

Object-Oriented Programming - course description

General information	
Course name	Object-Oriented Programming
Course ID	11.3-WE-AiRP-PO
Faculty	Faculty of Computer Science, Electrical Engineering and Automatics
Field of study	Automatic Control and Robotics / Industrial Control
Education profile	academic
Level of studies	First-cycle studies leading to Engineer's degree
Beginning semester	winter term 2016/2017

Course information	
Semester	2
ECTS credits to win	5
Course type	obligatory
Teaching language	polish
Author of syllabus	dr hab. inż. Paweł Majdzik, prof. UZ

Classes forms					
The class form	Hours per semester (full-time)	Hours per week (full-time)	Hours per semester (part-time)	Hours per week (part-time)	Form of assignment
Lecture	30	2	18	1,2	Credit with grade
Laboratory	30	2	18	1,2	Credit with grade

Aim of the course

- zapoznanie studentów z podstawowymi pojęciami programowania obiektowego: enkapsulacja, klasa, obiekt, metoda składowa
- zapoznanie studentów z metodami hermetyzacji danych, funkcji i klas ukształtowanie wśród studentów umiejętności projektowania programów obiektowych
- zapoznanie studentów z metodami implementacji programów zorientowanych obiektowo: dziedziczenie, polimorfizm

Prerequisites

Programowanie z elementami algorytmiki

Scope

Wstęp do programowania obiektowego. Pojęcie abstrakcyjnego typu danych. Definicja klas. Enkapsulacja - deklaracja i definicja metod składowych klas. Przekazywanie parametrów do funkcji składowych: przez wartość i przez referencję.

Składowe prywatne i publiczne klasy. Przeciążenie funkcji. Konstruktory: konstruktor domniemany, konstruktor kopiujący.

Lista inicjalizacyjna konstruktora. Konstruktory syntezywane. Destruktry. Przeciążenie operatorów. Funkcje zaprzyjaźnione. Funkcje typu inline. Konwersje zdefiniowane przez użytkownika: funkcja konwertująca, konstruktor konwertujący.

Dziedziczenie. Zasady dziedziczenia. Składowe typu protected. Dziedziczenie wielokrotne i wielobazowe. Problem nazw zmiennych w dziedziczeniu wielobazowym.

Polimorfizm. Funkcje wirtualne. Funkcje czysto wirtualne. Wczesne i późne wiązanie funkcji. Koszty czasowe i pamięciowe związane ze stosowaniem polimorfizmu.

Klasy abstrakcyjne. Definiowanie i przykłady zastosowań klas abstrakcyjnych w programach zorientowanych obiektowo. Destructory wirtualne. Szablony funkcji. Definicja funkcji szablonowych. Funkcje specjalizowane. Etapy dopasowania do funkcji. Szablony klas. Definicja szablonów klas. Szablony klas, a makrodefinicje. Składniki statyczne w szablonie klas. Dziedziczenie szablonów klas. Przykłady zastosowań szablonów klas. Klasy specjalizowane. Obsługa wyjątków.

Stosowanie wzorców z biblioteki STL – wzorce pojemnikowe, iteratory, pojemniki kojarzące (mapy, zbiory).

Teaching methods

wykład: wykład konwencjonalny

laboratorium: ćwiczenia laboratoryjne

Learning outcomes and methods of theirs verification

Outcome description	Outcome symbols	Methods of verification	The class form
potrafi zdefiniować i zaimplementować podstawowe elementy składowe klasy: konstruktory, funkcje operatorowe, destruktry	• K_W05	• an exam - oral, descriptive, test and other	• Lecture • Laboratory
rozumie różnicę pomiędzy programowaniem obiektowym i strukturalnym i potrafi zdefiniować wady i zalety obydwu metod	• K_W05	• an exam - oral, descriptive, test and other	• Lecture • Laboratory

Outcome description	Outcome symbols	Methods of verification	The class form
rozumie podstawowe pojęcia związane z programowaniem obiekowym: enkapsulację, hermetyzację	• K_W05	• an exam - oral, descriptive, test and other	• Lecture • Laboratory
potrafi zastosować gotowe biblioteki klas w swoich programach	• K_W05	• an evaluation test	• Laboratory
potrafi zaprojektować i zaimplementować diagram klas oraz zastosować polimorfizm	• K_W05	• an evaluation test	• Laboratory

Assignment conditions

Wykład – warunkiem zaliczenia jest uzyskanie pozytywnej oceny ze sprawdzianu w formie pisemnej

Laboratorium – warunkiem zaliczenia jest uzyskanie pozytywnych ocen ze wszystkich sprawdzianów

Składowe oceny końcowej = wykład: 50% + laboratorium: 50%

Recommended reading

- Eckel B.: Thinking in C++, Hellion, Warszawa, 2002.
- Kerighan B., Ritchie D.: Programowanie w języku C, WNT, Warszawa, 2000.
- Kisilewicz J.: Język. w środowisku Borland C++. Wydanie IV, Oficyna Wydawnicza Politechniki Wrocławskiej, Wrocław, 2003.
- Stroustrup B.: C++ Język programowania, WNT, Warszawa, 2001.

Further reading

- Lippman S.B.: Model w C++, WNT, Warszawa, 1996.
- Shalloway A., Trott J.R.: Projektowanie zorientowane obiektowo. Wzorce obiektowe II, Helion, Warszawa, 2005.

Notes

Modified by dr hab. inż. Paweł Majdzik, prof. UZ (last modification: 12-09-2016 13:20)

Generated automatically from SylabUZ computer system