



KARTA OPISU PRZEDMIOTU - SYLABUS

Nazwa przedmiotu

Programowanie wizualne [S1Inf1>PWIZ]

Przedmiot

Kierunek studiów
Informatyka

Rok/Semestr
4/7

Studia w zakresie (specjalność)
–

Profil studiów
ogólnoakademicki

Poziom studiów
pierwszego stopnia

Język oferowanego przedmiotu
polski

Forma studiów
stacjonarne

Wymagalność
obieralny

Liczba godzin

Wykład
30

Laboratorium
30

Inne
0

Ćwiczenia
0

Projekty/seminaria
0

Liczba punktów ECTS

4,00

Koordynatorzy

dr inż. Paweł Wojciechowski
pawel.wojciechowski@put.poznan.pl

Wykładowcy

Wymagania wstępne

Student rozpoczynający ten przedmiot powinien posiadać podstawową wiedzę z zakresu metodologii programowania, znać metodologię programowania obiektowego i omawiane uprzednio języki programowania obiektowego, znać podstawowe wzorce projektowe oraz architekturę aplikacji modułowych. Powinien również znać podstawowe pojęcia z zakresu algorytmiki, złożoności obliczeniowej, programowania systemów współbieżnych i współpracy z bazami danych. Powinien posiadać umiejętność rozwiązywania podstawowych problemów algorytmicznych, korzystania z zaawansowanych systemów programowania, budowania kodu wysokiej jakości oraz umiejętność pozyskiwania informacji ze wskazanych źródeł.

Cel przedmiotu

1. Przekazanie studentom podstawowej wiedzy dotyczącej zaawansowanego środowiska programowania wizualnego .NET i .NET Core, w zakresie: szczegółowej znajomości obiektowego języka programowania C#, zestawu klas i funkcji udostępnianych przez biblioteki Windows Presentation Foundation, technologii ADO.NET. 2. Rozwijanie u studentów umiejętności rozwiązywania podstawowych problemów algorytmicznych, umiejętności projektowania oraz implementowania aplikacji okienkowych. Celem przedmiotu jest również pogłębienie umiejętności stosowania wzorców projektowych, tworzenia bibliotek dynamicznych DLL, testowania tworzonych aplikacji, tworzenia aplikacji współbieżnych i rozproszonych oraz aplikacji współpracujących z bazami danych. 3. Kształtowanie u studentów umiejętności pracy z kodem, którego jakość pozwala na ponowne wykorzystanie części kodu w innych projektach programistycznych.

Przedmiotowe efekty uczenia się

Wiedza:

1. zna architekturę środowiska programowego .NET i posiada wiedzę o podstawowych technologiach udostępnianych przez to środowisko i sposobach tworzenia aplikacji konsolowych i okienkowych oraz przygotowywania bibliotek dynamicznych DLL.
2. zna szczegółowo język programowania obiektowego C# i jego zaawansowane konstrukcje (m. in. wyrażenia lambda, funkcje rozszerzające, wyrażenia LINQ)
3. zna podstawowe wzorce projektowe stosowane przy tworzeniu aplikacji ze szczególnym uwzględnieniem wzorca Model- View-ViewModel
4. zna biblioteki klas i funkcji środowiska programowego .NET umożliwiające tworzenie aplikacji okienkowych Windows Presentation Foundation
5. zna techniki programowania obiektowego, wykorzystania wybranych wzorców projektowych, budowy elastycznej architektury aplikacji i interfejsu użytkownika. Zna środowisko Visual Studio

Umiejętności:

1. potrafi wybrać sposób zabezpieczenia interfejsu użytkownika aplikacji okienkowej wykorzystującej biblioteki Windows Presentation Foundation przed wprowadzaniem błędnych danych, krytycznie ocenić wady i zalety zastosowanego rozwiązania
2. potrafi dokonać analizy sposobu funkcjonowania systemu informatycznego z interfejsem użytkownika, ocenić architekturę aplikacji w zakresie łatwości dokonywania w niej zmian. Potrafi ocenić architekturę oprogramowania z punktu możliwości zmiany źródła danych (aplikacja wielowarstwowa), niezależności warstwy prezentacji aplikacji od jej logiki funkcjonalnej (model MVVM)
3. potrafi, formułując i rozwiązując zadania informatyczne, zastosować odpowiednio dobrane metody dostępu do danych, zaprojektować architekturę aplikacji
4. potrafi - zgodnie z zadaną specyfikacją - zaprojektować oraz zrealizować prostą bazę danych opartą o relację Producent - Produkt używając właściwych metod, technik i narzędzi
5. Student potrafi zaimplementować algorytm filtrowania i sortowania danych w aplikacji z interfejsem użytkownika
6. potrafi znaleźć rozwiązanie wybranych problemów programistycznych w ramach technologii .NET i WPF

Kompetencje społeczne:

1. Student rozumie, że platforma .NET ciągle się rozwija i w kolejnych wersjach wprowadzane są nowe elementy języka C#
2. ma świadomość znaczenia wiedzy i zaawansowanych technik udostępnianych w ramach języka C# i bibliotek WCF i WPF w rozwiązywaniu problemów inżynierskich

Metody weryfikacji efektów uczenia się i kryteria oceny

Efekty uczenia się przedstawione wyżej weryfikowane są w następujący sposób:

Sprawdzanie założonych efektów kształcenia realizowane jest przez:

- ocenę wiedzy i umiejętności związanych z realizacją kolejnych zadań projektowych / laboratoryjnych,
- ocenę realizacji kolejnych etapów projektu zaliczeniowego. Projekt zaliczeniowy składa się z czterech obligatoryjnych i jednego opcjonalnego etapu. Każdy z nich jest sprawdzany i omawiany.
- ocenę wiedzy zdobytej na wykładzie w formie albo krótkich, 10 minutowych testów przeprowadzanych na wykładzie i obejmujących 1-2 wykłady wstecz z wykorzystaniem platformy moodle albo w formie kolokwium po zakończeniu wykładów.

