

Politechnika Białostocka
Wydział Informatyki

Program studiów podyplomowych

Tester Oprogramowania

edycja 2022/23

Sylwetka absolwenta

Studia poświęcone są zagadnieniom dotyczącym jakości oprogramowania, metodyk i narzędzi testowania oprogramowania i technologii powiązanych. Uczestnik poznaje od podstaw pojęcia związane z cyklem życia oprogramowania, uczy się narzędzi oraz poznaje podstawy języka JavaScript i jego możliwości istotnych w opracowywaniu skryptów testów automatycznych.

Absolwent studiów podyplomowych będzie przygotowany do podjęcia pracy na stanowiskach określanych na rynku pracy jako Junior Tester oprogramowania, QA Tester, Software Quality Assurance, Software QA Tester w firmach związanych z wytwarzaniem oprogramowania, w szczególności aplikacji WWW i mobilnych. Będzie posiadał także wiedzę i umiejętności z zakresu technologii i podstawowej analizy ruchu sieciowego, standardów i technologii WWW oraz pracy w zespole developerskim. Uczestnik zdobywa wiedzę w oparciu o praktyczne przykłady oraz doświadczenie kadry dydaktycznej. Po zakończeniu studiów uczestnik będzie posiadał m.in. wiedzę i umiejętności:

- związane z cyklem życia oprogramowania i teoretycznych podstawach procesu wytwarzania oprogramowania;
- pozwalające na pracę z wykorzystaniem kluczowych dla branży IT procesów, metodyk i narzędzi testowania;
- posługiwania się narzędziami wykorzystywanymi we współczesnej pracy specjalisty Software Quality Assurance oraz dobrymi praktykami prowadzenia testów;
- w zakresie wybranych tematów testowania automatycznego i aplikacji mobilnych, zrozumienia procesu oraz wykorzystania wybranego narzędzia do prowadzenia testów automatycznych;
- dotyczące narzędzi i frameworków testowania automatycznego i będzie potrafił używać w pracy wybrane środowisko zintegrowane, należące do grupy najczęściej używanych;
- na temat metodyk pracy przy projektach - Agile i Scrum, kultury pracy deweloperów i tworzenia kodu oraz całego procesu;
- dotyczące podstaw programowania w języku JavaScript i jego możliwości istotnych w opracowywaniu skryptów testów automatycznych;
- w zakresie wykorzystania wybranych narzędzi developerskich w pracy testera oprogramowania, m.in. Gita, Visual Studio Code oraz przygotowywania środowiska do testów, etc.

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 PRK 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 związanych z obsługą systemów operacyjnych i instalacją oprogramowania. Nie jest wymagane posiadanie wykształcenia technicznego. Wymagane jest natomiast posiadanie znajomości języka angielskiego na poziomie minimum średnio zaawansowanym, gdyż brak znajomości tego języka znacznie utrudni proces zdobywania wiedzy i umiejętności przez uczestnika.

Program studiów

Studia podyplomowe **Tester Oprogramowania** trwają 2 semestry i umożliwiają uzyskanie kwalifikacji częściowych na poziomie **6 PRK**. Łączna liczba punktów ECTS: **30**. Łączna liczba godzin zajęć: **214**.

Plan studiów TESTER OPROGRAMOWANIA

Lp.	Nazwa przedmiotu	Kod	Liczba ECTS	Liczba godzin w semestrze			Forma zaliczenia
				W	Ć	PS	
SEMESTR I							
1	Wprowadzenie do testowania oprogramowania i technologii ICT	SPTO01	2	16	0	0	ZO
2	Standardy WWW i podstawy ruchu sieciowego	SPTO02	3	4	0	12	ZO
3	Podstawowe narzędzia developerskie testera	SPTO03	3	8	0	16	ZO
4	Testowanie manualne	SPTO04	3	6	0	18	ZO
5	Wstęp do programowania w JavaScript	SPTO05	3	6	0	18	ZO
6	Testowanie automatyczne I - Wprowadzenie	SPTO06	1	3	0	9	ZO
RAZEM W SEMESTRZE			15	43	0	73	Razem godz. 116
SEMESTR II							
1	Agile i Scrum - Metodyka pracy zwinnej w zespole	SPTO07	3	5	0	15	ZO
2	Testowanie Aplikacji Mobilnych	SPTO08	2	8	0	8	ZO
3	Programowanie obiektowe w JavaScript	SPTO09	3	3	0	12	ZO
4	Testowanie automatyczne II - Narzędzia	SPTO10	4	6	0	18	ZO
5	Podstawy SQL w pracy testera aplikacji	SPTO11	3	5	0	18	ZO
RAZEM W SEMESTRZE			15	27	0	71	Razem godz. 98
RAZEM KIERUNEK [w podziale na FORMY ZAJĘCIOWE]							
ŁĄCZNIE ECTS/GODZ. W TRAKCIE STUDIÓW			30	70	0	144	Razem godz. 214

ZO - zaliczenie na ocenę

Zestawienie efektów uczenia się

Zestawienie tabelaryczne kierunkowych efektów uczenia się odnoszących się do charakterystyk drugiego stopnia (poziom 6 PRK) określonych na podstawie Ustawy z dnia 22 grudnia 2015 r. o Zintegrowanym Systemie Kwalifikacji.

Objaśnienia oznaczeń:

P6 – poziom 6 PRK (Polska Rama 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

N – narzędzia i materiały

U – umiejętności

I – informacje

W – wykorzystanie wiedzy

K – komunikowanie się

O – organizacja pracy

U – uczenie się i rozwój zawodowy

N – narzędzia i materiały

K – kompetencje społeczne

K – krytyczna ocena

O – odpowiedzialność

R – rola zawodowa

P – przestrzeganie reguł

W – współpraca

SQA – Tester Oprogramowania

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

Załącznik nr 1 do „Wytyczne do tworzenia programów studiów podyplomowych”

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			
SQA_W01	wybrane pojęcia związane z branżą IT, cyklem życia oprogramowania i podstawami procesu wytwarzania oprogramowania, ze szczególnym naciskiem na procesy planowania i prowadzenia testów oraz zapewnienia jakości.	P6S_WG, P6S_WK	P6Z_WT, P6Z_WZ, P6Z_WO
SQA_W02	w zaawansowanym stopniu wybrane procesy, metodyki i narzędzia testowania oprogramowania, ze szczególnym uwzględnieniem aplikacji WWW i mobilnych.	P6S_WG	P6Z_WT, P6Z_WZ, P6Z_WO
SQA_W03	w zaawansowanym stopniu wybrane narzędzia i jeden z frameworków testowania automatycznego oraz wybrane elementy języka skryptowego wspierającego automatyzację.	P6S_WG	P6Z_WT, P6Z_WZ, P6Z_WO
SQA_W04	w zaawansowanym stopniu zasady pracy w zespole projektowym wykorzystującym framework Scrum i wytyczne manifestu Agile.	P6S_WG	P6Z_WT, P6Z_WZ, P6Z_WO
SQA_W05	elementy języka SQL i zasady wykorzystywania wybranych narzędzi DevOps we współczesnej pracy testera oprogramowania oraz praktyki utrzymania systemów informatycznych.	P6S_WG, P6S_WK	P6Z_WT, P6Z_WZ, P6Z_WO
SQA_W06	mechanizmy ruchu sieciowego w Internecie, bezpieczeństwa aplikacji oraz komunikacji za pomocą API w systemach rozproszonych.	P6S_WG, P6S_WK	P6Z_WT, P6Z_WZ, P6Z_WO
Umiejętności: absolwent potrafi			
SQA_U01	identyfikować klasy i źródła problemów: sieć, aplikacja front-end / back-end i warstwa prezentacji HTML/CSS oraz wykorzystywać API aplikacji webowych z poziomu specjalistycznych narzędzi.	P6S_UW, P6S_UO	P6Z_UI, P6Z_UN, P6Z_UO
SQA_U02	projektować scenariusze testowe oraz prawidłowo raportować błędy i przeprowadzać ich późniejszą weryfikację.	P6S_UW, P6S_UO	P6Z_UI, P6Z_UN, P6Z_UO
SQA_U03	wykorzystać w pracy testera wybrane narzędzia i frameworki testowania automatycznego i posługiwać się wybranym środowiskiem do automatyzacji scenariuszy testowych.	P6S_UW, P6S_UK, P6S_UU	P6Z_UI, P6Z_UN, P6Z_UO

