

Karta opisu zajęć - Sylabus			
Państwowa Wyższa Szkoła Techniczno-Ekonomiczna im. ks. Bronisława Markiewicza w Jarosławiu			
I. INFORMACJE PODSTAWOWE			
Nazwa zajęć: Inżynieria oprogramowania		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 kształcenia kierunkowego	
Rok studiów: II		Semestr: 4	
Liczba punktów ECTS przypisana zajęciom: 4		Koordynator zajęć	
Jednostka organizacyjna: Instytut Inżynierii Technicznej / Zakład Informatyki			
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:	30	Wykład:	
Ćwiczenia:		Ćwiczenia:	
Laboratorium:	30	Laboratorium:	
Lektorat:		Lektorat:	
Projekt:		Projekt:	
Zajęcia praktyczne:		Zajęcia praktyczne:	
Seminarium:		Seminarium:	
Zajęcia terenowe:		Zajęcia terenowe:	
Praktyki zawodowe:		Praktyki zawodowe:	
Inna forma (jaka):		Inna forma (jaka):	
RAZEM:	60	RAZEM:	
II. INFORMACJE SZCZEGÓŁOWE			
Wymagania wstępne i dodatkowe:Znajomość podstawowych zagadnień w zakresie programowania komputerów oraz algorytmiki.			
Cel (cele) kształcenia dla zajęć: Zapoznanie studentów z zasadami organizacji prac i tworzenia oprogramowania. Zapoznanie z metodami, technikami i narzędziami wykorzystywanymi w procesie rozwoju systemów informatycznych. Wyrobienie umiejętności specyfikowania, analizy oraz projektowania rozwiązań informatycznych.			

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)
Wiedzy - zna i rozumie		
C09_01	Potrafi wymienić i omówić podstawowe etapy procesu wytwórczego systemów informatycznych	K_W06, K_W12,
C09_02	Potrafi wskazać na zastosowania i możliwości wykorzystania metod, technik i narzędzi analizy i projektowania systemów informatycznych	K_W06, K_W08
C09_03	Potrafi opracować projekt systemu informatycznego oraz przygotować specyfikację systemową.	K_W06, K_W08 , K_W10,
Umiejętności - potrafi		
C09_04	Potrafi wykorzystać odpowiednie metody, techniki i narzędzia do analizy i specyfikacji wymagań przy projektowaniu i tworzeniu systemów informatycznych	K_U03, K_U12, K_U18, K_U21, K_U13
C09_05	Potrafi wykorzystać odpowiednie oprogramowanie CASE-narzędzia do opracowania projektu systemu informatycznego	K_U09, K_U12
C09_06	Potrafi zaprojektować interfejs użytkownika dla aplikacji	K_U12
C09_07	Posiada umiejętności w zakresie oceny wydajności i kontroli jakości SI.	K_U15
Kompetencji społecznych - jest gotów do		
C05_08	Samodzielnej pracy z planowaniem zadań	K_K03
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 #
		wykład		
TK-01	Pojęcia inżynierii oprogramowania: podstawowe metody i techniki (specyfikowanie, projektowanie i wytwarzanie oprogramowania). Cykl życia oprogramowania, modele cyklu życia (kaskadowy, spiralny, prototypowanie, montaż z gotowych komponentów).		wykład	egzamin
TK-02	Specyfikacja projektu. Określanie wymagań (metody specyfikacji wymagań, wymagania funkcjonalne, budowa diagramu hierarchii funkcji). Tabela wymagań niefunkcjonalnych. Diagram kontekstowy aplikacji.		wykład	egzamin
TK-03	Metody strukturalne analizy i projektowania SO. Diagramy przepływu danych (Data Flow Diagram- DFD). Diagram związków encji (Entity Relationship Diagram (ERD)). Wspomaganie procesu analizy i projektowania z zastosowaniem narzędzi CASE.		wykład	egzamin

