

Politechnika Białostocka
Wydział Informatyki

Program studiów podyplomowych

JavaScript Developer

Białystok, kwiecień 2019 r.

PROGRAM STUDIÓW PODYPLOMOWYCH

JAVASCRIPT DEVELOPER

Studia podyplomowe **JavaScript Developer** trwają 2 semestry i umożliwiają uzyskanie kwalifikacji częściowych na poziomie **6 PRK**. Łączna liczba punktów ECTS: **30**. Łączna liczba godzin zajęć: **257**.

Plan studiów JAVASCRIPT DEVELOPER

Lp.	Nazwa przedmiotu	Kod	Liczba ECTS			Liczba godzin w semestrze			Forma zaliczenia
			C	K	P	W	Ć	PS	
SEMESTR I									
1	Programowanie w języku JavaScript	SPJSD01	3	1,24	2,68	8		18	zaliczenie na ocenę
2	Angular od podstaw - Programowanie aplikacji Web	SPJSD02	3	1,24	2,68	8		18	zaliczenie na ocenę
3	Framework i ekosystem React - Wprowadzenie	SPJSD03	3	1,24	2,68	8		18	zaliczenie na ocenę
4	NodeJS - środowisko i technologia ServerSide 1	SPJSD04	3	1,24	2,68	8		18	zaliczenie na ocenę
5	Podstawy Agile i Scrum	SPJSD05	2	0,6	1,75	6		6	zaliczenie na ocenę
6	Metodyka DevOps - produkcja i utrzymanie oprogramowania	SPJSD06	1	0,6	0,88	3		9	zaliczenie na ocenę
RAZEM W SEMESTRZE			15	6,16	13,35	41		87	Razem godz. 128
SEMESTR II									
1	Zaawansowane zagadnienia języka JavaScript	SPJSD07	3	1,16	2,76	6		18	zaliczenie na ocenę
2	Zawansowane programowanie aplikacji w Angular	SPJSD08	3	1,16	2,76	6		18	zaliczenie na ocenę
3	Framework i ekosystem React - Aplikacja serwerowa	SPJSD09	3	1,16	2,76	6		18	zaliczenie na ocenę
4	NodeJS - środowisko i technologia ServerSide 2	SPJSD10	3	1,16	2,76	6		18	zaliczenie na ocenę
5	Wytwarzanie oprogramowania z wykorzystaniem Agile i Scrum	SPJSD11	1	0,48	0,76	6		6	zaliczenie na ocenę
6	Aplikacje mobilne w technologii React Native	SPJSD12	2	1,04	1,76	6		15	zaliczenie na ocenę
RAZEM W SEMESTRZE			15	6,16	13,56	36		93	Razem godz. 129
	RAZEM KIERUNEK [w podziale na FORMY ZAJĘCIOWE]								
ŁĄCZNIE ECTS/GODZ. W TRAKCIE STUDIÓW			30	12,32	26,91	77		180	Razem godz. 257

Objaśnienia do punktów ECTS: C – Całkowita wartość punktowa, K – Punkty kontaktowe, P – Punkty praktyczne

Sylwetka absolwenta

Studia poświęcone są zagadnieniom dotyczącym wytwarzania nowoczesnych aplikacji internetowych i systemów z wykorzystaniem języka JavaScript i technologii powiązanych. Niniejsze studia podyplomowe skierowane są w głównej mierze do osób, które nie miały styczności z programowaniem. Autorzy planu studiów zakładają, że niezbędną wiedzę programistyczną słuchacz zdobędzie w trakcie nauki na obu semestrach studiów. Przed rozpoczęciem studiów wymagane będzie jednak ukończenie kilku lekcji z kursu "JavaScript | Codecademy". Pozytywne zaliczenie tych lekcji będzie podstawą do podjęcia decyzji przez samego słuchacza i definiuje wymagania wstępne dla kierunku studiów. Wymagany zakres to: zmienne, ciągi znaków, konsola, instrukcje warunkowe i funkcje.

Absolwent studiów podyplomowych będzie przygotowany do podjęcia pracy na stanowiskach określanych na rynku pracy jako JavaScript Junior Developer w firmach związanych z wytwarzaniem aplikacji WWW. Będzie posiadał wiedzę i umiejętności z zakresu programowania języka JavaScript, budowy stron internetowych, standardów i technologii WWW oraz pracy w zespole developerskim. Słuchacz zdobywa wiedzę w oparciu o praktyczne przykłady oraz doświadczenie kadry dydaktycznej. Po zakończeniu studiów słuchacz będzie posiadał m.in. wiedzę i umiejętności:

- dotyczące programowania w języku JavaScript, poczynając od podstaw a kończąc na programowaniu obiektowym i zagadnieniach zaawansowanych;
- na temat metodyk pracy przy projektach - Agile i Scrum i kultury pracy deweloperów i tworzenia kodu oraz całego procesu;
- projektowania aplikacji JavaScript z wykorzystaniem wytycznych organizacja kodu, wydajności oraz wzorców projektowych;
- o bieżących standardach kodowania w języku HTML5/CSS3 oraz technikach wytwarzania stron WWW w kontekście aplikacji JavaScript;
- na temat programowania w języku JavaScript wykorzystując do tego popularne frameworki m.in. AngularJS, React;
- projektowania aplikacji ServerSide, programowania i zarządzania systemami z wykorzystaniem środowiska i technologii node.js
- projektowania i programowania aplikacji mobilnych z wykorzystaniem frameworka wieloplatformowego React Native;
- o testowaniu i walidacji budowanych rozwiązań;
- w posługiwaniu się narzędziami wykorzystywanymi we współczesnej pracy JavaScript Developera oraz praktykami Dev-Ops w wytwarzaniu i utrzymaniu systemów.

Opis kompetencji oczekiwanych od kandydata ubiegającego się o przyjęcie na studia podyplomowe

Uczestnikiem studiów podyplomowych może być osoba, która posiada kwalifikację pełną co najmniej na poziomie 6 uzyskaną w systemie szkolnictwa wyższego i nauki.

Kandydaci ubiegający się o przyjęcie na studia podyplomowe powinni mieć podstawową wiedzę i umiejętności z zakresu obsługi komputera i urządzeń peryferyjnych oraz znajomość podstawowych zagadnień związanych z technologiami informacyjnymi, w szczególności związane z obsługą systemów operacyjnych, instalacją oprogramowania, posługiwaniem się konsolą wiersza poleceń MS Windows lub powłoką systemu Linux (shell). Kandydaci nie muszą posiadać wykształcenia technicznego. Niezbędną wiedzę programistyczną dotyczącą języka JavaScript, słuchacz zdobędzie od początku, w trakcie nauki tego przedmiotu na obu semestrach niniejszych studiów. Należy jednak zaznaczyć, że jest to kurs intensywny. Dla osób, które nie miały dotychczas do czynienia z podstawami programowania, udział w zajęciach będzie wymagał dodatkowych nakładów pracy.

Przed rozpoczęciem studiów wymagane jest ukończenie kilku lekcji z kursu JavaScript, np. w serwisach e-learningowych Codecademy, Udemy Pluralsight lub podobnych. Pozytywne zaliczenie tych lekcji będzie podstawą do podjęcia decyzji przez samego słuchacza i definiuje wymagania wstępne dla kierunku studiów. Wymagany zakres to zrozumienie pojęć: zmienne, ciągi znaków, konsola, instrukcje warunkowe i funkcje.