SQA_U04	pracować zespołowo i wykorzystywać w pracy wybrane narzędzia procesu Agile i Scrum, komunikować się z zespołem używając pojęć oraz przygotować dokumentację procesów testowania.	P6S_UW, P6S_UK, P6S_UO	P6Z_UI, P6Z_UN, P6Z_UO
SQA_U05	brać udział w procesie testowania oprogramowania wykorzystując narzędzia i środowisko DevOps oraz projektować wybrane skrypty SQL służące do wybierania i generowania danych.	P6S_UW	P6Z_UI, P6Z_UN, P6Z_UO
SQA_U06	interpretować znaczenie składni wybranego języka skryptowego oraz opracowywać proste skrypty wykorzystywane m.in. do automatyzacji scenariuszy testowych.	P6S_UW	
Kompetencje społeczne: absolwent jest gotów do			
SQA_K01	rozwiązywania problemów w sposób poznawczy i praktyczny.	P6S_KK	P6Z_KP

Ramowe programy przedmiotów

Wydział Informatyki										
Kierunek studiów	Tester Oprogramowania						Poziom i forma studiów		Studia podyplomowe	
Specjalność / ścieżka dyplomowania							Profil kształcenia			
Nazwa przedmiotu	Wprowadzenie do testowania oprogramowania i technologii ICT						Kod przedmiotu		SPT001	
							Rodzaj przedmiotu		obowiązkowy	
Formy zajęć i liczba godzin	W	C	L	P	Ps	T	S	Semestr		1
	16							Punkty ECTS		2
Przedmioty wprowadzające										
Cele przedmiotu	Celem przedmiotu jest zapoznanie uczestnika z podstawowymi pojęciami związanymi z cyklem życia oprogramowania i teoretycznymi podstawami procesu wytwarzania oprogramowania.									
Treści programowe	<p>Wykład:</p> <ol style="list-style-type: none"> 1. Podstawowe zagadnienia związane z oprogramowaniem: <ol style="list-style-type: none"> a. Kod źródłowy, języki programowania; b. Skrypty, kompilacja; c. Podstawy teoretyczne procesu wytwarzania oprogramowania ze szczególnym uwzględnieniem zapewnienia jakości; d. Pojęcia z branży IT, najważniejsze podejścia do wytwarzania oprogramowania oraz do testów - m.in. planowanie, typy testów oraz podstawowe techniki ich projektowania; e. Aplikacje webowe, lokalne, serwerowe; f. Modele wytwarzania oprogramowania, Test-Driven Development / Behavior-Driven Development; g. Błędy: rodzaje, przyczyny i skutki. 2. Role w zespole projektowym i pojęcia potrzebne do zrozumienia procesu tworzenia oprogramowania: <ol style="list-style-type: none"> a. Programiści, QA - Quality Assurance, management, projektanci, analitycy, UX – User Experience; b. Scenariusze testowe; c. Poziomy i typy testów. 									
Metody dydaktyczne	1. prezentacja multimedialna; 2. analiza wybranych przypadków; 3. dyskusje dydaktyczne.									
Forma zaliczenia	Wykład: test przeprowadzony w systemie e-learningowym									
Symbol efektu uczenia się	Zakładane efekty uczenia się							Odniesienie do kierunkowych efektów uczenia się		
EU1	Zna kluczowe pojęcia dla branży IT związane z realizacją procesów wytwarzania oprogramowania.							SQA_W01, SQA_W02		

EU2	Zna pojęcia potrzebne do komunikacji z pozostałymi osobami w zespole projektowym (programiści, managerowie, projektanci).	SQA_W01, SQA_W02
EU3	Posiada wiedzę nt. teorii zapewniania jakości w procesie tworzenia oprogramowania, oraz poszczególnych rodzajów i metodyk testowania.	SQA_W01, SQA_W02
EU4	Posiada wiedzę pozwalającą na dobór odpowiedniego podejścia do testowania.	SQA_W01, SQA_W02
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	Test przeprowadzony w systemie e-learningowym	W
EU4	Test przeprowadzony w systemie e-learningowym	W
Bilans nakładu pracy uczestnika (w godzinach)		Liczba godz.
Wyliczenie	Udział w wykładach	16
	Udział w konsultacjach związanych z wykładem	4
	Przygotowanie do zajęć wykładowych	20
	Przygotowanie do zaliczenia wykładu	10
	RAZEM:	50
Wskaźniki ilościowe		GODZINY ECTS
Nakład pracy uczestnika związany z zajęciami wymagającymi bezpośredniego udziału nauczyciela		20 0,80
Nakład pracy uczestnika związany z zajęciami o charakterze praktycznym		0 0,00
Literatura podstawowa	<ol style="list-style-type: none"> 1. Ian Sommerville, Inżynieria oprogramowania, Warszawa: Wydaw. Naukowe PWN, 2020; 2. Bogumiła Hnatkowska Red., Zbigniew Huzar, Inżynieria oprogramowania: metody wytwarzania i wybrane zastosowania, Warszawa: Wydaw. Naukowe PWN, 2008; 3. Karolina Zmitrowicz, Jakość projektów informatycznych: rozwój i testowanie oprogramowania, Gliwice: Helion, 2015; 	
Literatura uzupełniająca	<ol style="list-style-type: none"> 1. Adam Roman Red.; Karolina Zmitrowicz; Wojciech Anzel; Maciej Chmielarz; Aleksandra Kornecka; Natalia Krawczyk; Ewa Marchewka; Rafał Nazwalski; Adam Romanowicz Pirek; Bartłomiej Prędko Sasin Stankiewicz, Testowanie oprogramowania w praktyce: studium przypadków 2.0, Warszawa: Wydaw. Naukowe PWN, 2018; 2. Adam Roman, Testowanie i jakość oprogramowania: modele, techniki, narzędzia, Warszawa : Wydaw. Naukowe PWN, 2015; 3. Radosław Smilgin, Zawód tester: od decyzji do pierwszych kroków w pracy, Warszawa: Wydaw. Naukowe PWN, 2016; 	
Jednostka realizująca	Wydział Informatyki Politechniki Białostockiej	Data opracowania programu
Program opracował	<i>inż. Hubert Zub, dr inż. Paweł Tadejko</i>	18.05.2021

Wydział Informatyki									
Kierunek studiów	Tester Oprogramowania							Poziom i forma studiów	Studia podyplomowe
Specjalność / ścieżka dyplomowania								Profil kształcenia	
Nazwa przedmiotu	Standardy WWW i podstawy ruchu sieciowego							Kod przedmiotu	SPT002
								Rodzaj przedmiotu	obowiązkowy
Formy zajęć i liczba godzin	W	C	L	P	Ps	T	S	Semestr	1
	4				12			Punkty ECTS	3
Przedmioty wprowadzające									
Cele przedmiotu	Celem przedmiotu jest przekazanie uczestnikowi wiedzy i umiejętności z zakresu technologii internetowych, podstawowej analizy ruchu sieciowego oraz standardów i technologii WWW.								
Treści programowe	<p>Wykład:</p> <ol style="list-style-type: none"> Podstawowe mechanizmy: Klient, serwer, żądanie i odpowiedź; Klasyczne elementy budulcowe i podejścia: front-end, back-end, cienki/gruby klient; Podstawy protokołu HTTP; Nagłówki, treść żądania a treść odpowiedzi. Podstawowe technologie używane w Internecie: HTML i CSS jako języki opisowy stron; JavaScript jako język skryptowy; Notacja JSON, XML jako składnia opisu danych. <p>Pracownia specjalistyczna:</p> <ol style="list-style-type: none"> Domeny, DNS, adresy IP – podstawy. Przeglądarka a aplikacja mobilna. Podstawy security w sieci: <ol style="list-style-type: none"> HTTPS (dlaczego, na prostych przykładach); Tokeny (na przykładzie JWT); Local storage i cookies. Narzędzia: <ol style="list-style-type: none"> Chrome Developer Tools: <ol style="list-style-type: none"> Spoofing: urządzenia, warunki sieciowe; Inspekcja elementów; Konsola deweloperska; Inspekcja wywołań sieciowych. Postman: <ol style="list-style-type: none"> Podstawy korzystania - tworzenie i inspekcja żądań; Tworzenie i prowadzenie biblioteki testów. BrowserStack – Testowanie aplikacji w różnych przeglądarkach. 								
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: zadania realizowane podczas zajęć								
Symbol efektu uczenia się	Zakładane efekty uczenia się							Odniesienie do kierunkowych efektów uczenia się	

