learning）方法介绍UWP开发技术



关于我

相比于一名“教师”，我更觉得自己是一名“开发者开发爱好者”

2019年春

开设“全栈开发技术”课程，采用项目式学习（Problem Based Learning）方法介绍Xamarin开发技术

2021年秋

整合两门选修课，通过增加项目学时推动更深入的技术学习

2022年秋

重构“全栈开发技术”课程，介绍.NET MAUI开发技术



关于我

相比于一名“教师”，我更觉得自己是一名“**开发者开发爱好者**”

2023年春

开设“**微服务架构与设计模式**”课程，以eShopOnContainers参考应用为蓝本介绍.NET云原生开发技术

2023年秋-2026年秋

尚不可公开

很荣幸在个人成长的道路上一直有.NET相伴

也很荣幸能通过.NET将个人所学传递给未来的开发者们



(软件) 工程教育

“复杂工程问题解决能力”是中国乃至世界工程教育所关注的核心

复杂工程问题解决能力

并非单独的一项能力，而是一系列专业技能、跨学科技能、以及包括自我引导学习技能、沟通技能、管理技能、批判性思维技能，创造性问题解决技能在内的可转换技能的有机融合，而非线性相加

“全栈”

从软件与互联网业界的视角阐述了解决复杂的工程问题需要具备哪些技能，为培养复杂工程问题解决能力提供了完备的能力框架与技术体系

采用“全栈”思想培养复杂工程问题解决能力

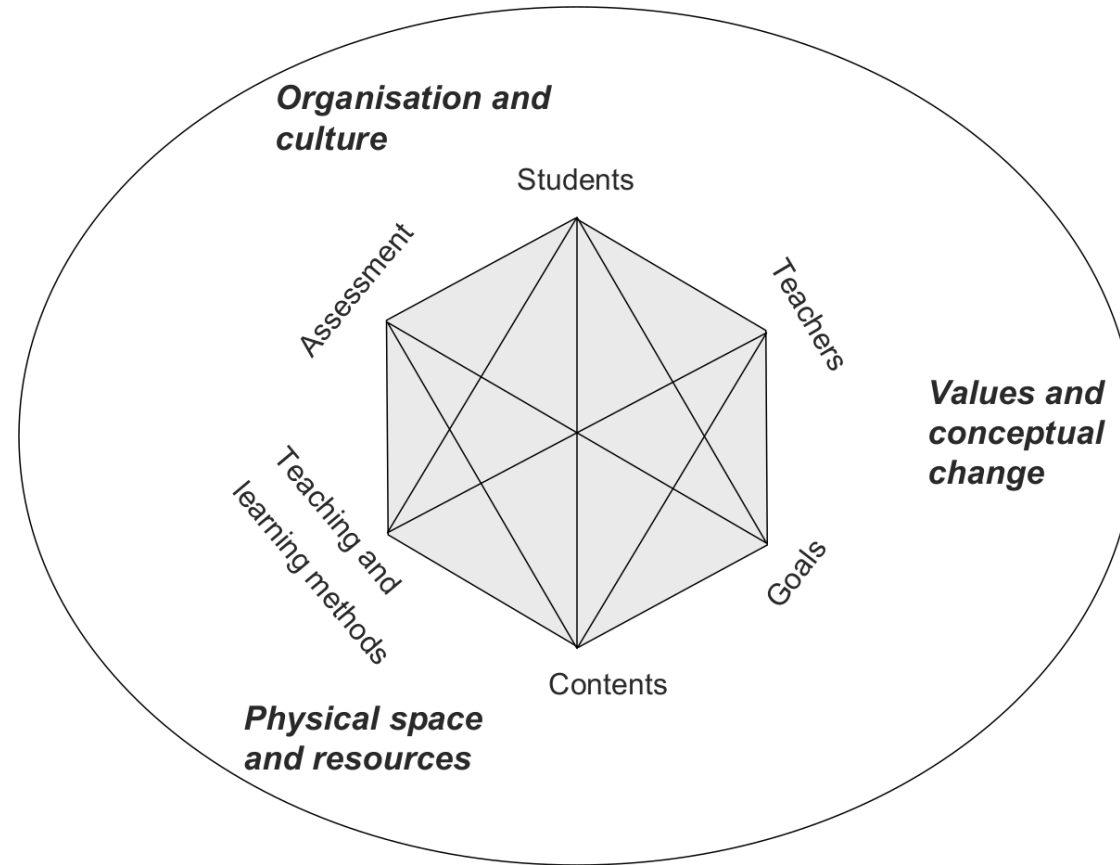
为中国乃至世界的工程教育提供切实可行的思路与方法



东北大学浑南校区风雨操场

由2022年“全栈开发技术”课程同学原榛良拍摄

工程教育设计



.NET与工程教育设计

为采用“全栈”思想开展工程教育设计提供了完备的解决方案

教学目标：培养“复杂工程问题解决能力”

顶流的开发平台，不仅为解决各类复杂工程问题提供了完整的技术栈与工具链，也为培养复杂工程问题解决能力规划了完备的知识、技能与能力体系。教师可以轻松形成明确的教学目标，并确保教学目标与实际需求紧密衔接。

教师：可持续的技术演进

持续而稳定的技术演进在推动教师不断更新自身技术的同时，也避免过大的技术变动从而给教师带来过重的学习负担。只有教师跟上技术的发展，才能确保新技术持续不断地进入高校课程体系，避免出现技术断层，导致教育与实际脱节。

学生：降低学习负担，推进深入学习

宁静致远。统一的语言、技术与思想让学生“一次学习，多处复用”，避免将时间浪费在反复学习不同的语言、技术和思维方式上，从而将注意力集中在“解决问题”而非“学习技术”上，推进学习的深度。



东北大学浑南校区1号教学楼
由东北大学软件学院同学拍摄

.NET

为采用“全栈”思想开展工程教育设计提供了完备的解决方案

教学内容：规范而优雅参考项目

学生会反复模仿教学项目，因此教学内容的示范性至关重要。官方和社区提供大量规范而优雅的参考项目，支持教师形成具有高度示范性的教学项目，确保学习过程符合行业的需求和期待。

学习与考核：资料全面，避免抄袭

高质量的文档涵盖了各类型的问题，帮助学生学会自我引导学习。持续的技术演进不仅避免了“以题搜题”，推动学生独立解决问题，还让学生学会过滤垃圾信息，培养批判性思维。

