



SYLABUS PRZEDMIOTU

Nazwa przedmiotu: Podstawy Programowania

Kod przedmiotu: PRG1

Kierunek / Profil: Informatyka / praktyczny

Tryb studiów: stacjonarny

Rok / Semestr: 1 / 1

Charakter: obowiązkowy

Odpowiedzialny: mgr inż. Adam Urbanowicz

Wersja z dnia: 19.02.2026

1. Godziny zajęć i punkty ECTS

Wykłady	Ćwiczenia	Laboratoria	Z prowadzącym	Praca własna	Łącznie	ECTS
30 h	—	60 h	56 h	94 h	150 h	6

2. Forma zajęć

Forma zajęć Sposób zaliczenia

Laboratorium Zaliczenie z oceną

3. Cel dydaktyczny

Celem zajęć jest opanowanie przez studentów podstaw klasycznych technik programowania strukturalnego na bazie języka C#. Wprowadzone zostają również elementy programowania obiektowego. Zajęcia mają na celu rozwijanie umiejętności abstrakcyjnego myślenia oraz rozwiązywanie prostych problemów programistycznych z wykorzystaniem platformy .NET i aplikacji konsolowych.

4. Treści programowe

1. Pojęcie algorytmu. Sposoby przedstawiania algorytmów. Zmienne, typy, operator przypisania, instrukcje Console.WriteLine i Console.ReadLine, pierwszy program w C#.
2. Wprowadzenie do środowiska .NET i Visual Studio. Tworzenie projektu, edycja kodu źródłowego, komplikacja, uruchamianie i debugowanie aplikacji konsolowych. Implementacja prostych obliczeń matematycznych.
3. Instrukcja warunkowa if, if else, zagnieźdzona instrukcja if. Instrukcja switch.
4. Implementacja programów wymagających użycia instrukcji warunkowych i wielowariantowych.
5. Pętle for, while, do while. Różnice i zastosowania. Zagnieźdzanie pętli. Instrukcje break i continue.
6. Implementacja programów wymagających użycia pętli.
7. Typy całkowite i zmiennoprzecinkowe. Typ znakowy i logiczny. Konwersje typów i rzutowania. Kwalifikator const. Operatory arytmetyczne, relacyjne i logiczne. Priorytety operatorów.
8. Implementacja programów z wykorzystaniem zmiennych różnych typów oraz operatorów.
9. Tablice jednowymiarowe i wielowymiarowe. Inicjalizacja tablic. Podstawowe algorytmy sortowania.
10. Implementacja programów wykorzystujących tablice jednowymiarowe i wielowymiarowe.
11. Metody. Przekazywanie parametrów. Zwracanie wartości. Metody void. Przeciążanie metod. Metody biblioteczne.
12. Definiowanie i stosowanie metod. Implementacja aplikacji wykorzystujących metody.
13. Rekurencja. Wady i zalety stosowania rekurencji. Klasa string i jej metody.
14. Implementacja programów wykorzystujących rekurencję oraz operacje na łańcuchach znakowych.
15. Typy referencyjne i wartościowe. Wprowadzenie do kolekcji: List, ArrayList. Zarządzanie pamięcią w .NET. Garbage Collector.
16. Implementacja programów wykorzystujących kolekcje i dynamiczne struktury danych.
17. Wstęp do testowania jednostkowego

5. Efekty kształcenia

Wiedza

- Wymienia zasady programowania strukturalnego i obiektowego w języku C#, w tym techniki iteracji, rekurencji, organizacji pamięci zarządzanej, obsługi typów wartościowych i referencyjnych. Opisuje podstawowe algorytmy wyszukiwania i sortowania danych.

Umiejętności

- Pozyskuje informacje dotyczące programowania z dokumentacji technicznej, literatury i Internetu. Konstruuje rozwiązania prostych zadań programistycznych w języku C# wymagających użycia instrukcji warunkowych, pętli, rekurencji, metod z odpowiednim sposobem przekazywania parametrów, tablic i kolekcji. Testuje poprawność wykonywanego oprogramowania oraz wykorzystuje zintegrowane środowisko programistyczne do konfigurowania, edytowania i testowania aplikacji konsolowych.

6. Kryteria oceny

- Ćwiczenia / Laboratorium/Lektorat:
- rozwiązywanie zadań programistycznych
- Kryteria oceny
- Ćwiczenia/Laboratorium/Projekt/Lektorat
- Zadania laboratoryjne (czas oddawania: następne zajęcia) - 20%
- Kolokwia (dwa): 60%
- Projekt programistyczny: 20%
- Brak

7. Metody dydaktyczne

Wykład, laboratoria, praca własna studenta.

8. Literatura

Podstawowa:

- Andrew Troelsen, Phil Japikse, Pro C# 10 with .NET 6, Apress, 2022
- Ian Griffiths, Programming C# 10, O'Reilly Media, 2022

Uzupełniająca:

- Jon Skeet, C# in Depth, Fourth Edition, Manning Publications, 2019
- Robert C. Martin, Czysty kod. Podręcznik dobrego programisty, Helion, 2010