Rozkład materiału — rok 1

3 godziny tygodniowo: 1 h wykładu + 2 h pracowni. 30 wykładów (prezentacje PPTX) i 34 zadania na pracownię (PDF + wzór sprawozdania DOCX). C# 14, .NET 10 LTS, Visual Studio 2026.

Wykłady

Nr	Temat	Zakres
1.01	**Platforma .NET i Visual Studio 2026**	czym jest .NET 10 LTS, CLR, kompilacja do IL; instalacja VS 2026 Community i obciążeń (.NET desktop, .NET MAUI); pierwszy projekt konsolowy; Solution Explorer; uruchamianie F5/Ctrl+F5
1.02	**Struktura programu w C#**	instrukcje najwyższego poziomu i metoda Main; przestrzenie nazw, using, implicit usings; pliki .csproj i .sln/.slnx; komentarze; konwencje nazewnictwa
1.03	**Typy danych i zmienne**	typy wartościowe i referencyjne; int, double, decimal, bool, char, string; var; stałe; konwersje jawne/niejawne; Parse, TryParse, Convert; null i string?
1.04	**Operatory i formatowanie**	operatory arytmetyczne, porównania, logiczne, przypisania złożone; dzielenie całkowite i reszta; klasa Math; interpolacja $"", formaty liczb i dat, CultureInfo pl-PL
1.05	**Instrukcje warunkowe**	if/else if/else; operator warunkowy ?:; switch klasyczny; wyrażenie switch i dopasowanie wzorców (relacyjne, logiczne and/or/not)
1.06	**Pętle**	for, while, do-while, foreach; break, continue; pętle zagnieżdżone; typowe algorytmy: suma, licznik, maksimum; unikanie pętli nieskończonych
1.07	**Tablice**	tablice jednowymiarowe, indeksy, Length; inicjalizacja; tablice wielowymiarowe i postrzępione; Array.Sort, Array.IndexOf; zakresy i indeksy od końca ^1, ..
1.08	**Kolekcje**	List<T>: Add, Remove, Insert, Contains, Count; Dictionary<TKey,TValue>: TryGetValue, ContainsKey; HashSet<T>; Queue<T>, Stack<T>; wyrażenia kolekcji [ ]; wybór kolekcji
1.09	**Metody**	definicja, parametry, zwracanie wartości; void; parametry domyślne i nazwane; ref, out, in; przeciążanie; metody statyczne; rekurencja
1.10	**Łańcuchy znaków**	niezmienność string; metody Substring, IndexOf, Split, Join, Replace, Trim, ToUpper; porównywanie z StringComparison; StringBuilder; char i jego metody; teksty wielowierszowe """
1.11	**Obsługa wyjątków**	try/catch/finally; hierarchia wyjątków; throw; filtrowanie when; własne wyjątki; kiedy nie łapać wyjątków; using i IDisposable
1.12	**Debugowanie w Visual Studio**	punkty przerwania (warunkowe), krokowanie F10/F11, okna Locals/Watch/Call Stack/Immediate; Hot Reload; Debug.WriteLine; czytanie komunikatów błędów i stosu wywołań
1.13	**Klasy i obiekty**	programowanie obiektowe; klasa jako szablon; pola, metody, konstruktory, this; tworzenie obiektów new; referencje; konstruktor podstawowy (primary constructor)
1.14	**Właściwości i hermetyzacja**	modyfikatory dostępu; właściwości get/set/init, auto-właściwości, walidacja w set, pole field; składowe statyczne i licznik instancji; ToString
1.15	**Dziedziczenie**	klasa bazowa i pochodna; base; virtual/override; abstract; sealed; protected; typ object; is/as
1.16	**Polimorfizm i interfejsy**	interfejs jako kontrakt; implementacja wielu interfejsów; IComparable<T>, IEquatable<T>; polimorfizm przez interfejs; metody domyślne interfejsu
1.17	**Rekordy, struktury i wyliczenia**	record i record struct, porównywanie wartościowe, with; struct vs class; enum i [Flags]; typy nullable int? i operatory ?. ?? ??=
1.18	**Delegaty, lambdy i zdarzenia**	delegat; Func, Action, Predicate; wyrażenia lambda; metody jako parametry; zdarzenia event i EventHandler; subskrypcja += i odpinanie
1.19	**LINQ**	zapytania na kolekcjach: Where, Select, OrderBy, GroupBy, Sum, Average, Count, First/FirstOrDefault, Any/All; składnia metod i zapytań; wykonanie odroczone

