

Karta opisu zajęć - Sylabus			
Państwowa Wyższa Szkoła Techniczno-Ekonomiczna im. ks. Bronisława Markiewicza w Jarosławiu			
I. INFORMACJE PODSTAWOWE			
Nazwa zajęć: Programowanie w języku Swift		Cykl kształcenia rozpoczynający się w roku akademickim 2023/2024	
Nazwa kierunku studiów, poziom i profil kształcenia: Informatyka, studia pierwszego stopnia, profil praktyczny			
Język wykładowy: polski		Rodzaj zajęć: zajęcia specjalistyczne	
Rok studiów: 3		Semestr: 5	
Liczba punktów ECTS przypisana zajęciom: 2		Koordynator zajęć Tomasz Lewandowski, dr inż., tomasz.lewandowski@pwste.edu.pl	
Jednostka organizacyjna: Instytut Inżynierii Technicznej			
FORMA PROWADZENIA ZAJĘĆ I LICZBA GODZIN			
Ogólna liczba godzin zajęć dydaktycznych na studiach stacjonarnych i niestacjonarnych z podziałem na formy:			
Studia stacjonarne		Studia niestacjonarne	
Wykład:		Wykład:	
Ćwiczenia:		Ćwiczenia:	
Laboratorium:		Laboratorium:	
Lektorat:		Lektorat:	
Projekt:		Projekt:	
Zajęcia praktyczne:	30	Zajęcia praktyczne:	
Seminarium:		Seminarium:	
Zajęcia terenowe:		Zajęcia terenowe:	
Praktyki zawodowe:		Praktyki zawodowe:	
Inna forma (jaka):		Inna forma (jaka):	
RAZEM:	30	RAZEM:	
II. INFORMACJE SZCZEGÓŁOWE			
Wymagania wstępne i dodatkowe: Podstawy programowania, Programowanie obiektowe, Bazy danych			
Cel (cele) kształcenia dla zajęć: Celem przedmiotu jest dostarczenie studentom umiejętności programowania aplikacji w języku Swift.			
EFEKTY UCZENIA SIĘ OKREŚLONE DLA ZAJĘĆ I ICH ODNIESIENIE DO EFEKTÓW UCZENIA SIĘ OKREŚLONYCH DLA KIERUNKU STUDIÓW			
Efekty uczenia się określone dla zajęć w kategorii wiedza, umiejętności oraz kompetencje społeczne oraz metody weryfikacji efektów uczenia się			
UWAGA: Dzielimy efekty uczenia się określone dla zajęć na kategorie wiedzy, umiejętności i kompetencji społecznych. Określone dla zajęć efekty uczenia się nie muszą obejmować wszystkich trzech kategorii i zależą one od formy zajęć.			

Symbol efektów uczenia się określonego dla zajęć*	Treść efektu uczenia się. Po zakończeniu zajęć i potwierdzeniu osiągnięcia efektów uczenia się, student w kategorii:	Odniesienie do efektów uczenia się określonych dla kierunku studiów (symbol efektów uczenia się)		
Umiejętności - potrafi				
M_01	Określić specyfikację, zaprojektować i wykonać aplikację na zgodnie ze standardami programistycznymi. Wykorzystać paradygmaty programowania obiektowego w praktyce programistycznej.	K_U01, K_U06, K_U09, K_U20		
M_02	Dokonać kompilacji, testowania oraz optymalizacji napisanych aplikacji.	K_U08		
Kompetencji społecznych - jest gotów do				
M_03	dalszego dokształcania się w celu profesjonalnego projektowania aplikacji mobilnych.	K_K01		
UWAGA! Zaleca się, aby w zależności od liczby godzin zajęć, liczba efektów uczenia się zawierała się w przedziale: 3-7, ale są to wartości umowne.				
TREŚCI PROGRAMOWE I ICH ODNIESIENIE DO FORM ZAJĘĆ I METOD OCENIANIA				
Treści programowe (uszczegółowione, zaprezentowane z podziałem na poszczególne formy zajęć, tj. wykład, ćwiczenia, laboratoria, projekty, seminaria i inne):				
Symbol treści programowych	Opis treści programowych	Forma zajęć	Metody dydaktyczne prowadzenia zajęć umożliwiające osiągnięcie założonych efektów uczenia się *	Metody weryfikacji osiągnięcia efektów uczenia się przypisanych do zajęć #
		zajęcia praktyczne		
TP-01	Zapoznanie ze środowiskiem programistycznym. Podstawy tworzenia aplikacji.		objaśnienie, ćwiczenia oparte na wykorzystaniu różnych źródeł wiedzy	projekt, prezentacja
TP-02	Konstrukcje języka Swift, typy danych, funkcje, struktury, klasy.		objaśnienie, ćwiczenia oparte na wykorzystaniu różnych źródeł wiedzy	projekt, prezentacja
TP-03	Wykorzystanie plików, bazy danych i innych danych zewnętrznych.		objaśnienie, ćwiczenia oparte na wykorzystaniu różnych źródeł wiedzy	projekt, prezentacja

Metody weryfikacji osiągnięcia efektów uczenia się określonych dla zajęć, powinny być zróżnicowane w zależności od kategorii, tj. inne dla kategorii wiedza i inne dla kategorii umiejętności i kompetencje społeczne.

