

SYLABUS ZAJ /GRUPY ZAJ

Dane ogólne:

Jednostka organizacyjna:	Katedra Informatyki				
Kierunek studiów:	Informatyka				
Specjalno /Specjalizacja:	Inżynieria systemów inteligentnych				
Nazwa zaj / grupy zaj :	Zaawansowane programowanie systemów mobilnych				
Course / group of courses:	Advanced Mobile Programming				
Forma studiów:	stacjonarne				
Nazwa katalogu:	WP-IN-I-20/21Z-IS				
Nazwa bloku zaj :					
Kod zaj /grupy zaj :	105884	Kod Erasmus:			
Punkty ECTS:	3	Rodzaj zaj :		obowi zkowy	
Rok studiów:	3	Semestr:		5	
Rok	Semestr	Forma zaj	Liczba godzin	Forma zaliczenia	ECTS
3	5	LO	30	Zaliczenie z ocen	2
		W	15	Zaliczenie z ocen	1
Razem			45		3
Koordinator:	magister in ynier Tomasz Gryl				
Prowadz cy zaj cia:					
J zyk wykładowy:	semestr: 5 - j zyk polski				

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:			
Kurs poprzedzaj cy Programowanie w Javie, Programowanie w Javie II, Programowanie systemów mobilnych, Programowanie Systemów Mobilnych II, Narz dzia i rodowiska programistyczne, Narz dzia i rodowiska programistyczne II.			
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	Posiada wiedz w zakresie architektur systemów komputerowych, w tym systemów wbudowanych, zarówno w zakresie warstwy sprz towej jak i programowej; rozumie cykl ycia systemów informatycznych oraz urz dze wykorzystywanych w informatyce.	IN1_W02	kolokwium, ocena aktywno ci, obserwacja zachowa
2	Dysponuje wiedz w zakresie metodyk wytwarzania oprogramowania oraz doboru modelu procesu wytwarzania do specyfikacji przedsi wzi cia.	IN1_W04	ocena aktywno ci, obserwacja zachowa
3	Ma zaawansowan wiedz dotycz c systemów operacyjnych, niezbdn do instalacji, obsługi, utrzymania oraz ich modyfikowania dla potrzeb systemów scentralizowanych oraz rozproszonych.	IN1_W05	ocena aktywno ci
4	Posiada zaawansowan wiedz w zakresie j zyków, metod, algorytmów oraz paradygmatów programowania, ma wiedz w zakresie modelowania, analizowania oraz przetwarzania danych.	IN1_W07	kolokwium, ocena aktywno ci, obserwacja zachowa

5	Zna i rozumie cykl życia oprogramowania oraz etapy wytwarzania w zakresie projektowania, implementacji, testowania oraz wdrożenia, ma wiedzę w zakresie tworzenia oprogramowania dla różnych zastosowań informatyki.	IN1_W08	ocena aktywności, obserwacja zachowa
6	Pozyskuje informacje z literatury, baz danych i innych źródeł; potrafi integrować uzyskane informacje, dokonywać ich krytycznej interpretacji, a także wyciąga wnioski oraz formułować i uzasadniać opinie, korzysta ze standardów i norm inżynierskich.	IN1_W08	kolokwium, ocena aktywności, obserwacja zachowa
7	Konstruuje interfejs komunikacji człowiek-maszyna, także z wykorzystaniem narzędzi wspomagających tworzenie graficznych interfejsów użytkownika.	IN1_U02, IN1_U01	kolokwium, ocena aktywności, obserwacja zachowa
8	Posiada umiejętności językowe w zakresie dziedzin nauki i dyscyplin naukowych, właściwych dla studiowanego kierunku studiów, zgodne z wymaganiami określonymi dla poziomu B2 Europejskiego Systemu Opisu Kształcenia Językowego, posługuje się językiem angielskim w stopniu wystarczającym do czytania ze zrozumieniem dokumentacji technicznej, instrukcji obsługi urządzeń i narzędzi informatycznych oraz podobnych dokumentów.	IN1_U12	ocena aktywności, obserwacja zachowa
9	Przestrzega zasad etyki zawodowej, jest świadomy własności zachowania w sposób profesjonalny.	IN1_K05	kolokwium, ocena aktywności, obserwacja zachowa

Stosowane metody osiągnięcia zakładanych efektów uczenia się (metody dydaktyczne)

metody praktyczne (ćwiczenia laboratoryjne prowadzone na podstawie instrukcji (udostępnianych w sposób elektroniczny) z przykładami i zadaniami do samodzielnej realizacji, które podlegają ocenie.), metody podające (Przedstawione treści na wykładzie w oparciu o prezentację multimedialną oraz analizę oraz omawianie praktycznych przykładów.)

