

SYLABUS ZAJ /GRUPY ZAJ

Dane ogólne:

Jednostka organizacyjna:	Katedra Informatyki				
Kierunek studiów:	Informatyka				
Specjalno /Specjalizacja:					
Nazwa zaj / grupy zaj :	Narzędzia i środowiska programistyczne II				
Course / group of courses:	Tools and Programming Environments II				
Forma studiów:	stacjonarne				
Nazwa katalogu:	WP-IN-I-20/21Z				
Nazwa bloku zaj :					
Kod zaj /grupy zaj :	105989	Kod Erasmus:			
Punkty ECTS:	1	Rodzaj zaj :		obowiązkowy	
Rok studiów:	2	Semestr:		4	
Rok	Semestr	Forma zaj	Liczba godzin	Forma zaliczenia	ECTS
2	4	P	15	Zaliczenie z ocen	1
Razem			15		1
Koordynator:	magister Tomasz Gódek				
Prowadzący zajęcia:					
Język wykładowy:	semestr: 4 - język polski				

Objaśnienia:

Rodzaj zaj : obowiązkowe, do wyboru.

Forma prowadzenia zaj : W - wykład, - wyćzenia audytoryjne, L - lektorat, S – seminarium/ zajęcia seminaryjne, P - wyćzenia praktyczne (w tym zajęcia wf), M - wyćzenia specjalistyczne (medyczne/ kliniczne), LO – wyćzenia laboratoryjne, LI - laboratorium informatyczne, ZTI - zajęcia z technologii informacyjnych, P – wyćzenia projektowe, ZT – zajęcia terenowe, T - wyćzenia terenowe na obozach programowych, SK - samokształcenie (i inne), PR - praktyka zawodowa

Dane merytoryczne

Wymagania wstępne:			
Znajomość języków i technik programowania, algorytmów i struktur danych przewidzianych programem studiów. Znajomość języka angielskiego w stopniu umożliwiającym studiowanie literatury fachowej.			
Szczegółowe efekty uczenia się			
Lp.	Student, który zaliczył zajęcia zna i rozumie/potrafi/jest gotowy do:	Kod efektu dla kierunku studiów	Sposób weryfikacji efektu uczenia się
1	Umie projektować i implementować podstawowe funkcje prostej aplikacji z wykorzystaniem różnych technik i języków programowania oraz różnych narzędzi i środowisk wytwarzania oprogramowania.	IN1_U01, IN1_U05	wykonanie zadania
2	Umie posługiwać się wybranymi narzędziami i środowiskami wspierającymi programowanie.	IN1_U05	wykonanie zadania
3	Rozumie trendy rozwoju nowoczesnych technologii, występujących we współczesnej praktyce programowania.	IN1_U11, IN1_U13	wykonanie zadania
4	Umie posługiwać się wybranymi narzędziami wersjonowania oprogramowania oraz narzędziami zarządzania cyklem wytwarzania oprogramowania.	IN1_U13	wykonanie zadania

5	Rozumie trendy rozwoju nowoczesnych technologii, występujących we współczesnej praktyce programowania.	IN1_K01	wykonanie zadania
Stosowane metody osiągnięcia zakładanych efektów uczenia się (metody dydaktyczne)			
metody praktyczne (Podstawowe formy zajęć: ćwiczenia projektowe wymagające samodzielnej analizy zadania projektowego, wykonania projektu, implementacji, testowania i dokumentacji prostej aplikacji lub systemu informatycznego.)			
Kryteria oceny i weryfikacji efektów uczenia się			
umiejętności: ocena wykonania zadania (Projekt)			
kompetencje społeczne: ocena wykonania zadania (Projekt)			
Warunki zaliczenia			
1. Samodzielne przygotowanie projektu oraz implementacji prostej aplikacji lub systemu informatycznego. 2. Przygotowanie i ocena dokumentacji projektowej według podanych założeń.			
Treści programowe (opis skrócony)			
1. Narzędzia i środowiska wytwarzania oprogramowania. 2. Narzędzia wspomagające wytwarzanie aplikacji serwerowych 3. Wersjonowanie oprogramowania. 4. Cykle życia oprogramowania.			
Content of the study programme (short version)			
1. Tools and environments for software development. 2. Development tools for server applications. 3. Software versioning. 4. Application development life-cycle.			
Treści programowe			
			Liczba godzin
Semestr: 4			
Forma zajęć: ćwiczenia projektowe			
1. Narzędzia i środowiska wytwarzania oprogramowania: środowiska i narzędzia implementacyjne, zintegrowane środowiska programistyczne (IntelliJ IDEA), tworzenie projektów, mechanizmy edycyjne, zarządzanie kodem, kompilacja, uruchamianie i debugowanie projektów, refaktoryzacja kodu. 2. Narzędzia wspomagające wytwarzanie aplikacji serwerowych: Spring Framework (podstawowa konfiguracja i uruchomienie aplikacji), REST, POSTMAN (testowanie API), MAVEN, integracja MAVEN z zintegrowanym środowiskiem wytwarzania oprogramowania IDE, wstrzykiwanie zależności, protokół HTTP, idea aplikacji klient-server. 3. Wersjonowanie oprogramowania: Podstawowe pojęcia i koncepcje wersjonowania oprogramowania, dobre praktyki w wersjonowaniu oprogramowania, systemy kontroli wersji, narzędzia wersjonowania, instalacja i konfiguracja systemu GIT, podstawowe polecenia systemu GIT, zaawansowane polecenia systemu GIT (rozgałęzianie, scalanie), wykorzystanie GIT w środowisku serwerowym i rozproszonym, integracja systemu GIT z zintegrowanym środowiskiem wytwarzania oprogramowania IDE, gitflow (przepływ pracy oraz rodzaje branchy). 4. Cykle życia oprogramowania: Narzędzia zarządzania cyklem wytwarzania aplikacji (MAVEN).			15
Literatura			
Podstawowa			
B. Eckel, Thinking in Java. Wyd. IV, Helion 2004			
C. Walls, Spring w akcji. Wyd. IV, Helion 2015			
W. Gajda, Git. Rozproszony system kontroli wersji, Helion 2013			
Uzupełniająca			

Przyporzkowanie zaj /grup zaj do dyscypliny naukowej/artystycznej		informatyka techniczna i telekomunikacja	
Sposób okre lenia liczby punktów ECTS			
Forma nakładu pracy studenta (udział w zaj ciach, aktywno , przygotowanie sprawozdania, itp.)		Obci enia studenta [w godz.]	
Udział w zaj ciach		15	
Konsultacje z prowadz cym		0	
Udział w egzaminie		0	
Bezpo redni kontakt z nauczycielem - inne		3	
Przygotowanie do laboratorium, wicze , zaj		7	
Przygotowanie do kolokwii i egzaminu		0	
Indywidualna praca własna studenta z literatur , wykładami itp.		0	
Inne		0	
Sumaryczne obci enie prac studenta		25	
Liczba punktów ECTS			
Liczba punktów ECTS		1	
Zaj cia wymagaj ce bezpo redniego udziału nauczyciela akademickiego		L. godzin	ECTS
		18	0,7
Zaj cia o charakterze praktycznym		L. godzin	ECTS
		25	1,0

1 godz = 45 minut; 1 punkt ECTS = 25-30 godzin

W sekcji 'Liczba punktów ECTS' suma punktów ECTS zaj wymagaj cych bezpo redniego udziału nauczyciela akademickiego i o charakterze praktycznym mo e si ró ni od ł cznej liczby punktów ECTS dla zaj /grup zaj .