Dla wykładu:

* np. wykład podający, wykład problemowy, ćwiczenia oparte na wykorzystaniu różnych źródeł wiedzy

np. egzamin ustny, test, prezentacja, projekt

Zaleca się podanie przykładowych zadań (pytań) służących weryfikacji osiągnięcia efektów uczenia się określonych dla zajęć.

ZALECANA LITERATURA (w tym pozycje w języku obcym)

Literatura podstawowa (powinna być dostępna dla studenta w uczelnianej bibliotece):

1. Dokumentacja internetowa i użytkowa języka Swift
2. Hoffman J.: Swift 4: koduj jak mistrz, Helion 2018
3. Atanasov E.: Poznaj Swifta, tworząc aplikacje: profesjonalne projekty dla systemu iOS, Helion 2019

Literatura uzupełniająca:

1. Pasternak P.: Swift od podstaw. Praktyczny przewodnik, Helion 2017
2. Moon K., Barker C.: Swift Cookbook - Second Edition, Packt Publishing 2021

III. INFORMACJE DODATKOWE

BILANS PUNKTÓW ECTS

OBCIĄŻENIE PRACĄ STUDENTA (godziny)

Forma aktywności	Liczba godzin *
Godziny zajęć (według harmonogramu) z nauczycielem akademickim lub inną osobą prowadzącą zajęcia	30
Praca własna studenta	25
SUMA GODZIN:	55

OBCIĄŻENIE PRACĄ STUDENTA (punkty ECTS)

		Liczba punktów ECTS	
SUMARYCZNA LICZBA PUNKTÓW ECTS PRZYPISANYCH DO ZAJĘĆ	Praca studenta wymagająca bezpośredniego kontaktu z nauczycielem akademickim lub inną osobą prowadzącą zajęcia	Ogółem: 2	1
	Praca własna studenta		1

* godziny lekcyjne, czyli 1 godz. oznacza 45 min;

OPIS PRACY WŁASNEJ STUDENTA:

Praca własna studenta musi być precyzyjnie opisana, uwzględniając charakter praktyczny zajęć. Należy podać symbol efektu uczenia się, którego praca własna dotyczy oraz metody weryfikacji efektów uczenia się stosowane w ramach pracy własnej.

Przykładowe formy aktywności: (1) przygotowanie do zajęć, (2) opracowanie wyników, (3) czytanie wskazanej literatury, (4) napisanie raportu z zajęć, (5) przygotowanie do egzaminu, opracowanie projektu.

M_01 – M_02	przygotowanie do zajęć, opracowanie zadania, projektu	ćwiczenie praktyczne, projekt, prezentacja
M_03	czytanie wskazanej literatury	projekt, prezentacja

KRYTERIA OCENIANIA

Ocena kształtująca:

1. Zrozumienie celu zajęć przez studenta.
2. Raportowanie postępu realizacji ćwiczenia
3. Zaangażowanie studenta w wykonywane ćwiczenie
4. Samoocena i ocena koleżeńska.

Ocena podsumowująca:

Na ocenę dostateczną student wykorzystuje w stopniu zadowalającym wiedzę i umiejętności praktyczne zdobyte w trakcie realizacji modułu do tworzenia aplikacji w języku Swift, realizuje funkcjonalności aplikacji, interfejs użytkownika.

Na ocenę dobrą student samodzielnie wykorzystuje wiedzę i umiejętności praktyczne zdobyte w trakcie realizacji modułu do tworzenia aplikacji w języku Swift, prawidłowo realizuje funkcjonalności aplikacji, interfejs użytkownika, tworzy i wykorzystuje połączenie z bazą danych

Na ocenę bardzo dobrą student samodzielnie wykorzystuje wiedzę i umiejętności praktyczne zdobyte w trakcie realizacji modułu oraz pozyskaną samodzielnie do tworzenia aplikacji w języku Swift, prawidłowo realizuje funkcjonalności aplikacji, interfejs użytkownika, tworzy i wykorzystuje połączenie z bazą danych, wykorzystuje dodatkowe możliwości aplikacji

**INFORMACJA O PRZEWIDYWANEJ MOŻLIWOŚCI WYKORZYSTANIA KSZTAŁCENIA NA
ODLEGŁOŚĆ**

istnieje

.....
(data, podpis Koordynatora
odpowiedzialnego za zajęcia)

.....
(data, podpis Kierownika Zakładu/
Kierownika Jednostki Międzyinstytutowej)

Uwaga:

Karta opisu zajęć (sylabus) musi być dostępna dla studenta.