

Platforma.NET - opis przedmiotu

Informacje ogólne	
Nazwa przedmiotu	Platforma.NET
Kod przedmiotu	11.3-WI-INFP-PNET
Wydział	Wydział Informatyki, Elektrotechniki i Automatyki
Kierunek	Informatyka
Profil	ogólnoakademicki
Rodzaj studiów	pierwszego stopnia z tyt. inżyniera
Semestr rozpoczęcia	semestr zimowy 2020/2021

Informacje o przedmiocie	
Semestr	6
Liczba punktów ECTS do zdobycia	6
Typ przedmiotu	obieralny
Język nauczania	polski
Sylabus opracował	dr hab. inż. Marek Sawerwain, prof. UZ

Formy zajęć					
Forma zajęć	Liczba godzin w semestrze (stacjonarne)	Liczba godzin w tygodniu (stacjonarne)	Liczba godzin w semestrze (niestacjonarne)	Liczba godzin w tygodniu (niestacjonarne)	Forma zaliczenia
Wykład	30	2	18	1,2	Egzamin
Laboratorium	30	2	18	1,2	Zaliczenie na ocenę
Projekt	15	1	9	0,6	Zaliczenie na ocenę

Cel przedmiotu

- ukształtowanie wśród studentów zrozumienia i świadomości roli pojęcia platformy informatycznej w codziennej praktyce programisty rozwiązań informatycznych
- zapoznanie studentów z podstawowymi informacjami o platformie .NET (w tym także w odmianie .NET Core) jako o środowisku do tworzenia programów funkcjonujących w tradycyjnym środowisku użytkownika (aplikacje desktop) oraz w sieci Internet (technologia ASP.NET)
- nauka podstawowych umiejętności w zakresie tworzenia programów w języku C#, wykorzystywania systemów baz danych, opisu danych za pomocą XML, tworzenia dynamicznych stron WWW w technologii ASP.NET
- ukształtowanie podstawowych umiejętności w zakresie tworzenia usług sieciowych w tym bezpieczeństwa aplikacji tworzonych za pomocą technologii .NET, a także poinformowanie o dostępności innych niż C# języków programowania oferowanych przez platformę .NET.

Wymagania wstępne

Podstawy programowania, Programowanie obiektowe, Algorytmy i struktury danych, Bazy danych

Zakres tematyczny

Wprowadzenie do platformy .NET. Przedstawienie struktury platformy .NET. Rodzaje i typy platformy .NET. Zarys środowiska .NET Framework oraz .NET Core.

Przegląd języków programowania dla platformy.NET. Charakterystyka pakietu Microsoft Visual Studio.

Prezentacja środowiska programistycznego. Edycja formularzy. Uruchamianie aplikacji. Tworzenie przykładowej aplikacji dla systemu Windows.

Środowisko CLR (ang. Common Language Runtime). Podstawowe funkcje i usługi CLR. Zarządzanie pamięcią i zasobami. Zarządzanie wątkami. Struktura i układ metadanych. Obsługa błędów przy użyciu wyjątków. Integracja z bibliotekami Windows DLL. Metody interakcji między aplikacjami.

Podstawy programowania w języku C#. Omówienie składni języka: instrukcje, zmienne, operatory i typy danych. Zasady tworzenia klas, metod, konstruktorów oraz obiektów tych klas. Użycie tablic.

Przegląd podstawowych narzędzi zawartych w SDK (ang. Software Development Kit).

Zaawansowane elementy języka C#. Dyrektywy preprocesora. Obsługa zdarzeń. Obsługa błędów za pomocą wyjątków. Operacje na łańcuchach znaków. Korzystanie z komponentów interfejsu Windows. Wykorzystanie wyrażeń regularnych. Zdalne wywoływanie obiektów. Dostęp i operacje na plikach. Wątki i ich synchronizacja. Konstrukcje programowania asynchronicznego async oraz await.

Omówienie BCL (ang. Base Class Library). Budowanie komponentów .NET. Zasady tworzenia, projektowanie, implementacja i testowanie komponentów.

Współpraca z komponentami COM i COM+.

Wykorzystanie języka XML na potrzeby platformy .NET. Sposoby wymiany informacji z wykorzystaniem dokumentów XML. Przegląd klas wspomagających przetwarzanie dokumentów XML.

Dostęp do danych przy użyciu ADO.NET. Przegląd obiektów ADO.NET. Metody dostępu do baz danych. Prezentacja danych z baz danych na witrynach internetowych.

Obsługa baz danych za pomocą odwzorowania obiektowego .NET Entity Framework. Język zapytań zintegrowanych – LINQ.

Technologia ASP.NET (ang. Active Server Pages .NET). Klasy bazowe i podstawowe obiekty ASP.NET. Użycie języka XML w połączeniu z ASP.NET. Tworzenie stron WWW zawierających komponenty ASP.NET.

Tworzenie usług sieciowych przy użyciu WEB Services, Wykorzystanie protokołów SOAP (ang. Simple Object Access Protocol) i UDDI (ang. Universal Description, Discovery and Integration).

Bezpieczeństwo aplikacji ASP.NET: kontrola dostępu, autoryzacja, szyfrowanie danych.

Metody kształcenia

Wykład: wykład konwencjonalny/tradycyjny.

Laboratorium: ćwiczenia laboratoryjne, wg listy zadań.

Projekt: praca w grupach, metoda projektu.

