**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. 2016/2017

**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:  | 225h |
| Ćwiczenia:  | 0h |
| Laboratorium:  | 0h |
| Projekt:  | 225h |
| 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++ i Java. Znajomość podstaw UML.

**Treści kształcenia:**

Podstawowe pojęcia języka obiektowego: obiekt i klasa. Notacja C++ a Java: 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. Dwa kolokwia zaliczeniowe.

Projekt dotyczący zagadnień przetwarzania obrazów. Dla każdego projektu przygotowany jest zestaw wymagań oraz zbiór danych początkowych (testowych). W trakcie realizacji projektu należy wyko-nać następujące etapy: stworzyć model UML, dokonać implementacji, testowania i poprawek na danych początkowych, stworzyć dokumentację.

**Metody oceny:**

Suma punktów z dwóch kolokwiów zaliczeniowych (40%) i projektu (60%) – oceniana jest dokumentacja projekty, kod zródłowy programu a następnie oceniane jest działanie samego programu, jego niezawodność i poprawność wyników jakie program 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 Naukowo-Techniczne, Warszawa 2002.

**Witryna www przedmiotu:**

brak

**Uwagi:**

## Efekty przedmiotowe

### Profil ogólnoakademicki - wiedza

**Efekt IPSM\_W01:**

Znajomość zagadnień z obszaru programowania obiektowego C++/Java

Weryfikacja:

Zaliczenie kolokwium w trakcie wykładu

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

**Powiązane efekty obszarowe:** T1A\_W02, T1A\_W02, T1A\_W04

**Efekt 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 efekty kierunkowe:** K\_W04, K\_W05

**Powiązane efekty obszarowe:** T1A\_W02, T1A\_W02, T1A\_W04

### Profil ogólnoakademicki - umiejętności

**Efekt IPSM\_U01:**

Potrafi zaprojektować i zaimplementować algorytmy w języku obiektowym C++

Weryfikacja:

Zaliczenie projektu programistycznego C++

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

**Powiązane efekty obszarowe:** T1A\_U01, T1A\_U01, T1A\_U03, T1A\_U04, T1A\_U06, T1A\_U05, T1A\_U16

**Efekt IPSM\_U02:**

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

Weryfikacja:

Zaliczenie projektu programistycznego C++

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

**Powiązane efekty obszarowe:** T1A\_U02, T1A\_U07, T1A\_U04, T1A\_U01, T1A\_U03, T1A\_U04, T1A\_U06

### Profil ogólnoakademicki - kompetencje społeczne

**Efekt IPSM\_S01:**

Potrafi zaprojektować, zaimplementować, udokumentować i zaprezentować swój projekt programistyczny

Weryfikacja:

Zaliczenie projektu programistycznego C++

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

**Powiązane efekty obszarowe:** T1A\_K01, T1A\_K03, T1A\_K04, T1A\_K05