



# POLSKO-JAPOŃSKA AKADEMIA TECHNIK KOMPUTEROWYCH

## SYLABUS PRZEDMIOTU

<b>Nazwa przedmiotu:</b>	Programowanie Obiektowe w C#
<b>Kod przedmiotu:</b>	POJ
<b>Kierunek / Profil:</b>	Informatyka / praktyczny
<b>Tryb studiów:</b>	stacjonarny
<b>Rok / Semestr:</b>	1 / 2
<b>Charakter:</b>	obowiązkowy
<b>Odpowiedzialny:</b>	mgr inż. Adam Urbanowicz
<b>Wersja z dnia:</b>	15.02.2025

### 1. Godziny zajęć i punkty ECTS

Wykłady	Ćwiczenia	Laboratori	Z prowa- dzącym	Praca własna	Łącznie	ECTS
30 h	—	30 h	60 h	40 h	100 h	3

### 2. Forma zajęć

Forma zajęć	Sposób zaliczenia
Laboratorium	Zaliczenie z oceną
Wykład	Nieoceniany

### 3. Cel dydaktyczny

Celem zajęć jest zapoznanie studentów z podstawowymi koncepcjami programowania obiektowego na przykładzie języka C# i platformy .NET. Omawiane są pojęcia interfejs, klasa, obiekt, dziedziczenie, polimorfizm, enkapsulacja i kompozycja. Prezentowane są koncepcje i praktyka związana z wykorzystaniem kolekcji generycznych, wyjątków, delegatów i zdarzeń, operacji wejścia/wyjścia oraz wybranych elementów nowoczesnego C# (m.in. LINQ).

## 4. Przedmioty wprowadzające

---

Przedmiot	Wymagane zagadnienia
PRG1	znajomość i umiejętność stosowania podstaw programowania strukturalnego (instrukcja warunkowa, pętle, tablice, funkcje i procedury)

---

## 5. Treści programowe

---

1. Wprowadzenie do programowania w C#: typy danych, definicja zmiennych, instrukcje sterujące. Tworzenie prostych programów z wykorzystaniem metody Main oraz klasy Console.
2. Metody statyczne, tablice i ich praktyczne zastosowanie; interakcja człowiek – komputer z użyciem Console.ReadLine/WriteLine, parsowanie danych wejściowych.
3. Programowanie obiektowe: definicja klas, tworzenie obiektów, konstruktory, pola i właściwości (properties). Implementacja metod niestatycznych.
4. Enkapsulacja, modyfikatory dostępu, zasady przechowywania danych w pamięci: stos i sarta w kontekście .NET; typy wartościowe i referencyjne.
5. Dziedziczenie: override oraz overload, metody wirtualne, przesłanianie składowych, dziedziczenie po klasie object (ToString, Equals, GetHashCode).
6. Klasy abstrakcyjne, polimorfizm, kompozycja; wprowadzenie do przestrzeni nazw i organizacji projektu.
7. Parametry opcjonalne i nazwane, params (zmienna liczba parametrów) oraz omówienie wyjątków (try/catch/finally), własne wyjątki.
8. Wyrażenia regularne w .NET, operacje na plikach (System.IO), serializacja w podstawowym zakresie (np. JSON jako format danych – wprowadzenie).
9. Interfejsy w C#, implementacja wielokrotna interfejsów; interfejs IComparable, wprowadzenie do typów generycznych.
10. Kolekcje generyczne: List, Dictionary, HashSet; iteracja (foreach), porównywanie i sortowanie; wprowadzenie do LINQ (Where/Select/OrderBy).
11. Delegaty i zdarzenia: model zdarzeniowy w C#, Action/Func, event; podstawy programowania zdarzeniowego na prostych przykładach.
12. Omówienie wybranych wzorców projektowych (np. Singleton, Factory, Strategy) oraz wstęp do programowania asynchronicznego w C# (async/await) w kontekście prostych operacji I/O.

## 6. Efekty kształcenia

---

### Wiedza

- Student zna i rozumie podstawowe konstrukcje programistyczne oraz struktury danych, jak też ich podstawowe implementacje w języku C# i środowisku .NET

## Umiejętności

- Student potrafi ocenić poprawność konstrukcji obiektowych w programach w języku C#.
- Potrafi skonstruować i uruchomić program obiektowy w języku C# (aplikacja konsolowa) z wykorzystaniem platformy .NET.
- Student potrafi znajdować błędy w tworzonych programach obiektowych przy użyciu wybranych środowisk uruchomieniowych i debuggera.
- Potrafi korzystać z bibliotek .NET (m.in. System, System.Collections.Generic, System.IO, System.Text.RegularExpressions).
- Student potrafi korzystać ze środowiska developerskiego przeznaczonego do tworzenia programów w języku C# (np. Visual Studio / Rider / VS Code), zaplanować prostą hierarchię klas oraz zastosować proste wzorce obiektowe zależnie od przedstawionego problemu.

## 7. Kryteria oceny

---

- rozwiązywanie zadań
- Kryteria oceny
- Ocena pracy podczas ćwiczenia – ocena zadań wykonanych przez studentów oddanych do końca zajęć.
- Skala ocen:
- Poniżej 50% - ndst
- Od 50% - dst
- Od 60% - dst+
- Od 70% - db
- Od 80% - db+
- Od 90% - bdb
- Brak

## 8. Metody dydaktyczne

---

Wykład, laboratoria, praca własna studenta.

## 9. Literatura

---

**Podstawowa:**

- Brak danych.

**Uzupełniająca:**

- Andrew Troelsen, Phil Japikse - Pro C# with .NET, Apress (najnowsze wydanie)
- Mark J. Price - C# (najnowsze wydanie, Packt)

- Microsoft Learn - Dokumentacja języka C# i platformy .NET
- Jon Skeet - C# in Depth, Manning Publications

