

# WROCŁAWSKA WYŻSZA SZKOŁA INFORMATYKI STOSOWANEJ

## KARTA OPISU PRZEDMIOTU

<b>Wydział</b>		<b>Informatyki</b>	
<b>Kierunek</b>		<b>Informatyka</b>	
<b>Specjalność</b>		<b>Programowanie</b>	
<b>Semestr</b>	<b>V</b>	<b>Program studiów, dla którego obowiązuje sylabus</b>	<b>2023/2024</b>
<b>Stopień studiów</b>	<b>I</b>		

Nazwa przedmiotu	Tworzenie aplikacji desktopowych			
Kod przedmiotu	TAD			
Łączna liczba godzin	18	Tryb	stacjonarny	niestacjonarny
Profil kształcenia	Ogólnoakademicki (A)		Praktyczny (P)	
Forma zajęć	laboratorium			
Język przedmiotu	polski			
Liczba punktów ECTS	2			

<b>Wymagania wstępne</b>	Podstawowa znajomość języków programowania, takich jak Java lub C++. Umiejętność obsługi komputera i środowisk programistycznych. Znajomość podstawowych algorytmów i struktur danych.
<b>Założenia i cele przedmiotu</b>	Przedmiot ma na celu zaznajomienie studentów z procesem tworzenia aplikacji desktopowych od wyboru platform i środowiska, przez projektowanie intuicyjnych interfejsów użytkownika, integrację z systemem operacyjnym, aż po implementację logiki biznesowej i testowanie. Studenci poznają techniki debugowania, testowania oraz optymalizacji wydajności aplikacji działających w środowisku lokalnym.
<b>Metody dydaktyczne</b>	<ol style="list-style-type: none"> <li>1. Prezentacje multimedialne.</li> <li>2. Pokazy przykładowych rozwiązań problemów.</li> <li>3. Rozwiązywanie zadań praktycznych.</li> </ol>

<b>Efekty uczenia się (odniesienie do charakterystyk poziomów Polskiej Ramy Kwalifikacji)</b>		<b>Odniesienie do efektów dla kierunku</b>	<b>Odniesienie do efektów uczenia się wg Polskiej Ramy Kwalifikacji</b>
<b>WIEDZA</b> – absolwent zna i rozumie:	W01. Różnice między platformami i środowiskami wykorzystywanymi do tworzenia aplikacji desktopowych (np. .NET, JavaFX, Qt).	K_W02 K_W06 K_W07 K_W10 K_W13 K_W20	P6S_WG P6S_WG_INŻ
	W02. Zasady projektowania czytelnych i użytecznych interfejsów graficznych, w tym znaczenie UX/UI w aplikacjach desktopowych.		
	W03. Mechanizmy integracji aplikacji z systemem operacyjnym, w tym operacje na plikach, wywoływanie procesów i powiadomienia systemowe.		
	W04. Wzorce projektowe i paradygmaty programowania stosowane w implementacji logiki biznesowej aplikacji desktopowych.		

## WROCŁAWSKA WYŻSZA SZKOŁA INFORMATYKI STOSOWANEJ

	W05. Podstawy testowania aplikacji desktopowych, w tym testy jednostkowe, integracyjne i metody debugowania.		
UMIEJĘTNOŚCI – absolwent potrafi:	<p>U01. Tworzyć proste aplikacje desktopowe w wybranym języku programowania i środowisku.</p> <p>U02. Zaprojektować i zaimplementować interfejs graficzny, korzystając z komponentów UI oraz dostosować go do potrzeb użytkowników.</p> <p>U03. Integrować aplikację z funkcjami systemu operacyjnego, takimi jak zarządzanie plikami czy wywoływanie procesów.</p> <p>U04. Testować i debugować aplikacje desktopowe, wykrywając i usuwając błędy na etapie rozwoju.</p> <p>U05. Porównać różne rozwiązania projektowe pod kątem wydajności, stabilności i łatwości utrzymania, a następnie zastosować najlepsze praktyki.</p>	<p>K_U01 K_U02 K_U03 K_U04 K_U08 K_U09 K_U13 K_U15 K_U21 K_U23</p>	<p>P6S_UW P6S_UW_INŻ P6S_UO P6S_KK P6S_UK</p>
KOMPETENCJE SPOŁECZNE – absolwent jest gotów do	<p>K01. Pracy w zespole, przyjmując w nim różne role.</p> <p>K02. Krytycznej oceny możliwości urządzeń oprogramowania i systemów dostępnych na rynku IT.</p> <p>K03. Ciągłego samokształcenia się w celu dostosowywania się do dynamicznie zmieniających się technologii.</p>	<p>K_K04 K_K05 K_K06</p>	<p>P6S_UO P6S_KR P6S_KK</p>

Lp.	Tematyka zajęć	Liczba godzin
1	Wprowadzenie do aplikacji desktopowych. Platformy i środowiska.	3
2	Projektowanie interfejsu użytkownika. Tworzenie intuicyjnych i estetycznych interfejsów dla aplikacji desktopowych.	3
3	Integracja z systemem operacyjnym. Dostęp do plików, zarządzanie procesami, powiadomienia.	3
4	Tworzenie funkcjonalności aplikacji. Implementacja logiki biznesowej, obsługa zdarzeń.	6
5	Testowanie i debugowanie aplikacji. Techniki testowania jednostkowego, integracyjnego i narzędzia do debugowania. Zaliczenie.	3

<b>Forma i warunki zaliczenia przedmiotu</b>	Wykonanie projektów. Częstkowe prezentacje, zdawanie raportów, obrona projektów.	
<b>Metody weryfikacji efektów uczenia się</b>		<b>Nr efektu uczenia się z sylabusu</b>

## WROCŁAWSKA WYŻSZA SZKOŁA INFORMATYKI STOSOWANEJ

<b>Forma i warunki zaliczenia przedmiotu</b>	Wykonanie projektów. Częstkowe prezentacje, zdawanie raportów, obrona projektów.	
	Ocena projektów i częściowych prezentacji.	W01-W05, U01-U05, K01-K03

<b>Literatura podstawowa</b>	<ol style="list-style-type: none"> <li>1. R. C. Martin, <i>Czysty kod. Podręcznik dobrego programisty</i>, Helion, Gliwice 2010.</li> <li>2. J. Roszkowski, <i>Analiza i projektowanie strukturalne</i>, Helion, Gliwice, 2004.</li> <li>3. N. Wirth, <i>Algorytmy + struktury danych = programy</i>, WNT, Warszawa 2002.</li> <li>4. A. Roman, <i>Testowanie i jakość oprogramowania. Modele, techniki, narzędzia</i>, Wydawnictwo Naukowe PWN, Warszawa 2015.</li> </ol>
<b>Literatura uzupełniająca</b>	<ol style="list-style-type: none"> <li>1. S. Prata, <i>Język C++. Szkoła programowania</i>. Wydanie VI, Helion, Gliwice 2019.</li> <li>2. B. Eckel, <i>Thinking in Java. Edycja polska</i>, Helion, Gliwice 2006.</li> </ol>

Nakład pracy studenta	
	Liczba godzin
Zajęcia dydaktyczne	18
Przygotowanie się do zajęć	9
Studiowanie literatury	9
Udział w konsultacjach	2
Przygotowanie projektu / eseju / prezentacji itp.	22
Przygotowanie się do egzaminu / zaliczenia	-
Inne	-
<b>ŁĄCZNY nakład pracy studenta w godz.</b>	<b>60</b>
<b>Liczba punktów ECTS</b>	<b>2</b>