Kryteria oceny i weryfikacji efektów uczenia się

wiedza:

- ocena kolokwium (Kolokwium)
- obserwacja zachowa (Obserwacja)
- ocena aktywności (Aktywność na zajęciach)

umiejętności:

- ocena kolokwium (Kolokwium)
- obserwacja zachowa (Obserwacja)
- ocena aktywności (Aktywność na zajęciach)

kompetencje społeczne:

- ocena kolokwium (Kolokwium)
- obserwacja zachowa (Obserwacja)
- ocena aktywności (Aktywność na zajęciach)

Warunki zaliczenia

Wykład:
Zaliczenie na podstawie obecności.
Laboratorium:
Zaliczenie z oceną wystawioną na podstawie pozytywnego zaliczenia kolokwium z zadania, aktywności na zajęciach (w rozwijaniu zadań i problemów). Ocena końcowa jest średnią ważoną ocen częściowych uzyskanych z w/w zadań.
Zaliczanie zajęć jest oceniane zgodnie ze skalą ocen określonych w Regulaminie Studiów PWSZ.

Treści programowe (opis skrócony)

Blok ten kładzie nacisk na zdobywanie umiejętności praktycznych związanych z wytwarzaniem oprogramowania na systemy mobilne. Zwrócona jest szczególna uwaga na zastosowanie wcześniej zdobytej wiedzy z zakresu programowania obiektowego oraz programowania systemów mobilnych.

Content of the study programme (short version)

This block puts emphasis on acquiring practical skills related to software development for mobile systems. Particular attention is paid to the application of previously acquired knowledge in the field of object-oriented programming and programming of mobile systems.

Treści programowe

	Liczba godzin
Semestr: 5	
Forma zajęć: wykład	
Uzyskanie umiejętności programowania systemów mobilnych (telefon komórkowy, tablet, telewizor) na przykładzie urządzeń wyposażonych w wybrany system operacyjny.	
1. Przegląd i porównanie aktualnie najpopularniejszych mobilnych systemów operacyjnych. Podstawowe kryteria programowania urządzeń mobilnych, bezpieczeństwo i dystrybucja aplikacji.	15
2. Architektura mobilnych systemów operacyjnych Android oraz iOS.	

3. Wykorzystanie React Native do budowy zaawansowanych systemów mobilnych.	15
4. Kierunki rozwoju mobilnych systemów operacyjnych oraz ich znaczenie we współczesnym społeczeństwie. Elementy interfejsu graficznego – widżety, grafika 3D.	
5. Wykorzystanie protokołu HTTP w aplikacjach mobilnych do przetwarzania danych.	
6. Dystrybucja aplikacji – Google Play oraz App Store.	
7. Wykorzystanie mechanizmu Redux do zarządzania stanem aplikacji.	

Forma zajęć : **wiczenia laboratoryjne**

1. Uzyskanie umiejętności zaawansowanego programowania systemów mobilnych (telefon komórkowy, tablet, telewizor), przy pomocy urządzeń wyposażonych w system operacyjny Android oraz iOS w oparciu o React Native.	30
2. Przegląd i porównanie aktualnie najpopularniejszych mobilnych systemów operacyjnych. Zaawansowane kryteria programowania urządzeń mobilnych, bezpieczeństwa i dystrybucji aplikacji.	
3. Multimedia w systemie Android – dźwięk, sekwencje wideo.	
4. Współpraca z siecią Internet. Modele aplikacji klient-serwer. Elementy bezpieczeństwa aplikacji sieciowych.	
5. Współpraca z czujnikami specyficznymi dla mobilnych systemów operacyjnych takimi jak: akcelerometr, cyfrowy kompas, żyroskop itp.	
6. Zarządzanie stanem aplikacji przy pomocy mechanizmu Redux.	
7. Wykorzystywanie wybranych bibliotek trzecich producentów do konstruowania zaawansowanych systemów mobilnych.	
8. Współpraca z wybranymi protokołami komunikacji (Bluetooth, NFC, itp.).	

Literatura

Podstawowa

A. D. Scott, Wszechstronny JavaScript. Technologie: GraphQL, React, React Native i Electron, Helion 2020

B. Eisenman, React Native. Tworzenie aplikacji mobilnych w języku JavaScript., Helion 2018

D. Griffiths, D. Griffiths, Android. Programowanie aplikacji. Rusz głowę!, Helion 2018

D. Jemerov, S. Isakova, Kotlin w akcji, Helion 2018

E. Atanasov, Poznaj Swifta, tworząc aplikacje. Profesjonalne projekty dla systemu iOS, Helion 2019

M. Neuburg, iOS 12. Wprowadzenie do programowania w Swiftcie., Helion 2019

Uzupełniająca

Dane jakościowe

Przyporządkowanie zajęć/grup zajęć do dyscypliny naukowej/artystycznej	informatyka techniczna i telekomunikacja
Sposób określania liczby punktów ECTS	
Forma nakładu pracy studenta (udział w zajęciach, aktywność, przygotowanie sprawozdania, itp.)	Obciążenie studenta [w godz.]
Udział w zajęciach	45
Konsultacje z prowadzącym	0
Udział w egzaminie	0
Bezpośredni kontakt z nauczycielem - inne	0
Przygotowanie do laboratorium, wicze, zajęcia	10
Przygotowanie do kolokwium i egzaminu	10
Indywidualna praca własna studenta z literatury, wykładami itp.	10

Inne	0	
Sumaryczne obciążenie prac studenta	75	
Liczba punktów ECTS		
Liczba punktów ECTS	3	
Zajęcia wymagające bezpośredniego udziału nauczyciela akademickiego	L. godzin	ECTS
	45	1,8
Zajęcia o charakterze praktycznym	L. godzin	ECTS
	45	1,8

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.