

KARTA PRZEDMIOTU DLA NABORU 2021/2022 FORMA: STUDIA STACJONARNE					
INFORMACJE OGÓLNE					
1. Nazwa przedmiotu Inżynieria oprogramowania					
2. Nazwa kierunku Informatyka					
3. Poziom studiów studia pierwszego stopnia					
4. Liczba punktów ECTS 3					
5. Liczba godzin w semestrze					
semestr	w	ćw	lab/lek	prj/zp	prk
VI	30		30		
6. Język wykładowy polski					
7. Wykładowca mgr inż. Piotr Szprychel					
INFORMACJE SZCZEGÓŁOWE					
8. Wymagania wstępne					
1. Podstawowa znajomość programowania (najlepiej obiektowego) zdobyta m.in. na przedmiocie Języki programowania.					
9. Cele przedmiotu					
C1 Zapoznanie studentów z podstawami języka UML.					
C2 Zaznajomienie studentów z podstawowymi zagadnieniami inżynierii oprogramowania: tj. modele procesów wytwarzania oprogramowania oraz zasady ich wyboru w zależności od specyfiki realizowanego przedsięwzięcia, specyfikowanie wymagań dotyczących tworzonego oprogramowania, zasady projektowania oprogramowania, walidacja i testowanie oprogramowania, zarządzanie konfiguracją oprogramowania, ewolucja oprogramowania, wprowadzenie do zarządzania projektem informatycznym, zarządzania ryzykiem projektu itp.					
C3 Pozyskanie praktycznych umiejętności tworzenia projektu systemu informatycznego.					
C4 Zrozumienie przez studentów złożoności procesu produkcji oprogramowania i faktu, że programowanie jest tylko elementem składowym tego procesu.					
C5 Wykształcenie u studentów umiejętności sporządzania i interpretacji dokumentacji projektów systemów informatycznych.					
10. Efekty uczenia się w zakresie wiedzy, umiejętności i kompetencji społecznych					
Student, który zaliczył przedmiot:				odniesienie do kierunkowych efektów uczenia się	
WIEDZA					
EU01	Student zna i umie korzystać z narzędzi do projektowania systemów informatycznych			K_W08 K_W20 K_W21	
EU02	Student zna i umie tworzyć diagramy UML oraz BPMN opisujące praktyczne projekty				
UMIEJĘTNOŚCI					
EU03	Student umie wybierać odpowiednie modele procesów			K_U01	

	wytwarzania oprogramowania w zależności od specyfiki realizowanego projektu	K_U02 K_U03 K_U09 K_U19
EU04	Student umie projektować oprogramowanie i tworzyć odpowiednią dokumentację projektową, właściwie interpretować dokumentację projektów systemów informatycznych	
EU05	Student umie przygotować proces wytwarzania oprogramowania i określać wymagania, walidować i testować oprogramowanie	
KOMPETENCJE SPOŁECZNE		
EU06	rozumie potrzebę ciągłego doskonalenia się	K_K01
11. Treści programowe		
Forma zajęć – wykłady/ ćwiczenia/laboratoria/zajęcia praktyczne itp.		
<p>Wykłady:</p> <ol style="list-style-type: none"> Przedmiot i cele inżynierii oprogramowania Cykl życia oprogramowania Inżynieria wymagań dla systemów informatycznych UML BPMN Modelowanie i projektowanie systemów Implementacja systemu Testowanie, weryfikacja i walidacja oprogramowania Dokumentowanie, instalacja, wdrażanie oraz konserwacja oprogramowania Wiarygodność systemów informatycznych Podsumowanie <p>Laboratorium:</p> <ol style="list-style-type: none"> Zapoznanie z narzędziami do wspomagania inżynierii oprogramowania Przygotowanie zadania projektowego – opis systemu Tworzenie wymagań funkcjonalnych – Burza mózgów Tworzenie wymagań funkcjonalnych dla projektu Tworzenie wymagań нефункциональных dla projektu Uzupełnienie projektu o diagramy UML Uzupełnienie projektu o diagramy BPML Uzupełnienie projektu o Architekturę systemu Uzupełnienie projektu o Metody implementacji systemu Uzupełnienie projektu o Testowanie, weryfikacja i walidacja oprogramowania Uzupełnienie projektu o Dokumentowanie systemu Omówienie projektów Podsumowanie. 		
12. Narzędzia/metody dydaktyczne		
1. Wykład: wykorzystanie prezentacji multimedialnej, filmów szkoleniowych		
2. Laboratorium: pokaz praktyczny, Specjalistyczne oprogramowanie komputerowe		
13. Sposoby oceny (częstkowe, końcowe)		
1. Ocena bieżącego przygotowania do zajęć laboratoryjnych i aktywności w trakcie zajęć - ocenianie ciągłe.		
2. Ocena projektów częściowych wykonywanych na laboratoriach		
3. Egzamin w formie testu z materiału przekazanego na wykładach.		
14. Obciążenie pracą studenta		
	Forma aktywności	liczba godzin

1. Zajęcia z bezpośrednim udziałem nauczyciela oraz konsultacje	65
2. Nakład pracy studenta	10
suma	75
liczba punktów ECTS	3
15. Literatura	
Literatura podstawowa:	
1. Wprowadzenie do inżynierii oprogramowania, K. Subieta, Wydawnictwo PJWSTK, 2002.	
2. Inżynieria oprogramowania, I. Sommerville, WNT, 2003	
3. Nowoczesna inżynieria oprogramowania. Stosowanie skutecznych technik szybszego rozwoju oprogramowania wyższej jakości. D. Farley, Wydawnictwo Helion, 2023	
4. Software Requirements, K. Wiegers, J. Beatty, Microsoft Press, 2013	
Literatura uzupełniająca:	
1. Zarządzanie projektami informatycznymi, M. Flasiński, PWN, 2006	
2. UML w kropelce, M. Fowler, Oficyna Wydawnicza LTP, 2005 (wyd. III, UML wersja 2.0).	
3. Język UML 2.0 w modelowaniu systemów informatycznych, S. Wrycza, B. Marcinkowski, K. Wyrzykowski, Helion 2006.	
16. Formy oceny – szczegóły	
Warunki uzyskania zaliczenia przedmiotu: Wykład kończy się testem z całego materiału. Ocena na podstawie poniższych kryteriów: 5.0 – 97% lub więcej poprawnych odpowiedzi 4.5 – 89% - 96% poprawnych odpowiedzi 4.0 – 79% - 88% poprawnych odpowiedzi 3.5 – 69% - 78% poprawnych odpowiedzi 3.0 – 51% - 68% poprawnych odpowiedzi 2.0 – 50% lub mniej poprawnych odpowiedzi Laboratorium kończy się oceną przekazanego projektu. Ocenie podlega zawartość merytoryczna oraz systematyczność oddawanych rozdziałów projektu.	
17. Inne przydatne informacje o przedmiocie	
1. Bezpośrednich informacji o problematyce zajęć i treściach programowych udziela Prowadzący w trakcie zajęć i podczas konsultacji	
2. Zajęcia odbywać się będą w Akademii Białskiej im. Jana Pawła II	
3. Zajęcia odbywać się będą zgodnie z aktualnym planem zajęć	
4. Konsultacje odbywać się będą zgodnie z obowiązującym terminarzem	