**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:**

Obieralne

**Kod przedmiotu:**

**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:  | 0h |
| Projekt:  | 15h |
| Lekcje komputerowe:  | 0h |

**Wymagania wstępne:**

Znajomość programowania strukturalnego i obiektowego w wybranym języku wyższego poziomu.

**Limit liczby studentów:**

**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, 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:**

**Uwagi:**

brak

## Charakterystyki przedmiotowe