EU1	Zna podstawowe mechanizmy działania Internetu i sposoby identyfikacji problemów w warstwie prezentacji: HTML i CSS.	SQA_W02, SQA_W03	
EU2	Ma wiedzę o fundamentalnych zasadach bezpieczeństwa aplikacji.	SQA_W05, SQA_W06	
EU3	Rozumie mechanizmy i zasady ruchu sieciowego, identyfikacje klasy i źródła problemów: sieć, aplikacja front-end / back-end.	SQA_U01, SQA_U05, SQA_K01	
EU4	Potrafi wykorzystać serwis postman w testowaniu komunikacji z API aplikacji webowych z poziomu narzędzi z pominięciem przeglądarki.	SQA_U05, SQA_K01	
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	Zadania realizowane podczas zajęć i prac domowych	Ps	
EU4	Zadania realizowane podczas zajęć i prac domowych	Ps	
Bilans nakładu pracy uczestnika (w godzinach)		Liczba godz.	
Wyliczenie	Udział w wykładach	4	
	Udział w pracowni specjalistycznej	12	
	Przygotowanie do pracowni specjalistycznej	20	
	Udział w konsultacjach związanych z pracownią	5	
	Realizacja zadań projektowych i zadań domowych	34	
	RAZEM:	75	
Wskaźniki ilościowe		GODZINY	ECTS
Nakład pracy uczestnika związany z zajęciami wymagającymi bezpośredniego udziału nauczyciela		21	0,84
Nakład pracy uczestnika związany z zajęciami o charakterze praktycznym		71	2,84
Literatura podstawowa	<ol style="list-style-type: none"> 1. Mikołaj Jacek Łuczak, Komunikacja w społeczeństwie sieci: technologia, bezpieczeństwo i zmiana społeczna, Poznań: Wydaw. Naukowe SILVA RERUM, 2019 2. Russell Bradford, Podstawy sieci komputerowych, Warszawa: Wydaw. Komunikacji i Łączności, 2009 3. Laura Lemay, Rafe Colburn, Jennifer Kyrmin, HTML, CSS i JavaScript dla każdego, Gliwice: Helion, 2017 		
Literatura uzupełniająca	<ol style="list-style-type: none"> 1. Paco Hope, Ben Walther, Testowanie bezpieczeństwa aplikacji internetowych: receptury, Gliwice: Helion, 2010 2. Testowanie API - Postman dla początkujących. RST Software, [18.05.2021], https://rst.software.pl/blog/2018/04/testowanie-api-postman-dla-poczatkujacych/ 3. Testowanie aplikacji REST przy użyciu Postmana - Blog testuj.pl, [18.05.2021], https://testuj.pl/blog/testowanie-aplikacji-rest-przy-uzyciu-postmana/ 		
Jednostka realizująca	Wydział Informatyki Politechniki Białostockiej	Data opracowania programu	
Program opracował	<i>Marcin Ambrożej, dr inż. Paweł Tadejko</i>	18.05.2021	

Wydział Informatyki									
Kierunek studiów	Tester Oprogramowania							Poziom i forma studiów	Studia podyplomowe
Specjalność / ścieżka dyplomowania								Profil kształcenia	
Nazwa przedmiotu	Podstawowe narzędzia developerskie testera							Kod przedmiotu	SPT003
								Rodzaj przedmiotu	obowiązkowy
Formy zajęć i liczba godzin	W	C	L	P	Ps	T	S	Semestr	1
	8				16			Punkty ECTS	3
Przedmioty wprowadzające									
Cele przedmiotu	Celem przedmiotu jest przekazanie uczestnikowi wiedzy i nauka wykorzystania w praktyce wybranych narzędzi developerskich w pracy testera oprogramowania.								
Treści programowe	<p>Wykład:</p> <ol style="list-style-type: none"> 1. Podstawy systemów operacyjnych. 2. Przygotowywanie środowiska do testów. 3. Systemy kontroli wersji. 4. Zintegrowane środowiska developerskie. <p>Pracownia specjalistyczna:</p> <ol style="list-style-type: none"> 1. Podstawy wiersza poleceń systemu operacyjnego: <ol style="list-style-type: none"> a. operacje na plikach i katalogach; b. zmienne środowiskowe; c. operacje na procesach. 2. Warsztaty z wykorzystania narzędzia kontroli wersji – Git. 3. Wykorzystanie implementacji kontenerów na przykładzie Dockera, jako narzędzia do testowania w różnych środowiskach i konfiguracjach. 4. Przygotowywanie środowiska do testów. 5. Zintegrowane narzędzia developerskie: <ol style="list-style-type: none"> a. Visual Studio Code; b. Visual Studio Community Edition. 								
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: zadania realizowane podczas zajęć								
Symbol efektu uczenia się	Zakładane efekty uczenia się							Odniesienie do kierunkowych efektów uczenia się	
EU1	Zna wybrane pojęcia związane z działaniem systemów operacyjnych.							SQA_W01, SQA_W02	
EU2	Ma wiedzę o tym jaka idea stoi za wykorzystaniem mechanizmów i narzędzi do budowy środowisk testowych.							SQA_W03, SQA_W04, SQA_W05	
EU3	Potrafi w zakresie wybranych operacji korzystać z narzędzia kontroli wersji Git oraz zintegrowanych narzędzi developerskich.							SQA_U04, SQA_U05	

EU4	Potrafi operować wybranymi poleceniami systemu operacyjnego i wywołaniem wybranych narzędzi testera.	SQA_U01	
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	Zadania realizowane podczas zajęć i prac domowych	Ps	
EU4	Zadania realizowane podczas zajęć i prac domowych	Ps	
Bilans nakładu pracy uczestnika (w godzinach)		Liczba godz.	
Wyliczenie	Udział w wykładach	8	
	Udział w pracowni specjalistycznej	16	
	Przygotowanie do pracowni specjalistycznej	16	
	Udział w konsultacjach związanych z pracownią	5	
	Realizacja zadań projektowych i zadań domowych	30	
	RAZEM:	75	
Wskaźniki ilościowe		GODZINY	ECTS
Nakład pracy uczestnika związany z zajęciami wymagającymi bezpośredniego udziału nauczyciela		29	1,16
Nakład pracy uczestnika związany z zajęciami o charakterze praktycznym		67	2,68
Literatura podstawowa	<ol style="list-style-type: none"> Adam Roman, Testowanie i jakość oprogramowania: modele, techniki, narzędzia, Warszawa : Wydaw. Naukowe PWN, 2015; Radosław Smilgin, Zawód tester: od decyzji do pierwszych kroków w pracy, Warszawa: Wydaw. Naukowe PWN, 2016; Bogdan Bereza-Jarociński, Bolesław Szomański, Inżynieria oprogramowania : jak zapewnić jakość tworzonej aplikacjom, Gliwice : Helion, 2009; 		
Literatura uzupełniająca	<ol style="list-style-type: none"> Adam Roman Red.; Karolina Zmitrowicz; Wojciech Anzel; Maciej Chmielarz; Aleksandra Kornecka; Natalia Krawczyk; Ewa Marchewka; Rafał Nazwalski; Adam Romanowicz Pirek; Bartłomiej Prędko Sasin Stankiewicz, Testowanie oprogramowania w praktyce: studium przypadków 2.0, Warszawa: Wydaw. Naukowe PWN, 2018; Testowanie aplikacji REST przy użyciu Postmana - Blog testuj.pl, [18.05.2021], https://testuj.pl/blog/testowanie-aplikacji-rest-przy-uzyciu-postmana/ Automatyzacja testów funkcjonalnych przy pomocy Cypress.io, [18.05.2021], https://testuj.pl/blog/automatyzacja-testow-funkcjonalnych-przy-pomocy-cypress-io/ 		
Jednostka realizująca	Wydział Informatyki Politechniki Białostockiej	Data opracowania programu	
Program opracował	<i>Ernest Czajkowski, dr inż. Paweł Tadejko</i>	18.05.2021	

