



KARTA OPISU PRZEDMIOTU - SYLABUS

Nazwa przedmiotu

Programowanie obiektowe [S1Mech2>PO]

Przedmiot

Kierunek studiów
Mechatronika

Rok/Semestr
2/3

Studia w zakresie (specjalność)
–

Profil studiów
ogólnoakademicki

Poziom studiów
pierwszego stopnia

Język oferowanego przedmiotu
polski

Forma studiów
stacjonarne

Wymagalność
obligatoryjny

Liczba godzin

Wykład	Laboratorium	Inne
15	30	0
Ćwiczenia	Projekty/seminaria	
0	0	

Liczba punktów ECTS

3,00

Koordynatorzy

dr hab. inż. Wojciech Pietrowski prof. PP
wojciech.pietrowski@put.poznan.pl

Wykładowcy

Wymagania wstępne

Podstawowa wiedza w zakresie informatyki, systemów operacyjnych, algorytmów i struktur danych, podstaw programowania. Obsługa: komputera, systemu operacyjnego Windows oraz umiejętność programowania strukturalnego w języku C++.

Cel przedmiotu

Poznanie teoretycznych i praktycznych zagadnień związanych z programowaniem obiektowym w języku C#.

Przedmiotowe efekty uczenia się

Wiedza:

Student zna podstawowe pojęcia z zakresu programowania obiektowego. Student zna w zakresie podstawowym języki programowania zorientowanych obiektowo: C#. Student zna podstawowe techniki programowania obiektowego w wybranych środowiskach programistycznych. Student zna struktury programu, struktury modułu, wybranych komponentów wizualnych.

Umiejętności:

Student potrafi pozyskiwać informacje z literatury technicznej oraz Internetu dotyczącej programowania obiektowego. Student potrafi definiować i posługiwać się typem klasowym, szablonami funkcji, techniką dziedziczenia, polimorfizmu. Student potrafi budować programy komputerowe w środowiskach zintegrowanych z wykorzystaniem techniki programowania obiektowego. Student potrafi testować opracowane programy oraz oceniać poprawność ich funkcjonowania.

Kompetencje społeczne:

Rozumie znaczenie podnoszenia kompetencji zawodowych, osobistych i społecznych; ma świadomość, że wiedza i umiejętności w obszarze mechatroniki szybko ewoluują. Rozumie znaczenie wiedzy w rozwiązywaniu problemów z zakresu mechatroniki; jest świadomy konieczności wykorzystania wiedzy ekspertów podczas rozwiązywania zadań inżynierskich w zakresie wykraczającym poza własne kompetencje.

Metody weryfikacji efektów uczenia się i kryteria oceny

Efekty uczenia się przedstawione wyżej weryfikowane są w następujący sposób:

Wykład: zaliczenie na podstawie kolokwium składającego się z pytań ogólnych i testowych. Skala ocen 51-60% pkt. - dst, 61-70% pkt dst+, 71-80% pkt. - db, 81-90% pkt. - db+, 91-100% pkt. - bdb.

Laboratorium: premiowanie praktycznej wiedzy zdobytej w trakcie poprzednich ćwiczeń laboratoryjnych, sprawdzenie praktycznych umiejętności programowania w języku C# (kolokwium zaliczeniowe), ocena wiedzy i umiejętności związanych z realizacją indywidualnych i grupowych projektów programistycznych.

Uzyskiwanie punktów dodatkowych za aktywność podczas zajęć, a szczególnie za: umiejętność współpracy w ramach zespołu praktycznie realizującego zadanie szczegółowe w laboratorium, wykorzystanie elementów i technik wykraczających poza materiał z zakresu prowadzonego wykładu i ćwiczeń laboratoryjnych, staranność estetyczną zrealizowanych projektów.

Treści programowe

Język C#. Programowanie obiektowe w środowisku Microsoft Visual Studio. Edytory kodów źródłowych. Debugger. Programowanie obiektowe. Klasy, obiekty, dziedziczenie, hermetyzacja danych, polimorfizm oraz metody wirtualne, klasy abstrakcyjne, szablony funkcji, szablony klas. Tworzenie i przetwarzanie obiektów, obiekty jako argumenty funkcji. Komponenty, formularze, właściwości, zdarzenia, obsługa wyjątków. Wykorzystanie programowania obiektowego do symulacji wybranych stanów pracy elementów wykonawczych mechatroniki.

Tematyka zajęć

Wykład:

I. Programowanie zorientowane obiektowo: Zastosowanie języków obiektowych, Reguły modelowania obiektowego, Analiza i projektowanie.

