**Nazwa przedmiotu:**

Integracja programowa systemów multimedialnych I

**Koordynator przedmiotu:**

dr inż. Robert Sitnik

**Status przedmiotu:**

Obowiązkowy

**Poziom kształcenia:**

Studia I stopnia

**Program:**

Mechatronika

**Grupa przedmiotów:**

Obowiązkowe

**Kod przedmiotu:**

IPSM1

**Semestr nominalny:**

6 / rok ak. 2020/2021

**Liczba punktów ECTS:**

3

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

1) Liczba godzin bezpośrednich – 30 godz., w tym:
• wykład 15 godz.,
• konsultacje projektowe z prowadzącym 15 godz.,
2) Praca własna studenta – 55 godz. w tym:
• zapoznanie się z literaturą i dokumentacją 20 godz.
• projekt i implementacja aplikacji, optymalizacja, testowanie i dokumentacja 35 godz.
 RAZEM 85 godz. = 3 ECTS

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

1 punkt ECTS- Liczba godzin bezpośrednich – 30 godz., w tym:
• wykład 15,
• konsultacje projektowe z prowadzącym 15,

**Język prowadzenia zajęć:**

polski

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

2 punkty ECTS - 50 godz., w tym:
• projekt i implementacja aplikacji, optymalizacja, testowanie i dokumentacja 35 godz.
• konsultacje projektowe z prowadzącym 15 godz.,

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

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

**Wymagania wstępne:**

Kurs inżynierski matematyki. Podstawy programowania strukturalnego.

**Limit liczby studentów:**

60

**Cel przedmiotu:**

Znajomość podstaw programowania obiektowego na przykładzie języków C++. Znajomość podstaw UML.

**Treści kształcenia:**

(W) Podstawowe pojęcia języka obiektowego: obiekt i klasa. Notacja C++: typy, deklaracje, wskaźniki, struktury, tablice, wyrażenia. Przykłady implementacji dla przetwarzania obrazów.
Omówienie podstawowych zasad programowania obiektowego: abstrakcji i kapsułkowania danych, dziedziczenia i polimorfizmu. Wstęp do wyjątków. Rozwinięcie przykładów dla przetwarzania obrazów.
Wprowadzenie do UML. Definicje podstawowych pojęć: przypadki użycia, diagramy klas, definiowanie właściwości dynamicznych systemu, diagramy implementacji. Dokumentacja. Przykład dla przetwarzania obrazów.
Omówienie podstawowej funkcjonalności biblioteki standardowej C++. Przykład projektu oraz implementacji prostej aplikacji do przetwarzania obrazów.

(P) Projekt dotyczący zagadnień związanych z programowaniem obiektowym. Dla każdego projektu przygotowany jest zestaw wymagań, a gdy zaistnieje taka potrzeba wskazywany jest bądź dostarczany zbiór danych początkowych (testowych). W trakcie realizacji projektu należy wykonać następujące etapy: stworzyć projekt, dokonać implementacji, testowania i poprawek na danych początkowych, stworzyć dokumentację.

**Metody oceny:**

Średnia ważona z kolokwium zaliczeniowego (40%) i projektu (60%). W projekcie oceniana jest Suma punktów z kolokwium zaliczeniowego przeprowadzanego w formie pisemnej ( wariant stacjonarny ) lub w formie testu jednokrotnego i wielokrotnego wyboru ( wariant zdalny ) - (40%) i projektu (60%), w którym oceniana jest jego dokumentacja, kod źródłowy programu, a następnie oceniane jest działanie samego programu, jego niezawodność i poprawność wyników jakie on generuje.

**Egzamin:**

nie

**Literatura:**

1. B. Strostroup, Język C++, Wydawnictwa Naukowo-Techniczne, Warszawa 1997.
2. B. Eckel, Thinking in C++, edycja polska, Helion, Warszawa 2004.
3. G. Booch, J. Rumbaugh, I. Jacobson, UML przewodnik użytkownika, Wydawnictwa NaukowoTechniczne, Warszawa 2002.
4. B. Strostroup, Język C++. Kompendium wiedzy. Wydanie IV, Helion, Warszawa 2014.
5. M. Bancila, Nowoczesny C++ Zbiór praktycznych zadań dla przyszłych ekspertów, Helion, Warszawa 2015.

**Witryna www przedmiotu:**

brak

**Uwagi:**

## Charakterystyki przedmiotowe

### Profil ogólnoakademicki - wiedza

**Charakterystyka IPSM\_W01:**

Znajomość zagadnień z obszaru programowania obiektowego C++/Java

Weryfikacja:

Zaliczenie kolokwium w trakcie wykładu

**Powiązane charakterystyki kierunkowe:** K\_W04, K\_W05

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

**Charakterystyka IPSM\_W02:**

Zna podstawowe techniki projektowania (UML) i dokumentacji (DOXYGEN) programów komputerowych tworzonych w językach wyższego poziomu C++/Java

Weryfikacja:

Zaliczenie kolokwium w trakcie wykładu

**Powiązane charakterystyki kierunkowe:** K\_W05, K\_W04

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

### Profil ogólnoakademicki - umiejętności

**Charakterystyka IPSM\_U01:**

Potrafi zaprojektować i zaimplementować algorytmy w języku obiektowym C++

Weryfikacja:

Zaliczenie projektu programistycznego C++

**Powiązane charakterystyki kierunkowe:** K\_U01, K\_U04, K\_U05, K\_U18

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

**Charakterystyka IPSM\_U02:**

Potrafi opracować projekt (UML) aplikacji i dokumentację (DOXYGEN) kodów źródłowych

Weryfikacja:

Zaliczenie projektu programistycznego C++

**Powiązane charakterystyki kierunkowe:** K\_U02, K\_U03, K\_U04

**Powiązane charakterystyki obszarowe:** P6U\_U, I.P7S\_UW.o, I.P6S\_UK, I.P6S\_UW.o

### Profil ogólnoakademicki - kompetencje społeczne

**Charakterystyka IPSM\_S01:**

Potrafi zaprojektować, zaimplementować, udokumentować i zaprezentować swój projekt programistyczny

Weryfikacja:

Zaliczenie projektu programistycznego C++

**Powiązane charakterystyki kierunkowe:** K\_K01, K\_K04

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