



## SYLABUS PRZEDMIOTU

**Nazwa przedmiotu:** Projektowanie systemów informacyjnych

**Kod przedmiotu:** PRI

**Kierunek / Profil:** Informatyka / praktyczny

**Tryb studiów:** niestacjonarny

**Rok / Semestr:** 3 / 6

**Charakter:** obowiązkowy

**Odpowiedzialny:** dr hab. Bartosz Marcinkowski

**Wersja z dnia:** 19.02.2026

### 1. Godziny zajęć i punkty ECTS

Wykłady	Ćwiczenia	Laboratoria	Z prowadzącym	Praca własna	Łącznie	ECTS
16 h	—	16 h	32 h	93 h	125 h	6

### 2. Forma zajęć

Forma zajęć	Sposób zaliczenia
Laboratorium	Zaliczenie z oceną
Wykład	Nieoceniany

### 3. Cel dydaktyczny

Celem przedmiotu jest zapoznanie słuchaczy z podstawowymi technikami analityczno-projektowymi, narzędziami CASE oraz założeniami metodologicznymi wytwarzania rozwiązań IT. Student musi wykazywać się znajomością najczęściej wykorzystywanych dokumentów przedprojektowych, rdzeniowych kategorii modelowania i diagramów języka UML oraz metod odwzorowania modelu obiektowego w wybranym języku programowania/schemacie relacyjnym z wykorzystaniem narzędziowych technik transformacji.

## 4. Przedmioty wprowadzające

Przedmiot	Wymagane zagadnienia
Programowanie obiektowe w Javie	Relacyjne bazy danych
Podstawowe umiejętności w zakresie programowania obiektowego	Umiejętność projektowania podstawowych algorytmów
Ogólna znajomość problematyki relacyjnych baz danych	—

## 5. Treści programowe

1. Pojęcia i metody inżynierii oprogramowania. Cykl życia produktu informatycznego
2. Faza przedprojektowa i planowanie projektu; Dokument Założeń Wstępnych (DZW)
3. Inżynieria wymagań; pojęcie wymagania; kategorie i przykłady wymagań. Specyfikacja wymagań i dokument Specyfikacji Wymagań Systemowych (SWS)
4. Praktyczne przykłady konstrukcji SWS
5. Cechy narzędzi CASE. Enterprise Architect
6. Graficzne techniki modelowania wymagań systemowych na przykładzie diagramów wymagań systemowych języka SysML. Strukturyzacja złożonych modeli z wykorzystaniem pakietów
7. Model przypadków użycia
8. Praktyka modelowania i specyfikacji przypadków użycia. Techniki specyfikacji scenariuszy przypadków użycia
9. Modelowanie dynamiki systemu w języku UML – wprowadzenie. Techniki modelowania interakcji w systemie
10. Diagramy sekwencji UML jako uniwersalna technika modelowania interakcji pomiędzy składowymi systemu
11. Alternatywne techniki modelowania interakcje na przykładzie diagramu komunikacji. Wstęp do transformacji w narzędziu Enterprise Architect
12. Specyfikowanie przypadków użycia z wykorzystaniem diagramów czynności
13. Modelowanie struktury systemu: obiekty, klasy, właściwości klasy, dziedziczenie, polimorfizm, agregacja. Wyrażanie struktury systemu w języku UML
14. Zaawansowane przykłady modelowania diagramu klas UML
15. Przekształcanie modeli klas w kod szkieletowy, modele w wybranych frameworkach i trwałe struktury bazodanowe w narzędziu Enterprise Architect
16. Diagramy maszyny stanowej jako technika zarządzania złożonością

## 6. Efekty kształcenia

## Wiedza

- Student zna i rozumie cechy najistotniejszych cykli wytwarzania/ewolucji oprogramowania
- Student zna i rozumie składnię rdzeniowych diagramów języka UML
- Student zna i rozumie ograniczenia występujące w procesie wytwarzania oprogramowania
- Student rozumie wagę wymagań systemowych w procesie wytwarzania oprogramowania
- Student zna i rozumie techniki specyfikowania wymagań. Student zna i rozumie umiejscowienie poszczególnych elementów testowania w zależności od obranego cyklu

## Umiejętności

- Student potrafi zaprojektować małoskalowe rozwiązanie IT w sposób holistyczny
- Student potrafi skorzystać z narzędzi klasy CASE na poziomie średniozaawansowanym
- Student potrafi specyfikować wymagania systemowe
- Student potrafi projektować algorytmikę przypadków użycia oraz interakcje pomiędzy składowymi przypadku użycia. Student potrafi projektować architekturę systemu wyrażoną klasami. Student potrafi iteracyjnie udoskonalać komponenty projektowe systemu

## 7. Kryteria oceny

---

- Kryteria oceny
- prezentacja projektu i dokumentacji (60%)
- aktywność na zajęciach (10%)
- kolokwium końcowe (30%)

## 8. Metody dydaktyczne

---

Wykład, laboratoria, praca własna studenta.

## 9. Literatura

---

### Podstawowa:

- Wrycza S., Marcinkowski B., Wyrzykowski K. (2006): Język UML 2.0 w modelowaniu systemów informatycznych. Helion, Gliwice.

### Uzupełniająca:

- Wrycza S., Marcinkowski B., Maślankowski J. (2012): UML. Ćwiczenia zaawansowane. Helion, Gliwice.