Zestawienie efektów uczenia się

Zestawienie tabelaryczne kierunkowych efektów uczenia się odnoszących się do charakterystyk drugiego stopnia określonych na podstawie ustawy z dnia 22 grudnia 2015 r. o zintegrowanym systemie kwalifikacji na poziomie 6 PRK

Objaśnienia oznaczeń:

P6 – poziom 6 PRK (Polskie Ramy Kwalifikacji)

S – charakterystyka typowa dla kwalifikacji uzyskiwanych w ramach szkolnictwa wyższego

W – wiedza

T – teorie, zasady

Z – zjawiska i procesy

O – organizacja pracy

G – głębia i zakres

K – kontekst

U – umiejętności

I – informacje

W – wykorzystanie wiedzy

K – komunikowanie się

O – organizacja pracy

U – uczenie się

K – kompetencje społeczne

K – krytyczna ocena

O – odpowiedzialność

R – rola zawodowa

JSD – JavaScript Developer

1, 2, 3 i kolejne – numery efektu uczenia się

Symbol	Efekty Uczenia się dla kierunku studiów podyplomowych	Odniesienie do charakterystyk drugiego stopnia określonych na podstawie art.7 ust. 3 Ustawy z dn. 22 grudnia 2015 r. o Zintegrowanym Systemie Kwalifikacji na poziomie 6 PRK	Odniesienie do charakterystyk drugiego stopnia określonych na podstawie art.7 ust. 4 Ustawy z dn. 22 grudnia 2015 r. o Zintegrowanym Systemie Kwalifikacji na poziomie 6 PRK
Wiedza: absolwent zna i rozumie			
JSD_W01	pojęcia i standardy związane z językiem JavaScript, standardem ES6, wzorcami projektowymi i wybranymi frameworkami JavaScript oraz ich zastosowaniem.	P6S_WG, P6S_WK	P6Z_WT, P6Z_WZ, P6Z_WO
JSD_W02	wybrane konstrukcje, funkcje i zastosowanie języka HTML i w szerokim zakresie języka JavaScript.	P6S_WG, P6S_WK	P6Z_WT, P6Z_WZ, P6Z_WO
JSD_W03	wybrane współczesne frameworki JavaScript tworzenia wysoce skalowalnych aplikacji internetowych w języku JavaScript w oparciu o paradygmat SOA i microservices.	P6S_WG, P6S_WK	P6Z_WT, P6Z_WZ, P6Z_WO
JSD_W04	zasady zarządzania projektem, frameworkiem Scrum, manifestem Agile oraz narzędziami do szacowania prac developerskich.	P6S_WG, P6S_WK	P6Z_WT, P6Z_WZ, P6Z_WO
JSD_W05	zasady wykorzystywania narzędzi we współczesnej pracy programisty oraz praktyki Dev-Ops w wytwarzaniu i utrzymaniu systemów.	P6S_WG, P6S_WK	P6Z_WT, P6Z_WZ, P6Z_WO
Umiejętności: absolwent potrafi			
JSD_U01	posługiwać się językiem JavaScript zgodnie ze standardem ES6, wzorcami projektowymi i wybranymi frameworkami.	P6S_UW	P6Z_UI, P6Z_UN, P6Z_UO
JSD_U02	wykorzystać proste konstrukcje HTML oraz język JavaScript w tworzeniu dynamicznych elementów stron internetowych.	P6S_UW	P6Z_UI, P6Z_UN, P6Z_UO
JSD_U03	budować skalowalne aplikacje internetowe z wykorzystaniem frameworków JavaScript w oparciu o paradygmat SOA i microservices.	P6S_UW, P6S_UK, P6S_UU	P6Z_UI, P6Z_UN, P6Z_UO
JSD_U04	pracować zespołowo, zbudować środowisko pracy i wykorzystywać w pracy wybrane narzędzia procesu Agile i Scrum oraz przygotować dokumentację techniczną.	P6S_UW, P6S_UK, P6S_UO	P6Z_UI, P6Z_UN, P6Z_UO
JSD_U05	wykorzystać narzędzia i brać udział w procesie tworzenia oprogramowania wykorzystując narzędzia i środowisko Dev-Ops.	P6S_UW	P6Z_UI, P6Z_UN, P6Z_UO

Kompetencje społeczne: absolwent jest gotów do			
JSD_S01	rozwiązywania problemów w sposób poznawczy i praktyczny.	P6S_KK	P6Z_KP
JSD_S02	wejścia na rynek pracy jako Junior JavaScript Developer i odpowiedzialnego pełnienia roli zawodowej zgodnie z zasadami etyki oraz dbałością o dorobek zawodu	P6S_KR, P6S_KO	P6Z_KP, P6Z_KW

Matryca efektów uczenia się

Załącznik do Uchwały Senatu PB nr 328/XVIII/XV/2018

Załącznik nr 2 do Wytyczne do tworzenia programów studiów podyplomowych

Nazwa studiów podyplomowych: JavaScript Developer																	
MATRYCA POKRYCIA EFEKTÓW UCZENIA SIĘ																	
WIEDZA					UMIEJĘTNOŚCI					KOM. SPOŁ.							
Lp.	Nazwa przedmiotu	Kod przedmiotu	semestr	JSD_W01	JSD_W02	JSD_W03	JSD_W04	JSD_W05	Kod przedmiotu	JSD_U01	JSD_U02	JSD_U03	JSD_U04	JSD_U05	JSD_S01	JSD_S02	Kod przedmiotu
1	Programowanie w języku JavaScript	SPJSD01	pierwszy	+	+				SPJSD01	+	+						SPJSD01
2	Angular od podstaw - Programowanie aplikacji Web	SPJSD02			+				SPJSD02		+						SPJSD02
3	Framework i ekosystem React - Wprowadzenie	SPJSD03			+				SPJSD03		+						SPJSD03
4	NodeJS - środowisko i technologia ServerSide 1	SPJSD04		+	+			+	SPJSD04	+	+			+			SPJSD04
5	Podstawy Agile i Scrum	SPJSD05			+		+		SPJSD05		+		+				SPJSD05
6	Metodyka DevOps - produkcja i utrzymanie oprogramowania	SPJSD06				+		+	SPJSD06			+		+			SPJSD06
1	Zaawansowane zagadnienia języka JavaScript	SPJSD07	drugi	+	+				SPJSD07	+		+			+	+	SPJSD07
2	Zawansowane programowanie aplikacji w Angular	SPJSD08			+	+			SPJSD08		+	+					SPJSD08
3	Framework i ekosystem React - Aplikacja serwerowa	SPJSD09			+	+			SPJSD09		+	+					SPJSD09
4	NodeJS - środowisko i technologia ServerSide 2	SPJSD10			+	+		+	SPJSD10		+	+		+	+	+	SPJSD10
5	Wytwarzanie oprogramowania z wykorzystaniem Agile i Scrum	SPJSD11					+		SPJSD11				+		+	+	SPJSD11
6	Aplikacje mobilne w technologii React Native	SPJSD12			+	+			SPJSD12			+					SPJSD12
	SUMA			3	9	5	2	3		3	8	6	2	3	3	3	

Ramowe programy przedmiotów

Wydział Informatyki									
Kierunek studiów	JavaScript Developer						Poziom i forma studiów	Studia podyplomowe	
Specjalność / ścieżka dyplomowania							Profil kształcenia		
Nazwa przedmiotu	Programowanie w języku JavaScript						Kod przedmiotu	SPJSD01	
							Rodzaj przedmiotu	obowiązkowy	
Formy zajęć i liczba godzin	W	C	L	P	Ps	T	S	Semestr	1
	8				18			Punkty ECTS	3
Przedmioty wprowadzające									
Cele przedmiotu	Celem przedmiotu jest zapoznanie słuchacza z podstawami programowania JavaScript w standardzie ES6. Słuchacz nabędzie umiejętność efektywnego używania podstawowych elementów języka, planowania i tworzenia kodu JS odpowiadającego zadany wymaganiom oraz dobierania wzorców do zadanego problemu.								
Treści programowe	Wykład i pracownia specjalistyczna: 1. Składnia oraz zastosowanie języka JavaScript w standardzie ES6. 2. Nested Scope. 3. Modułowość. 4. Funkcja zwrotne. 5. Organizacja kodu, wydajność oraz wzorce projektowe.								
Metody dydaktyczne	1. prezentacja multimedialna; 2. analiza wybranych przypadków; 3. samodzielne opracowanie wskazanych zagadnień; 4. ćwiczenia indywidualne; 5. dyskusje dydaktyczne.								
Forma zaliczenia	Wykład: test przeprowadzony w systemie e-learningowym Pracownia specjalistyczna: zaliczenie projektem								
Symbol efektu uczenia się	Zakładane efekty uczenia się							Odniesienie do kierunkowych efektów uczenia się	
EU1	Zna podstawy programowania JavaScript standard ES6							JSD_W01, JSD_W02	
EU2	Potrafi efektywnie używać podstawowych elementów języka							JSD_W01, JSD_W02	
EU3	Potrafi zaplanować i stworzyć kod JS odpowiadający zadany wymaganiom							JSD_U01, JSD_U02	
EU4	Potrafi dobrać wzorzec do zadanego problemu							JSD_U02	

