**Nazwa przedmiotu:**

Integracja programowa systemów multimedialnych I

**Koordynator przedmiotu:**

Dr inż. Robert Sitnik, adiunkt

**Status przedmiotu:**

Obowiązkowy

**Poziom kształcenia:**

Studia I stopnia

**Program:**

Mechatronika

**Grupa przedmiotów:**

Obowiązkowe

**Kod przedmiotu:**

**Semestr nominalny:**

6 / rok ak. 2009/2010

**Liczba punktów ECTS:**

3

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

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

**Język prowadzenia zajęć:**

polski

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

**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:**

**Cel przedmiotu:**

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

**Treści kształcenia:**

(W) 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.

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

**Metody oceny:**

(W) Dwa kolokwia
(P) Ocena z projektu

**Egzamin:**

**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:**

**Uwagi:**

## Efekty przedmiotowe