

**KARTA PRZEDMIOTU**

Cykl kształcenia od roku akademickiego: 2022/2023

**I. Dane podstawowe**

Nazwa przedmiotu	Modelowanie i analiza systemów informatycznych
Nazwa przedmiotu w języku angielskim	Modeling and analysis of information systems
Kierunek studiów	Informatyka
Poziom studiów (I, II, jednolite magisterskie)	II stopnia
Forma studiów (stacjonarne, niestacjonarne)	Stacjonarne
Dyscyplina	Informatyka
Język wykładowy	Polski

Koordinator przedmiotu	dr Viktor Melnyk, prof. KUL
------------------------	-----------------------------

Forma zajęć ( <i>katalog zamknięty ze słownika</i> )	Liczba godzin	semestr	Punkty ECTS
wykład	30	I	6
konwersatorium			
ćwiczenia			
laboratorium	30	I	
warsztaty			
seminarium			
proseminarium			
lektorat			
praktyki			
zajęcia terenowe			
pracownia dyplomowa			
translatorium			
wizyta studyjna			

Wymagania wstępne	W1 – znajomość podstawowych zagadnień dotyczących inżynierii oprogramowania (cykl życia oprogramowania, inżynieria wymagań, zarządzanie projektem). W2 – znajomość z podstawami programowania.
-------------------	---

**II. Cele kształcenia dla przedmiotu**

C1 – Przedstawienie zaawansowanych metod modelowania i analizy systemów informatycznych (strukturalnych i obiektowych).
C2 – Zapoznanie studentów z językiem modelowania UML i praktycznymi jego zastosowaniami.
C3 – Wyrobienie u studentów umiejętności posługiwania się metodami modelowania i analizy systemów informatycznych oraz oceny związanych z nimi artefaktów.

### III. Efekty uczenia się dla przedmiotu wraz z odniesieniem do efektów kierunkowych

Symbol	Opis efektu przedmiotowego	Odniesienie do efektu kierunkowego
<b>WIEDZA</b>		
W_01	Student zna i rozumie proces wytwarzania produktu informatycznego, modele wytwarzania produktów informatycznych, cykl życia systemu informatycznego.	K_W01, K_W04, K_W06
W_02	Student potrafi opisać proces modelowania i analizy systemu informatycznego oraz omówić artefakty tego procesu, jak np. diagramy języka UML.	K_W01, K_W04, K_W06
W_03	Student zna język modelowania systemów informatycznych UML i ma podstawową wiedzę języka SysML.	K_W01, K_W04
W_04	Student ma wiedzę o użyteczności graficznych interfejsów użytkownika i rozumie zasady ich tworzenia.	K_W04
W_05	Student ma wiedzę o technologii ponownego użycia artefaktów systemów informatycznych.	K_W04
<b>UMIEJĘTNOŚCI</b>		
U_01	Student potrafi tworzyć modele systemów informatycznych i umiejętnie się nimi posługiwać, zastosować i rozwijać zdobytą wiedzę na temat systemów informatycznych i ich wykorzystania w pokrewnych dziedzinach nauki i gospodarki (m.in. w zarządzaniu przedsiębiorstwem).	K_U01, K_U02, K_U04, K_U05, K_U10, K_U17
U_02	Student potrafi analizować cechy systemów informatycznych i związane z nimi artefakty.	K_U01, K_U02, K_U05, K_U10
U_03	Student ma podstawowe umiejętności modelowania i analizy systemów informatycznych za pomocą znanych narzędzi CASE.	K_U03, K_U04, K_U17
<b>KOMPETENCJE SPOŁECZNE</b>		
K_01	Umiejętnie rozwiązywać złożone problemy, z którymi może spotkać się w życiu, stosując poznane metody modelowania i analizy systemów informatycznych, obiektywnie oceniając uzyskane wyniki.	K_K01, K_K02, K_K03
K_02	Postępować ze standardami etycznymi obowiązującymi w branży IT.	K_K04, K_K05
K_03	Pracować efektywnie, sprawnie, zespołowo i indywidualnie, umiejętnie oceniając priorytety w realizacji projektu.	K_K01, K_K02, K_K03