1.20	**Pliki i JSON**	File.ReadAllLines/WriteAllText, StreamReader/StreamWriter; ścieżki Path.Combine; CSV; System.Text.Json: Serialize/Deserialize; obsługa błędów plików
1.21	**Testy jednostkowe**	po co testy; projekt MSTest w VS; [TestClass], [TestMethod], [DataRow]; Assert.AreEqual/IsTrue/ThrowsException; wzorzec AAA; Test Explorer; xUnit jako alternatywa; wymagania INF.04
1.22	**Aplikacje desktopowe w .NET**	WinForms, WPF, WinUI 3, .NET MAUI — porównanie; architektura WPF; projekt WPF w VS; App.xaml, MainWindow.xaml i code-behind; projektant XAML i Hot Reload
1.23	**Język XAML**	elementy i atrybuty; składnia element-właściwość; przestrzenie nazw xmlns i x:; x:Name; drzewo logiczne; zawartość (Content); właściwości dołączone
1.24	**Układ okna w WPF**	Grid (wiersze, kolumny, *, Auto), StackPanel, DockPanel, WrapPanel, Border, ScrollViewer; Margin, Padding, HorizontalAlignment; skalowanie okna
1.25	**Podstawowe kontrolki WPF**	TextBlock, Label, TextBox, Button, CheckBox, RadioButton, ComboBox, ListBox, Slider, Image, ProgressBar; najważniejsze właściwości
1.26	**Zdarzenia w WPF**	obsługa Click, TextChanged, SelectionChanged, Checked; code-behind; sender i RoutedEventArgs; zdarzenia trasowane (bubbling); IsEnabled, Visibility
1.27	**Walidacja i komunikaty**	sprawdzanie danych wejściowych (TryParse, zakresy, puste pola); MessageBox (przyciski, ikony, wynik); wyróżnianie błędów; komunikaty zrozumiałe dla użytkownika; dostępność
1.28	**Style i zasoby**	Resources w oknie i App.xaml; StaticResource vs DynamicResource; Style z TargetType, Setter, BasedOn; style domyślne; ResourceDictionary; kolory i czcionki aplikacji
1.29	**Wiele okien i okna dialogowe**	tworzenie nowego okna; Show vs ShowDialog; DialogResult; Owner; przekazywanie danych między oknami; OpenFileDialog, SaveFileDialog (Microsoft.Win32)
1.30	**Projektowanie aplikacji**	od wymagań do aplikacji: specyfikacja, makieta okna, podział na klasy logiki i interfejs; nazewnictwo; dokumentacja i zrzuty (INF.04); przygotowanie do projektu rocznego

Zadania na pracownię

Nr	Temat	Wykład	Cel
01	**Pierwszy program w Visual Studio**	1.01	Przygotujesz stanowisko pracy: sprawdzisz instalację Visual Studio 2026 i .NET 10, utworzysz pierwszą aplikację konsolową i zapiszesz ją w lokalnym repozytorium Git.
02	**Dane z klawiatury i obliczenia**	1.02–1.03	Napiszesz dwa programy obliczeniowe, które bezpiecznie wczytują liczby z klawiatury i odrzucają błędne dane.
03	**Konwersje i typy liczbowe**	1.03	Napiszesz przeliczniki jednostek i sprawdzisz w praktyce różnice między typami `int`, `double` i `decimal` oraz sposobami konwersji.
04	**Formatowanie wyników — paragon**	1.04	Wydrukujesz w konsoli czytelny paragon sklepowy z wyrównanymi kolumnami, kwotami w złotych, rabatem i podatkiem VAT.
05	**Warunki i wyrażenie switch**	1.05	Zapiszesz decyzje w programie instrukcją `if`, operatorem warunkowym i wyrażeniem `switch` z wzorcami, dbając o poprawne granice przedziałów.
06	**Pętle**	1.06	Rozwiążesz cztery krótkie problemy, dobierając odpowiedni rodzaj pętli: `for`, `while` lub `do…while`.
07	**Tablice jednowymiarowe — statystyki ocen**	1.07	Obliczysz statystyki ocen klasy zapisanych w tablicy, napiszesz własne sortowanie bąbelkowe i porównasz je z metodą `Array.Sort`.
08	**Tablice dwuwymiarowe — kółko i krzyżyk**	1.07	Napiszesz grę w kółko i krzyżyk dla dwóch graczy w konsoli, przechowując planszę w tablicy dwuwymiarowej.
09	**Kolekcje List i Dictionary**	1.08	Wykorzystasz listę i słownik w trzech programach: liście zakupów z menu, teście słówek i zliczaniu słów w tekście.
10	**Metody i rekurencja**	1.09	Zbudujesz statyczną klasę `Matematyka` z metodami obliczeniowymi w wersjach iteracyjnych i rekurencyjnych oraz użyjesz parametrów `out` i `ref`.
11	**Przetwarzanie tekstu**	1.10	Napiszesz klasę `Tekst` z metodami do sprawdzania palindromów, liczenia samogłosek, szyfrowania szyfrem Cezara i weryfikacji numeru PESEL.
12	**Wyjątki i odporny program**	1.11	Napiszesz kalkulator, który nie kończy się błędem przy żadnych danych, obsługując wyjątki wbudowane i własny wyjątek.
13	**Debugowanie programu**	1.12	Znajdziesz i poprawisz sześć błędów logicznych w gotowym programie, korzystając z debugera Visual Studio zamiast zgadywania.
14	**Pierwsza klasa — konto bankowe**	1.13	Zaprojektujesz klasę `KontoBankowe` z hermetyzacją danych, konstruktorem, metodami wpłaty i wypłaty oraz historią operacji.
15	**Właściwości i składowe statyczne — klasa Kosc**	1.14	Napiszesz klasę `Kosc` w stylu zadania egzaminacyjnego INF.04 — z polem statycznym liczącym obiekty, właściwościami i dokumentacją — oraz program testowy gry w kości.
16	**Dziedziczenie — pracownicy**	1.15	Zbudujesz hierarchię klas pracowników z abstrakcyjną klasą bazową i metodą wirtualną `ObliczPensje`, a następnie przetworzysz listę różnych pracowników jedną pętlą.
17	**Interfejsy i polimorfizm — figury**	1.16	Zdefiniujesz interfejs `IFigura` i kilka implementujących go klas, a następnie posortujesz listę figur według pola dzięki interfejsowi `IComparable<T>`.
18	**Rekordy i wyliczenia — kawiarnia**	1.17	Zamodelujesz zamówienia w kawiarni za pomocą rekordów i typów wyliczeniowych, z kontrolą

