

AKADEMIA TECHNICZNO-INFORMATYCZNA W NAUKACH STOSOWANYCH

KARTA OPISU PRZEDMIOTU

Wydział		Informatyki	
Kierunek		Informatyka	
Specjalność		Programowanie	
Semestr	V	Program studiów, dla którego obowiązuje sylabus	2025/2026
Stopień studiów	I		

Nazwa przedmiotu	Tworzenie aplikacji desktopowych			
Kod przedmiotu	TAD			
Łączna liczba godzin	30	Tryb	stacjonarny	niestacjonarny
Profil kształcenia	Ogólnoakademicki (A)		Praktyczny (P)	
Forma zajęć	laboratorium			
Język przedmiotu	polski			
Liczba punktów ECTS	2			

Prowadzący zajęcia	
Forma prowadzonych zajęć	Laboratorium
Wymiar zajęć	30 h
Stopień (tytuł) naukowy	
Imię	
Nazwisko	

Wymagania wstępne	Podstawowa znajomość języków programowania, takich jak Java lub C++. Umiejętność obsługi komputera i środowisk programistycznych. Znajomość podstawowych algorytmów i struktur danych.
Założenia i cele przedmiotu	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.
Metody dydaktyczne	<ol style="list-style-type: none"> 1. Prezentacje multimedialne. 2. Pokazy przykładowych rozwiązań problemów. 3. Rozwiązywanie zadań praktycznych.

Efekty uczenia się (odniesienie do charakterystyk poziomów Polskiej Ramy Kwalifikacji)		Odniesienie do efektów dla kierunku	Odniesienie do efektów uczenia się wg Polskiej Ramy Kwalifikacji
WIEDZA – 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	P6S_WG P6S_WG_INŻ

AKADEMIA TECHNICZNO-INFORMATYCZNA W NAUKACH STOSOWANYCH

	<p>W02. Zasady projektowania czytelnych i użytecznych interfejsów graficznych, w tym znaczenie UX/UI w aplikacjach desktopowych.</p> <p>W03. Mechanizmy integracji aplikacji z systemem operacyjnym, w tym operacje na plikach, wywoływanie procesów i powiadomienia systemowe.</p> <p>W04. Wzorce projektowe i paradygmaty programowania stosowane w implementacji logiki biznesowej aplikacji desktopowych.</p> <p>W05. Podstawy testowania aplikacji desktopowych, w tym testy jednostkowe, integracyjne i metody debugowania.</p>	<p>K_W10 K_W13 K_W20</p>	
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.	6
2	Projektowanie interfejsu użytkownika. Tworzenie intuicyjnych i estetycznych interfejsów dla aplikacji desktopowych.	6
3	Integracja z systemem operacyjnym. Dostęp do plików, zarządzanie procesami, powiadomienia.	6

AKADEMIA TECHNICZNO-INFORMATYCZNA W NAUKACH STOSOWANYCH

4	Tworzenie funkcjonalności aplikacji. Implementacja logiki biznesowej, obsługa zdarzeń.	8
5	Testowanie i debugowanie aplikacji. Techniki testowania jednostkowego, integracyjnego i narzędzia do debugowania. Zaliczenie.	4

Forma i warunki zaliczenia przedmiotu	Wykonanie projektów. Częstkowe prezentacje, zdawanie raportów, obrona projektów.	
Metody weryfikacji efektów uczenia się		Nr efektu uczenia się z sylabusu
	Ocena projektów i częściowych prezentacji.	W01-W05, U01-U05, K01-K03

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

Nakład pracy studenta	
	Liczba godzin
Zajęcia dydaktyczne	30
Przygotowanie się do zajęć	5
Studiowanie literatury	5
Udział w konsultacjach	2
Przygotowanie projektu / eseju / prezentacji itp.	18
Przygotowanie się do egzaminu / zaliczenia	-
Inne	-
ŁĄCZNY nakład pracy studenta w godz.	60
Liczba punktów ECTS	2