



UNIWERSYTET
IM. ADAMA MICKIEWICZA
W POZNANIU

Języki programowania JavaScript

Sylabus zajęć

Informacje podstawowe

Kierunek studiów Informatyka		Cykl dydaktyczny 2023/24	
Specjalność -		Kod zajęć 06INFN.310S.02434.23	
Jednostka organizacyjna Wydział Matematyki i Informatyki		Języki wykładowe polski	
Poziom studiów studia inżynierskie pierwszego stopnia		Obligatoryjność Fakultatywny	
Forma studiów studia niestacjonarne		Blok zajęciowy Przedmioty specjalnościowe	
Profil studiów profil ogólnoakademicki			
Koordynator zajęć	Marek Gałązka		
Prowadzący zajęcia	Marek Gałązka		
Okres Semestr 5	Forma zajęć / liczba godzin / forma zaliczenia • Laboratorium: 15, Zaliczenie z oceną		Liczba punktów ECTS 3

Cele kształcenia dla zajęć

Kod	Cel
C1	Celem przedmiotu jest zapoznanie studentów z tworzeniem oprogramowania w technologiach opartych na JavaScript/TypeScript, w szczególności w ramach następujących obszarów: - podstawowe koncepcje i konstrukcje językowe związane z technologią JavaScript/TypeScript, - wybrane przykłady stosów technologicznych wykorzystywanych przy tworzeniu aplikacji JavaScript/TypeScript oraz zagadnienia związane z cyklem życia tych aplikacji, - wybrany przykład lub przykłady bibliotek i platform programistycznych JavaScript/TypeScript.

Wymagania wstępne

Przydatne są efekty uczenia się osiągnięte w ramach następujących przedmiotów: Algorytmy i struktury danych, Programowanie aplikacji internetowych oraz Programowanie obiektowe.

W obszarze kompetencji społecznych cenna jest chęć współpracy w zespole, rozwijania własnych umiejętności oraz pomagania innym.

Efekty uczenia się dla zajęć

Kod	Efekty uczenia się dla zajęć w zakresie	Efekty uczenia się dla kierunku	Metody weryfikacji osiągnięcia efektów uczenia się dla zajęć
Wiedzy - Student/ka:			
W1	opisuje wybrane koncepcje i struktury języka JavaScript/TypeScript	INF_K3_W02, INF_K3_W03, INF_K3_W04_inz	Test, Projekt
W2	identyfikuje wybrane zastosowania technologii JavaScript/TypeScript	INF_K3_W02, INF_K3_W03, INF_K3_W04_inz	Test, Projekt
W3	wskazuje metody projektowania, wdrażania i testowania aplikacji JavaScript/TypeScript	INF_K3_W02, INF_K3_W03, INF_K3_W04_inz	Projekt
Umiejętności - Student/ka:			
U1	tworzy proste aplikacje w technologii JavaScript/TypeScript z wykorzystaniem wybranych bibliotek i platform programistycznych	INF_K3_U03, INF_K3_U05_inz, INF_K3_U08	Projekt
U2	rozwiązuje problemy i zadania jako uczestnik zespołu	INF_K3_U02, INF_K3_U03, INF_K3_U04_inz, INF_K3_U08	Projekt
U3	analizuje dostępne materiały i źródła wiedzy w celu rozwiązywania problemów programistycznych	INF_K3_U08, INF_K3_U09	Test, Projekt
Kompetencji społecznych - Student/ka:			
K1	jest gotowy/a do prezentacji i omówienia wypracowanych rozwiązań	INF_K3_K01, INF_K3_K02, INF_K3_K05	Projekt

Treści programowe dla zajęć

Lp.	Treści programowe dla zajęć	Efekty uczenia się dla zajęć	Formy zajęć
1.	Wprowadzenie do JavaScript/TypeScript. Podstawowe zastosowania i struktury językowe.	W1, W2	Laboratorium
2.	Architektura aplikacji klient - serwer w kontekście programowania JavaScript/TypeScript.	W1, W2, U3	Laboratorium
3.	Wprowadzenie do tworzenia prostych aplikacji klienckich JavaScript/TypeScript. Przykładowa architektura i podstawowe komponenty aplikacji klienckiej JavaScript/TypeScript.	W1, W2, U1	Laboratorium

Lp.	Treści programowe dla zajęć	Efekty uczenia się dla zajęć	Formy zajęć
4.	Korzystanie z API REST z poziomu aplikacji klienckiej JavaScript/TypeScript.	W2, U1	Laboratorium
5.	Routing w aplikacjach klienckich JavaScript/TypeScript.	W2, W3, U1, U3	Laboratorium
6.	Debugowanie, testowanie i obsługa błędów w aplikacjach klienckich JavaScript/TypeScript.	W1, W3, U1	Laboratorium
7.	Wprowadzenie do tworzenia prostych aplikacji serwerowych JavaScript/TypeScript. Przykładowa architektura i podstawowe komponenty aplikacji serwerowej JavaScript/TypeScript.	W2, W3, U1	Laboratorium
8.	Debugowanie, testowanie i obsługa błędów w aplikacjach serwerowych JavaScript/TypeScript.	W1, W3, U1, U3	Laboratorium
9.	Podsumowanie rozwiązań problemów wypracowanych w trakcie semestru.	U2, K1	Laboratorium

