

<b>KARTA OPISU MODUŁU KSZTAŁCENIA</b>		
Nazwa modułu/przedmiotu <b>Wybrane technologie internetowe</b>		Kod <b>1010331561010337132</b>
Kierunek studiów <b>Informatyka</b>	Profil kształcenia (ogólnoakademicki, praktyczny) <b>(brak)</b>	Rok / Semestr <b>3 / 6</b>
Ścieżka obieralności/specjalność <b>Technologie informatyczne</b>	Przedmiot oferowany w języku: <b>polski</b>	Kurs (obligatoryjny/obieralny) <b>obligatoryjny</b>
Stopień studiów: <b>I stopień</b>	Forma studiów (stacjonarna/niestacjonarna) <b>stacjonarna</b>	
Godziny Wykłady: <b>30</b> Ćwiczenia: - Laboratoria: <b>30</b> Projekty/seminaria: -		Liczba punktów <b>5</b>
Status przedmiotu w programie studiów (podstawowy, kierunkowy, inny) <b>(brak)</b>		(ogólnouczelniany, z innego kierunku) <b>(brak)</b>
Obszar(y) kształcenia i dziedzina(y) nauki i sztuki		Podział ECTS (liczba i %)
<b>Odpowiedzialny za przedmiot / wykładowca:</b>		
dr inż. Andrzej Szwabe email: Andrzej.Szwabe@put.poznan.pl tel. 61 665 3958 Wydział Elektryczny ul. Piotrowo 3A 60-965 Poznań		
<b>Wymagania wstępne w zakresie wiedzy, umiejętności, kompetencji społecznych:</b>		
1	<b>Wiedza:</b>	zna i rozumie metodyki wytwarzania oprogramowania oraz podstawowe wzorce projektowe, a także podstawowe procesy zachodzące w cyklu życia oprogramowania zna i rozumie budowę, działanie i podstawowe procesy zachodzące w cyklu życia sieci komputerowych zna i rozumie w zaawansowanym stopniu wiedzę w zakresie baz i hurtowni danych oraz podstawowe procesy zachodzące w cyklu życia baz i hurtowni danych [K1_W12 (P6S_WG), K1_W07 (P6S_WG), K1_W08 (P6S_WG)]
2	<b>Umiejętności:</b>	potrafi posłużyć się środowiskami i platformami programistycznymi do pisania, wykonywania i testowania prostych programów kodowanych w językach programowania imperatywnego, obiektowego i deklaratywnego, wykorzystać w tym celu metody analityczne, symulacyjne i eksperymentalne potrafi zaprojektować ? zgodnie z zadaną specyfikacją ? oraz wykonać prostą bazę danych lub hurtownię danych, posługiwać się prostymi zapytaniami, używając odpowiednio dobranych metod, technik i narzędzi [K1_U10 (P6S_UW), K1_U12 (P6S_UW)]
3	<b>Kompetencje społeczne</b>	jest gotów do odpowiedzialnego pełnienia ról zawodowych, w tym przestrzegania zasad etyki zawodowej i wymagania tego od innych [K1_K04 (P6S_KR)]
<b>Cel przedmiotu:</b>		
Głównym celem przedmiotu jest dostarczenie studentowi wiedzy i umiejętności z zakresu technologii budowy zaawansowanych systemów świadczenia usług w Internecie ? w szczególności zgodnych z paradygmatem mikrousługowym (ang. microservices) i paradygmatem tzw. konteneryzacji (wirtualizacji poziomu systemu operacyjnego, ang. operating-system-level virtualization). Jako najważniejszy z uwzględnianych scenariuszy aplikacyjnych przyjmuje się wykorzystanie w usłudze internetowej funkcji tzw. uczenia maszynowego ? z tej perspektywy zakres programowy przedmiotu stanowi dopełnienie zakresów przedmiotów poświęconych technologiom sieciowym, systemom sztucznej inteligencji i systemom rekomendacyjnym.		
