Ustawienia i dziennik zdarzeń

Czas: 2 × 45 min | praca indywidualna | Wykład 2.17

Cel: Dodasz do swojej aplikacji zapamiętywanie ustawień użytkownika w pliku JSON w AppData, listę ostatnich plików, dziennik zdarzeń i przechwytywanie nieobsłużonych wyjątków.

Po zajęciach potrafisz:

- zapisuje i odczytuje ustawienia w folderze danych aplikacji
- odpornie obsługuje brak i uszkodzenie pliku ustawień
- prowadzi dziennik zdarzeń z poziomami
- przechwytuje nieobsłużone wyjątki aplikacji WPF

Sprzęt i materiały

- komputer z Windows 11 i dostępem do internetu
- pendrive lub konto w repozytorium do zapisania projektu

Oprogramowanie

- **Visual Studio Community 2026 (obciążenie „Programowanie aplikacji klasycznych platformy .NET")** — darmowe dla uczniów i nauczycieli; w pracowni zainstalowane visualstudio.microsoft.com/pl/downloads
- **.NET 10 SDK (LTS)** — instalowany razem z Visual Studio; sprawdzenie: `dotnet --version` dotnet.microsoft.com/pl-pl/download/dotnet/10.0

Przygotowanie przed zajęciami

- Notatnik z roku 1 (zadanie 30) lub edytor notatek z zadania 8.

Bezpieczeństwo

- Zapisuj projekt w swoim folderze (np. `C:\Users\<login>\source\repos`), a na koniec zajęć skopiuj go na pendrive lub wyślij do repozytorium.
- Nie uruchamiaj programów i pakietów z nieznanych źródeł; pakiety NuGet instaluj tylko te, które wskazuje zadanie.
- Rób przerwy dla oczu (zasada 20-20-20) — praca z projektantem okien wymaga długiego patrzenia w ekran.
- Nie zapisuj w dzienniku haseł, numerów PESEL ani treści notatek użytkownika.

Przebieg zadania

1 Ścieżki i ustawienia

Klasy bez WPF.

1. `Sciezki`: folder w `ApplicationData` (ustawienia) i w `LocalApplicationData` (dzienniki), tworzone przy pierwszym użyciu.
2. `Ustawienia`: szerokość, wysokość, zmaksymalizowane, ciemny motyw, lista ostatnich plików; metoda `DodajOstatni` (bez duplikatów, max 5).
3. `UstawieniaService`: Wczytaj (brak pliku / uszkodzony JSON → domyślne), Zapisz.

2 Dziennik

Klasa statyczna `Dziennik`.

```
public static void Blad(string t, Exception ex) =>
    Zapisz("BŁĄD", $"{t}: {ex.GetType().Name}: {ex.Message}{Environment.NewLine}
{ex.StackTrace}");
```

1. Metody Info, Ostrzezenie, Blad (z typem wyjątku i stosem wywołań).
2. Linia: data z milisekundami, poziom, tekst; zapis do pliku z `lock` i do okna Output.
3. Błąd zapisu dziennika nie może zatrzymać programu.

3 Połączenie z aplikacją

App.xaml.cs (instrukcja w pliku INSTRUKCJA.txt rozwiązania).

1. Usuń StartupUri; w OnStartup wczytaj ustawienia i utwórz okno z zapamiętanym rozmiarem i stanem.
2. Przy zamykaniu zapisz rozmiar (`RestoreBounds` dla zmaksymalizowanego okna) i stan.
3. Menu Plik → Ostatnie z listą plików; brakujący plik — komunikat i usunięcie z listy.
4. `DispatcherUnhandledException`, `TaskScheduler.UnobservedTaskException`, `AppDomain.UnhandledException` — wpis do dziennika i komunikat.

4 Sprawdzenie

Scenariusze.

1. Zmień rozmiar okna, zamknij, uruchom — rozmiar zapamiętany.
2. Zepsuj ręcznie plik ustawień — program startuje z domyślnymi i zapisuje ostrzeżenie.
3. Wywołaj wyjątek przyciskiem testowym (tylko Debug) — wpis w dzienniku, program działa dalej.
4. Zatwierdź pracę w Git.

W sprawozdaniu umieść

- ☐ kod klas Sciezki, Ustawienia, Dziennik
- ☐ zawartość pliku ustawień
- ☐ fragment dziennika
- ☐ wyniki scenariuszy

Pytania kontrolne

1. Dlaczego ustawień nie zapisujemy w folderze programu?
2. Czym różnią się foldery ApplicationData i LocalApplicationData?
3. Po co lock w metodzie zapisu dziennika?
4. Czy DispatcherUnhandledException zastępuje try/catch? Uzasadnij.

Kryteria oceny

Kryterium	Punkty
Ustawienia z odpornym odczytem	5
Lista ostatnich plików	3
Dziennik	4
Połączenie z aplikacją i wyjątki	5
Pytania	3
Razem	**20**

Sprawozdanie przygotuj na wzorze dołączonym do zadania (plik .docx o tym samym numerze).