


**KAPITAŁ LUDZKI**  
 NARODOWA STRATEGIA SPÓJNOŚCI

 Projekt współfinansowany przez  
 Unię Europejską w ramach  
 Europejskiego Funduszu  
 Społecznego

**UNIA EUROPEJSKA**  
 EUROPEJSKI  
 FUNDUSZ SPOŁECZNY


<b>Nazwa przedmiotu</b>		<b>Kod ECTS</b>	
Podstawy inżynierii oprogramowania		11.3.1573	
<b>Nazwa jednostki prowadzącej przedmiot</b>			
Instytut Informatyki			
<b>Studia</b>			
<b>wydział</b>	<b>kierunek</b>	<b>poziom</b>	<b>pierwszego stopnia</b>
Wydział Matematyki, Fizyki i Informatyki	Informatyka	<b>forma</b>	niestacjonarne (zaoczne)
		<b>moduł</b>	wszystkie
		<b>specjalnościowy</b>	wszystkie
		<b>specjalizacja</b>	wszystkie
<b>Nazwisko osoby prowadzącej (osób prowadzących)</b>			
dr Adam Kostulak; prof. UG, dr hab. Viktoriia Onyshchenko; prof. UG, dr hab. Bartosz Marcinkowski			
<b>Formy zajęć, sposób ich realizacji i przypisana im liczba godzin</b>		<b>Liczba punktów ECTS</b>	
<b>Formy zajęć</b>		3 Przedmiot w wymiarze 10h wykładu i 10h laboratorium + praca własna studenta	
Wykład, Ćw. laboratoryjne			
<b>Sposób realizacji zajęć</b>			
zajęcia on-line, zajęcia w sali dydaktycznej			
<b>Liczba godzin</b>			
Wykład: 10 godz., Ćw. laboratoryjne: 10 godz.			
<b>Termin realizacji przedmiotu</b>			
2022/2023 letni			
<b>Status przedmiotu</b>		<b>Język wykładowy</b>	
obowiązkowy		polski	
<b>Metody dydaktyczne</b>		<b>Forma i sposób zaliczenia oraz podstawowe kryteria oceny lub wymagania egzaminacyjne</b>	
<ul style="list-style-type: none"> <li>- Dyskusja</li> <li>- Praca w grupach</li> <li>- Rozwiązywanie zadań</li> <li>- Wykład z prezentacją multimedialną</li> </ul>		<b>Sposób zaliczenia</b>	
		<ul style="list-style-type: none"> <li>- Zaliczenie na ocenę</li> <li>- Zaliczenie (zal)</li> </ul>	
		<b>Formy zaliczenia</b>	
		<ul style="list-style-type: none"> <li>- ustalenie oceny zaliczeniowej na podstawie ocen cząstkowych otrzymywanych w trakcie trwania semestru</li> <li>- egzamin pisemny (dłuższa wypowiedź pisemna / rozwiązanie problemu)</li> </ul>	
		<b>Podstawowe kryteria oceny</b>	
		Laboratorium: Ocena zależy od sumy punktów zdobytych za zrealizowane zadania. Wykład: Wynik zaliczenia laboratorium - 50% oceny. Egzamin pisemny lub pisemne zadanie zaliczeniowe - 50% oceny.	
<b>Sposób weryfikacji założonych efektów uczenia się</b>			

zakładany efekt kształcenia	egzamin	kolokwium	projekt	referat	raport	aktywność w dyskusji	obserwacja postawy studenta
<b>Wiedza</b>							
K_W04		X	X				
P_W01			X				
P_W02			X				
P_W03			X				
<b>Umiejętności</b>							
K_U05			X				X
K_U07			X				X
P_U01			X				X
P_U02			X				X
P_U03			X				X
<b>Kompetencje</b>							
K_K02			X				X
							P_K01
		X					X

**Określenie przedmiotów wprowadzających wraz z wymogami wstępnymi****A. Wymagania formalne**

Aktywny udział w zajęciach.

**B. Wymagania wstępne**

Języki programowania, algorytmy i struktury danych, bazy danych.

**Cele kształcenia**

Celem przedmiotu jest zapoznanie studentów z podstawowymi problemami, metodami, technikami i narzędziami produkcji oprogramowania wysokiej jakości. Nacisk jest położony na obiektowe podejście do projektowania i modelowanie systemu w UML.