<b>Efekty kształcenia i odniesienie do kierunkowych efektów kształcenia</b>		
<b>Wiedza:</b>		
1. zna i rozumie w zaawansowanym stopniu wiedzę w zakresie technologii internetowych - [K1_W11 (P6S_WG)] 2. zna i rozumie podstawowe procesy zachodzące w cyklu życia urządzeń, obiektów i systemów informatycznych - [K1_W19 (P6S_WG)]		
<b>Umiejętności:</b>		

1. potrafi zaprojektować ? zgodnie z zadaną specyfikacją ? oraz wykonać podstawowe zadania dotyczące portali i usług internetowych, używając odpowiednio dobranych metod, technik i narzędzi - [K1_U15 (P6S_UW)]
2. potrafi oszacować czas potrzebny na realizację zleconego zadania; opracować i zrealizować harmonogram prac zapewniający dotrzymanie terminów; planować i organizować pracę ? indywidualną oraz w zespole - [K1_U02 (P6S_UO)]
<b>Kompetencje społeczne:</b>
1. jest gotów do odpowiedzialnego pełnienia ról zawodowych, w tym przestrzegania zasad etyki zawodowej i wymagania tego od innych - [K1_K04 (P6S_KR )]

<b>Sposoby sprawdzenia efektów kształcenia</b>
Wykład: egzamin pisemny z punktowanymi pytaniami, zaliczenie od 50,1% punktów.
Laboratorium: punktowane wyniki pracy na każdym zajęciach, zaliczenie od 50,1% punktów.
<b>Treści programowe</b>
<p>Wykłady:</p> <p>Ważnym celem wykładu jest przekazanie studentowi wiedzy na temat technologii wytwarzania mikrousług, a w szczególności technologii projektowania, implementacji (tj. przygotowania kodu, w tym wykorzystania techniki programowania tzw. dekoratorów) w języku programowania Python (wybór języka jest podyktowany dominującą pozycją wśród języków programowania, które umożliwiają komercyjne wdrażanie usług sieciowych i usług predykcyjnych opartych na technologiach Machine Learning) oraz testowania mikrousług wyposażonych w interfejsy typu Web API lub typu REST. Przedstawiana jest metoda pracy nad implementacją umożliwiającą efektywne testowanie wstępnej implementacji poza środowiskiem kontenera i bez użycia serwera HTTP zapewniającego wielowątkowość, a następnie przeprowadzenie nieskomplikowanej transformacji wstępnej implementacji do postaci skonteneryzowanej (z użyciem technologii Docker) zapewniającej wielowątkową obsługę zapytań (dzięki zastosowaniu tzw. produkcyjnego serwera HTTP). W toku prezentacji technologii konteneryzacji szczególną uwagę zwraca się na funkcje współdzielenia przestrzeni dyskowej i komunikacji między kontenerami. Student uzyskuje również wiedzę na temat serwerów baz danych, serwerów indeksujących i ich aplikacji klienckich - ze szczególnym uwzględnieniem technologii NoSQL (w tym Redis i Elasticsearch) i ich zastosowań do realizacji funkcji pamięci podręcznej (ang. cache) oraz technologii budowy modeli predykcyjnych, usług generowania predykcji z użyciem predefiniowanych modeli predykcyjnych (określanych jako tzw. serwowanie modelu, ang. model serving) i usług przygotowywania danych.</p> <p>Laboratoria</p> <p>Tematy ćwiczeń laboratoryjnych obejmują: przygotowywanie kodu (w języku Python) usługi Web (w postaci interfejsu Web API i/lub usługi REST) m.in. z użyciem techniki programistycznej tzw. dekoratorów, dwuetapowe testowanie kodu mikrousług (wyposażonych w interfejsy typu Web API lub usług sieciowych typu REST) prowadzone w sposób umożliwiający testowanie wstępnej implementacji poza środowiskiem wirtualizacji poziomu systemu operacyjnego i bez użycia serwera HTTP zapewniającego wielowątkowość obsługi (ang. multithreading), a następnie nieskomplikowane przekształcenie implementacji do postaci skonteneryzowanej z użyciem technologii Docker zapewniającej wielowątkowość obsługi zapytań (zrealizowanej z użyciem tzw. produkcyjnego serwera HTTP), zastosowania technologii Flask, Requests i CherryPy, techniki konteneryzacji, (w szczególności funkcji współdzielenia przestrzeni dyskowej i międzykontenerowej komunikacji sieciowej), zastosowanie technologii bazodanowych NoSQL (m.in. Redis) i serwerów indeksowania (m.in. Elasticsearch) do realizacji architektur mikrousługowych ? ze szczególnym uwzględnieniem mechanizmu pamięci podręcznej (ang. cache), mikrousługi sterowania budową modeli predykcyjnych, generowania predykcji z użyciem uprzednio zbudowanych modeli predykcyjnych (określanych jako serwowanie modeli, ang. model serving), mikrousługi przygotowywania danych niezbędnych do budowy modeli predykcyjnych, zastosowania technologii TensorFlow (w szczególności TensorFlow Estimator API i TensorFlow ModelServer).