Wydział Informatyki									
Kierunek studiów	Tester Oprogramowania							Poziom i forma studiów	Studia podyplomowe
Specjalność / ścieżka dyplomowania								Profil kształcenia	
Nazwa przedmiotu	Testowanie manualne							Kod przedmiotu	SPT004
								Rodzaj przedmiotu	obowiązkowy
Formy zajęć i liczba godzin	W	C	L	P	Ps	T	S	Semestr	1
	6				18			Punkty ECTS	3
Przedmioty wprowadzające									
Cele przedmiotu	Celem przedmiotu jest zapoznanie uczestnika z podstawami testowania, czyli zagadnieniami testów manualnych oraz nauczenie go wykorzystania w pracy wybranych narzędzi używanych na różnych etapach procesu testowania.								
Treści programowe	<p>Wykład:</p> <ol style="list-style-type: none"> 1. Czym są testy manualne? - omówienie podstaw testowania oprogramowania. 2. Wprowadzenie - omówienie pracy i niezbędnych umiejętności testera oprogramowania: <ol style="list-style-type: none"> a. Modele wytwarzania oprogramowania, b. Poziomy i typy testów, c. Cykl życia 'buga', d. Zasady i techniki testowania. 3. Testowanie statyczne i dynamiczne. 4. Omówienie certyfikacji testerskiej. <p>Pracownia specjalistyczna:</p> <ol style="list-style-type: none"> 1. Praca z najczęściej wykorzystywanymi narzędziami testera. 2. Ćwiczenia z projektowania i opisywania przypadków i scenariuszy testowych. 3. Sposoby opisywania i raportowania znalezionych defektów. 4. Testowanie manualne oraz raportowanie błędów w praktyce. 5. Zarządzanie procesem testowym, planowanie testów. 								
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: zadania realizowane podczas zajęć								
Symbol efektu uczenia się	Zakładane efekty uczenia się							Odniesienie do kierunkowych efektów uczenia się	
EU1	Zna teorię dotyczącą testów manualnych oraz praktyk testowania manualnego.							SQA_W01, SQA_W02	
EU2	Posiada wiedzę na temat podstawowych narzędzi pracy testera manualnego.							SQA_W02, SQA_W04	
EU3	Umie tworzyć scenariusze testowe.							SQA_U04, SQA_U05	
EU4	Potrafi prawidłowo raportować błędy oraz przeprowadzać ich późniejszą weryfikację.							SQA_U01, SQA_U02, SQA_K01	

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	Zadania realizowane podczas zajęć i prac domowych	Ps	
EU4	Zadania realizowane podczas zajęć i prac domowych	Ps	
Bilans nakładu pracy uczestnika (w godzinach)		Liczba godz.	
Wyliczenie	Udział w wykładach	6	
	Udział w pracowni specjalistycznej	18	
	Przygotowanie do pracowni specjalistycznej	20	
	Udział w konsultacjach związanych z pracownią	5	
	Realizacja zadań projektowych i zadań domowych	26	
	RAZEM:	75	
Wskaźniki ilościowe		GODZINY	ECTS
Nakład pracy uczestnika związany z zajęciami wymagającymi bezpośredniego udziału nauczyciela		29	1,16
Nakład pracy uczestnika związany z zajęciami o charakterze praktycznym		69	2,76
Literatura podstawowa	<ol style="list-style-type: none"> 1. Bogdan Bereza-Jarociński, Bolesław Szomański, Inżynieria oprogramowania : jak zapewnić jakość tworzonym aplikacjom, Gliwice : Helion, 2009; 2. Alistair Cockburn, Jak pisać efektywne przypadki użycia, Warszawa: Wydawnictwa Naukowo-Techniczne, 2004; 3. Dean Leffingwell, Don Widrig, Zarządzanie wymaganiami, Warszawa: Wydawnictwa Naukowo-Techniczne, 2003 ; 		
Literatura uzupełniająca	<ol style="list-style-type: none"> 1. Michael E Bays, Metodyka wprowadzania oprogramowania na rynek, Warszawa: Wydawnictwa Naukowo-Techniczne, 2001; 2. Karolina Zmitrowicz, Jakość projektów informatycznych: rozwój i testowanie oprogramowania, Gliwice: Helion, 2015; 3. Adam Roman, Testowanie i jakość oprogramowania: modele, techniki, narzędzia, Warszawa : Wydaw. Naukowe PWN, 2015; 		
Jednostka realizująca	Wydział Informatyki Politechniki Białostockiej	Data opracowania programu	
Program opracował	<i>mgr Bartosz Andrejczuk, dr inż. Paweł Tadejko</i>	18.05.2021	

Wydział Informatyki									
Kierunek studiów	Tester Oprogramowania							Poziom i forma studiów	Studia podyplomowe
Specjalność / ścieżka dyplomowania								Profil kształcenia	
Nazwa przedmiotu	Wstęp do programowania w JavaScript							Kod przedmiotu	SPT005
								Rodzaj przedmiotu	obowiązkowy
Formy zajęć i liczba godzin	W	C	L	P	Ps	T	S	Semestr	1
	6				18			Punkty ECTS	3
Przedmioty wprowadzające									
Cele przedmiotu	Celem przedmiotu jest zapoznanie uczestnika z podstawami programowania JavaScript w standardzie ES6. Uczestnik nabędzie umiejętność analizowania podstawowych konstrukcji języka oraz nabędzie podstawową wiedzę jak wykorzystać język skryptowy w testach automatycznych.								
Treści programowe	<p>Wykład:</p> <ol style="list-style-type: none"> 1. Składnia oraz zastosowanie języka JavaScript w standardzie ES6. 2. Organizacja kodu i podstawowe wzorce projektowe. 3. Pojęcia programowania obiektowego i cel ich wykorzystania. 4. Wiedza o klasach, instancjach, modułach, abstrakcjach. <p>Pracownia specjalistyczna:</p> <ol style="list-style-type: none"> 1. Implementacja prostych skryptów z wykorzystaniem języka JavaScript. 2. Analizowanie możliwości wykorzystania języków skryptowych do automatyzacji testów. 3. Ćwiczenia z analizy i zrozumienie składni podstawowych. 								
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 zaliczeniowy								
Symbol efektu uczenia się	Zakładane efekty uczenia się							Odniesienie do kierunkowych efektów uczenia się	
EU1	Zna podstawowe konstrukcje języka skryptowego JavaScript.							SQA_W03	
EU2	Posiada wiedzę jak wykorzystać język skryptowy w pracy z narzędziami do automatyzacji testów.							SQA_W03, SQA_W06	
EU3	Umie czytać ze zrozumieniem wybrane konstrukcje podstawowych języków skryptowych, na przykładzie JavaScript.							SQA_U06	
EU4	Potrafi posługiwać się pojęciami pozwalającymi na podstawy komunikacji z programistami i innymi członkami zespołu projektowego.							SQA_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	Realizacja projektu	Ps	
EU4	Realizacja projektu	Ps	
Bilans nakładu pracy uczestnika (w godzinach)		Liczba godz.	
Wyliczenie	Udział w wykładach	6	
	Udział w pracowni specjalistycznej	18	
	Przygotowanie do pracowni specjalistycznej	20	
	Udział w konsultacjach związanych z projektem	5	
	Realizacja zadań projektowych (zadań domowych)	26	
	RAZEM:	75	
Wskaźniki ilościowe		GODZINY	ECTS
Nakład pracy uczestnika związany z zajęciami wymagającymi bezpośredniego udziału nauczyciela		29	1,16
Nakład pracy uczestnika związany z zajęciami o charakterze praktycznym		69	2,76
Literatura podstawowa	<ol style="list-style-type: none"> 1. Kirupa Chinnathambi, JavaScript : przewodnik dla absolutnie początkujących, Gliwice: Helion, 2017; 2. Laura Lemay, Rafe Colburn, Jennifer Kyrmin, HTML, CSS i JavaScript dla każdego, Gliwice: Helion, 2017; 3. JavaScript Tutorial for Beginners: Learn Javascript Step by Step, [18.05.2021], https://www.guru99.com/interactive-javascript-tutorials.html 		
Literatura uzupełniająca	<ol style="list-style-type: none"> 1. Nicholas C Zakas, ECMAScript 6: przewodnik po nowym standardzie języka JavaScript, Gliwice: Helion, 2017; 2. Kurs Javascript dla superbohaterów, [18.05.2021], https://kursjs.pl/ 3. Mirosław Zelent, Kurs JavaScript. Programowanie frontendowe – YouTube, [18.05.2021], https://www.youtube.com/playlist?list=PLOYHgt8dldoxTUYuHS9ZYNIcJq5R3jBsC 		
Jednostka realizująca	Wydział Informatyki Politechniki Białostockiej	Data opracowania programu	
Program opracował	<i>inż. Huber Zub, dr inż. Paweł Tadejko</i>	18.05.2021	