资源

完整、高质量、免费且持续更新的工具体系将学生的学习与未来的工作无缝衔接。富有活力的社区极大地降低了获取新知识的成本，有效拓宽学生的视野。

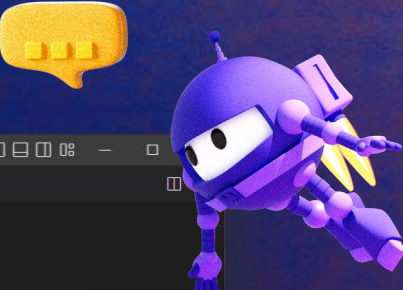


东北大学浑南校区1号教学楼
由东北大学软件学院同学拍摄

教学设计实例



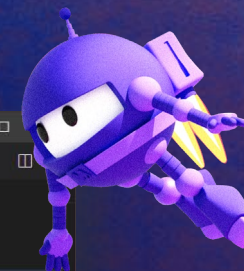
基本用法



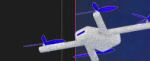
```
code sample.cs x
D: > LiveEvents > dotNetConf > 2022 > code sample.cs
1 // DailyPoetryM
2
3 public class DetailPageViewModel : ObservableObject {
4     public DetailPageViewModel(IFavoriteStorage favoriteStorage) {
5         _favoriteStorage = favoriteStorage;
6         _lazyFavoriteToggledCommand =
7             new Lazy<AsyncRelayCommand>(
8                 new AsyncRelayCommand(FavoriteToggledCommandFunction));
9     }
10
11     public Poetry? Poetry {
12         get => _poetry;
13         set => SetProperty(ref _poetry, value);
14     }
15
16     private Poetry? _poetry;
17
18     public AsyncRelayCommand NavigatedToCommand =>
19         _lazyNavigatedToCommand.Value;
20
21     private Lazy<AsyncRelayCommand> _lazyNavigatedToCommand;
22
23     public async Task NavigatedToCommandFunction() {
24         IsLoading = true;
25         var favorite = await _favoriteStorage.GetFavoriteAsync(Poetry.Id) ??
26             new Favorite {PoetryId = Poetry.Id};
27         IsFavorite = favorite.IsFavorite;
28         Favorite = favorite;
29         IsLoading = false;
30     }
31 }
32
33 }
```