</p> <p>Zastosowane metody kształcenia - wykład:</p> <ul style="list-style-type: none"><li>- prezentacja teorii z częstymi odniesieniami do praktycznych przykładów implementacji,</li><li>- wykład z prezentacjami multimedialnymi i prezentacjami przykładów kodu źródłowego w języku programowania Python, jego modyfikacji i wyników uruchomienia,</li><li>- dyskusje inicjowane w trakcie wykładu.</li></ul> <p>Zastosowane metody kształcenia - laboratorium:</p> <ul style="list-style-type: none"><li>- indywidualna praca studenta (wg treści skryptu danego ćwiczenia),</li><li>- praca konfiguracyjna i prowadzenie testów/eksperymentów w środowisku sieciowym złożonym z wielu kontenerów,</li><li>- praca z narzędziami o otwartym kodzie źródłowym (ang. open source) i składnikami oprogramowania opracowanymi w projektach badawczych Politechniki Poznańskiej udostępnianymi studentom przez prowadzącego zajęcia,</li><li>- na koniec każdego zajęć jawna oceną wyników dokonana przez wykładownicę,</li><li>- na koniec semestru przygotowanie sprawozdania z realizacji wszystkich zadań.</li></ul>

<b>Literatura podstawowa:</b>		
<p>1. Sam Newman? Building Microservices. Designing Fine-Grained Systems, O'Reilly Media, 2015, <a href="https://books.google.pl/books/about/Building_Microservices.html">https://books.google.pl/books/about/Building_Microservices.html</a></p> <p>2. Mark Richards, Microservices vs. Service-Oriented Architecture, O'Reilly Media, 2016</p> <p>3. Leonard Richardson and Sam Ruby. 2007. Restful Web Services (First ed.). O'Reilly, <a href="http://restfulwebapis.org/RESTful_Web_Services.pdf">http://restfulwebapis.org/RESTful_Web_Services.pdf</a></p> <p>4. Heng-Tze Cheng et al., Wide &amp; Deep Learning for Recommender Systems. In Proceedings of the 1st Workshop on Deep Learning for Recommender Systems (DLRS 2016). ACM, New York, NY, USA, 2016, 7-10. DOI: <a href="http://dx.doi.org/10.1145/2988450.2988454">http://dx.doi.org/10.1145/2988450.2988454</a>, <a href="https://arxiv.org/pdf/1606.07792.pdf">https://arxiv.org/pdf/1606.07792.pdf</a></p> <p>5. D. Baylor et al, TFX: A TensorFlow-Based Production-Scale Machine Learning Platform, In Proceedings of the 23rd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD '17), ACM, New York, NY, USA, 1387-1395, 2017, DOI: <a href="https://doi.org/10.1145/3097983.3098021">https://doi.org/10.1145/3097983.3098021</a>, <a href="http://stevenwhang.com/tfx_paper.pdf">http://stevenwhang.com/tfx_paper.pdf</a></p>		
<b>Literatura uzupełniająca:</b>		
<p>1. M. Abadi et al., TensorFlow: a system for large-scale machine learning. In Proceedings of the 12th USENIX conference on Operating Systems Design and Implementation (OSDI'16). USENIX Association, Berkeley, CA, USA, 265-283, <a href="https://www.usenix.org/system/files/conference/osdi16/osdi16-abadi.pdf">https://www.usenix.org/system/files/conference/osdi16/osdi16-abadi.pdf</a></p> <p>2. G. Zaccane, Getting Started with TensorFlow, Packt Publishing, July 2016, ISBN: 9781786468574, <a href="https://www.packtpub.com/free-ebook/getting-started-tensorflow">https://www.packtpub.com/free-ebook/getting-started-tensorflow</a></p> <p>3. Sieci komputerowe i intersieci, D.E. Comer, Helion, Warszawa, 2012.</p> <p>4. Sieci komputerowe, A. Tanenbaum, Helion, Gliwice, 2012.</p>		
<b>Bilans nakładu pracy przeciętnego studenta</b>		
<b>Czynność</b>	<b>Czas (godz.)</b>	
1. wykłady	30	
2. ćwiczenia laboratoryjne	30	
3. egzamin i konsultacje	10	
4. przygotowanie do egzaminu	10	
5. prace przygotowawcze do laboratorium	45	
<b>Obciążenie pracą studenta</b>		
<b>forma aktywności</b>	<b>godzin</b>	<b>ECTS</b>
Łączny nakład pracy	125	5
Zajęcia wymagające bezpośredniego kontaktu z nauczycielem	50	2
Zajęcia o charakterze praktycznym	75	3