### IV. Opis przedmiotu/ treści programowe

<p>WYKŁAD</p> <ol style="list-style-type: none"> <li>1. Wprowadzenie. Historia. Źródła pozyskiwania oprogramowania. Cykl życia systemu informatycznego.</li> <li>2. Faza planowania i wyboru. Faza określania wymagań. Modele wytwarzania produktu informatycznego. UML.</li> <li>3. Model przypadków użycia.</li> <li>4. Model obiektowy: diagramy obiektów i klas.</li> <li>5. Model obiektowy: związki asocjacji i generalizacji-specjalizacji.</li> <li>6. Model obiektowy: inne formy związków.</li> <li>7. Metody budowy i transformacji diagramu klas.</li> <li>8. Model dynamiczny: diagramy maszyny stanów i czynności.</li> <li>9. Model dynamiczny: diagramy interakcji - diagramy komunikacji i sekwencji. Generyczne</li> </ol>
--

diagramy interakcji.  
 10. Model dynamiczny: diagramy harmonogramowania i przeglądu interakcji.  
 11. Model strukturalny: diagramy komponentów, wdrożeniowe (rozlokowania), pakietów oraz struktur złożonych.  
 12. Wstęp do fazy projektowania. Przejście z modelu obiektowego na model relacyjny.  
 13. Narzędzia CASE.  
 14. Użyteczność (usability) graficznych interfejsów użytkownika.  
 15. Technologia ponownego użycia.  
 16. Krytyka UML. SysML.

#### ĆWICZENIA

1. Modelowanie funkcjonalności systemu informatycznego. Opisy przypadków użycia. Diagramy przypadków użycia UML.  
 2. Modelowanie struktury systemu informatycznego na poziomie konceptualnym i implementacyjnym. Diagramy obiektów, klas i pakietów UML.  
 3. Modelowanie zachowania w systemie informatycznym. Diagramy czynności UML.  
 4. Modelowanie interakcji w systemie informatycznym. Diagramy interakcji UML.  
 4. Modelowanie zmienności stanów w systemie informatycznym. Diagramy maszyny stanowej i harmonogramowania UML.  
 5. Modelowanie fizycznego aspektu systemu informatycznego. Diagramy komponentów i wdrażania UML.  
 6. Zagadnienia z zakresu zarządzania projektem konstrukcji systemu informatycznego.

#### V. Metody realizacji i weryfikacji efektów uczenia się

Symbol efektu	Metody dydaktyczne <i>(lista wyboru)</i>	Metody weryfikacji <i>(lista wyboru)</i>	Sposoby dokumentacji <i>(lista wyboru)</i>
<b>WIEDZA</b>			
W_01, W_04, W_05	Wykład konwencjonalny	Egzamin / Zaliczenie pisemne	Karta egzaminacyjna / Karta zaliczeniowa
W_02, W_03	Wykład konwencjonalny, Ćwiczenia praktyczne	Egzamin / Zaliczenie pisemne, Kolokwium / Test / Sprawdzian pisemny, Przygotowanie / wykonanie projektu, Prezentacja	Karta egzaminacyjna / Karta zaliczeniowa, Uzupełnione i ocenione kolokwium / Test / Sprawdzian pisemny, Protokół / Wydruk / Plik sprawozdania
<b>UMIEJĘTNOŚCI</b>			
U_01 - U_03	Ćwiczenia laboratoryjne, Ćwiczenia praktyczne design thinking	Kolokwium / Test / Sprawdzian pisemny, Przygotowanie / wykonanie projektu, Prezentacja	Uzupełnione i ocenione kolokwium / Test / Sprawdzian pisemny, Protokół / Wydruk / Plik sprawozdania
<b>KOMPETENCJE SPOŁECZNE</b>			
K_01, K_02	Wykład konwencjonalny, Ćwiczenia praktyczne design thinking	Egzamin / Zaliczenie pisemne, Sprawdzenie umiejętności praktycznych, Obserwacja	Karta egzaminacyjna / Karta zaliczeniowa, Uzupełnione i ocenione kolokwium / Test / Sprawdzian pisemny,

			Protokół / Wydruk / Plik sprawozdania
K_03	Ćwiczenia laboratoryjne design thinking	Sprawdzenie umiejętności praktycznych, Obserwacja	Uzupełnione i ocenione kolokwium / Test / Sprawdzian pisemny, Protokół / Wydruk / Plik sprawozdania

## VI. Kryteria oceny, wagi...

Zaliczenie ćwiczeń - kolokwia (40% oceny końcowej), projekt (60% oceny końcowej).