Symbol efektu uczenia się	Sposoby weryfikacji efektów uczenia się	Forma zajęć, na której zachodzi weryfikacja	
EU1	Test przeprowadzony w systemie e-learningowym	W	
EU2	Test przeprowadzony w systemie e-learningowym	W	
EU3	Zaliczenie projektem	Ps	
EU4	Zaliczenie projektem	Ps	
Bilans nakładu pracy słuchacza (w godzinach)		Liczba godz.	
Wyliczenie	Udział w wykładach	8	
	Udział w pracowni specjalistycznej	18	
	Przygotowanie do pracowni specjalistycznej	20	
	Opracowanie sprawozdań z pracowni	10	
	Udział w konsultacjach związanych z projektem	5	
	Realizacja zadań projektowych (zadań domowych)	14	
	RAZEM:	75	
Wskaźniki ilościowe		GODZINY	ECTS
Nakład pracy słuchacza związany z zajęciami wymagającymi bezpośredniego udziału nauczyciela		31	1,24
Nakład pracy słuchacza związany z zajęciami o charakterze praktycznym		67	2,68
Literatura podstawowa	1. Kirupa Chinnathambi. JavaScript. Przewodnik dla absolutnie początkujących, Helion, 2017 2. Nicolas Bevacqua. Nowoczesny JavaScript. Poznaj ES6 i praktyczne zastosowania nowych rozwiązań, Helion, 2018 3. Luis Atencio. Programowanie funkcyjne z JavaScriptem. Sposoby na lepszy kod, Helion, 2017 4. Simon Timms. JavaScript i wzorce projektowe. Programowanie dla zaawansowanych. Wydanie II, Helion, 2017		
Literatura uzupełniająca	1. JavaScript Tutorial, [20.06.2018] https://www.w3schools.com/JS/default.asp 2. Oficjalna strona dokumentacji Mozilli, [20.06.2018] https://developer.mozilla.org/pl/docs/Web/JavaScript 3. ES6 przykłady: [20.06.2018] http://es6-features.org/		
Jednostka realizująca	Wydział Informatyki Politechniki Białostockiej	Data opracowania programu	
Program opracował	dr inż. Paweł Tadejko, mgr inż. Karol Rogowski	22.02.2019	

Wydział Informatyki									
Kierunek studiów	JavaScript Developer						Poziom i forma studiów	Studia podyplomowe	
Specjalność / ścieżka dyplomowania							Profil kształcenia		
Nazwa przedmiotu	Angular od podstaw - Programowanie aplikacji Web						Kod przedmiotu	SPJSD02	
							Rodzaj przedmiotu	obowiązkowy	
Formy zajęć i liczba godzin	W	C	L	P	Ps	T	S	Semestr	1
	8				18			Punkty ECTS	3
Przedmioty wprowadzające									
Cele przedmiotu	Celem przedmiotu jest zapoznanie słuchacza z podstawami programowania w technologii Angular oraz z architekturą aplikacji opartych na komponentach. Słuchacz nabędzie umiejętność wymaganą do tworzenia nowoczesnych aplikacji Web.								
Treści programowe	Wykład i pracownia specjalistyczna: 1. Podstawy języka TypeScript 2. Wprowadzenie do technologii Angular 3. Narzędzia Angular CLI i analiza struktury projektu 4. Budowa komponentów 5. Wyświetlanie danych w komponencie i dostępne modele komunikacji między komponentami 6. Style w komponentach – zasady działania 7. Działanie serwisów logiki aplikacji 8. Budowanie routingu w aplikacji 9. Implementacja formularzy 10. Podział aplikacji na moduły oraz wdrażanie aplikacji na produkcję								
Metody dydaktyczne	1. prezentacja multimedialna; 2. analiza wybranych przypadków; 3. samodzielne opracowanie wskazanych zagadnień; 4. ćwiczenia indywidualne 5. dyskusje dydaktyczne								
Forma zaliczenia	Wykład: test przeprowadzony w systemie e-learningowym Pracownia specjalistyczna: zaliczenie projektem								
Symbol efektu uczenia się	Zakładane efekty uczenia się							Odniesienie do kierunkowych efektów uczenia się	
EU1	Zna podstawy programowania w języku TypeScript							JSD_W02	
EU2	Zna dostępne niezbędne rozwiązania w technologii Angular do stworzenia aplikacji							JSD_W02	
EU3	Potrafi tworzyć aplikacje w architekturze zorientowanej na komponenty							JSD_U02	
EU4	Potrafi wykorzystywać możliwości technologii Angular do tworzenia aplikacji Web							JSD_U02	

Symbol efektu uczenia się	Sposoby weryfikacji efektów uczenia się	Forma zajęć, na której zachodzi weryfikacja	
EU1	Test przeprowadzony w systemie e-learningowym	W	
EU2	Test przeprowadzony w systemie e-learningowym	W	
EU3	Zaliczenie projektem	Ps	
EU4	Zaliczenie projektem	Ps	
Bilans nakładu pracy słuchacza (w godzinach)		Liczba godz.	
Wyliczenie	Udział w wykładach	8	
	Udział w pracowni specjalistycznej	18	
	Przygotowanie do pracowni specjalistycznej	10	
	Opracowanie sprawozdań z pracowni	14	
	Udział w konsultacjach związanych z projektem	5	
	Realizacja zadań projektowych (zadań domowych)	20	
	RAZEM:	75	
Wskaźniki ilościowe		GODZINY	ECTS
Nakład pracy słuchacza związany z zajęciami wymagającymi bezpośredniego udziału nauczyciela		31	1,24
Nakład pracy słuchacza związany z zajęciami o charakterze praktycznym		67	2,68
Literatura podstawowa	1. Yakov Fain, Anton Moiseev, Angular 2. Programowanie z użyciem języka TypeScript, Helion 2017 2. Gion Kunz, Angular 2. Tworzenie interaktywnych aplikacji internetowych, Helion 2017 3. Official Angular Tutorial: https://angular.io/tutorial		
Literatura uzupełniająca	1. Angular 4 Tutorial: https://www.tutorialspoint.com/angular4/index.htm 2. Minko Gechev, Getting Started with Angular, Second Edition, Packt Publishing 2017 3. Angular: Getting Started: https://www.pluralsight.com/courses/angular-2-getting-started-update 4. Nate Murray, Felioe Coury, Ari Lerner, Carlos Taborda, ng-book, Fullstack.io 2017		
Jednostka realizująca	Wydział Informatyki Politechniki Białostockiej	Data opracowania programu	
Program opracował	<i>dr inż. Paweł Tadejko, Kamil Mijacz</i>	22.02.2019	

Wydział Informatyki									
Kierunek studiów	JavaScript Developer							Poziom i forma studiów	Studia podyplomowe
Specjalność / ścieżka dyplomowania								Profil kształcenia	
Nazwa przedmiotu	Framework i ekosystem React - Wprowadzenie							Kod przedmiotu	SPJSD03
								Rodzaj przedmiotu	obowiązkowy
Formy zajęć i liczba godzin	W	C	L	P	Ps	T	S	Semestr	1
	8				18			Punkty ECTS	3
Przedmioty wprowadzające									
Cele przedmiotu	Celem przedmiotu jest zapoznanie słuchacza z frameworkiem React.js. Słuchacz pozna czym jest React.js i dlaczego warto go używać. Słuchacz nabędzie umiejętność budowania dynamicznych aplikacji internetowych.								
Treści programowe	Wykład i pracownia specjalistyczna 1. Środowisko React.js 2. Komponenty - ich struktury, stany i komunikacja między nimi 3. Składnia JSX oraz narzędzia developerskie 4. Tworzenie prostej aplikacji internetowej z wykorzystaniem środowiska React.js								
Metody dydaktyczne	1. prezentacja multimedialna; 2. analiza wybranych przypadków; 3. samodzielne opracowanie wskazanych zagadnień; 4. ćwiczenia indywidualne 5. dyskusje dydaktyczne.								
Forma zaliczenia	Wykład: test przeprowadzony w systemie e-learningowym Pracownia specjalistyczna: zaliczenie projektem								
Symbol efektu uczenia się	Zakładane efekty uczenia się							Odniesienie do kierunkowych efektów uczenia się	
EU1	Zna podstawy oraz strukturę frameworka React.js							JSD_W01, JSD_W02	
EU2	Zna zasady tworzenia komponentów oraz realizacji komunikacji między nimi							JSD_W01, JSD_W02	
EU3	Potrafi stworzyć prostą aplikację							JSD_U02	
EU4	Potrafi używać narzędzi developerskich							JSD_U02	
Symbol efektu uczenia się	Sposoby weryfikacji efektów uczenia się							Forma zajęć, na której zachodzi weryfikacja	
EU1	Test przeprowadzony w systemie e-learningowym							W	
EU2	Test przeprowadzony w systemie e-learningowym							W	
EU3	Zaliczenie projektem							Ps	
EU4	Zaliczenie projektem							Ps	