			dozwolonych zmian statusu zamówienia.
19	**Delegaty i zdarzenia — czujnik temperatury**	1.18	Zbudujesz klasę czujnika, która powiadamia inne części programu o nowych pomiarach i przekroczeniu progu za pomocą zdarzeń, oraz napiszesz filtr przyjmujący warunek jako parametr `Func<T,bool>`.
20	**LINQ w praktyce — wyniki uczniów**	1.19	Przeanalizujesz listę uczniów i ich ocen zapytaniami LINQ: filtrowaniem, sortowaniem, grupowaniem i agregacją.
21	**Pliki CSV i JSON — dziennik wydatków**	1.20	Wczytasz dziennik wydatków z pliku CSV z pominięciem błędnych wierszy, obliczysz sumy w kategoriach, zapiszesz raport oraz wyeksportujesz dane do JSON i odczytasz je z powrotem.
22	**Testy jednostkowe MSTest**	1.21	Utworzysz projekt testów MSTest dla klas `Kosc` (zadanie 15) i `Matematyka` (zadanie 10), korzystając z testów parametryzowanych i testów wyjątków.
23	**Aplikacja konsolowa jak na egzaminie — parking**	1.13–1.21	W 90 minut wykonasz zadanie w stylu części konsolowej egzaminu INF.04: klasę z polem statycznym i walidacją, program główny, testy jednostkowe i dokumentację ze zrzutami.
24	**Pierwsza aplikacja WPF**	1.22–1.23	Utworzysz pierwszą aplikację okienkową WPF, poznasz pliki projektu i projektant XAML oraz obsłużysz kliknięcie przycisku.
25	**Układ okna formularza**	1.24	Zbudujesz okno formularza rejestracji, które poprawnie zachowuje się przy zmianie rozmiaru, używając kontenerów Grid, StackPanel i DockPanel.
26	**Kontrolki wyboru — konfigurator pizzy**	1.25	Zbudujesz konfigurator pizzy z przyciskami opcji, polami wyboru, listą rozwijaną i suwakiem, a cenę obliczy osobna klasa logiki.
27	**Kalkulator WPF**	1.26	Zbudujesz kalkulator z przyciskami cyfr i działań, w którym jedna metoda obsługuje wszystkie przyciski, a stan kalkulatora przechowuje osobna klasa.
28	**Walidacja formularza**	1.27	Zbudujesz formularz zgłoszenia, który sprawdza wszystkie pola naraz, wyróżnia błędne pola i pokazuje czytelne komunikaty; reguły walidacji sprawdzisz testami jednostkowymi.
29	**Style i zasoby**	1.28	Ujednolicisz wygląd aplikacji z zadania 26 za pomocą zasobów i stylów w App.xaml oraz dodasz przełącznik motywu jasnego i ciemnego.
30	**Wiele okien i pliki — notatnik**	1.29	Napiszesz prosty notatnik z menu Plik, systemowymi oknami Otwórz i Zapisz, oknem „O programie” oraz pytaniem o zapis zmian przy zamykaniu.
31	**Gra w kości w WPF**	1.22–1.29	Zbudujesz aplikację okienkową gry w kości w stylu części desktopowej egzaminu INF.04, wykorzystując klasę `Kosc` z zadania 15 i obrazy kości.
32	**Projekt roczny — specyfikacja**	1.30	Przygotujesz specyfikację własnej aplikacji WPF: wymagania, makiety okien, projekt klas, plan testów i harmonogram.
33	**Projekt roczny — implementacja**	1.30	Zrealizujesz aplikację WPF zgodnie ze specyfikacją z zadania 32, pracując małymi krokami z historią w Git, i przeprowadzisz przegląd kodu w parach.
34	**Projekt roczny — testy i prezentacja**	1.30	Przetestujesz aplikację testami jednostkowymi i testami ręcznymi, przygotujesz dokumentację w stylu INF.04 i zaprezentujesz projekt.