Ln 33, Col 1 Spaces: 4 UTF-8 CRLF C# Go Live

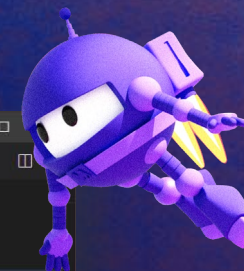
单元测试



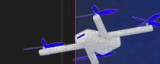
```
code sample.cs x
D: > LiveEvents > dotNetConf > 2022 > code sample.cs
1 // DailyPoetryM
2
3 [Fact]
4 public async Task NavigatedToCommandFunction_Default() {
5     var poetry = new Poetry {Id = 0};
6     var favoriteToReturn =
7         new Favorite {PoetryId = poetry.Id, IsFavorite = true};
8
9     var favoriteStorageMock = new Mock<IFavoriteStorage>();
10    favoriteStorageMock
11        .Setup(p => p.GetFavoriteAsync(favoriteToReturn.PoetryId))
12        .ReturnsAsync(favoriteToReturn);
13    var mockFavoriteStorage = favoriteStorageMock.Object;
14
15    var detailPageViewModel = new DetailPageViewModel(mockFavoriteStorage);
16    detailPageViewModel.Poetry = poetry;
17
18    var loadingList = new List<bool>();
19    detailPageViewModel.PropertyChanged += (sender, args) => {
20        if (args.PropertyName == nameof(DetailPageViewModel.IsLoading)) {
21            loadingList.Add(detailPageViewModel.IsLoading);
22        }
23    };
24
25    await detailPageViewModel.NavigatedToCommandFunction();
26
27    favoriteStorageMock.Verify(
28        p => p.GetFavoriteAsync(favoriteToReturn.PoetryId), Times.Once);
29    Assert.Same(favoriteToReturn, detailPageViewModel.Favorite);
30    Assert.True(detailPageViewModel.IsFavorite);
31    Assert.Equal(2, loadingList.Count);
32    Assert.True(loadingList.First());
33    Assert.False(loadingList.Last());
34 }
```



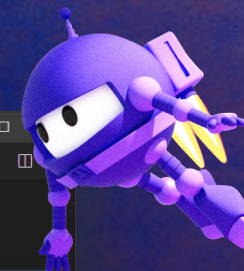
数据绑定的原理



```
code sample.cs x
D: > LiveEvents > dotNetConf > 2022 > code sample.cs
1 // CommunityToolkit.Mvvm
2
3 protected bool SetProperty<T>([NotNullIfNotNull("newValue")] ref T field, T newValue, [CallerMemberName] string? propertyName =
4 null)
5 {
6     if (EqualityComparer<T>.Default.Equals(field, newValue))
7         return false;
8     this.OnPropertyChanging(propertyName);
9     field = newValue;
10    this.OnPropertyChanged(propertyName);
11    return true;
12 }
13
14 protected virtual void OnPropertyChanged(PropertyChangedEventArgs e)
15 {
16     CommunityToolkit.Mvvm.ArgumentNullException.ThrowIfNull((object) e, nameof(e));
17     PropertyChangedEventHandler propertyChanged = this.PropertyChanged;
18     if (propertyChanged == null)
19         return;
20     propertyChanged((object) this, e);
21 }
22
23
24
25
26
27
28
29
30
```