Bilans nakładu pracy słuchacza (w godzinach)		Liczba godz.	
Wyliczenie	Udział w wykładach	8	
	Udział w pracowni specjalistycznej	18	
	Przygotowanie do pracowni specjalistycznej	14	
	Opracowanie sprawozdań z pracowni	10	
	Udział w konsultacjach związanych z projektem	5	
	Realizacja zadań projektowych (zadań domowych)	20	
	RAZEM:	75	
Wskaźniki ilościowe		GODZINY	ECTS
Nakład pracy słuchacza związany z zajęciami wymagającymi bezpośredniego udziału nauczyciela		31	1,24
Nakład pracy słuchacza związany z zajęciami o charakterze praktycznym		67	2,68
Literatura podstawowa	1. Stoyan Stefanov, React w działaniu. Tworzenie aplikacji internetowych, Helion, 2017 2. Cassio de Sousa Antonio, React dla zaawansowanych, Helion, 2017 3. Stoyan Stefanov, React: Up & Running. Building Web Applications (ebook), O'Reilly Media, 2016		
Literatura uzupełniająca	1. Tutorial: Intro To React : https://reactjs.org/tutorial/tutorial.html 2. React Guide: https://scotch.io/guides/react 3. Kurs React.js: https://typeofweb.com/kurs/react-js/		
Jednostka realizująca	Wydział Informatyki Politechniki Białostockiej	Data opracowania programu	
Program opracował	dr inż. Paweł Tadejko, mgr inż. Przemysław Wiszowaty	22.02.2019	

Wydział Informatyki									
Kierunek studiów	JavaScript Developer							Poziom i forma studiów	Studia podyplomowe
Specjalność / ścieżka dyplomowania								Profil kształcenia	
Nazwa przedmiotu	NodeJS - środowisko i technologia ServerSide 1							Kod przedmiotu	SPJSD04
								Rodzaj przedmiotu	obowiązkowy
Formy zajęć i liczba godzin	W	C	L	P	Ps	T	S	Semestr	1
	8				18			Punkty ECTS	3
Przedmioty wprowadzające									
Cele przedmiotu	Celem przedmiotu jest zapoznanie słuchacza z podstawami środowiska zaprojektowanego do tworzenia wysoce skalowalnych aplikacji internetowych w języku JavaScript. Słuchacz nabędzie umiejętność tworzenia własnego modułu/pakietu.								
Treści programowe	Wykład i pracownia specjalistyczna 1. Środowisko NodeJS - czym jest i jak działa 2. Programowanie asynchroniczne. 3. Podstawy programowania w środowisku NodeJS. 4. Moduł/pakiet - omówienie 5. Tworzenie własnego modułu/pakietu.								
Metody dydaktyczne	1. prezentacja multimedialna; 2. analiza wybranych przypadków; 3. samodzielne opracowanie wskazanych zagadnień; 4. ćwiczenia indywidualne 5. dyskusje dydaktyczne.								
Forma zaliczenia	Wykład: test przeprowadzony w systemie e-learningowym Pracownia specjalistyczna: zaliczenie projektem								
Symbol efektu uczenia się	Zakładane efekty uczenia się							Odniesienie do kierunkowych efektów uczenia się	
EU1	Zna założenie programowania asynchronicznego							JSD_W01, JSD_W02	
EU2	Zna podstawy programowania w środowisku NodeJS							JSD_W01, JSD_W02, JSD_W05	
EU3	Potrafi korzystać z zewnętrznych modułów/pakietów							JSD_U01, JSD_U02, JSD_U05	
EU4	Potrafi stworzyć własny moduł/pakiet.							JSD_U02, JSD_U05	
Symbol efektu uczenia się	Sposoby weryfikacji efektów uczenia się							Forma zajęć, na której zachodzi weryfikacja	
EU1	Test przeprowadzony w systemie e-learningowym							W	
EU2	Test przeprowadzony w systemie e-learningowym							W	
EU3	Zaliczenie projektem							Ps	
EU4	Zaliczenie projektem							Ps	

Bilans nakładu pracy słuchacza (w godzinach)		Liczba godz.	
Wyliczenie	Udział w wykładach	8	
	Udział w pracowni specjalistycznej	18	
	Przygotowanie do pracowni specjalistycznej	20	
	Opracowanie sprawozdań z pracowni	10	
	Udział w konsultacjach związanych z projektem	5	
	Realizacja zadań projektowych (zadań domowych)	14	
	RAZEM:	75	
Wskaźniki ilościowe		GODZINY	ECTS
Nakład pracy słuchacza związany z zajęciami wymagającymi bezpośredniego udziału nauczyciela		31	1,24
Nakład pracy słuchacza związany z zajęciami o charakterze praktycznym		67	2,68
Literatura podstawowa	1. David Herron, Platforma Node.js Przewodnik webdevelopera, Helion 2017 2. Mike Cantelon, Marc Harter, TJ Holowaychuk, Nathan Rajlich, Node.js w akcji, Helion 2015 3. Oficjalna strona dokumentacji: https://nodejs.org/en/docs/		
Literatura uzupełniająca	1. NodeJS tutorial: https://www.w3schools.com/nodejs/ 2. Mardan Azat, Node.js w praktyce., Helion 2015 3. Mardan Azat, Express.js Tworzenie aplikacji sieciowych Node.js, Helion 2016		
Jednostka realizująca	Wydział Informatyki Politechniki Białostockiej	Data opracowania programu	
Program opracował	dr inż. Paweł Tadejko, mgr Piotr Tarasiuk	21.02.2019	

Wydział Informatyki									
Kierunek studiów	JavaScript Developer						Poziom i forma studiów	Studia podyplomowe	
Specjalność / ścieżka dyplomowania							Profil kształcenia		
Nazwa przedmiotu	Podstawy Agile i Scrum						Kod przedmiotu	SPJSD05	
							Rodzaj przedmiotu	obowiązkowy	
Formy zajęć i liczba godzin	W	C	L	P	Ps	T	S	Semestr	1
	6				6			Punkty ECTS	2
Przedmioty wprowadzające									
Cele przedmiotu	Celem przedmiotu jest zapoznanie słuchacza z frameworkiem Scrum, manifestem Agile oraz narzędziami do szacowania prac developerskich. Słuchacz nabędzie umiejętność szacowania prac developerskich.								
Treści programowe	Wykład i pracownia specjalistyczna - Główne wytyczne frameworka Agile. - Role i odpowiedzialność członków zespołu wynikających z manifestu Agile. - Sposoby pracy w zespole - Szacowanie prac developerskich.								
Metody dydaktyczne	1. prezentacja multimedialna; 2. analiza wybranych przypadków; 3. samodzielne opracowanie wskazanych zagadnień; 4. ćwiczenia indywidualne, 5. dyskusje dydaktyczne.								
Forma zaliczenia	Wykład: test przeprowadzony w systemie e-learningowym Pracownia specjalistyczna: zaliczenie projektem								
Symbol efektu uczenia się	Zakładane efekty uczenia się							Odniesienie do kierunkowych efektów uczenia się	
EU1	Zna i rozumie zasady manifestu Agile i Frameworka Scrum							JSD_W02, JSD_W04	
EU2	Zna role i obowiązki członków zespołu developerskiego według Frameworka Scrum							JSD_W02, JSD_W04	
EU3	Potrafi wyszacować czasochłonność wyznaczonych zadań							JSD_U02, JSD_U04	
EU4	Potrafi zastosować wytyczne Scrum w pracy w projekcie.							JSD_U02, JSD_U04	
Symbol efektu uczenia się	Sposoby weryfikacji efektów uczenia się							Forma zajęć, na której zachodzi weryfikacja	
EU1	Test przeprowadzony w systemie e-learningowym							W	
EU2	Test przeprowadzony w systemie e-learningowym							W	
EU3	Zaliczenie projektem							Ps	
EU4	Zaliczenie projektem							Ps	