Wydział Informatyki									
Kierunek studiów	Tester Oprogramowania							Poziom i forma studiów	Studia podyplomowe
Specjalność / ścieżka dyplomowania								Profil kształcenia	
Nazwa przedmiotu	Testowanie automatyczne I - Wprowadzenie							Kod przedmiotu	SPT006
								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 uczestnika z podstawowymi zagadnieniami i procesami testowania automatycznego oraz nauczenie go wykorzystania w pracy testera przeglądarki internetowej i użycia selektorów CSS i XPath.								
Treści programowe	<p>Wykład:</p> <ol style="list-style-type: none"> 1. Czym jest testowanie automatyczne. 2. Testowanie automatyczne w procesie WEB development. <p>Pracownia specjalistyczna:</p> <ol style="list-style-type: none"> 1. Wprowadzenie i analiza struktury DOM stron WWW. 2. Poznanie selektorów CSS i Xpath i ich rola w testowaniu automatycznym. 3. Wykorzystanie narzędzi przeglądarkowych do automatyzacji testów. 								
Metody dydaktyczne	1. prezentacja multimedialna; 2. analiza wybranych przypadków; 3. samodzielne opracowanie wskazanych zagadnień; 4. ćwiczenia indywidualne; 5. dyskusje dydaktyczne.								
Forma zaliczenia	<p>Wykład: test przeprowadzony w systemie e-learningowym</p> <p>Pracownia specjalistyczna: projekt zaliczeniowy</p>								
Symbol efektu uczenia się	Zakładane efekty uczenia się							Odniesienie do kierunkowych efektów uczenia się	
EU1	Ma wiedzę na temat sposobów wykorzystania środowiska zintegrowanego Selenium IDE.							SQA_W03, SQA_W04	
EU2	Zna selektory CSS i selektory XPath.							SQA_W06	
EU3	Potrafi posługiwać się selektorami by odfiltrować potrzebne elementy z strony WWW.							SQA_U02, SQA_U06, SQA_K01	
EU4	Potrafi posługiwać się narzędziami developerskimi przeglądarki do analizy selektorów.							SQA_U02, SQA_U03, SQA_U05, SQA_K01	
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	Realizacja projektu	Ps	
EU4	Realizacja projektu	Ps	
Bilans nakładu pracy uczestnika (w godzinach)		Liczba godz.	
Wyliczenie	Udział w wykładach	3	
	Udział w pracowni specjalistycznej	9	
	Przygotowanie do pracowni specjalistycznej	4	
	Udział w konsultacjach związanych z projektem	3	
	Realizacja zadań projektowych (zadań domowych)	6	
	RAZEM:	25	
Wskaźniki ilościowe		GODZINY	ECTS
Nakład pracy uczestnika związany z zajęciami wymagającymi bezpośredniego udziału nauczyciela		15	0,60
Nakład pracy uczestnika związany z zajęciami o charakterze praktycznym		22	0,88
Literatura podstawowa	<ol style="list-style-type: none"> Adam Roman, Testowanie i jakość oprogramowania: modele, techniki, narzędzia, Warszawa : Wydaw. Naukowe PWN, 2015; Dariusz Woźniak, TDD [Test-Driven-Development]: techniki programowania sterowanego testami, Gliwice: Helion, 2018; Prashanth Sams, Selenium: automatyczne testowanie aplikacji, Gliwice: Helion, 2017; Automatyzacja testów funkcjonalnych przy pomocy Cypress.io, [18.05.2021], https://testuj.pl/blog/automatyzacja-testow-funkcjonalnych-przy-pomocy-cypress-io/ 		
Literatura uzupełniająca	<ol style="list-style-type: none"> Tilo Linz, Testowanie w procesie Scrum: przewodnik po zarządzaniu jakością oprogramowania w świecie programowania zwinnego, Warszawa : APN PROMISE, 2014; JavaScript Tutorial for Beginners: Learn Javascript Step by Step, [18.05.2021], https://www.guru99.com/interactive-javascript-tutorials.html Kurs Javascript dla superbohaterów, [18.05.2021], https://kursis.pl/ 		
Jednostka realizująca	Wydział Informatyki Politechniki Białostockiej	Data opracowania programu	
Program opracował(a)	<i>Ernest Czajkowski, dr inż. Paweł Tadejko</i>	18.05.2021	

Wydział Informatyki									
Kierunek studiów	Tester Oprogramowania							Poziom i forma studiów	Studia podyplomowe
Specjalność / ścieżka dyplomowania								Profil kształcenia	
Nazwa przedmiotu	Agile i Scrum - Metodyka pracy zwinnej w zespole							Kod przedmiotu	SPT007
								Rodzaj przedmiotu	obowiązkowy
Formy zajęć i liczba godzin	W	C	L	P	Ps	T	S	Semestr	2
	5				15			Punkty ECTS	3
Przedmioty wprowadzające									
Cele przedmiotu	Celem przedmiotu jest zapoznanie uczestnika z frameworkiem Scrum, manifestem Agile oraz narzędziami do szacowania prac developerskich. Uczestnik nabędzie umiejętność szacowania prac developerskich.								
Treści programowe	<p>Wykład:</p> <ol style="list-style-type: none"> Główne wytyczne frameworka Agile. Role i odpowiedzialność członków zespołu wynikająca z manifestu Agile. Sposoby pracy w zespole. Wprowadzenie do szacowanie prac developerskich. <p>Pracownia specjalistyczna:</p> <ol style="list-style-type: none"> Warsztaty pracy w zespole. Szacowanie prac developerskich. Analiza różnic między różnymi metodami implementacji. Opracowywanie założeń i realizacja wybranych elementów projektu zgodnie z metodyką pracy Agile – artefakty, role, wydarzenia. 								
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: zadania realizowane podczas zajęć								
Symbol efektu uczenia się	Zakładane efekty uczenia się							Odniesienie do kierunkowych efektów uczenia się	
EU1	Zna i rozumie zasady manifestu Agile i frameworka Scrum oraz innych podejść zwinnych: Scrum, XP, Kanban, Lean.							SQA_W02	
EU2	Zna zasady, role i obowiązki członków zespołu developerskiego według frameworka Scrum.							SQA_W02	
EU3	Potrafi określić zasady, koncepcje i podstawy Scruma oraz wyszacować czasochłonność wybranych rodzajów zadań.							SQA_U02, SQA_K02	
EU4	Potrafi zastosować wytyczne Scrum w pracy w projekcie.							SQA_U02, SQA_U04, SQA_K01	
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	Zadania realizowane podczas zajęć i prac domowych	Ps	
EU4	Zadania realizowane podczas zajęć i prac domowych	Ps	
Bilans nakładu pracy uczestnika (w godzinach)		Liczba godz.	
Wyliczenie	Udział w wykładach	5	
	Udział w pracowni specjalistycznej	15	
	Przygotowanie do pracowni specjalistycznej	15	
	Udział w konsultacjach związanych z pracownią	5	
	Realizacja zadań projektowych i zadań domowych	35	
	RAZEM:	75	
Wskaźniki ilościowe		GODZINY	ECTS
Nakład pracy uczestnika związany z zajęciami wymagającymi bezpośredniego udziału nauczyciela		25	1
Nakład pracy uczestnika związany z zajęciami o charakterze praktycznym		70	2,80
Literatura podstawowa	<ol style="list-style-type: none"> 1. Andrew Stellman, Jennifer Greene, Agile: przewodnik po zwinnych metodykach programowania: [poznaj nowoczesne podejście do wytwarzania oprogramowania!], Gliwice: Helion, 2015; 2. Adam Roman, Testowanie i jakość oprogramowania: modele, techniki, narzędzia, Warszawa : Wydaw. Naukowe PWN, 2015; 3. Piotr Wróblewski, Zarządzanie projektami informatycznymi dla praktyków, Gliwice: Helion, 2005; 4. Kenneth S Rubin, Scrum: praktyczny przewodnik po najpopularniejszej metodyce Agile, Gliwice: Helion, 2014; 		
Literatura uzupełniająca	<ol style="list-style-type: none"> 1. Dariusz Woźniak, TDD [Test-Driven-Development]: techniki programowania sterowanego testami, Gliwice: Helion, 2018; 2. Radosław Smilgin, Zawód tester: od decyzji do pierwszych kroków w pracy, Warszawa: Wydaw. Naukowe PWN, 2016; 3. Karolina Zmitrowicz, Jakość projektów informatycznych: rozwój i testowanie oprogramowania, Gliwice: Helion, 2015; 		
Jednostka realizująca	Wydział Informatyki Politechniki Białostockiej	Data opracowania programu	
Program opracował	<i>mgr inż. Milena Sejnowska, dr inż. Paweł Tadejko</i>	18.05.2021	

