Rozkład materiału — rok 2

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
2.01	**Wiązanie danych**	Binding: źródło, ścieżka, tryby OneWay/TwoWay/OneTime, UpdateSourceTrigger; DataContext; ElementName; INotifyPropertyChanged
2.02	**Kolekcje w interfejsie**	ObservableCollection<T>; ItemsSource; ListBox, ListView, ComboBox z danymi; DataTemplate; DisplayMemberPath; SelectedItem
2.03	**DataGrid**	DataGrid: AutoGenerateColumns, kolumny Text/CheckBox/ComboBox/Template; sortowanie, edycja, zaznaczenie; formatowanie StringFormat
2.04	**Konwertery i walidacja w wiązaniu**	IValueConverter; StringFormat; ValidationRule, INotifyDataErrorInfo; Validation.ErrorTemplate; wyświetlanie błędów
2.05	**Wzorzec MVVM**	Model–View–ViewModel; ViewModel jako DataContext; ICommand i RelayCommand; CanExecute; brak logiki w code-behind; zalety dla testów
2.06	**CommunityToolkit.Mvvm**	pakiet NuGet; ObservableObject, [ObservableProperty], [RelayCommand], generatory kodu; partial; NotifyPropertyChangedFor; komendy asynchroniczne
2.07	**Widoki i nawigacja**	UserControl jako widok; ContentControl i DataTemplate dla ViewModelu; nawigacja między widokami w MVVM; komunikaty (Messenger)
2.08	**Menu, karty i paski**	Menu, ContextMenu, ToolBar, StatusBar, TabControl, Expander, TreeView; skróty klawiszowe InputBindings; ikony
2.09	**Grafika i animacje**	Canvas, Shape (Rectangle, Ellipse, Line, Path); pędzle i gradienty; transformacje; Storyboard i DoubleAnimation; DispatcherTimer
2.10	**Szablony i wyzwalacze**	ControlTemplate, TemplateBinding; Trigger, DataTrigger, MultiTrigger; VisualStateManager; własny wygląd przycisku
2.11	**Programowanie asynchroniczne**	wątek interfejsu; Task, async/await; Task.Run dla obliczeń; IProgress<T>; CancellationToken; typowe błędy (async void, .Result)
2.12	**SQLite i ADO.NET**	baza w pliku; Microsoft.Data.Sqlite; połączenie, polecenie, parametry, DataReader; CREATE TABLE z kodu; ochrona przed wstrzykiwaniem SQL przez parametry
2.13	**Entity Framework Core**	ORM; DbContext, DbSet; konfiguracja SQLite; migracje (Add-Migration, Update-Database); CRUD; SaveChanges
2.14	**EF Core: relacje i zapytania**	relacje 1:N i N:M; właściwości nawigacyjne; Include; LINQ to Entities; AsNoTracking; wyświetlanie w DataGrid
2.15	**Wstrzykiwanie zależności**	luźne powiązania; interfejsy usług; wzorzec repozytorium; Microsoft.Extensions.DependencyInjection w WPF; czas życia Singleton/Transient
2.16	**Aplikacja i REST API**	HttpClient; GetFromJsonAsync; modele DTO; obsługa błędów sieci i limitu czasu; publiczne API (np. NBP); asynchroniczne ładowanie danych
2.17	**Ustawienia i dzienniki**	zapis ustawień użytkownika w JSON w AppData; Environment.GetFolderPath; ILogger i logowanie do pliku/okna Debug; obsługa nieprzechwyconych wyjątków
2.18	**Testowanie aplikacji MVVM**	testy jednostkowe ViewModeli; atrapy (fake) usług przez interfejsy; testy parametryzowane; pokrycie kodu; testy w INF.04
2.19	**Lokalizacja i dostępność**	pliki .resx i wersje językowe; CultureInfo; dostępność: AutomationProperties.Name, nawigacja klawiaturą TabIndex, kontrast, skalowanie
2.20	**Publikowanie aplikacji**	konfiguracje Debug/Release; dotnet publish; framework-dependent vs self-contained; single-file; ikona i wersja; MSIX i instalatory