Warunkiem zaliczenia ćwiczeń jest obecność studenta na zajęciach dydaktycznych, wykonanie ćwiczeń i otrzymywanie ocen, uzyskanie pozytywnej oceny z kolokwiów pisemnych, oraz uzyskanie pozytywnej oceny z projektu.

Egzamin (dla osób, które zaliczyły ćwiczenia) polega na przeprowadzeniu testu z wiedzy przekazanej na wykładzie. Ocena egzaminacyjna jest formowana na podstawie dwóch składowych: 70 % - odpowiedzi pisemne na zadania testowe i odpowiedzi ustne w razie wątpliwości, 30% - ocena otrzymana z ćwiczeń.

Ocenianie jest wykonywane według następującej skali:

91 – 100% - bardzo dobry (5.0),

81 – 90% - plus dobry (4.5),

71 – 80% - dobry (4.0),

61 – 70% - plus dostateczny (3.5),

50 – 60% - dostateczny (3.0),

poniżej 50% - niedostateczny (2.0).

Szczegółowe zasady oceniania są podawane studentom z każdą edycją przedmiotu.

## VII. Obciążenie pracą studenta

Forma aktywności studenta	Liczba godzin
Liczba godzin kontaktowych z nauczycielem	<b>60</b>
Liczba godzin indywidualnej pracy studenta	Przygotowanie do zajęć <b>20</b> Studiowanie literatury <b>20</b> Przygotowanie do kolokwium <b>25</b> Przygotowanie do projektu <b>35</b> Łączna liczba godzin <b>100</b>

## VIII. Literatura

Literatura podstawowa
1. J. Płodzień, E. Stemposz – Analiza i projektowanie systemów informatycznych, Wydawnictwo PJWSTK, Warszawa 2005
2. M. Trzaska: „Modelowanie i implementacja systemów informatycznych”. Rok 2008. Wydawnictwo PJWSTK.
3. S. Wrycza, B. Marcinkowski, K. Wyrzykowski – Język UML 2.0 w modelowaniu systemów

informatycznych, Helion, Gliwice 2005

4. M. Śmiałek – Zrozumieć UML 2.0. Metody modelowania obiektowego, Helion, Gliwice 2005

5. J. S. Valacich, J. F. George, J. A. Hoffer, Essentials of Systems Analysis and Design, Pearson, Boston 2012

6. S. Wrycza (red.), UML 2.1. Ćwiczenia, Helion, Gliwice 2006

7. S. Wrycza, B. Marcinkowski, J. Maślankowski – UML 2.x. Ćwiczenia zaawansowane, Helion, Gliwice 2012

8. H. Tańska, H. Pikus, Ćwiczenia z analizy i projektowania systemów informatycznych, Wydawnictwo UW-M, Olsztyn 2004

9. P. Beynon-Davies, Inżynieria systemów informacyjnych. Wprowadzenie, WN-T, Warszawa 2004

Literatura uzupełniająca

1. C. Larman, UML i wzorce projektowe. Analiza i projektowanie obiektowe oraz iteracyjny model wytwarzania aplikacji. Wydanie III, Helion, Gliwice 2011

2. D. Pitone, N. Pitman – UML 2.0. Almanach, Helion, Gliwice 2007

3. [www.omg.org](http://www.omg.org) – specyfikacje UML

4. K. E. Kendall, J. E. Kendall, Systems Analysis and Design, Prentice Hall, Boston 2011

5. G. Booch, J. Rumbaugh, I. Jacobson – UML. Przewodnik użytkownika, WNT, Warszawa 2002