Wydział Informatyki									
Kierunek studiów	Tester Oprogramowania							Poziom i forma studiów	Studia podyplomowe
Specjalność / ścieżka dyplomowania								Profil kształcenia	
Nazwa przedmiotu	Testowanie Aplikacji Mobilnych							Kod przedmiotu	SPT008
								Rodzaj przedmiotu	obowiązkowy
Formy zajęć i liczba godzin	W	C	L	P	Ps	T	S	Semestr	2
	8				8			Punkty ECTS	2
Przedmioty wprowadzające	Standardy WWW i podstawy ruchu sieciowego, Podstawowe narzędzia developerskie testera, Testowanie manualne								
Cele przedmiotu	Celem przedmiotu jest zapoznanie uczestnika z podstawami testowania, czyli zagadnieniami testów manualnych oraz nauczenie go wykorzystania w pracy wybranych narzędzi wykorzystywanych na różnych etapach procesu testowania.								
Treści programowe	<p>Wykład:</p> <ol style="list-style-type: none"> Historia i omówienie mobilnych systemów operacyjnych oraz możliwości sprzętu na przestrzeni ostatnich lat (podobieństwa i różnice pomiędzy systemami, dotykowość ekranów, etc.). Wytyczna RWD – Responsive Web Design. Responsywność, skalowanie, landscape/portrait, etc. Aplikacje webowe, PWA, aplikacje natywne, online/offline; "sklepy" z aplikacjami i alternatywne źródła instalacji aplikacji do testowania (np. testflight). <p>Pracownia specjalistyczna:</p> <ol style="list-style-type: none"> Accessibility (TalkBack - Android, VoiceOver - iOS). Sposoby na radzenie sobie z dużą liczbą urządzeń mobilnych - automatyzacja pracy manualnej (ghostlab), narzędzia webowe (BrowserStack). Automatyzacja testów aplikacji mobilnej na przykładzie Appium i Selenium. 								
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 zaliczeniowy								
Symbol efektu uczenia się	Zakładane efekty uczenia się							Odniesienie do kierunkowych efektów uczenia się	
EU1	Zna różnice między najważniejszymi mobilnymi systemami operacyjnymi.							SQA_W01, SQA_W02	
EU2	Zna techniki testowania aplikacji mobilnych, w tym np. aplikacji dla osób z zaburzeniami wzroku.							SQA_W02	
EU3	Potrafi dobrać reprezentatywną grupę sprzętową do testów.							SQA_U02	
EU4	Umie zorganizować warsztat pracy manualnej z większym zestawem sprzętowym (ghostlab).							SQA_U02, SQA_U04, SQA_U06	
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	Realizacja projektu	Ps	
EU4	Realizacja projektu	Ps	
Bilans nakładu pracy uczestnika (w godzinach)		Liczba godz.	
Wyliczenie	Udział w wykładach	8	
	Udział w pracowni specjalistycznej	8	
	Przygotowanie do pracowni specjalistycznej	6	
	Udział w konsultacjach związanych z projektem	5	
	Realizacja zadań projektowych (zadań domowych)	23	
	RAZEM:	50	
Wskaźniki ilościowe		GODZINY	ECTS
Nakład pracy uczestnika związany z zajęciami wymagającymi bezpośredniego udziału nauczyciela		21	0,84
Nakład pracy uczestnika związany z zajęciami o charakterze praktycznym		42	1,68
Literatura podstawowa	<ol style="list-style-type: none"> 1. Dariusz Woźniak, TDD [Test-Driven-Development]: techniki programowania sterowanego testami, Gliwice: Helion, 2018; 2. John Ferguson Smart, BDD w działaniu: sterowanie zachowaniem w rozwoju aplikacji, Gliwice: Helion, 2016; 3. Andrew Stellman, Jennifer Greene, Agile: przewodnik po zwinnych metodykach programowania: [poznaj nowoczesne podejście do wytwarzania oprogramowania!], Gliwice: Helion, 2015; 4. Testowanie za pomocą narzędzia Cypress - Aplikacje Mobilne dla Firm Webowe i Natywne iOS & Android itCraftApps, [18.05.2021], https://itcraftapps.com/pl/blog/testowanie-cypress/ 		
Literatura uzupełniająca	<ol style="list-style-type: none"> 1. Michał Sobczak, Jakość oprogramowania: podręcznik dla profesjonalistów, Gliwice: Helion, 2020; 2. JavaScript Tutorial for Beginners: Learn Javascript Step by Step, [18.05.2021], https://www.guru99.com/interactive-javascript-tutorials.html 3. Kurs Javascript dla superbohaterów, [18.05.2021], https://kursis.pl/ 		
Jednostka realizująca	Wydział Informatyki Politechniki Białostockiej	Data opracowania programu	
Program opracował	<i>mgr Bartosz Andrejczuk, dr inż. Paweł Tadejko</i>	18.05.2021	

Wydział Informatyki									
Kierunek studiów	Tester Oprogramowania							Poziom i forma studiów	Studia podyplomowe
Specjalność / ścieżka dyplomowania								Profil kształcenia	
Nazwa przedmiotu	Programowanie obiektowe w JavaScript							Kod przedmiotu	SPT009
								Rodzaj przedmiotu	obowiązkowy
Formy zajęć i liczba godzin	W	C	L	P	Ps	T	S	Semestr	2
	3				12			Punkty ECTS	3
Przedmioty wprowadzające	Wstęp do programowania w JavaScript								
Cele przedmiotu	Celem przedmiotu jest zapoznanie uczestnika z wybranymi zasadami programowania obiektowego JavaScript. Uczestnik nabyte umiejętności używania podstawowych elementów języka, planowania i tworzenia kodu oraz dobierania wzorców do danego problemu.								
Treści programowe	<p>Wykład:</p> <ol style="list-style-type: none"> 1. Wiedza o klasach, instancjach, modułach, abstrakcjach. 2. Składnia oraz zastosowanie języka JavaScript. <p>Pracownia specjalistyczna:</p> <ol style="list-style-type: none"> 1. Implementacja prostych założeń programowania obiektowego i ich praktyczne wykorzystanie w skryptach. 2. Zastosowanie języka JavaScript w standardzie ES6. 3. Projekt i implementacja prostej klasy w JavaScript. 4. Pojęcie Funkcji. Organizacja kodu 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: projekt zaliczeniowy								
Symbol efektu uczenia się	Zakładane efekty uczenia się							Odniesienie do kierunkowych efektów uczenia się	
EU1	Zna podstawowe pojęcia programowania obiektowego.							SQA_W03, SQA_W06	
EU2	Wie jak wyglądają najprostsze konstrukcje języka skryptowego.							SQA_W03, SQA_W06	
EU3	Posiada umiejętność czytania i rozumienia składni podstawowych języków skryptowych i wykorzystania frameworków testowania automatycznego.							SQA_U03, SQA_U06	
EU4	Umie komunikować się z programistami za pomocą podstawowych pojęć związanych z programowaniem obiektowym.							SQA_U03, SQA_U06	
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	Realizacja projektu	Ps	
EU4	Realizacja projektu	Ps	
Bilans nakładu pracy uczestnika (w godzinach)		Liczba godz.	
Wyliczenie	Udział w wykładach	3	
	Udział w pracowni specjalistycznej	12	
	Przygotowanie do pracowni specjalistycznej	20	
	Udział w konsultacjach związanych z projektem	5	
	Realizacja zadań projektowych (zadań domowych)	35	
	RAZEM:	75	
Wskaźniki ilościowe		GODZINY	ECTS
Nakład pracy uczestnika związany z zajęciami wymagającymi bezpośredniego udziału nauczyciela		20	0,80
Nakład pracy uczestnika związany z zajęciami o charakterze praktycznym		72	2,88
Literatura podstawowa	<ol style="list-style-type: none"> 1. Nicolás Bevacqua, Nowoczesny JavaScript : poznaj ES6 i praktyczne zastosowania nowych rozwiązań, Gliwice: Helion, 2018; 2. Laura Lemay, Rafe Colburn, Jennifer Kyrnin, HTML, CSS i JavaScript dla każdego, Gliwice: Helion, 2017; 3. Ved Antani, Stoyan Stefanov, Programowanie zorientowane obiektowo w języku JavaScript, Gliwice: Helion, 2018; 		
Literatura uzupełniająca	<ol style="list-style-type: none"> 1. Nicholas C Zakas, ECMAScript 6: przewodnik po nowym standardzie języka JavaScript, Gliwice: Helion, 2017; 2. JavaScript Tutorial for Beginners: Learn Javascript Step by Step, [18.05.2021], https://www.guru99.com/interactive-javascript-tutorials.html 3. Kurs Javascript dla superbohaterów, [18.05.2021], https://kursjs.pl/ 		
Jednostka realizująca	Wydział Informatyki Politechniki Białostockiej	Data opracowania programu	
Program opracował	<i>inż. Hubert Zub, dr inż. Paweł Tadejko</i>	18.05.2021	