2.21	**Git w Visual Studio**	repozytorium, commit, gałęzie, scalanie, konflikty; GitHub; pull request; .gitignore dla Visual Studio; dobre komunikaty commitów
2.22	**.NET MAUI — wprowadzenie**	jedna baza kodu dla Android/iOS/Windows/macOS; struktura projektu (Platforms, Resources); MauiProgram; emulator Androida w VS; uruchamianie na Windows
2.23	**MAUI: strony, układy i kontrolki**	ContentPage; Grid, VerticalStackLayout, HorizontalStackLayout, FlexLayout; Label, Entry, Editor, Button, Image, Switch, CheckBox, Slider, Picker
2.24	**MAUI: zdarzenia, obrazy i style**	obsługa Clicked, TextChanged, Toggled; TapGestureRecognizer; obrazy w Resources/Images; czcionki; Style i ResourceDictionary; Opacity
2.25	**MAUI: MVVM i wiązanie danych**	BindingContext; CommunityToolkit.Mvvm w MAUI; komendy; x:DataType i kompilowane wiązania
2.26	**MAUI: nawigacja Shell**	AppShell; TabBar, Flyout; trasy Routing.RegisterRoute; GoToAsync; przekazywanie parametrów QueryProperty; przycisk Wstecz
2.27	**MAUI: listy danych**	CollectionView, ItemTemplate, SelectionMode; EmptyView; RefreshView; SwipeView; grupowanie
2.28	**MAUI: dane i funkcje urządzenia**	Preferences, SecureStorage, pliki w FileSystem.AppDataDirectory; SQLite w MAUI; uprawnienia; funkcje urządzenia (Connectivity, Geolocation, MediaPicker)
2.29	**Egzamin INF.04 w praktyce**	budowa arkusza: aplikacja konsolowa, desktopowa lub mobilna, testy, dokumentacja; zasady oceniania; planowanie 180 minut; zrzuty i archiwa; typowe błędy
2.30	**Jakość i bezpieczeństwo kodu**	czytelny kod i refaktoryzacja; zasady SOLID w praktyce; walidacja danych; hasła tylko jako skrót (PBKDF2); sekrety poza kodem; przegląd kodu

Zadania na pracownię

Nr	Temat	Wykład	Cel
01	**Wiązanie danych — przelicznik jednostek**	2.01	Zbudujesz przelicznik odległości i temperatury, w którym kontrolki nie mają ani jednej metody obsługi zdarzeń — wszystko łączy wiązanie danych z modelem implementującym INotifyPropertyChanged.
02	**Lista z ObservableCollection — ToDo**	2.02	Zbudujesz listę zadań z dodawaniem, usuwaniem i oznaczaniem wykonanych, korzystając z ObservableCollection, szablonu elementu i licznika aktualizowanego automatycznie.
03	**Katalog w DataGrid**	2.03	Zbudujesz katalog książek w DataGrid z polskimi nagłówkami, formatowaniem walut, edycją, filtrowaniem tekstem i podsumowaniem pod tabelą.
04	**Konwertery i walidacja w wiązaniu**	2.04	Zbudujesz formularz rejestracji, który waliduje dane w modelu przez INotifyDataErrorInfo, pokazuje komunikaty pod polami i używa konwerterów do koloru i widoczności.
05	**MVVM od podstaw — kalkulator BMI**	2.05	Zbudujesz kalkulator BMI we wzorcu MVVM z własną klasą RelayCommand: okno bez kodu w zdarzeniach, ViewModel bez typów WPF i testy jednostkowe ViewModelu.
06	**CommunityToolkit.Mvvm — ToDo**	2.06	Przepiszesz listę zadań z zadania 2 na MVVM z pakietem CommunityToolkit.Mvvm: właściwości i komendy wygeneruje generator kodu, a okno nie będzie miało żadnej metody obsługi zdarzeń.
07	**Widoki i nawigacja w MVVM**	2.07	Zbudujesz aplikację z menu bocznym i trzema widokami UserControl przełączanymi przez ViewModel, a widoki wymienią dane komunikatami Messenger.
08	**Menu, karty i skróty — edytor notatek**	2.08	Zbudujesz edytor notatek z menu, paskiem narzędzi, paskiem stanu i kartami TabControl, w którym menu, przyciski i skróty klawiszowe wywołują te same komendy.
09	**Grafika i animacja — zegar analogowy**	2.09	Narysujesz zegar analogowy na Canvas z gradientem i transformacjami obrotu, napędzany DispatcherTimer, oraz dodasz animację Storyboard.
10	**Własny wygląd kontrolek**	2.10	Zaprojektujesz własne szablony przycisku i przełącznika ze wszystkimi stanami oraz pokolorujesz wiersze tabeli wyzwalaczami danych.
11	**Operacje asynchroniczne**	2.11	Zbudujesz aplikację liczącą liczby pierwsze w dużym zakresie bez zawieszania interfejsu: z paskiem postępu, anulowaniem i pomiarem czasu.
12	**SQLite i ADO.NET — książka adresowa**	2.12	Zbudujesz książkę adresową z bazą SQLite: tabela tworzona z kodu, dodawanie, edycja, usuwanie i wyszukiwanie wyłącznie przez polecenia z parametrami.
13	**Entity Framework Core — CRUD**	2.13	Zbudujesz bibliotekę książek opartą na EF Core z SQLite: model, kontekst, migracje i pełne operacje CRUD z interfejsu.
14	**EF Core — relacje: dziennik ocen**	2.14	Zbudujesz dziennik ocen z trzema powiązanymi encjami: listą uczniów ze średnimi, szczegółami ocen wybranego ucznia i średnimi przedmiotów liczonymi w bazie.
15	**Wstrzykiwanie zależności i repozytorium**	2.15	Zrefaktoryzujesz dziennik z zadania 14: dostęp do danych za interfejsem repozytorium, okna dialogowe za interfejsem usługi, obiekty tworzone przez kontener DI.
16	**Klient REST API — kursy walut NBP**	2.16	Zbudujesz aplikację pobierającą aktualną tabelę kursów walut z API Narodowego Banku Polskiego, z przelicznikiem, historią notowań i odporną na brak sieci.
17	**Ustawienia i dziennik zdarzeń**	2.17	Dodasz do swojej aplikacji zapamiętywanie ustawień użytkownika w pliku JSON w AppData, listę ostatnich

