

Programowanie obiektowe - opis przedmiotu

Informacje ogólne	
Nazwa przedmiotu	Programowanie obiektowe
Kod przedmiotu	11.3-WI-INFP-PO
Wydział	Wydział Informatyki, Elektrotechniki i Automatyki
Kierunek	Informatyka
Profil	ogólnoakademicki
Rodzaj studiów	pierwszego stopnia z tyt. inżyniera
Semestr rozpoczęcia	semestr zimowy 2019/2020

Informacje o przedmiocie	
Semestr	2
Liczba punktów ECTS do zdobycia	5
Typ przedmiotu	obowiązkowy
Język nauczania	polski
Sylabus opracował	• dr hab. inż. Paweł Majdzik

Formy zajęć					
Forma zajęć	Liczba godzin w semestrze (stacjonarne)	Liczba godzin w tygodniu (stacjonarne)	Liczba godzin w semestrze (niestacjonarne)	Liczba godzin w tygodniu (niestacjonarne)	Forma zaliczenia
Laboratorium	30	2	18	1,2	Zaliczenie na ocenę
Wykład	30	2	18	1,2	Egzamin

Cel przedmiotu

- 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
- ukształtowanie umiejętności projektowania programów z zastosowaniem wzorców projektowych

Wymagania wstępne

Algorytmy i struktury danych, Podstawy programowania.

Zakres tematyczny

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 syntezowane. Destruktory. 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. Destruktory 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, itp.)

Metody kształcenia

wykład: wykład konwencjonalny

laboratorium: ćwiczenia laboratoryjne

Efekty uczenia się i metody weryfikacji osiągnięcia efektów uczenia się

Opis efektu	Symbole efektów	Metody weryfikacji	Forma zajęć
-------------	-----------------	--------------------	-------------

Opis efektu	Symbole efektów	Metody weryfikacji	Forma zajęć
rozumie podstawowe pojęcia związane z programowaniem obiektowym: enkapsulację, hermetyzację	<ul style="list-style-type: none"> • K_W09 	<ul style="list-style-type: none"> • aktywność w trakcie zajęć • egzamin - ustny, opisowy, testowy i inne • kolokwium 	<ul style="list-style-type: none"> • Wykład • Laboratorium
potrafi zdefiniować i zaimplementować podstawowe elementy składowe klasy: konstruktory, funkcje operatorowe, destruktory	<ul style="list-style-type: none"> • K_W09 	<ul style="list-style-type: none"> • bieżąca kontrola na zajęciach • egzamin - ustny, opisowy, testowy i inne 	<ul style="list-style-type: none"> • Wykład • Laboratorium
potrafi zastosować gotowe biblioteki klas w swoich programach	<ul style="list-style-type: none"> • K_W05 	<ul style="list-style-type: none"> • bieżąca kontrola na zajęciach • obserwacja i ocena aktywności na zajęciach 	<ul style="list-style-type: none"> • Laboratorium
zna podstawowe wzorce projektowe oraz rozumie ich znaczenie w projektowaniu elastycznych programów	<ul style="list-style-type: none"> • K_W05 	<ul style="list-style-type: none"> • aktywność w trakcie zajęć • bieżąca kontrola na zajęciach 	<ul style="list-style-type: none"> • Laboratorium
rozumie różnicę pomiędzy programowaniem obiektowym i strukturalnym i potrafi zdefiniować wady i zalety obydwu metod	<ul style="list-style-type: none"> • K_W09 	<ul style="list-style-type: none"> • egzamin - ustny, opisowy, testowy i inne • kolokwium 	<ul style="list-style-type: none"> • Wykład • Laboratorium

Warunki zaliczenia

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

Laboratorium - warunkiem zaliczenia jest uzyskanie pozytywnych ocen ze wszystkich sprawdzianów

Literatura podstawowa

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

Literatura uzupełniająca

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

Uwagi

Zmodyfikowane przez prof. dr hab. inż. Andrzej Obuchowicz (ostatnia modyfikacja: 05-05-2019 23:10)

Wygenerowano automatycznie z systemu SylabUZ