

**KARTA PRZEDMIOTU****I. Dane podstawowe**

Nazwa przedmiotu	Programowanie aplikacji internetowych
Nazwa przedmiotu w języku angielskim	Internet applications development
Kierunek studiów	Informatyka
Poziom studiów (I, II, jednolite magisterskie)	I stopnia
Forma studiów (stacjonarne, niestacjonarne)	stacjonarne
Dyscyplina	Informatyka
Język wykładowy	polski

Koordynator przedmiotu	Dr Rafał Stęgiński
------------------------	--------------------

Forma zajęć ( <i>katalog zamknięty ze słownika</i> )	Liczba godzin	semestr	Punkty ECTS
wykład			5
konwersatorium	30	IV	
ćwiczenia			
laboratorium	30	IV	
warsztaty			
seminarium			
proseminarium			
lektorat			
praktyki			
zajęcia terenowe			
pracownia dyplomowa			
translatorium			
wizyta studyjna			

Wymagania wstępne	1. Podstawy algorytmiki i programowania 2. Projektowanie stron WWW 3. Programowanie obiektowe
-------------------	---

**II. Cele kształcenia dla przedmiotu**

C1 - Zapoznanie się z podstawami języka PHP
C2 - Zapoznanie się z technikami programowania aplikacji webowych
C3 - Zapoznanie się z technikami programowania aplikacji konsolowych
C4 - Zapoznanie się z podstawowymi wzorcami projektowymi
C5 - Zapoznanie się ze strukturą i cyklem wykonania aplikacji webowych

### III. Efekty uczenia się dla przedmiotu wraz z odniesieniem do efektów kierunkowych

Symbol	Opis efektu przedmiotowego	Odniesienie do efektu kierunkowego
<b>WIEDZA</b>		
W_01	Ma wiedzę na temat algorytmiki i programowania i jest w stanie wykorzystać ją w kontekście projektowania, tworzenia i wdrażania aplikacji internetowych	K_W06
<b>UMIEJĘTNOŚCI</b>		
U_01	Potrafi posługiwać się narzędziami i środowiskami związanymi z tworzeniem serwisów internetowych	K_U04, K_U02
U_02	Potrafi projektować strony www i serwisy internetowe	K_U05, K_U02
<b>KOMPETENCJE SPOŁECZNE</b>		
K_01	Potrafi ocenić poziom swojej wiedzy w kontekście tworzenia aplikacji internetowych	K_K01

### IV. Opis przedmiotu/ treści programowe

<p>1. Sieć WWW</p> <ol style="list-style-type: none"> <li>a. Protokół http</li> <li>b. Zapytania GET oraz POST</li> <li>c. Przetwarzanie po stronie klienta</li> <li>d. Przetwarzanie po stronie serwera</li> <li>e. REST</li> </ol> <p>2. EcmaScript</p> <ol style="list-style-type: none"> <li>a. historia standardu</li> <li>b. składnia języka</li> <li>c. sposób budowy aplikacji</li> <li>d. wykrywanie i obsługa błędów</li> </ol> <p>3. TypeScript</p> <ol style="list-style-type: none"> <li>a. różnice w odniesieniu do EcmaScript</li> <li>b. wykorzystanie w aplikacjach internetowych</li> </ol> <p>4. NodeJS</p> <ol style="list-style-type: none"> <li>a. architektura</li> <li>b. podstawowe składowe</li> <li>c. bazowe biblioteki</li> <li>d. frameworki</li> <li>e. testowanie i rozwój oprogramowania</li> <li>f. orm</li> </ol>
--

### V. Metody realizacji i weryfikacji efektów uczenia się

Symbol efektu	Metody dydaktyczne <i>(lista wyboru)</i>	Metody weryfikacji <i>(lista wyboru)</i>	Sposoby dokumentacji <i>(lista wyboru)</i>
<b>WIEDZA</b>			
W_01	Wykład konwersatoryjny/Praca pod kierunkiem	Egzamin	Protokół

UMIEJĘTNOŚCI			
U_01	Ćwiczenia praktyczne design thinking	Projekt	Karta oceny projektu
U_02	Ćwiczenia praktyczne design thinking	Projekt	Karta oceny projektu
K_01	Metoda projektu design thinking	Projekt	Karta oceny projektu

## VI. Kryteria oceny, wagi...

Na ocenę 3 student:

W1 - omówić składnię języka EcmaScript i opisać mechanizm uruchamiania aplikacji internetowej i sposobu komunikacji w oparciu o nią

U1 - uruchomić przykładowe aplikacje internetowe wykonane w różnych frameworkach/językach

U2 - tworzyć podstawowe aplikacje w NodeJS operujące na danych z plików lub formularzy

K1 - potrafi indywidualnie zaplanować pracę nad aplikacją

Na ocenę 4 student:

W1 - zna model obiektowy EcmaScript i TypeScript

U1 - implementować obiektowe biblioteki rozwiązujące bardziej zaawansowane zadania

U2 - wykorzystywać własne biblioteki do implementacji aplikacji

K1 - pracować indywidualnie i grupowo zaplanować pracę nad aplikacją

Na ocenę 5 student potrafi:

W1 - podać przykłady użycia omówionych wzorców projektowych

U1 - stosować w praktyce wzorzec MVC

## VII. Obciążenie pracą studenta

Forma aktywności studenta	Liczba godzin
Liczba godzin kontaktowych z nauczycielem	90
Liczba godzin indywidualnej pracy studenta	60

## VIII. Literatura

Literatura podstawowa
Ethan Brown, Tworzenie aplikacji internetowych z użyciem Node i Express. Korzystanie ze stosu JavaScript. Wydanie II, Helion
David Herron, Platforma Node.js. Przewodnik webdevelopera. Wydanie III, Helion
Literatura uzupełniająca
Adam Freeman, TypeScript. Od początkującego do profesjonalisty. <a href="https://nodejs.org/en/docs/">https://nodejs.org/en/docs/</a>