

Języki programowania - opis przedmiotu

Informacje ogólne	
Nazwa przedmiotu	Języki programowania
Kod przedmiotu	11.3-WE-EiTP-Jęz.Prog.
Wydział	Wydział Informatyki, Elektrotechniki i Automatyki
Kierunek	Elektronika i telekomunikacja
Profil	ogólnoakademicki
Rodzaj studiów	pierwszego stopnia z tyt. inżyniera
Semestr rozpoczęcia	semestr zimowy 2017/2018

Informacje o przedmiocie	
Semestr	3
Liczba punktów ECTS do zdobycia	6
Typ przedmiotu	obowiązkowy
Język nauczania	polski
Sylabus opracował	dr hab. inż. Paweł Majdzik, prof. UZ

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
Wykład	30	2	-	-	Egzamin
Laboratorium	30	2	-	-	Zaliczenie na ocenę

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
- ukształtowanie umiejętności projektowania programów z zastosowaniem wzorców projektowych

Wymagania wstępne

Algorytmy i struktury danych, Metody i techniki programowania I i II.

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 syntezywane. Destruktry. Przeciążenie operatorów. Funkcje zaprzyjaźnione. Funkcje typu inline. Konwersje zdefiniowane przez użytkownika: funkcja konwertująca, konstruktor 70 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. Destruktry 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)

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
potrafi zdefiniować i zaimplementować podstawowe elementy składowe klasy: konstruktory, funkcje operatorowe, destruktry	• K_U22	• bieżąca kontrola na zajęciach • sprawdzian	• Laboratorium
Rozumie różnicę pomiędzy programowaniem obiektowym i strukturalnym i potrafi zdefiniować wady i zalety obydwu metodyk	• K_W09	• bieżąca kontrola na zajęciach • sprawdzian	• Laboratorium
Rozumie podstawowe pojęcia związane z programowaniem obiektowym: enkapsulację, hermetyzację	• K_W09	• aktywność w trakcie zajęć • sprawdzian	• Wykład • Laboratorium

Opis efektu	Symbole efektów	Metody weryfikacji	Forma zajęć
Potrafi zastosować gotowe biblioteki klas w swoich programach	K_U10	- bieżąca kontrola na zajęciach - sprawdzian	Laboratorium
Zna podstawowe wzorce projektowe oraz rozumie ich znaczenie w projektowaniu elastycznych programów	K_W09	- bieżąca kontrola na zajęciach - sprawdzian	- 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

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

Literatura podstawowa

1. Eckel B.: Thinking in C++, Helion, 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.

Literatura uzupełniająca

1. Stroustrup B.: C++ Język programowania, WNT, Warszawa, 2001.
2. Shalloway A., Trott J.R.: Projektowanie zorientowane obiektowo. Wzorce obiektowe II, Helion, Warszawa, 2005.

Uwagi

Zmodyfikowane przez dr inż. Emil Michta, prof. UZ (ostatnia modyfikacja: 28-04-2017 09:08)

Wygenerowano automatycznie z systemu SylabUZ