Wydział Informatyki									
Kierunek studiów	Tester Oprogramowania							Poziom i forma studiów	Studia podyplomowe
Specjalność / ścieżka dyplomowania								Profil kształcenia	
Nazwa przedmiotu	Testowanie automatyczne II - Narzędzia							Kod przedmiotu	SPTO10
								Rodzaj przedmiotu	obowiązkowy
Formy zajęć i liczba godzin	W	C	L	P	Ps	T	S	Semestr	2
	6				18			Punkty ECTS	4
Przedmioty wprowadzające	Standardy WWW i podstawy ruchu sieciowego, Podstawowe narzędzia developerskie testera, Wstęp do programowania w JavaScript, Testowanie automatyczne I - Wprowadzenie								
Cele przedmiotu	Celem przedmiotu jest poszerzenie wiedzy i umiejętności uczestnika w obszarze testowania automatycznego. Uczestnik dowie się i nauczy jak wykorzystać w pracy testera wybrane narzędzia i środowisko zintegrowane używane na różnych etapach procesu testowania.								
Treści programowe	<p>Wykład:</p> <ol style="list-style-type: none"> Pojęcia automatyzacji procesu testowania i scenariuszy testowych. Framework Cypress i jego charakterystyka. Analiza możliwości Cypress Dashboard. Budowanie ścieżek testowych w testach automatycznych. <p>Pracownia specjalistyczna:</p> <ol style="list-style-type: none"> Poznanie zasad używania i metodyki pracy z frameworkiem Cypress. Warsztaty praktyczne z wykorzystaniem narzędzia Cypress Dashboard. Tworzenie skryptów testowych z użyciem Cypress - ćwiczenia praktyczne. Realizacja ćwiczeń z automatyzacji scenariuszy testowych. Wykorzystywanie zdobytej wiedzy w kompleksowym procesie budowania ścieżek testowych. Utrzymywanie środowiska i procesu testów automatycznych. 								
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	Posiada wiedzę nt. działania frameworka Cypress.							SQA_W03, SQA_W06	
EU2	Posiada wiedzę nt. mechanizmów działania i sposobu pracy w środowisku Cypress Dashboard.							SQA_W05, SQA_W06	
EU3	Potrafi posługiwać się narzędziem Cypress Dashboard do automatyzacji scenariuszy testowych.							SQA_U03, SQA_U06, SQA_K01	
EU4	Potrafi realizować działania związane z utrzymaniem środowiska testów automatycznych.							SQA_U03, SQA_K02	
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	Realizacja projektu	Ps	
EU4	Realizacja projektu	Ps	
Bilans nakładu pracy uczestnika (w godzinach)		Liczba godz.	
Wyczenie	Udział w wykładach	6	
	Udział w pracowni specjalistycznej	18	
	Przygotowanie do pracowni specjalistycznej	20	
	Udział w konsultacjach związanych z projektem	5	
	Realizacja zadań projektowych (zadań domowych)	51	
	RAZEM:	100	
Wskaźniki ilościowe		GODZINY	ECTS
Nakład pracy uczestnika związany z zajęciami wymagającymi bezpośredniego udziału nauczyciela		29	1,16
Nakład pracy uczestnika związany z zajęciami o charakterze praktycznym		94	3,76
Literatura podstawowa	<ol style="list-style-type: none"> 1. Dariusz Woźniak, TDD [Test-Driven-Development]: techniki programowania sterowanego testami, Gliwice: Helion, 2018; 2. Andrew Stellman, Jennifer Greene, Agile: przewodnik po zwinnych metodykach programowania: [poznaj nowoczesne podejście do wytwarzania oprogramowania!], Gliwice: Helion, 2015; 3. Prashanth Sams, Selenium: automatyczne testowanie aplikacji, Gliwice: Helion, 2017; 4. Automatyzacja testów funkcjonalnych przy pomocy Cypress.io, [18.05.2021], https://testuj.pl/blog/automatyzacja-testow-funkcjonalnych-przy-pomocy-cypress-io/ 		
Literatura uzupełniająca	<ol style="list-style-type: none"> 1. Piotr Wróblewski, Zarządzanie projektami informatycznymi dla praktyków, Gliwice: Helion, 2005; 2. Michał Sobczak, Jakość oprogramowania: podręcznik dla profesjonalistów, Gliwice: Helion, 2020; 3. JavaScript Tutorial for Beginners: Learn Javascript Step by Step, [18.05.2021], https://www.guru99.com/interactive-javascript-tutorials.html 4. Kurs Javascript dla superbohaterów, [18.05.2021], https://kursjs.pl/ 		
Jednostka realizująca	Wydział Informatyki Politechniki Białostockiej	Data opracowania programu	
Program opracował	<i>Ernest Czajkowski, dr inż. Paweł Tadejko</i>	18.05.2021	

Wydział Informatyki										
Kierunek studiów	Tester Oprogramowania							Poziom i forma studiów	Studia podyplomowe	
Specjalność / ścieżka dyplomowania								Profil kształcenia		
Nazwa przedmiotu	Podstawy SQL w pracy testera aplikacji							Kod przedmiotu	SPTO11	
								Rodzaj przedmiotu	obowiązkowy	
Formy zajęć i liczba godzin	W	C	L	P	Ps	T	S	Semestr	2	
	5				18			Punkty ECTS	3	
Przedmioty wprowadzające	Wprowadzenie do testowania oprogramowania i technologii ICT									
Cele przedmiotu	Celem przedmiotu jest zapoznanie uczestnika z podstawami języka SQL, dzięki którym zdobędzie umiejętności w zakresie budowy i wykorzystania wybranych zapytań SQL w realizacji zadań i procesach testowania oprogramowania.									
Treści programowe	<p>Wykład:</p> <ol style="list-style-type: none"> 1. Bazy danych. Co to baza danych? - typy bazy danych, podstawy relacyjnej bazy danych. 2. System zarządzania bazą danych - RDBMS Architektura client-server na przykładzie PostgreSQL. 3. Język zapytań baz danych - SQL na przykładzie PostgreSQL. 4. Rodzaje operacji języka SQL – operacje na danych (DML) i definiowanie struktur (DDL). <p>Pracownia specjalistyczna:</p> <ol style="list-style-type: none"> 1. DML - zapytania SELECT (projekcje, filtrowanie, łączenie, agregowanie, agregacje, aliasy). 2. DML - operacje na danych (INSERT, UPDATE, DELETE); Pojęcie i mechanizm transakcji. 3. DDL – operacje na strukturze bazy danych (CREATE). 4. Operacje związane z realizacją transakcji (BEGIN TRANSACTION, COMMIT, ROLLBACK); Uruchamianie procedur (EXECUTE); Operacje tworzenia kopii zapasowej i odzyskiwanie bazy danych - backup i restore. 5. Zapytania i podzapytania złożone. 6. Tworzenie skryptów do generowania danych testowych. 									
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: zadania realizowane podczas zajęć									
Symbol efektu uczenia się	Zakładane efekty uczenia się							Odniesienie do kierunkowych efektów uczenia się		
EU1	Zna podstawowe pojęcia relacyjnych baz danych i języka SQL.							SQA_W03, SQA_W05		
EU2	Posiada wiedzę nt. technologii client-server, mechanizmów działania transakcji i typów operacji SQL.							SQA_W03, SQA_W05		
EU3	Potrafi realizować wybrane typy zapytań SQL do przeglądania danych, także z wykorzystaniem zapytań złożonych: subquery, union oraz tworzyć proste skrypty generowania i ładowania danych.							SQA_U05, SQA_U06		
EU4	Potrafi wykonywać wybrane operacje zmiany danych za pomocą SQL,							SQA_U05, SQA_U06		

	wykonywać proste operacje administracyjne oraz uruchamiać procedury.		
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	Zadania realizowane podczas zajęć	Ps	
EU4	Zadania realizowane podczas zajęć	Ps	
Bilans nakładu pracy uczestnika (w godzinach)		Liczba godz.	
Wyliczenie	Udział w wykładach	5	
	Udział w pracowni specjalistycznej	18	
	Przygotowanie do pracowni specjalistycznej	15	
	Udział w konsultacjach	5	
	Realizacja zadań projektowych (zadań domowych)	32	
	RAZEM:	75	
Wskaźniki ilościowe		GODZINY	ECTS
Nakład pracy uczestnika związany z zajęciami wymagającymi bezpośredniego udziału nauczyciela		28	1,12
Nakład pracy uczestnika związany z zajęciami o charakterze praktycznym		70	2,80
Literatura podstawowa	<ol style="list-style-type: none"> 1. Krystyna Czapla, Bazy danych : podstawy projektowania i języka SQL, Gliwice: Helion, 2015; 2. Itzik Ben-Gan, Podstawy języka T-SQL : Microsoft SQL Server 2016 i Azure SQL Database, Warszawa: APN Promise, 2016; 3. Mariusz Rogulski, Bazy danych dla studentów : [podstawy projektowania i języka SQL], Warszawa: Witkom Salma Press, 2012; 		
Literatura uzupełniająca	<ol style="list-style-type: none"> 1. Kurs SQL - Samouczek Programisty, [18.05.2021], https://www.samouczekprogramisty.pl/kurs-sql/ 2. Kurs SQL online dla początkujących, [18.05.2021], https://www.sqlpedia.pl/kurs-sql/ 3. Michał Sobczak, Jakość oprogramowania: podręcznik dla profesjonalistów, Gliwice: Helion, 2020; 		
Jednostka realizująca	Wydział Informatyki Politechniki Białostockiej	Data opracowania programu	
Program opracował	<i>mgr inż. Jacek Wojtkowski, dr inż. Paweł Tadejko</i>	18.05.2021	

ZASOBY BIBLIOTECZNE ORAZ ELEKTRONICZNE ZASOBY WIEDZY OBEJMUJĄCE LITERATURĘ ZALECANĄ NA STUDIACH PODYPLOMOWYCH

Biblioteka Politechniki Białostockiej (PB) zapewnia dostęp do zasobów bibliotecznych oraz elektronicznych zasobów wiedzy obejmujących literaturę zalecaną na Studiach Podyplomowych „Tester Oprogramowania” oraz wielu innych dziedzin pokrewnych.