Bilans nakładu pracy słuchacza (w godzinach)		Liczba godz.	
Wyliczenie	Udział w wykładach	6	
	Udział w pracowni specjalistycznej	6	
	Przygotowanie do pracowni specjalistycznej	10	
	Opracowanie sprawozdań z pracowni	10	
	Udział w konsultacjach związanych z projektem	5	
	Realizacja zadań projektowych (zadań domowych)	13	
	RAZEM:	50	
Wskaźniki ilościowe		GODZINY	ECTS
Nakład pracy słuchacza związany z zajęciami wymagającymi bezpośredniego udziału nauczyciela		17	0,6
Nakład pracy słuchacza związany z zajęciami o charakterze praktycznym		44	1,75
Literatura podstawowa	1. Mariusz Chrapko, Scrum. O zwinnym zarządzaniu projektami, Wydanie II rozszerzone, Helion, 2015 2. Agile Manifesto, [20.06.2018] http://agilemanifesto.org/ 3. Scrum Guide, Przewodnik, [20.06.2018] http://www.scrumguides.org/docs/scrumguide/v2017/2017-Scrum-Guide-Polish.pdf		
Literatura uzupełniająca	1. Roman Pichler: Zarządzanie projektami ze Scrum. Twórz produkty, które pokochają klienci, Helion, 2014		
Jednostka realizująca	Wydział Informatyki Politechniki Białostockiej	Data opracowania programu	
Program opracował	dr inż. Paweł Tadejko, mgr inż. Łukasz Muszyński	21.02.2019	

Wydział Informatyki									
Kierunek studiów	JavaScript Developer						Poziom i forma studiów	Studia podyplomowe	
Specjalność / ścieżka dyplomowania							Profil kształcenia		
Nazwa przedmiotu	Zaawansowane zagadnienia języka JavaScript						Kod przedmiotu	SPJSD07	
							Rodzaj przedmiotu	obowiązkowy	
Formy zajęć i liczba godzin	W	C	L	P	Ps	T	S	Semestr	2
	6				18			Punkty ECTS	3
Przedmioty wprowadzające									
Cele przedmiotu	Celem przedmiotu jest zapoznanie słuchacza z zaawansowanymi elementami programowania JavaScript w standardzie ES6. Słuchacz nabędzie umiejętność programowanie wielowątkowego, organizacji kodu oraz wykorzystywania wzorców projektowych.								
Treści programowe	Wykład i pracownia specjalistyczna 1. Programowanie wielowątkowe 2. Organizacja kodu 3. Wzorce projektowe i wydajność								
Metody dydaktyczne	1. prezentacja multimedialna; 2. analiza wybranych przypadków; 3. samodzielne opracowanie wskazanych zagadnień; 4. ćwiczenia indywidualne, 5. dyskusje dydaktyczne.								
Forma zaliczenia	Wykład: test przeprowadzony w systemie e-learningowym Pracownia specjalistyczna: projekt końcowy								
Symbol efektu uczenia się	Zakładane efekty uczenia się							Odniesienie do kierunkowych efektów uczenia się	
EU1	Zna zaawansowane elementy programowania JavaScript standardzie ES6							JSD_W01, JSD_W02	
EU2	Zna efektywne metody używania zaawansowanych elementów języka							JSD_W01, JSD_S01	
EU3	Potrafi zaplanować i stworzyć kod JS odpowiadający zadanym wymaganiom przy użyciu wzorców projektowych i wiedzy optymalizacyjnej							JSD_U01, JSD_U03, JSD_S01, JSD_S02	
EU4	Potrafi dobrać wzorec do zadanego problemu							JSD_U03, JSD_S01, JSD_S02	
Symbol efektu uczenia się	Sposoby weryfikacji efektów uczenia się							Forma zajęć, na której zachodzi weryfikacja	
EU1	Test przeprowadzony w systemie e-learningowym							W	
EU2	Test przeprowadzony w systemie e-learningowym							W	
EU3	Projekt końcowy							Ps	
EU4	Projekt końcowy							Ps	

Bilans nakładu pracy słuchacza (w godzinach)		Liczba godz.	
Wyliczenie	Udział w wykładach	6	
	Udział w pracowni specjalistycznej	18	
	Przygotowanie do pracowni specjalistycznej	15	
	Opracowanie sprawozdań pracowni	11	
	Udział w konsultacjach związanych z projektem	5	
	Realizacja zadań projektowych (zadań domowych)	20	
	RAZEM:	75	
Wskaźniki ilościowe		GODZINY	ECTS
Nakład pracy słuchacza związany z zajęciami wymagającymi bezpośredniego udziału nauczyciela		29	1,16
Nakład pracy słuchacza związany z zajęciami o charakterze praktycznym		69	2,76
Literatura podstawowa	1. Kirupa Chinnathambi. JavaScript. Przewodnik dla absolutnie początkujących, Helion, 2017 2. Nicolas Bevacqua. Nowoczesny JavaScript. Poznaj ES6 i praktyczne zastosowania nowych rozwiązań, Helion, 2018 3. Luis Atencio. Programowanie funkcyjne z JavaScriptem. Sposoby na lepszy kod, Helion, 2017 4. Simon Timms. JavaScript i wzorce projektowe. Programowanie dla zaawansowanych. Wydanie II, Helion, 2017		
Literatura uzupełniająca	1. JavaScript Tutorial, [20.06.2018] https://www.w3schools.com/jS/default.asp 2. Oficjalna strona dokumentacji Mozilli, [20.06.2018] https://developer.mozilla.org/pl/docs/Web/JavaScript 3. ES6 przykłady: [20.06.2018] http://es6-features.org/		
Jednostka realizująca	Wydział Informatyki Politechniki Białostockiej	Data opracowania programu	
Program opracował	dr inż. Paweł Tadejko, mgr inż. Karol Rogowski	20.02.2019	

Wydział Informatyki									
Kierunek studiów	JavaScript Developer							Poziom i forma studiów	Studia podyplomowe
Specjalność / ścieżka dyplomowania								Profil kształcenia	
Nazwa przedmiotu	Framework i ekosystem React - Aplikacja serwerowa							Kod przedmiotu	SPJSD09
								Rodzaj przedmiotu	obowiązkowy
Formy zajęć i liczba godzin	W	C	L	P	Ps	T	S	Semestr	2
	6				18			Punkty ECTS	3
Przedmioty wprowadzające									
Cele przedmiotu	Celem przedmiotu jest rozszerzenie wiedzy słuchacza o podstawach pracy z frameworkiem React.js. Słuchacz nabędzie umiejętność stworzenia aplikacji komunikującej się z serwerem.								
Treści programowe	Wykład i pracownia specjalistyczna 1. Zasady działania routingu 2. Implementacja asynchroniczna komunikacji z API 3. Biblioteki do zarządzania stanami 4. Przepływy danych 5. Store 6. Przedstawienie dobrych praktyk pisania aplikacji 7. Wprowadzenie do testowania komponentów 8. Stworzenie aplikacji internetowej z wykorzystaniem React.js komunikującej się z API								
Metody dydaktyczne	1. prezentacja multimedialna; 2. analiza wybranych przypadków; 3. samodzielne opracowanie wskazanych zagadnień; 4. ćwiczenia indywidualne, 5. dyskusje dydaktyczne.								
Forma zaliczenia	Wykład: test przeprowadzony w systemie e-learningowym Pracownia specjalistyczna: projekt końcowy								
Symbol efektu uczenia się	Zakładane efekty uczenia się							Odniesienie do kierunkowych efektów uczenia się	
EU1	Zna zasady budowania aplikacji z wykorzystaniem routingu							JSD_W02, JSD_W03	
EU2	Zna zasady i ma wiedzę jak zarządzać danymi (stanem aplikacji)							JSD_W02, JSD_W03	
EU3	Potrafi zaimplementować routing i komunikację z API							JSD_U02, JSD_U03	
EU4	Potrafi zarządzać danymi (stanem aplikacji) i testować komponenty							JSD_U02, JSD_U03	
Symbol efektu uczenia się	Sposoby weryfikacji efektów uczenia się							Forma zajęć, na której zachodzi weryfikacja	
EU1	Test przeprowadzony w systemie e-learningowym							W	
EU2	Test przeprowadzony w systemie e-learningowym							W	
EU3	Projekt końcowy							Ps	
EU4	Projekt końcowy							Ps	

