**Nazwa przedmiotu:**

Inżynieria oprogramowania

**Koordynator przedmiotu:**

dr. inż. Paweł Wnuk

**Status przedmiotu:**

Obowiązkowy

**Poziom kształcenia:**

Studia I stopnia

**Program:**

Automatyka Robotyka i Informatyka Przemysłowa

**Grupa przedmiotów:**

Obowiązkowe

**Kod przedmiotu:**

IPR

**Semestr nominalny:**

7 / rok ak. 2020/2021

**Liczba punktów ECTS:**

2

**Liczba godzin pracy studenta związanych z osiągnięciem efektów uczenia się:**

1) Liczba godzin bezpośrednich – 32 godz., w tym:
• wykład - 15 godz.
• projekt – konsultacje na uczelni 15 godz.
• kolokwia - 2 godz.
2) Praca własna studenta – 30 godz., w tym:
• przygotowanie do kolokwiów zaliczeniowych - 5 godz.
• opracowanie projektu- 25 godz.
Razem: 62 godz. (2 ECTS)

**Liczba punktów ECTS na zajęciach wymagających bezpośredniego udziału nauczycieli akademickich:**

1 punkt ECTS - liczba godzin bezpośrednich: 32, w tym:
• wykład - 15 godz.
• projekt – konsultacje na uczelni – 15 godz.
• egzamin – 2 godz.

**Język prowadzenia zajęć:**

polski

**Liczba punktów ECTS, którą student uzyskuje w ramach zajęć o charakterze praktycznym:**

1,6 punktu ECTS – 40 godz., w tym:
• projekt – konsultacje na uczelni - 15 godz.
• opracowanie projektu- 25 godz.

**Formy zajęć i ich wymiar w semestrze:**

|  |  |
| --- | --- |
| Wykład:  | 15h |
| Ćwiczenia:  | 0h |
| Laboratorium:  | 15h |
| Projekt:  | 0h |
| Lekcje komputerowe:  | 0h |

**Wymagania wstępne:**

Znajomość programowania strukturalnego i obiektowego w wybranym języku wyższego poziomu.

**Limit liczby studentów:**

brak

**Cel przedmiotu:**

Przedstawienie procesu powstawania oprogramowania, ze szczególnym uwzględnieniem projektowania aplikacji. Zapoznanie studentów z głównym językiem modelowania w informatyce - UML. Omówienie wzorców projektowych.

**Treści kształcenia:**

1. Cykl tworzenia oprogramowania. Stosowane metodologie. Praca w zespole - role uczestników projektu. Komunikacja członków zespołu - narzędzia. 2. Analiza wymagań. Komunikacja z nie-informatykami, wspólne definicje, problemy. Schematy funkcjonalności 3. Modelowanie obiektowe - UML. Modele statyczne i dynamiczne systemu. Diagramy statyczne, ich związek z rzeczywistym kodem aplikacji. Generowanie kodu na podstawie diagramu i odwrotnie. Podstawowe narzędzia modelowania 4. Diagramy dynamiczne - zachowanie aplikacji. Maszyna stanów, przejścia między stanami. Diagramy UML opisujące dynamikę. 5. Wzorce projektowe - co to jest i dlaczego warto je stosować. Omówienie podstawowych wzorców wraz z przykładowymi implementacjami. 6. Projekt aplikacji

**Metody oceny:**

Końcowa ocena przedmiotu opiera się na projekcie zespołowym, składającym się z 4 niezależnie ocenianych etapów:
1: analiza funkcjonalna
2: model statyczny
3: model dynamiczny
4: prototyp

**Egzamin:**

tak

**Literatura:**

1. Materiały dodatkowe do przedmiotu
2. Roger S. Pressman "Praktyczne podejście do inżynierii oprogramowania", WNT 2004
3. Dick Hamlet, Joe Maybee "Podstawy techniczne inżynierii oprogramowania"

**Witryna www przedmiotu:**

iair.mchtr.pw.edu.pl/studenci

**Uwagi:**

## Charakterystyki przedmiotowe

### Profil ogólnoakademicki - wiedza

**Charakterystyka IP\_1ST\_W01:**

Posiada uporządkowaną i podbudowaną wiedzę o procesie powstawania oprogramowania

Weryfikacja:

realizacja projektu, egzamin

**Powiązane charakterystyki kierunkowe:** K\_W05

**Powiązane charakterystyki obszarowe:** P6U\_W

**Charakterystyka IP\_1ST\_W01:**

Posiada elementarną wiedzę w zakresie zarządzania cyklem tworzenia oprogramowania

Weryfikacja:

Egzamin, projekt

**Powiązane charakterystyki kierunkowe:** K\_W21

**Powiązane charakterystyki obszarowe:** P6U\_W, I.P6S\_WK

### Profil ogólnoakademicki - umiejętności

**Charakterystyka IP\_1ST\_U01:**

Potrafi przygotować kompletną dokumentację systemu informatycznego

Weryfikacja:

Projekt

**Powiązane charakterystyki kierunkowe:** K\_U02, K\_U03

**Powiązane charakterystyki obszarowe:** P6U\_U, I.P6S\_UW.o, I.P6S\_UK

**Charakterystyka IP\_1ST\_U02:**

Potrafi zaprojektować i zaprogramować system informatyczny przy wykorzystaniu wybranego języka wspierającego programowanie obiektowe, wykorzystujący komunikację sieciową i bazy danych

Weryfikacja:

Projekt

**Powiązane charakterystyki kierunkowe:** K\_U09, K\_U10, K\_U11

**Powiązane charakterystyki obszarowe:** I.P6S\_UW.o, III.P6S\_UW.o, I.P6S\_UW, P6U\_U

### Profil ogólnoakademicki - kompetencje społeczne

**Charakterystyka IP\_1ST\_K01:**

Potrafi przygotować oprogramowanie w zespole

Weryfikacja:

projekt

**Powiązane charakterystyki kierunkowe:** K\_K04

**Powiązane charakterystyki obszarowe:** P6U\_K, I.P6S\_KO, I.P6S\_KR

**Charakterystyka IP\_1ST\_K02:**

Potrafi oszacować koszt wytworzenia oprogramowania i jego wdrożenia w praktyce

Weryfikacja:

projekt

**Powiązane charakterystyki kierunkowe:** K\_K05

**Powiązane charakterystyki obszarowe:** P6U\_K, I.P6S\_KO