**Treści programowe**

1. Proces tworzenia Systemów Informatycznych
2. Pojęcie procesu i projektu (budowy) oprogramowania. Modele cyklu życia oprogramowania (model kaskadowy, realizacja kierowana dokumentami, V-model, prototypowanie, model przyrostowy, spiralny)
3. Określanie wymagań i analiza. Rodzaje i role notacji wykorzystywanych w fazie analizy, obiektowe i strukturalne metody analizy.
4. UML 2. – standard modelowania obiektowego: poziomy model i typy diagramów. Przykłady użycia (use case)- elementy diagramów, organizowanie. Model logiczny - diagramy klas, diagramy behawioralne (sekwencji, stanów, aktywności), Model implementacyjny – diagram komponentów (komponenty, podsystemy, porty) i model fizyczny (diagram montażowy).
5. Testowanie i inspekcja. Testowanie ukierunkowane na wyszukiwanie defektów – funkcjonalne, strukturalne (złożoność cyklomatyczna McCabe, pokrycie kodu - ścieżki niezależne, pokrycie danych, testowanie pętli). Testowanie obiektów. Estymacja kosztów. Metoda punktów funkcyjnych, model COCOMO 81 i COCOMO II. Model CMM. Miary niezawodności oprogramowania i techniki programowania dla systemów o dużej niezawodności.

**Wykaz literatury**

1. I. Sommerville: Inżynieria oprogramowania, WNT 2003
2. Szejko St. (red). Metody wytwarzania oprogramowania. MIKOM, 2002.
3. Jaszkievicz A.: Inżynieria oprogramowania. Helion, 1997.
4. Materiały umieszczone na platformie edukacyjnej.

Uzupełniająca:

5. Booch G., Rumbaugh J., Jacobson I.: UML podręcznik użytkownika. WNT, 2001
6. Dumnicki R., Kasprzyk A., Kozłowski M.: Analiza i projektowanie obiektowe. Helion, 1998
7. Eriksson H-E, Penker M.: UML Toolkit. Wiley Computer Publishing, John Wiley & Sons, Inc. 1998.
8. Pressman R.S.: Software Engineering. A Practitioner's Approach. McGraw-Hill, Inc. 1992.

**Kierunkowe efekty uczenia się**

K\_W04: ma uporządkowaną wiedzę w zakresie inżynierii oprogramowania, specyfikacji, walidacji i weryfikacji oprogramowania oraz narzędzi wspomagających proces

**Wiedza**

P\_W01: Student zna różne modele cyklu życia oprogramowania; (K\_W04)  
 P\_W02: Student zna metody wytwarzania oprogramowania; (K\_W04)  
 P\_W03: Student posiada wiedzę odnośnie podstaw wytworzenia dokumentacji,

<p>wytwarzania oprogramowania</p> <p>K_U04: potrafi pracować w zespole informatyków, zarządzać swoim czasem oraz podejmować zobowiązania i dotrzymywać terminy, porozumiewać się przy użyciu różnych technik w tym z wykorzystaniem dedykowanych narzędzi</p> <p>K_U06: potrafi projektować, tworzyć, uruchamiać i testować programy przy wykorzystaniu dedykowanych narzędzi oraz adekwatnych wzorców</p> <p>K_K02: rozumie potrzebę i docenia zalety pracy zespołowej, rozumie konieczność systematycznej pracy nad zespołowymi projektami informatycznymi, jest gotów do aktywnego działania w zespole</p>	<p>analizy wymagań, projektowania, testowania oprogramowania. (K_W04)</p>
	<p><b>Umiejętności</b></p> <p>P_U01: Student potrafi zarządzać analizą wymagań; (K_U04, K_U06)</p> <p>P_U02: Student potrafi tworzyć specyfikację ze zdefiniowanymi metrykami; (K_U04)</p> <p>P_U03: Student potrafi utworzyć projekt oprogramowania za pomocą języka UML. (K_U04, K_U06)</p>
	<p><b>Kompetencje społeczne (postawy)</b></p> <p>P_K01: Student posiada podstawowe umiejętności komunikacji w zespole informatycznym (K_K02)</p>
<p><b>Kontakt</b></p> <p>a.kostulak@inf.ug.edu.pl</p>	