Bilans nakładu pracy słuchacza (w godzinach)		Liczba godz.	
Wyliczenie	Udział w wykładach	6	
	Udział w pracowni specjalistycznej	18	
	Przygotowanie do pracowni specjalistycznej	15	
	Opracowanie sprawozdań pracowni	11	
	Udział w konsultacjach związanych z projektem	5	
	Realizacja zadań projektowych (zadań domowych)	20	
	RAZEM:	75	
Wskaźniki ilościowe		GODZINY	ECTS
Nakład pracy słuchacza związany z zajęciami wymagającymi bezpośredniego udziału nauczyciela		29	1,16
Nakład pracy słuchacza związany z zajęciami o charakterze praktycznym		69	2,76
Literatura podstawowa	1. Stoyan Stefanov, React w działaniu. Tworzenie aplikacji internetowych, Helion, 2017 2. Cassio de Sousa Antonio, React dla zaawansowanych, Helion, 2017 3. Stoyan Stefanov, React: Up & Running. Building Web Applications, O'Reilly Media, 2016		
Literatura uzupełniająca	1. Tutorial: Intro To React: [20.06.2018], https://reactjs.org/tutorial/tutorial.html 2. React Guide: [20.06.2018], https://scotch.io/guides/react 3. Kurs React.js: [20.06.2018], https://typeofweb.com/kurs/react-js/		
Jednostka realizująca	Wydział Informatyki Politechniki Białostockiej	Data opracowania programu	
Program opracował	dr inż. Paweł Tadejko, mgr inż. Przemysław Wiszowaty	21.02.2019	

Wydział Informatyki									
Kierunek studiów	JavaScript Developer						Poziom i forma studiów	Studia podyplomowe	
Specjalność / ścieżka dyplomowania							Profil kształcenia		
Nazwa przedmiotu	NodeJS - środowisko i technologia ServerSide 2						Kod przedmiotu	SPJSD10	
							Rodzaj przedmiotu	obowiązkowy	
Formy zajęć i liczba godzin	W	C	L	P	Ps	T	S	Semestr	2
	6				18			Punkty ECTS	3
Przedmioty wprowadzające									
Cele przedmiotu	Celem przedmiotu jest rozszerzenie wiedzy słuchacza o podstawach programowania asynchronicznego w środowisku NodeJS. Słuchacz nabędzie umiejętność tworzenia aplikacji sieciowych w środowisku NodeJS.								
Treści programowe	Wykład i pracownia specjalistyczna 1. Wprowadzenie do protokołów sieciowych 2. Tworzenie aplikacji sieciowych w środowisku NodeJS 3. Praca z wykorzystaniem framework'a Express 4. Stworzenie własnego mikroservisu 5. Testowanie aplikacji NodeJS								
Metody dydaktyczne	1. prezentacja multimedialna; 2. analiza wybranych przypadków; 3. samodzielne opracowanie wskazanych zagadnień; 4. ćwiczenia indywidualne, 5. dyskusje dydaktyczne.								
Forma zaliczenia	Wykład: test przeprowadzony w systemie e-learningowym Pracownia specjalistyczna: projekt końcowy								
Symbol efektu uczenia się	Zakładane efekty uczenia się							Odniesienie do kierunkowych efektów uczenia się	
EU1	Zna podstawy protokołów sieciowych							JSD_W02, JSD_W03	
EU2	Zna założenie działania aplikacji sieciowych w środowisku NodeJS							JSD_W02, JSD_W03, JSD_W05, JSD_S01	
EU3	Potrafi stworzyć prosty serwer HTTP w oparciu o dostarczone biblioteki							JSD_U02, JSD_U03	
EU4	Potrafi stworzyć testy do aplikacji napisanej w środowisku NodeJS							JSD_U03, JSD_U05, JSD_S02	
Symbol efektu uczenia się	Sposoby weryfikacji efektów uczenia się							Forma zajęć, na której zachodzi weryfikacja	
EU1	Test przeprowadzony w systemie e-learningowym							W	
EU2	Test przeprowadzony w systemie e-learningowym							W	
EU3	Projekt końcowy							Ps	
EU4	Projekt końcowy							Ps	

Bilans nakładu pracy słuchacza (w godzinach)		Liczba godz.	
Wyliczenie	Udział w wykładach	6	
	Udział w pracowni specjalistycznej	18	
	Przygotowanie do pracowni specjalistycznej	15	
	Opracowanie sprawozdań pracowni	11	
	Udział w konsultacjach związanych z projektem	5	
	Realizacja zadań projektowych (zadań domowych)	20	
	RAZEM:	75	
Wskaźniki ilościowe		GODZINY	ECTS
Nakład pracy słuchacza związany z zajęciami wymagającymi bezpośredniego udziału nauczyciela		29	1,16
Nakład pracy słuchacza związany z zajęciami o charakterze praktycznym		69	2,76
Literatura podstawowa	1. David Herron, Platforma Node.js Przewodnik webdevelopeera, Helion 2017 2. Mike Cantelon, Marc Harter, TJ Holowaychuk, Nathan Rajlich, Node.js w akcji, Helion 2015 3. Oficjalna strona dokumentacji: [20.06.2018], https://nodejs.org/en/docs/		
Literatura uzupełniająca	1. NodeJS tutorial: [20.06.2018], https://www.w3schools.com/nodejs/ 2. Mardan Azat, Node.js w praktyce., Helion 2015 3. Mardan Azat, Express.js Tworzenie aplikacji sieciowych Node.js, Helion 2016		
Jednostka realizująca	Wydział Informatyki Politechniki Białostockiej	Data opracowania programu	
Program opracował	dr inż. Paweł Tadejko, mgr Piotr Tarasiuk	21.02.2019	

Wydział Informatyki									
Kierunek studiów	JavaScript Developer						Poziom i forma studiów	Studia podyplomowe	
Specjalność / ścieżka dyplomowania							Profil kształcenia		
Nazwa przedmiotu	Wytwarzanie oprogramowania z wykorzystaniem Agile i Scrum						Kod przedmiotu	SPJSD11	
							Rodzaj przedmiotu	obowiązkowy	
Formy zajęć i liczba godzin	W	C	L	P	Ps	T	S	Semestr	2
	6				6			Punkty ECTS	1
Przedmioty wprowadzające									
Cele przedmiotu	Celem przedmiotu jest zapoznanie słuchacza z zasadami zarządzania sprintem. Słuchacz nabędzie umiejętność wyceny szacunkowej projektu.								
Treści programowe	Wykład i pracownia specjalistyczna 1. Sposobu tworzenia ścieżek użytkownika 2. Zarządzania backlogiem produktowym 3. Przygotowanie wyceny projektowej								
Metody dydaktyczne	1. prezentacja multimedialna; 2. analiza wybranych przypadków; 3. samodzielne opracowanie wskazanych zagadnień; 4. ćwiczenia indywidualne, 5. dyskusje dydaktyczne.								
Forma zaliczenia	Wykład: test przeprowadzony w systemie e-learningowym Pracownia specjalistyczna: projekt końcowy								
Symbol efektu uczenia się	Zakładane efekty uczenia się							Odniesienie do kierunkowych efektów uczenia się	
EU1	Zna i rozumie pojęcie sprintu a także jego składowych							JSD_W04, JSD_S01	
EU2	Zna i rozumie zasady zbierania wytycznych projektowych							JSD_W04, JSD_S01	
EU3	Potrafi zaplanować sprint i zmierzyć postęp prac							JSD_U04, JSD_S02	
EU4	Potrafi przygotować szacunkową wycenę projektu							JSD_U04, JSD_S02	
Symbol efektu uczenia się	Sposoby weryfikacji efektów uczenia się							Forma zajęć, na której zachodzi weryfikacja	
EU1	Test przeprowadzony w systemie e-learningowym							W	
EU2	Test przeprowadzony w systemie e-learningowym							W	
EU3	Projekt końcowy							Ps	
EU4	Projekt końcowy							Ps	