			plików, dziennik zdarzeń i przechwytywanie nieobsłużonych wyjątków.
18	**Testy ViewModeli**	2.18	Napiszesz testy jednostkowe ViewModeli z zadań 1, 5 i 15 — z powiadomieniami, komendami i atrapami usług — i ocenisz pokrycie kodu.
19	**Lokalizacja i dostępność**	2.19	Przygotujesz polską i angielską wersję aplikacji ToDo z zadania 6 w plikach zasobów oraz poprawisz jej dostępność: nazwy dla czytnika ekranu, kolejność klawisza Tab, kontrast i skalowanie.
20	**Publikowanie aplikacji**	2.20	Przygotujesz wydanie aplikacji z zadania 14/15: metadane, ikonę, dwie wersje publikacji (zależną od .NET i samodzielną), instrukcję dla użytkownika — i sprawdzisz je na innym komputerze.
21	**Praca zespołowa z Git**	2.21	W parach poprowadzicie wspólne repozytorium na GitHub: gałęzie funkcji, pull requesty z przeglądem, kontrolowany konflikt scalania i ochronę gałęzi main.
22	**Pierwsza aplikacja .NET MAUI**	2.22	Utworzysz pierwszą aplikację .NET MAUI, uruchomisz ją na Windows i w emulatorze Androida oraz zmienisz stronę główną na licznik kliknięć.
23	**Układy i kontrolki MAUI — rezerwacja**	2.23	Zbudujesz mobilny formularz rezerwacji sali gier z kontrolkami MAUI, walidacją w osobnej klasie i podsumowaniem z ceną.
24	**Gra w kości w MAUI (styl INF.04)**	2.24	Zbudujesz mobilną grę w kości według specyfikacji w stylu arkusza INF.04: obrazy kości, rzut, blokowanie dotknięciem z przezroczystością, wynik losowania i gry — z klasą Kosc z roku 1.
25	**MAUI i MVVM — lista zakupów**	2.25	Zbudujesz listę zakupów w .NET MAUI we wzorcu MVVM z CommunityToolkit.Mvvm: ViewModel wstrzykiwany przez kontener, kompilowane wiązania x:DataType, usuwanie przesunięciem.
26	**Nawigacja Shell — sklep**	2.26	Zbudujesz aplikację sklepu z TabBar (Produkty, Koszyk, O sklepie), stroną szczegółów otwieraną trasą z parametrem i koszykiem współdzielonym przez usługę.
27	**CollectionView — katalog produktów**	2.27	Zbudujesz katalog produktów z CollectionView: szablon z obrazem, wyszukiwanie, grupowanie po kategorii, odświeżanie gestem i usuwanie przesunięciem z potwierdzeniem.
28	**Dane w aplikacji mobilnej — notatnik**	2.28	Zbudujesz notatnik mobilny: notatki w bazie SQLite, ustawienia w Preferences, informacja o braku internetu i udostępnianie notatki.
29	**Motywy i różne ekrany**	2.29	Przygotujesz aplikację MAUI działającą w motywie jasnym i ciemnym (systemowym lub wybranym) oraz zmieniającą układ przy obrocie ekranu.
30	**Próbny egzamin INF.04 — część 1: konsola i testy**	2.29	W 90 minut wykonasz część konsolową próbnego arkusza: klasę KodRabatowy według specyfikacji, program testujący, testy jednostkowe i dokumentację.
31	**Próbny egzamin INF.04 — część 2: aplikacja z interfejsem**	2.29	W 90 minut wykonasz część z interfejsem próbnego arkusza: aplikację kasy (WPF lub MAUI — do wyboru) według makiety i specyfikacji, korzystającą z klasy z części 1, z dokumentacją.
32	**Projekt końcowy — specyfikacja**	2.30	W zespole przygotujecie specyfikację aplikacji MVVM z bazą danych: wymagania, makiety, architekturę warstw, plan testów i podział pracy w Git.
33	**Projekt końcowy — implementacja**	2.30	Zrealizujecie projekt zgodnie ze specyfikacją: warstwa danych, ViewModele, widoki i testy — w małych krokach z pull requestami i przeglądem kodu.
34	**Projekt końcowy — publikacja i obrona**	2.30	Przygotujecie wydanie projektu (publikacja, dokumentacja, testy na innym komputerze) i obronicie

rozwiązanie przed klasą i nauczycielem.