II. Fazy tworzenia oprogramowania: Poznawanie i analiza problemu, Tworzenie modelu obiektowego, projektowanie interfejsu i funkcji programu, Implementacja, Testowanie.

III. Programowanie na platformie dotNET: Struktura kodu, Typy proste, System typów, model obiektowy, Typy proste a typy referencyjne, boxing i unboxing.

IV. Programowanie na platformie dotNET: Testowanie i debugowanie w C#, Programowanie asynchroniczne, Programowanie wielowątkowe.

V. Programowanie zorientowane obiektowo: Klasy, Pola, Pola statyczne, Metody, Metody statyczne, Metody - przeciążanie.

VI. Programowanie zorientowane obiektowo: Metody - Konstruktory, Proporcje/Właściwości, Stałe, Indeksery, Przeciążanie operatorów.

VII. Programowanie zorientowane obiektowo: Dziedziczenie, Polimorfizm, Niszczenie obiektów - destruktor, Niszczenie obiektów - interfejs IDisposable, Interfejsy, Wyjątki, Delegaci, Moduły.

VIII. Kolokwium zaliczeniowe.

Laboratoria:

I. Wprowadzenie do języka C#: Przegląd języka C#, Konfiguracja środowiska programistycznego, Pierwszy program w C#

II. Klasy, obiekty i konstruktory: Definicja klas i obiektów, Tworzenie i używanie konstruktorów,

III. Hermetyzacja i właściwości: Modyfikatorzy dostępu, Gettery i settery, Właściwości automatyczne

IV. Dziedziczenie: Podstawowe zasady dziedziczenia, Klasy bazowe i pochodne, Przesłanianie metod

- V. Polimorfizm: Przeciążanie metod, Przesłanianie metod, Klasy abstrakcyjne i interfejsy
- VI. Delegaty: Definicja i zastosowanie delegatów, Zdarzenia i obsługa zdarzeń
- VII. Operacje na plikach XML: Odczyt i zapis plików XML, Serializacja i deserializacja XML
- VIII. Kolekcje: Listy, słowniki, kolejki, Podstawy LINQ do przetwarzania danych
- IX. Obsługa wyjątków: Mechanizm wyjątków w C#, Standardowe klasy wyjątków, Tworzenie własnych klas wyjątków
- X. Programowanie asynchroniczne: Programowanie współbieżne w C#, async i await - zarządzanie zadaniami, Obsługa wyjątków w kodzie asynchronicznym
- XI. Programowanie wielowątkowe: Wątki w C# (Thread, Task), Synchronizacja wątków (lock, Mutex, Semaphore), Komunikacja między wątkami
- XII. Zaawansowane LINQ i operacje na bazach danych: Zaawansowane zapytania LINQ, Wprowadzenie do Entity Framework, Podstawy komunikacji z bazą danych
- XIII. Wzorce projektowe w C#: Wzorzec Singleton, Factory, Repository, Dependency Injection w .NET, Dobre praktyki w projektowaniu aplikacji
- XIV. Testowanie i debugowanie w C#: Narzędzia do debugowania w Visual Studio, Testy jednostkowe (xUnit, NUnit, MSTest), Tworzenie i uruchamianie testów
- XV. Kolokwium zaliczeniowe

Metody dydaktyczne

Wykład: Prezentacja multimedialna oraz prezentacja pisania i wykonywania wybranych programów bezpośrednio w języku C#.

Ćwiczenia laboratoryjne: Ćwiczenia praktyczne dotyczące elementów języka C#, pisanie programów okienkowych.

Literatura

Podstawowa:

1. Albahari J. C# 9.0 w pigułce, Helion, 2022.
2. Troelsen A., Japikse P. Język C# 6.0 i platforma .NET 4.6, Helion, 2017.
3. Martin R. C. Czysty kod. Podręcznik dobrego programisty, Helion, 2010.

Uzupełniająca:

1. Gamma E., Helm R., Johnson R., Vlissides J. Wzorce projektowe. Elementy oprogramowania obiektowego wielokrotnego użytku, Helion, 2005.
2. Osherove R. Sztuka Unit Testowania. Przykłady w języku C#, Helion, 2014.

Bilans nakładu pracy przeciętnego studenta

	Godzin	ECTS
Łączny nakład pracy	75	3,00
Zajęcia wymagające bezpośredniego kontaktu z nauczycielem	45	2,00
Praca własna studenta (studia literaturowe, przygotowanie do zajęć laboratoryjnych/ćwiczeń, przygotowanie do kolokwium/egzaminu, wykonanie projektu)	30	1,00