Biblioteka PB jest największą biblioteką naukowo-techniczną w regionie północno-wschodnim Polski. Stanowi podstawę systemu biblioteczno-informacyjnego uczelni, w skład którego wchodzi: Biblioteka Główna, Biblioteka Wydziału Architektury, Biblioteka Wydziału Inżynierii Zarządzania, Biblioteka Instytutu Nauk Leśnych Wydziału Budownictwa i Nauk o Środowisku. Zadaniem Biblioteki PB jest przede wszystkim zaspokajanie potrzeb wszystkich pracowników i studentów w zakresie dostępu do literatury naukowej i dydaktycznej.

Biblioteki specjalistyczne obsługują zaś poszczególne wydziały Uczelni, gromadząc i udostępniając księgozbiór ściśle związany z ich potrzebami.

Zasoby biblioteczne obejmujące literaturę zalecaną na kierunku studiów „Tester Oprogramowania”, do których uczelnia zapewnia dostęp obejmują:

1. Ian Sommerville, Inżynieria oprogramowania, Warszawa: Wydaw. Naukowe PWN, 2020;
2. Bogumiła Hnatkowska Red., Zbigniew Huzar, Inżynieria oprogramowania: metody wytwarzania i wybrane zastosowania, Warszawa: Wydaw. Naukowe PWN, 2008;
3. Karolina Zmitrowicz, Jakość projektów informatycznych: rozwój i testowanie oprogramowania, Gliwice: Helion, 2015;
4. Adam Roman Red.; Karolina Zmitrowicz; Wojciech Anzel; Maciej Chmielarz; Aleksandra Kornecka; Natalia Krawczyk; Ewa Marchewka; Rafał Nazwalski; Adam Romanowicz Pirek; Bartłomiej Prędko Sasin Stankiewicz, Testowanie oprogramowania w praktyce: studium przypadków 2.0, Warszawa: Wydaw. Naukowe PWN, 2018;
5. Adam Roman, Testowanie i jakość oprogramowania: modele, techniki, narzędzia, Warszawa : Wydaw. Naukowe PWN, 2015;
6. Radosław Smilgin, Zawód tester: od decyzji do pierwszych kroków w pracy, Warszawa: Wydaw. Naukowe PWN, 2016;
7. Bogdan Bereza-Jarociński, Bolesław Szomański, Inżynieria oprogramowania : jak zapewnić jakość tworzonemu aplikacjom, Gliwice : Helion, 2009;
8. Alistair Cockburn, Jak pisać efektywne przypadki użycia, Warszawa: Wydawnictwa Naukowo-Techniczne, 2004;
9. Dean Leffingwell, Don Widrig, Zarządzanie wymaganiami, Warszawa: Wydawnictwa Naukowo-Techniczne, 2003 ;
10. Michael E Bays, Metodyka wprowadzania oprogramowania na rynek, Warszawa: Wydawnictwa Naukowo-Techniczne, 2001;
11. Mikołaj Jacek Łuczak, Komunikacja w społeczeństwie sieci: technologia, bezpieczeństwo i zmiana społeczna, Poznań : Wydaw. Naukowe SILVA RERUM, 2019;
12. Paco Hope, Ben Walther, Testowanie bezpieczeństwa aplikacji internetowych: receptury, Gliwice: Helion, 2010;
13. Russell Bradford, Podstawy sieci komputerowych, Warszawa: Wydaw. Komunikacji i Łączności, 2009
14. Kirupa Chinnathambi, JavaScript : przewodnik dla absolutnie początkujących, Gliwice: Helion, 2017;
15. Laura Lemay, Rafe Colburn, Jennifer Kyrnin, HTML, CSS i JavaScript dla każdego, Gliwice: Helion, 2017;
16. Nicholas C Zakas, ECMAScript 6: przewodnik po nowym standardzie języka JavaScript, Gliwice: Helion, 2017;
17. Dariusz Woźniak, TDD [Test-Driven-Development]: techniki programowania sterowanego testami, Gliwice: Helion, 2018;
18. Prashanth Sams, Selenium: automatyczne testowanie aplikacji, Gliwice: Helion, 2017;
19. Tilo Linz, Testowanie w procesie Scrum: przewodnik po zarządzaniu jakością oprogramowania w świecie programowania zwinnego, Warszawa : APN PROMISE, 2014;
20. Andrew Stellman, Jennifer Greene, Agile: przewodnik po zwinnych metodykach programowania: [poznaj nowoczesne podejście do wytwarzania oprogramowania!], Gliwice: Helion, 2015;
21. Piotr Wróblewski, Zarządzanie projektami informatycznymi dla praktyków, Gliwice: Helion, 2005;
22. Kenneth S Rubin, Scrum: praktyczny przewodnik po najpopularniejszej metodyce Agile, Gliwice: Helion, 2014;
23. John Ferguson Smart, BDD w działaniu: sterowanie zachowaniem w rozwoju aplikacji, Gliwice: Helion, 2016;
24. Michał Sobczak, Jakość oprogramowania: podręcznik dla profesjonalistów, Gliwice: Helion, 2020;
25. Nicolás Bevacqua, Nowoczesny JavaScript : poznaj ES6 i praktyczne zastosowania nowych rozwiązań, Gliwice: Helion, 2018;
26. Ved Antani, Stoyan Stefanov, Programowanie zorientowane obiektowo w języku JavaScript, Gliwice: Helion, 2018;
27. Krystyna Czaplą, Bazy danych : podstawy projektowania i języka SQL, Gliwice: Helion, 2015;
28. Itzik Ben-Gan, Podstawy języka T-SQL : Microsoft SQL Server 2016 i Azure SQL Database, Warszawa: APN Promise, 2016;
29. Mariusz Rogulski, Bazy danych dla studentów : [podstawy projektowania i języka SQL], Warszawa: Witkom Salma Press, 2012;

Elektroniczne zasoby wiedzy

Wybór źródeł elektronicznych został zweryfikowany i jest wykorzystywany tylko w sytuacjach, kiedy nie istnieje literatura książkowa w języku polskim w danym obszarze.

1. JavaScript Tutorial for Beginners: Learn Javascript Step by Step, [18.05.2021], <https://www.guru99.com/interactive-javascript-tutorials.html>
2. Kurs Javascript dla superbohaterów, [18.05.2021], <https://kursjs.pl/>
3. Mirosław Zelent, Kurs JavaScript. Programowanie frontendowe – YouTube, [18.05.2021], <https://www.youtube.com/playlist?list=PLOYHgt8dldoxTUYuHS9ZYNlcJq5R3jBsC>
4. Testowanie API - Postman dla początkujących. RST Software, [18.05.2021], <https://rst.software/pl/blog/2018/04/testowanie-api-postman-dla-poczatkujacych/>
5. Testowanie aplikacji REST przy użyciu Postmana - Blog testuj.pl, [18.05.2021], <https://testuj.pl/blog/testowanie-aplikacji-rest-przy-uzyciu-postmana/>
6. Automatyzacja testów funkcjonalnych przy pomocy Cypress.io, [18.05.2021], <https://testuj.pl/blog/automatyzacja-testow-funkcjonalnych-przy-pomocy-cypress-io/>
7. Testowanie za pomocą narzędzia Cypress - Aplikacje Mobilne dla Firm Webowe i Natywne iOS & Android | itCraftApps, [18.05.2021], <https://itcraftapps.com/pl/blog/testowanie-cypress/>
8. Kurs SQL - Samouczek Programisty, [18.05.2021], <https://www.samouczekprogramisty.pl/kurs-sql/>
9. Kurs SQL online dla początkujących, [18.05.2021], <https://www.sqlpedia.pl/kurs-sql/>