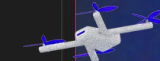


事件转命令的原理



```
code sample.cs x
D:\> LiveEvents > dotNetConf > 2022 > code sample.cs
1 // MauiBehaviors, ported from Behaviors.Forms to .NET MAUI
2 // For educational purpose only
3 private void RegisterEvent(string eventName) {
4     if (string.IsNullOrEmpty(eventName)) {
5         return;
6     }
7
8     var sourceObjectType = _resolvedSource.GetType();
9     var eventInfo = sourceObjectType.GetRuntimeEvent(eventName);
10    if (eventInfo is null) {
11        return;
12    }
13
14    var methodInfo = typeof(MauiEventHandlerBehavior).GetTypeInfo()
15        .GetDeclaredMethod("OnEvent");
16    _eventHandler = methodInfo.CreateDelegate(eventInfo.EventHandlerType,
17        this);
18    eventInfo.AddEventHandler(_resolvedSource, _eventHandler);
19 }
20
21
22
23
24
25
26
27
28
29
30
```

Ln 33, Col 1 Spaces: 4 UTF-8 CRLF C# Go Live



基于.NET的教学设计

实现可持续的教学改进

整合且贯通的知识体系

从客户端到服务器端，整合而一致的知识体系贯通软件工程、软件构造、设计模式、软件质量保证、计算机网络、操作系统、密码学、编程语言等十余门专业课，降低学习负担同时促进知识整合

稳定而持续的技术迭代

从MVVMLight到CommunityToolkit.Mvvm，从Xamarin到MAUI，从Simpleloc到Microsoft.Extensions.DependencyInjection，知识体系在保持稳定的同时不断进化

热情友好的社区氛围

.NET广大的社区成员提供友好而持续不断的改进思路，促进教学与现实需求始终紧密衔接



社区反馈

评论详情 ✕

 CoreDX **LV6**
2小时前 IP属地: 云南

你可能需要System.Linq.Async 🐱

    ⋮

相关回复共5条 ☰ 按时间

 CoreDX **LV6**
2小时前 IP属地: 云南

当然返回值也就变成IEnumerable<T>了 🤔

    ⋮

学生反馈



我欠我个未来 LVE

2小时前

来给张老师打call! 去年选了张老师的全栈课, 感觉是本科阶段学到东西最多的课了。抛开项目本身不说, 如何使用git, 如何写标准的单元测试, 接触一些有意思的技术问题, 这绝对是其他课程很难接触到的。我现在在阿里实习, 项目本身也很有东西, 这个课程的经历对我今年找实习帮助非常大, 不限于项目经历本身, 各种意义上的提升。想提升自己编程水平的赶紧冲!

[收起](#)



5



社区反馈



小祺吱吱 LVE

1小时前

少见的精品课程。一般开发中很完整的最佳实践指导，在公司里工作四五年的开发者者可能也未必掌握这么全面。非常感谢把这个课程分享出来 👍 🙌 期待MAUI出来后的新版课程！



教材建设

面向新工科专业建设计算机系列教材 丛书主编
计算机科学与技术 张尧学

Xamarin 全栈开发技术与实践 (微课版)

张引 赵玉丽 张斌 高克宁 编著

教学课件 / 教学大纲 / 教学计划 / 教学视频 / 配套素材



清华大学出版社

谢谢

感谢.NET社区对课程改革工作的付出

欢迎到访东北大学

关注我，B站@东北大学张引



东北大学浑南校区西门
由东北大学软件学院同学拍摄

基于.NET的教学设计

实现可持续的教学改进

整合且贯通的知识体系

从客户端到服务器端，整合而一致的知识体系贯通软件工程、软件构造、设计模式、软件质量保证、计算机网络、操作系统、密码学、编程语言等十余门专业课，降低学习负担同时促进知识整合

稳定而持续的技术迭代

从MVVMLight到CommunityToolkit.Mvvm，从Xamarin到MAUI，从Simpleloc到Microsoft.Extensions.DependencyInjection，知识体系在保持稳定的同时不断进化

热情友好的社区氛围

.NET广大的社区成员提供友好而持续不断的改进思路，促进教学与现实需求始终紧密衔接