Bilans nakładu pracy słuchacza (w godzinach)		Liczba godz.	
Wyliczenie	Udział w wykładach	6	
	Udział w pracowni specjalistycznej	6	
	Przygotowanie do pracowni specjalistycznej	5	
	Realizacja zadań projektowych (zadań domowych)	3	
	RAZEM:	25	
Wskaźniki ilościowe		GODZINY	ECTS
Nakład pracy słuchacza związany z zajęciami wymagającymi bezpośredniego udziału nauczyciela		12	0,48
Nakład pracy słuchacza związany z zajęciami o charakterze praktycznym		19	0,76
Literatura podstawowa	1. Krystian Kaczor, Scrum i nie tylko. Teoria i praktyka w metodykach Agile, Helion, 2015 2. Mike Cohn, Agile: metodyki zwinne w planowaniu projektów, Helion, 2018 3. Roman Pichler: Zarządzanie projektami ze Scrum. Twórz produkty, które pokochają klienci, Helion, 2014		
Literatura uzupełniająca	1. Andrew Stellman, Jennifer Greene, Agile. Przewodnik po zwinnych metodykach programowania, Helion, 2015 2. Mariusz Chrapko, Scrum. O zwinnym zarządzaniu projektami, Wydanie II rozszerzone, Helion, 2015 3. Scrum Guide, Przewodnik, [20.06.2018] http://www.scrumguides.org/docs/scrumguide/v2017/2017-Scrum-Guide-Polish.pdf		
Jednostka realizująca	Wydział Informatyki Politechniki Białostockiej	Data opracowania programu	
Program opracował	dr inż. Paweł Tadejko, mgr inż. Łukasz Muszyński	21.02.2019	

Wydział Informatyki									
Kierunek studiów	JavaScript Developer						Poziom i forma studiów	Studia podyplomowe	
Specjalność / ścieżka dyplomowania							Profil kształcenia		
Nazwa przedmiotu	Zawansowane programowanie aplikacji w Angular						Kod przedmiotu	SPJSD08	
							Rodzaj przedmiotu	obowiązkowy	
Formy zajęć i liczba godzin	W	C	L	P	Ps	T	S	Semestr	2
	6				18			Punkty ECTS	3
Przedmioty wprowadzające									
Cele przedmiotu	Celem przedmiotu jest utrwalenie zdobytej wiedzy słuchacza o programowania Angular. Słuchacz nabędzie umiejętność tworzenia w pełni funkcjonalnych aplikacji biznesowych. Słuchacz zdobędzie umiejętności istotne dla doświadczonych deweloperów, dotyczące utrzymania, rozwoju kodu i pisania testów komponentów.								
Treści programowe	Wykład i pracownia specjalistyczna 1. Proces rozruchu aplikacji 2. Konfiguracja wstrzykiwania zależności 3. Cykl życia komponentu 4. Zaawansowane elementy Angular: dyrektywy i potoki 5. Zaawansowany routing adresów 6. Komunikacja HTTP z serwerem 7. Testy jednostkowe komponentów 8. Optymalizacja ładowania zasobów serwera 9. Refactoring aplikacji								
Metody dydaktyczne	1. prezentacja multimedialna; 2. analiza wybranych przypadków; 3. samodzielne opracowanie wskazanych zagadnień; 4. ćwiczenia indywidualne, 5. dyskusje dydaktyczne.								
Forma zaliczenia	Wykład: test przeprowadzony w systemie e-learningowym Pracownia specjalistyczna: projekt końcowy								
Symbol efektu uczenia się	Zakładane efekty uczenia się							Odniesienie do kierunkowych efektów uczenia się	
EU1	Zna zasady działania technologii Angular							JSD_W02, JSD_W03	
EU2	Zna zaawansowane zagadnienia dotyczące logiki komponentów i ich cyklu życia							JSD_W02, JSD_W03	
EU3	Potrafi tworzyć złożone komponenty i pisać do nich testy							JSD_U02, JSD_U03	
EU4	Potrafi komunikować się z serwerem i optymalizować ładowanie zasobów serwera							JSD_U02, JSD_U03	
Symbol efektu	Sposoby weryfikacji efektów uczenia się							Forma zajęć, na której	

uczenia się		zachodzi weryfikacja	
EU1	Test przeprowadzony w systemie e-learningowym	W	
EU2	Test przeprowadzony w systemie e-learningowym	W	
EU3	Projekt końcowy	Ps	
EU4	Projekt końcowy	Ps	
Bilans nakładu pracy słuchacza (w godzinach)		Liczba godz.	
Wyliczenie	Udział w wykładach	6	
	Udział w pracowni specjalistycznej	18	
	Przygotowanie do pracowni specjalistycznej	16	
	Opracowanie sprawozdań z pracowni	10	
	Udział w konsultacjach związanych z projektem	5	
	Realizacja zadań projektowych (zadań domowych)	20	
	RAZEM:	75	
Wskaźniki ilościowe		GODZINY	ECTS
Nakład pracy słuchacza związany z zajęciami wymagającymi bezpośredniego udziału nauczyciela		29	1,16
Nakład pracy słuchacza związany z zajęciami o charakterze praktycznym		69	2,76
Literatura podstawowa	1. Yakov Fain, Anton Moiseev, Angular 2. Programowanie z użyciem języka TypeScript, Helion 2017 2. Gion Kunz, Angular 2. Tworzenie interaktywnych aplikacji internetowych, Helion 2017 3. Official Angular Tutorial: https://angular.io/tutorial		
Literatura uzupełniająca	1. Angular 4 Tutorial: https://www.tutorialspoint.com/angular4/index.htm 2. Adam Freeman, Angular. Profesjonalne techniki programowania. Wydanie II, Helion 2017 3. Angular Best Practices: https://www.pluralsight.com/courses/best-practices-angular 4. Nate Murray, Felioe Coury, Ari Lerner, Carlos Taborda, ng-book, Fullstack.io 2017		
Jednostka realizująca	Wydział Informatyki Politechniki Białostockiej	Data opracowania programu	
Program opracował	<i>dr inż. Paweł Tadejko, Kamil Mijacz</i>	21.02.2019	

Wydział Informatyki									
Kierunek studiów	JavaScript Developer							Poziom i forma studiów	Studia podyplomowe
Specjalność / ścieżka dyplomowania								Profil kształcenia	
Nazwa przedmiotu	Metodyka DevOps - produkcja i utrzymanie oprogramowania							Kod przedmiotu	SPJSD06
								Rodzaj przedmiotu	obowiązkowy
Formy zajęć i liczba godzin	W	C	L	P	Ps	T	S	Semestr	1
	3				9			Punkty ECTS	1
Przedmioty wprowadzające									
Cele przedmiotu	Celem przedmiotu jest zapoznanie słuchacza z podstawowymi narzędziami wykorzystywanymi w pracy z JavaScriptem tj. systemami kontroli wersji, narzędziami utrzymującymi jakość kodu, narzędziami pozwalającymi na automatyzację zadań. Słuchacz nabędzie umiejętność postawienie swojego serwera developerskiego.								
Treści programowe	Wykład i pracownia specjalistyczna 1. Działanie systemów kontroli wersji 2. Narzędzia automatyzacji pracy 3. Narzędzia utrzymujące jakość kodu 4. Wprowadzenie do używania GITa 5. Wprowadzenie do używania Gulp.js 6. Wprowadzenie do używania Eslint								
Metody dydaktyczne	1. prezentacja multimedialna; 2. analiza wybranych przypadków; 3. samodzielne opracowanie wskazanych zagadnień; 4. ćwiczenia indywidualne; 5. dyskusje dydaktyczne.								
Forma zaliczenia	Wykład: test przeprowadzony w systemie e-learningowym Pracownia specjalistyczna: zaliczenie projektem								
Symbol efektu uczenia się	Zakładane efekty uczenia się							Odniesienie do kierunkowych efektów uczenia się	
EU1	Zna podstawy systemów kontroli wersji							JSD_W03, JSD_W05	
EU2	Zna podstawowe narzędzia automatyzacji pracy							JSD_W03, JSD_W05	
EU3	Potrafi korzystać z GITa							JSD_U03, JSD_U05	
EU4	Potrafi stworzyć serwer developerski i własne skrypty							JSD_U03, JSD_U05	
Symbol efektu uczenia się	Sposoby weryfikacji efektów uczenia się							Forma zajęć, na której zachodzi weryfikacja	
EU1	Test przeprowadzony w systemie e-learningowym							W	
EU2	Test przeprowadzony w systemie e-learningowym							W	
EU3	Zaliczenie projektem							Ps	
EU4	Zaliczenie projektem							Ps	