Każdy z testów przeprowadzanych na wykładach oceniany jest tak samo, przy czym maksymalną liczbę punktów wyznacza najlepszy uzyskany wynik. Względem niego wyznaczana jest ocena procentowa. Aby uzyskać zaliczenie w tej formie, należy uzyskać średnią z testów min. 50%, przy czym dwa najniższe wyniki testów nie są brane pod uwagę. Student, który nie uzyskał wymaganych punktów ma obowiązek przystąpić do kolokwium zaliczeniowego po skończeniu wykładów.

Treści programowe

W ramach przedmiotu prezentowane są zaawansowane mechanizmy języka C# i platformy .NET z naciskiem na implementację graficznych interfejsów użytkownika (GUI) na przykładzie platformy .NET Core i bibliotek WPF, .NET MAUI oraz ADO.NET Core.

Tematyka zajęć

W ramach wykładu omawiane są następujące zagadnienia:

- Wprowadzenie do środowiska .NET Core i języka C#
- Podstawowe elementy języka C#: rodzaje danych, wbudowane typy zmiennych, funkcje i właściwości dostępne dla typów danych, łańcuchy znaków, tablice, struktury i klasy.
- Zaawansowane elementy języka: indeksatory, konwersja typów, typy parametryczne (generics), interfejsy, iteratory, typy i metody anonimowe, delegacje, wyrażenia lambda, zdarzenia, wyjątki, zapytania LINQ, współbieżność,
- Architektura ADO.NET Core (Razor, Blazor)
- współpraca z bazami danych Entity Framework (EF)
- Biblioteka WPF: architektura aplikacji, XAML, pojemniki, zdarzenia, Dependency Objects, wiązanie danych, style, tworzenie komponentów, zasoby, animacje, elementy graficzne, system komend, model MVVM, weryfikacja danych formularza, widoki danych.
- Biblioteka .NET MAUI

Laboratorium.

W początkowej części zajęć laboratoryjnych studenci implementują proste aplikacje stanowiące przykład omawianych na wykładach mechanizmów. Programy te są realizowane przez wszystkich studentów jako aplikacje konsolowe w celu pokazania zaawansowanych konstrukcji języka C#. Kolejnym etapem jest realizacja aplikacji okienkowych korzystających z wybranej biblioteki GUI z uwzględnieniem zaawansowanych mechanizmów oferowanych tę bibliotekę (m. in. mechanizmów wiązania (binding) i powiadamiania o zmianach właściwości (property change notification)). Równolegle, na bazie zdobytej wiedzy studenci rozwijają indywidualnie lub w niewielkich zespołach bardziej rozbudowaną, warstwową aplikację z interfejsem użytkownika, realizowaną zgodnie z modelem MVVM.

Metody dydaktyczne

Wykłady: prezentacje multimedialne, przedstawianie fragmentów kodów i wyników ich działania

Laboratoria: zadania do realizacji, pisanie kodu aplikacji na żywo, prezentacje multimedialne

Literatura

Podstawowa

1. Troelsen, Andrew W., Japikse, Philip F, Język C# 6.0 i platforma .NET 4.6, PWN, 2017.
2. Michaelis, Mark., Lippert, Eric., C# 6.0 : Kompletny przewodnik dla praktyków, Helion, 2016.
3. Raffaele Garofalo, Budowanie aplikacji biznesowych za pomocą Windows Presentation Foundation i wzorca Model View ViewM, PROMISE, 2011
4. Dokumentacja elektroniczna systemu programowania wizualnego Visual Studio.NET
5. Dokumentacja języka C# i bibliotek WPF i WCF

Uzupełniająca

1. Matthew MacDonald, Pro WPF 4.5 in C#: Windows Presentation Foundation in .NET 4.5, Apress, 2012.
2. Magennis, Troy, LINQ to Objects w C# 4.0 : wygodne operacje na danych!, Helion, 2012.
3. Metsker, Steven John, C# - wzorce projektowe, Helion, 2005.
4. WCF - Getting started tutorial <https://docs.microsoft.com/en-us/dotnet/framework/wcf/getting-started-tutorial>
5. C# 12 and .NET 8 - Modern Cross-Platform Development Fundamentals - Eighth Edition: Start building websites and services with ASP.NET Core 8, Blazor, and EF Core 8, Packt Publishing, 2023

Bilans nakładu pracy przeciętnego studenta

	Godzin	ECTS
Łączny nakład pracy	100	4,00
Zajęcia wymagające bezpośredniego kontaktu z nauczycielem	60	2,50
Praca własna studenta (studia literaturowe, przygotowanie do zajęć laboratoryjnych/ćwiczeń, przygotowanie do kolokwίων/egzaminu, wykonanie projektu)	40	1,50