

# SYLABUS ZAJ /GRUPY ZAJ

## Dane ogólne:

Jednostka organizacyjna:	Katedra Informatyki				
Kierunek studiów:	Informatyka				
Specjalno /Specjalizacja:					
Nazwa zaj / grupy zaj :	Programowanie reaktywne II				
Course / group of courses:	Reactive Programming II				
Forma studiów:	stacjonarne				
Nazwa katalogu:	WP-IN-I-20/21Z				
Nazwa bloku zaj :					
Kod zaj /grupy zaj :	106002	Kod Erasmus:			
Punkty ECTS:	1	Rodzaj zaj :		fakultatywny	
Rok studiów:	3	Semestr:		5	
Rok	Semestr	Forma zaj	Liczba godzin	Forma zaliczenia	ECTS
3	5	P	15	Zaliczenie z ocen	1
Razem			15		1
Koordinator:	magister in ynier Dariusz Piwko				
Prowadz cy zaj cia:					
J zyk wykładowy:	semestr: 5 - ---				

## Obja nienia:

Rodzaj zaj : obowi zkowe, do wyboru.

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

## Dane merytoryczne

Wymagania wst pne:			
Wiadomo ci z zakresu przedmiotu "Programowanie reaktywne"			
Szczegółowe efekty uczenia si			
Lp.	Student, który zaliczył zaj cia zna i rozumie/potrafi/jest gotowy do:	Kod efektu dla kierunku studiów	Sposób weryfikacji efektu uczenia si
1	Student posiada wiedz i umiej tno ci do stworzenia aplikacji internetowej opartej o poznane wzorcu.	IN1_U02	praca pisemna
2	Student posiada wiedz i umiej tno ci do u ycia poznanych j zyków programowania oraz bibliotek	IN1_U05	praca pisemna
3	Student posiada wiedz i umiej tnosci do korzystania z dokumentacji technicznej bibliotek oraz pakietów. Jest w stanie stworzy dokumentacj techniczn swojego projektu.	IN1_U11, IN1_U05, IN1_U07, IN1_U12	praca pisemna
4	Student ma wiadomo , jak współpracowa w projekcie zespół. Potrafi planowa i koordynowa prac zespół.	IN1_U13	praca pisemna
5	Student ma wiadomo , jak współpracowa w projekcie zespół. Potrafi planowa i koordynowa prac zespół.	IN1_K01, IN1_K05	praca pisemna, obserwacja zachowa