Efekty uczenia się i metody weryfikacji osiągnięcia efektów uczenia się

Opis efektu	Symbole efektów	Metody weryfikacji	Forma zajęć
Jest zorientowany w możliwościach platformy.NET w dziedzinie bezpieczeństwa aplikacji oraz zabezpieczania informacji tworzonej przez użytkowników systemu bądź aplikacji.NET.	K_W20	sprawdzian z progami punktowymi	Wykład
Student potrafi utworzyć aplikację działającą w ramach środowiska.NET wykorzystując jej możliwości.	K_U29	- obserwacja i ocena aktywności na zajęciach - sprawdzian z progami punktowymi	Laboratorium
Potrafi utworzyć dokumentację dla nowo tworzonych projektów oraz opisywania istniejących programów/bibliotek/pakietów.NET.	K_U29	- obserwacja i ocena aktywności na zajęciach - sprawdzian z progami punktowymi	Laboratorium
Potrafi przeprowadzić analizę i interpretację istniejących problemów, a następnie wskazać sposoby i techniki rozwiązywania tych problemów przy użyciu platformy.NET.	K_U29	- projekt - sprawozdanie z projektu	Projekt
Znane się mu podstawowe składowe platformy i potrafi scharakteryzować wady oraz zalety platformy.NET. Ma świadomość dynamicznego rozwoju platformy.NET.	K_W20	sprawdzian z progami punktowymi	Wykład
Potrafi samodzielnie pracować nad systemem informatycznym, a także znane są mu narzędzia do pracy w zespole dostępne w ramach platformy.NET.	K_U29	- projekt - wykonanie sprawozdań laboratoryjnych	Projekt
Znane są mu również zasady postępowania zgodnego z etyką oraz zagadnienia licencyjne korzystania z pakietów/komponentów/bibliotek innych producentów we własnych projektach.NET.	K_W20	sprawdzian z progami punktowymi	Wykład
Potrafi w podstawowym zakresie tworzyć nowe komponenty (i rozwijać istniejące), do rozwiązywania problemów informatycznych. Znana jest mu struktura komponentów.NET.	K_U29	sprawdzian z progami punktowymi	Laboratorium

Warunki zaliczenia

Wykład - warunkiem zaliczenia jest uzyskanie pozytywnej oceny z egzaminu przeprowadzonego w formie pisemnej.

Laboratorium - warunkiem zaliczenia jest uzyskanie pozytywnych ocen ze wszystkich ćwiczeń laboratoryjnych, przewidzianych do realizacji w ramach programu laboratorium.

Projekt - warunkiem zaliczenia jest uzyskanie pozytywnych ocen ze wszystkich zadań projektowych, przewidzianych do realizacji w ramach zajęć projektowych.

Składowe oceny końcowej = wykład: 40% + laboratorium: 30% + projekt: 30%

Literatura podstawowa

- Verma R., Visual Studio Extensibility Development, Apress 2020.
- Freeman A., Pro ASP.NET Core 3, Apress 2020.
- Sarcar V., Getting Started with Advanced C#, Apress, 2020.
- Ovais Mehboob Ahmed Khan. :C# 7 i .NET Core 2.0. Programowanie wielowątkowych i współbieżnych aplikacji, Helion 2019.
- Price M. J.: C# 7.1 i .NET Core 2.0 dla programistów aplikacji wieloplatformowych, Helion, 2018.
- Troelsen A., Japikse P.: Język C# 6.0 i platforma .NET 4.6, Wydawnictwo Naukowe PWN, 2017.
- Żydzik K., Rak T., C# 6.0 i MVC 5. Tworzenie nowoczesnych portali internetowych, Helion, 2015.
- Chadwick J., Snyder T., Panda H., ASP.NET MVC 4. Programowanie, Helion, 2013.
- Matulewski J., Visual Studio 2013. Podręcznik programowania w C# z zadaniami, Helion, 2014.

10. Löwy J., Programowanie usług WCF, Helion, 2012.
11. Magennis T., LINQ to Objects w C# 4.0, Helion, 2012.
12. Chappell D.: Zrozumieć platformę .NET, Helion, 2007.
13. Liberty J.: C# 2005. Wprowadzenie, O'Reilly, 2007.
14. Michelsen K.: Język C#. Szkoła programowania, Helion, 2007.
15. Duffy J.: .NET Framework 2.0, Zaawansowane programowanie, Helion, 2007.
16. Burton K.: .NET CLR. Księga eksperta, Helion, 2002.

Literatura uzupełniająca

1. Palermo J., .NET DevOps for Azure, Apress, 2019.
2. Sawhney R., Beginning Azure Functions, Apress, 2019.
3. Classon I., Migrating ASP.NET Microservices to ASP.NET Core, Apress, 2019.
4. Troelsen A., Japikse P.: Pro C# 7 With .NET and .NET Core, APress, 2017.
5. Johnson B., Professional Visual Studio 2017, Wiley Publishing, Inc., 2017.
6. Novák. I, Velvárt A., Granicz A., Balássy G., Hajdrik A., Sellers M., Hillar G.C., Molnár A., Kanjilal J.: Visual Studio 2010 and .NET 4 Six-in-One, Wiley Publishing, Inc., 2010.
7. Nash T.: Accelerated C# 2010, APress, 2010.
8. Solis D.M.: Illustrated C# 2010, APress, 2010.
9. Freeman A., Rattz J.C. Jr.: Pro LINQ: Language Integrated Query in C#, A-Press, 2010.
10. Richter J., CLR via C#, 3rd edition, Microsoft Press, 2010.
11. Chappell D., Understanding .NET (2nd Edition), Addison-Wesley Professional, 2nd edition, 2006.

Uwagi

Zmodyfikowane przez dr hab. inż. Marek Sawerwain, prof. UZ (ostatnia modyfikacja: 25-04-2020 16:29)

Wygenerowano automatycznie z systemu SyllabUZ