Bilans nakładu pracy słuchacza (w godzinach)		Liczba godz.	
Wyliczenie	Udział w wykładach	3	
	Udział w pracowni specjalistycznej	9	
	Przygotowanie do pracowni specjalistycznej	4	
	Opracowanie sprawozdań z pracowni	3	
	Udział w konsultacjach związanych z projektem	3	
	Realizacja zadań projektowych (zadań domowych)	3	
	RAZEM:	25	
Wskaźniki ilościowe		GODZINY	ECTS
Nakład pracy słuchacza związany z zajęciami wymagającymi bezpośredniego udziału nauczyciela		15	0,6
Nakład pracy słuchacza związany z zajęciami o charakterze praktycznym		22	0,88
Literatura podstawowa	1. Jon Loeliger, Matthew McCullough, Kontrola wersji z systemem Git. Narzędzia i techniki programistów, Wydanie II, O'Reilly, 2014 2. Travis Maynard, Getting Started with Gulp - Second Edition, 2017		
Literatura uzupełniająca	1. Oficjalna dokumentacja GITa: https://git-scm.com/docs 2. Oficjalna strona Gulp.js: https://gulpjs.com/ 3. Oficjalna dokumentacja Eslint: https://eslint.org/docs/user-guide/getting-started		
Jednostka realizująca	Wydział Informatyki Politechniki Białostockiej	Data opracowania programu	
Program opracował(a)	dr inż. Paweł Tadejko, mgr inż.. Dawid Kozak	21.02.2019	

Wydział Informatyki									
Kierunek studiów	JavaScript Developer						Poziom i forma studiów	Studia podyplomowe	
Specjalność / ścieżka dyplomowania							Profil kształcenia		
Nazwa przedmiotu	Aplikacje mobilne w technologii React Native						Kod przedmiotu	SPJSD12	
							Rodzaj przedmiotu	obowiązkowy	
Formy zajęć i liczba godzin	W	C	L	P	Ps	T	S	Semestr	2
	6				15			Punkty ECTS	2
Przedmioty wprowadzające									
Cele przedmiotu	Celem przedmiotu jest zapoznanie słuchacza z zasadami tworzenia aplikacji wieloplatformowych (cross platform) z wykorzystaniem frameworka React Native na urządzenia mobilne. Słuchacz nabędzie umiejętność stworzenia szkieletu własnej aplikacji React Native dla systemu Android/iOS.								
Treści programowe	Wykład i pracownia specjalistyczna 1. Wprowadzenie do mobilnych aplikacji wieloplatformowych. 2. Tworzenie aplikacji mobilnych w React Native. 3. Wykorzystanie frameworka react Native i łączenie go z aplikacją natywną na Android/iOS. 4. Zasady współpracy pomiędzy kodem napisanym w JavaScript a częścią natywną aplikacji. 5. Stworzenie szkieletu własnej aplikacji React Native dla Androida lub iOS.								
Metody dydaktyczne	1. prezentacja multimedialna; 2. analiza wybranych przypadków; 3. samodzielne opracowanie wskazanych zagadnień; 4. ćwiczenia indywidualne; 5. dyskusje dydaktyczne.								
Forma zaliczenia	Wykład: test przeprowadzony w systemie e-learningowym Pracownia specjalistyczna: projekt końcowy								
Symbol efektu uczenia się	Zakładane efekty uczenia się						Odniesienie do kierunkowych efektów uczenia się		
EU1	Zna założenia i problem aplikacji wieloplatformowych na urządzenia mobilne						JSD_W02, JSD_W03		
EU2	Zna założenie działania aplikacji napisanych w React Native i szkieletu aplikacji natywnych Android/iOS						JSD_W03		
EU3	Potrafi stworzyć prostą aplikację w React Native						JSD_U03		
EU4	Potrafi oprogramować podstawową komunikację pomiędzy częścią aplikacji JavaScript a natywnym szkieletem.						JSD_U03		
Symbol efektu uczenia się	Sposoby weryfikacji efektów uczenia się						Forma zajęć, na której zachodzi weryfikacja		
EU1	Test przeprowadzony w systemie e-learningowym						W		
EU2	Test przeprowadzony w systemie e-learningowym						W		
EU3	Projekt końcowy						Ps		
EU4	Projekt końcowy						Ps		

Bilans nakładu pracy słuchacza (w godzinach)		Liczba godz.	
Wyliczenie	Udział w wykładach	6	
	Udział w pracowni specjalistycznej	15	
	Przygotowanie do pracowni specjalistycznej	10	
	Opracowanie sprawozdań z pracowni	9	
	Udział w konsultacjach związanych z projektem	5	
	Realizacja zadań projektowych (zadań domowych)	5	
	RAZEM:	50	
Wskaźniki ilościowe		GODZINY	ECTS
Nakład pracy słuchacza związany z zajęciami wymagającymi bezpośredniego udziału nauczyciela		26	1,04
Nakład pracy słuchacza związany z zajęciami o charakterze praktycznym		44	1,76
Literatura podstawowa	1. David Herron, Platforma Node.js Przewodnik webdevelopera, Helion 2017 2. Mike Cantelon, Marc Harter, TJ Holowaychuk, Nathan Rajlich, Node.js w akcji, Helion 2015 3. Oficjalna strona dokumentacji: https://nodejs.org/en/docs/		
Literatura uzupełniająca	1. NodeJS tutorial: https://www.w3schools.com/nodejs/ 2. Mardan Azat, Node.js w praktyce., Helion 2015 3. Mardan Azat, Express.js Tworzenie aplikacji sieciowych Node.js, Helion 2016		
Jednostka realizująca	Wydział Informatyki Politechniki Białostockiej	Data opracowania programu	
Program opracował(a)	dr inż. Paweł Tadejko, mgr inż. Michał Kwilman	21.02.2019	

Zasoby biblioteczne

- Cassio de Souza Antonio, React dla zaawansowanych, Helion, 2017
- David Herron, Platforma Node.js Przewodnik webdevelopera, Helion 2017
- Gion Kunz, Angular 2. Tworzenie interaktywnych aplikacji internetowych, Helion 2017
- Kirupa Chinnathambi. JavaScript. Przewodnik dla absolutnie początkujących, Helion, 2017
- Luis Atencio. Programowanie funkcyjne z JavaScriptem. Sposoby na lepszy kod, Helion, 2017
- Mariusz Chrapko, Scrum. O zwinym zarządzaniu projektami, Wydanie II rozszerzone, Helion, 2015
- Mike Cohn, Agile: metodyki zwinne w planowaniu projektów, Helion, 2018
- Mike Cantelon, Marc Harter, TJ Holowaychuk, Nathan Rajlich, Node.js w akcji, Helion, 2015
- Nicolas Bevacqua. Nowoczesny JavaScript. Poznaj ES6 i praktyczne zastosowania nowych rozwiązań, Helion, 2018
- Stoyan Stefanov, React w działaniu. Tworzenie aplikacji internetowych, Helion, 2017
- Jon Loeliger, Matthew McCullough, Kontrola wersji z systemem Git. Narzędzia i techniki programistów, Wydanie II, O'Reilly, 2014
- Roman Pichler: Zarządzanie projektami ze Scrum. Twórz produkty, które pokochają klienci, Helion, 2014

Wszystkie powyższe pozycje dostępne są w Bibliotece PB.

Elektroniczne zasoby wiedzy

- <https://angular.io/tutorial>
- <https://angular.io/tutorial>
- <https://nodejs.org/en/docs/>
- <http://agilemanifesto.org/>
- <http://www.scrumguides.org/docs/scrumguide/v2017/2017-Scrum-Guide-Polish.pdf>