Stosowane metody osiągnięcia zakładanych efektów uczenia się (metody dydaktyczne)	
<p>metody praktyczne (Forma zajęć : Klasy mają charakter projektu. Głównym celem jest stworzenie aplikacji internetowej przy użyciu określonych narzędzi. Metody dydaktyczne: Postęp wniosku jest weryfikowany podczas konsultacji. Projekt jest podzielony na kilka etapów w celu lepszej motywacji studentów.)</p>	
Kryteria oceny i weryfikacji efektów uczenia się	
<p><b>umiejętności:</b></p> <p>ocena pracy pisemnej (ocena zadania projektowego)</p> <p><b>kompetencje społeczne:</b></p> <p>obserwacja zachowań (obserwacja zachowań indywidualnych i zespołowych pod kątem kompetencji społecznych)</p> <p>ocena pracy pisemnej (ocena zadania projektowego)</p>	
Warunki zaliczenia	
<p>Każdy student musi przygotować aplikację internetową razem z dokumentacją techniczną. Każdego projektu musi być wykonany niezależnie. Projekt musi zostać utworzony przy użyciu poznanej technologii podczas zajęć.</p>	
Treści programowe (opis skrócony)	
<ol style="list-style-type: none"> <li>Omówienie cyklu życia projektu.</li> <li>Wybranie i omówienie zakresu projektu.</li> <li>Określenie wymagań funkcjonalnych i нефункциональных.</li> <li>Zdefiniowanie architektury systemu.</li> <li>Opracowanie podstawowych diagramów opisujących system.</li> <li>Implementacja systemu w oparciu o przygotowanie modułów.</li> <li>Implementacja testów systemu.</li> <li>Opracowanie dokumentacji technicznej.</li> <li>Poznanie słownictwa technicznego przy opracowywaniu projektu.</li> </ol>	
Content of the study programme (short version)	
<ol style="list-style-type: none"> <li>Discuss project life cycle.</li> <li>Choose and discuss project scope.</li> <li>Specify functional and non-functional requirements.</li> <li>Define system architecture.</li> <li>Develop basic diagrams.</li> <li>Create system.</li> <li>Develop tests for the system.</li> <li>Create technical documentation.</li> <li>Improve technical vocabulary.</li> </ol>	
Treści programowe	
	Liczba godzin
Semestr: 5	
Forma zajęć : <b>wiczenia projektowe</b>	
<p>Zajęcia projektowe składają się z ogólnych zasad tworzenia oprogramowania w IT:</p> <ol style="list-style-type: none"> <li>Omówienie cyklu życia projektu. Prezentacja wszystkich faz i etapów projektowania IT. Omówienie planowania projektu, w tym ustalenie i udokumentowanie listy konkretnych celów projektu.</li> <li>Wybranie i omówienie zakresu projektu, dokumentowanie listy konkretnych zadań projektu. Użycie frameworków / bibliotek poznanych podczas zajęć.</li> <li>Określenie wymagań funkcjonalnych i нефункциональных systemu. Przygotowanie listy funkcji, które rozwiążą określone problemy dla użytkowników.</li> <li>Zdefiniowanie architektury systemu wysokiego poziomu z podziałem na aplikacje klienckie i serwerowe z bazami danych.</li> <li>Przygotowanie diagramów architektury pokazujące integrację między frontendem i backendem z wewnętrzną strukturą techniczną. Diagramy UML, które opisują przypadki użycia, diagram ERD opisujący bazy danych oraz opis Interfejsu aplikacji (API).</li> <li>Wdrożenie systemu przy użyciu określonych frameworków lub bibliotek (np. ReactJS, Java). System składa się z części klienta i aplikacji serwerowej, które współpracują z bazami danych (PostgreSQL, MySQL, MongoDB itp.).</li> <li>Określenie przepływu aplikacji testującej. Głównymi celami są zintegrowane testy, które eliminują wiele wspólnych błędów.</li> </ol>	15

8. Przygotowanie dokumentacji technicznej opisującej każdą fazę tworzonej aplikacji. 9. Poznanie słownictwa technicznego przy opracowywaniu projektu.	15
<b>Literatura</b>	
Podstawowa	
Craig Walls, Spring in Action, Fifth Edition, Manning Publications 2018	
Kirupa Chinnathambi, React i Redux. Praktyczne tworzenie aplikacji WWW, Addison-Wesley Professional 2018	
Michał Miszczyszyn, TypeScript na poważnie, Type of Web, Gdańsk 2020	
Mike Cantelon, Marc Harter, TJ Holowaychuk, Nathan Rajlich, Node.js in Action, Manning Publications 2014	
Dokumentacja techniczna dostępna w sieci Internet	
Uzupełniająca	

#### Dane jakościowe

Przyporządkowanie zajęć /grup zajęć do dyscypliny naukowej/artystycznej		informatyka techniczna i telekomunikacja	
Sposób określenia liczby punktów ECTS			
Forma nakładu pracy studenta (udział w zajęciach, aktywność, przygotowanie sprawozdania, itp.)		Obciążenie studenta [w godz.]	
Udział w zajęciach		15	
Konsultacje z prowadzącym		0	
Udział w egzaminie		0	
Bezpośredni kontakt z nauczycielem - inne		0	
Przygotowanie do laboratorium, ćwiczeń, zajęć		10	
Przygotowanie do kolokwium i egzaminu		0	
Indywidualna praca własna studenta z literatury, wykładami itp.		0	
Inne		0	
Sumaryczne obciążenie prac studenta		25	
Liczba punktów ECTS			
Liczba punktów ECTS		1	
Zajęcia wymagające bezpośredniego udziału nauczyciela akademickiego		L. godzin	ECTS
		15	0,6
Zajęcia 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 zajęć wymagających bezpośredniego udziału nauczyciela akademickiego i o charakterze praktycznym może się różnić od łącznej liczby punktów ECTS dla zajęć/grup zajęć.