Informacje dodatkowe

Forma zajęć	Metody i formy prowadzenia zajęć
Laboratorium	Dyskusja, Rozwiązywanie zadań (np.: obliczeniowych, artystycznych, praktycznych), Metoda ćwiczeniowa, Metoda laboratoryjna, Metoda projektu

Forma zajęć	Warunki zaliczenia zajęć
Laboratorium	W czasie semestru student ma możliwość zbierania punktów na zaliczenie. Maksymalnie może uzyskać 100 punktów, zaliczenie jest od 50. Jeśli w trakcie zajęć student nie zdoła uzyskać wystarczającej na zaliczenie liczby punktów, będzie miał taką możliwość w czasie sesji poprawkowej. Na łączny wynik (100 punktów) w trybie zwykłym składają się następujące elementy: - quiz – 30 punktów, - projekt zespołowy – 50 punktów - dokumentacja projektowa – 10 punktów, - prezentacja projektu zespołowego – 10 punktów. Na łączny wynik (100 punktów) w trybie poprawkowym składają się następujące elementy: - quiz – 30 punktów, - projekt zespołowy – 50 punktów - dokumentacja projektowa – 10 punktów, - prezentacja projektu zespołowego – 10 punktów. Skala ocen: 0 - 49 niedostateczny 50 - 59 dostateczny 60 - 69 dostateczny plus 70 - 79 dobry 80 - 89 dobry plus 90 - 100 bardzo dobry

Literatura

Obowiązkowa

1. Crockford D.: JavaScript - mocne strony, Wydawnictwo Helion, Gliwice, 2011

Dodatkowa

1. Duckett J.: JavaScript i jQuery. Interaktywne strony WWW dla każdego. Podręcznik Front-End Developera, Wydawnictwo Helion, Gliwice, 2015.
2. Wilken J.: Angular w akcji, Wydawnictwo Helion, Gliwice, 2019.
3. Banks A., Porcello E.: React od podstaw. Nowoczesne wzorce tworzenia aplikacji. Wydanie II, Wydawnictwo Helion, Gliwice, 2021.
4. Brown E.: Tworzenie aplikacji internetowych z użyciem Node i Express. Korzystanie ze stosu JavaScript. Wydanie II, Wydawnictwo Helion, Gliwice, 2020.

Nakład pracy studenta i punkty ECTS

Rodzaje zajęć studenta	Średnia liczba godzin* przeznaczonych na zrealizowane rodzaje zajęć
Laboratorium	15
Czytanie wskazanej literatury	10
Przygotowanie projektu	40
Przygotowanie do zaliczenia	10
Łączny nakład pracy studenta	Liczba godzin 75
Liczba punktów ECTS	ECTS 3

* godzina (lekcyjna) oznacza 45 minut

Efekty uczenia się dla kierunku

Kod	Treść
INF_K3_K01	Absolwent/ka jest gotów/gotowa do precyzyjnego formułowania pytań służących pogłębieniu własnego zrozumienia danego tematu lub odnalezienia brakujących elementów rozumowania
INF_K3_K02	Absolwent/ka jest gotów/gotowa do zasięgania opinii ekspertów przy rozwiązywaniu problemów teoretycznych i praktycznych
INF_K3_K05	Absolwent/ka jest gotów/gotowa do przedstawiania w sposób przystępny podstawowych zagadnień z zakresu informatyki, porozumiewania się przy użyciu słownictwa technicznego w środowisku zawodowym, również w języku angielskim oraz z wykorzystaniem narzędzi informatycznych
INF_K3_U02	Absolwent/ka potrafi pozyskiwać wiarygodne informacje z literatury, baz wiedzy, Internetu oraz innych źródeł, integrować je, interpretować oraz wyciągać wnioski i formułować opinie
INF_K3_U03	Absolwent/ka potrafi pracować indywidualnie i w zespole, zarządzać swoim czasem oraz podejmować zobowiązania i dotrzymywać terminów
INF_K3_U04_inz	Absolwent/ka potrafi opracować, przeanalizować, zaprojektować klasyczne algorytmy i systemy informatyczne
INF_K3_U05_inz	Absolwent/ka potrafi pisać, uruchamiać i testować programy w wybranym środowisku programistycznym
INF_K3_U08	Absolwent/ka potrafi posługiwać się językiem angielskim zgodnie z wymaganiami określonymi dla poziomu B2 Europejskiego Systemu Opisu Kształcenia Językowego oraz zna język angielski w stopniu umożliwiającym czytanie ze zrozumieniem dokumentacji oprogramowania, podręczników i artykułów informatycznych
INF_K3_U09	Absolwent/ka potrafi przygotować dokumentację, opracowania i raporty w języku polskim i języku angielskim, w tym z wykorzystaniem podstawowych ujęć teoretycznych, a także różnych źródeł
INF_K3_W02	Absolwent/ka zna i rozumie zaawansowane pojęcia i problemy formujące kanon dyscypliny informatyka
INF_K3_W03	Absolwent/ka zna i rozumie narzędzia, technologie i urządzenia informatyczne właściwe dla wybranych obszarów zastosowań oraz podstawy ich działania
INF_K3_W04_inz	Absolwent/ka zna i rozumie zaawansowane pojęcia, konstrukcje i procesy związane z językami programowania i inżynierią programowania