TK-04	Obiektowa metodologia analizy i projektowania systemów. Wprowadzenie do języka obiektowego modelowania Unified Modeling Language (UML). Tworzenie obiektowo zorientowanej specyfikacji systemowej z wykorzystaniem języka UML. Analiza funkcjonalna systemu, diagramy przypadków użycia.		wykład	egzamin
TK-05	Język UML. Analiza statyczna. Diagramy klas i obiektów. Model implementacji: diagram komponentów i diagram wdrożeniowy. Modele dynamiczne: diagram stanów, diagram aktywności i diagram interakcji.		wykład	egzamin
TK-06	Implementacja i obiektowe języki programowania. Weryfikacja i walidacja: wymagań i projektu. Testowanie statyczne i dynamiczne. Metody testowania. Jakość oprogramowania.		wykład	egzamin
TK-07	Jakość i bezpieczeństwo systemów oprogramowania. Sposoby przygotowania dokumentacji technicznej i użytkowej. Audyt oprogramowania.		wykład	egzamin
Laboratorium				
TK-08	Zbieranie i dokumentowanie i specyfikacja wymagań wobec systemu oprogramowania. Wymagania funkcjonalne – diagramy hierarchii funkcji.		laboratorium	kontrola realizacji projektu
TK-09	Modelowanie wymagań funkcjonalnych – Diagramy przypadków użycia.		laboratorium	kontrola realizacji projektu

TK-10	Modelowanie struktury statycznej systemu – diagram klas		laboratorium	kontrola realizacji projektu
TK-11	Projektowanie architektury systemu		laboratorium	kontrola realizacji projektu
TK-12	Projektowanie interfejsu użytkownika.		laboratorium	kontrola realizacji projektu
TK-13	Zarządzanie konfiguracją oprogramowania. Testowanie oprogramowania		laboratorium	kontrola realizacji projektu

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. Pressman R., Praktyczne podejście do inżynierii oprogramowania, WNT., 2004
2. Jaskiewicz A., UML. Inżynieria oprogramowania, Helion., 1997
3. Sommerville T., Inżynieria oprogramowania, WNT, Warszawa, 2020

Literatura uzupełniająca:

1. Stevens P., UML. Inżynieria oprogramowania, Helion., 2007
2. Barker R.: Modelowanie związków encji, WNT, Warszawa, 1996, 2005 - wyd. II
3. Fuglewicz P., Stąpor K., Trojnar A.: CASE dla ludzi, LUPUS, Warszawa, 1995
4. Wrycza S., Marcinkowski B., Maślankowski J., UML 2.x. Ćwiczenia zaawansowane, Helion., 2012
5. A. Cockburn, Jak pisać efektywne przypadki użycia, WNT, Warszawa 2004.
6. Yourdon E.: Marsz ku kłęsce : poradnik dla projektanta systemów, WNT, 2007
7. Brooks F.: V Mityczny osobomiesiąc : eseje o inżynierii oprogramowania, WNT, 2000
8. Larman C., UML i wzorce projektowe. Analiza i projektowanie obiektowe oraz iteracyjny model wytwarzania aplikacji, Helion., 2011
9. Subieta K.: Wprowadzenie do inżynierii oprogramowania, Wyd. PJWSTK, Warszawa, 2002

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	60
Praca własna studenta	40
SUMA GODZIN:	100

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:4	2
	Praca własna studenta		2
* 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 symbole 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.			
Przygotowanie do zajęć Opracowanie projektu Przygotowanie do egzaminu			
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:			
1. Na ocenę dostateczną student wykorzystuje w stopniu zadowalającym wiedzę i umiejętności praktyczne zdobyte w trakcie realizacji modułu 2. Na ocenę dobrą student samodzielnie wykorzystuje wiedzę i umiejętności praktyczne zdobyte w trakcie realizacji modułu 3. Na ocenę bardzo dobrą student samodzielnie wykorzystuje wiedzę i umiejętności praktyczne zdobyte w trakcie realizacji modułu oraz pozyskaną samodzielnie			
INFORMACJA O PRZEWIDYWANEJ MOżliWOści WYKORZYSTANIA KSZTAŁCENIA NA ODLEGŁOŚĆ			