Walidacja formularza

Czas: 2 × 45 min | praca indywidualna | Wykład 1.27

Cel: Zbudujesz formularz zgłoszenia, który sprawdza wszystkie pola naraz, wyróżnia błędne pola i pokazuje czytelne komunikaty; reguły walidacji sprawdzisz testami jednostkowymi.

Po zajęciach potrafisz:

- wydziela reguły walidacji do klasy testowalnej
- pokazuje wszystkie błędy jednocześnie (MessageBox i komunikaty pod formularzem)
- wyróżnia błędne pola ramką i podpowiedzią ToolTip
- pisze testy dla wartości granicznych reguł

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

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.

Przebieg zadania

1 Klasa Walidator

Utwórz projekt WPF `Zgloszenie` i statyczną klasę `Walidator`.

1. Metody zwracające komunikat błędu albo `null`: `SprawdzImie` (min. 2 litery), `SprawdzEmail` (wyrażenie regularne z lekcji), `SprawdzWiek` (13–120), `SprawdzZgode`.
2. Metoda `Sprawdz(...)` zwraca listę wszystkich błędów.
3. Komunikaty mają mówić, jak poprawić dane, np. „Wiek musi być liczbą od 13 do 120.".

2 Formularz

Okno na Gridzie.

```
private void Oznacz(Control pole, string? blad)
{
    pole.BorderBrush = blad is null ? _ramkaDomyslna : Brushes.Red;
    pole.ToolTip = blad;
}
```

1. Pola: imię, e-mail, wiek (wąskie, `MaxLength="3"`), CheckBox zgody, TextBlock na błędy (kolor czerwony), przyciski Wyczyść i Wyślij.
2. Po kliknięciu Wyślij: oznacz każde pole (czerwona ramka + ToolTip z opisem albo przywrócenie domyślnej ramki).
3. Gdy są błędy — pokaż je w TextBlocku i w jednym `MessageBox` z ikoną ostrzeżenia; w przeciwnym razie podziękowanie z ikoną informacji.
4. Domyślny kolor ramki zapamiętaj w zdarzeniu `Loaded` okna.

3 Czyszczenie

Przycisk Wyczyść.

1. Zapytaj `MessageBox` YesNo „Wyczyścić formularz?”.
2. Po potwierdzeniu wyczyść pola, odznacz zgodę, usuń oznaczenia błędów i ustaw kursor w polu imienia.

4 Testy jednostkowe

Projekt MSTest dla klasy Walidator (klasę przenieś do biblioteki klas lub dodaj plik jako link).

1. Test: poprawne dane — 0 błędów; puste dane — 4 błędy.
2. Test parametryzowany e-mail dla 4 adresów (poprawny, bez domeny, ze spacją, bez kropki).
3. Test parametryzowany wieku dla 12, 13, 120, 121 i tekstu.
4. Zrzut Eksploratora testów. Zatwierdź pracę w Git.

W sprawozdaniu umieść

- ☐ kod klasy Walidator i testów
- ☐ zrzut formularza z oznaczonymi błędami
- ☐ zrzut MessageBox
- ☐ zrzut Eksploratora testów

Pytania kontrolne

1. Dlaczego walidację umieszczamy w osobnej klasie?
2. Dlaczego pokazujemy wszystkie błędy naraz?
3. Jak przywrócić pierwotny wygląd pola po poprawieniu błędu?
4. Jakie kontrolki mogłyby zastąpić TextBox wieku, aby uniknąć błędów?

Kryteria oceny

Kryterium	Punkty
Klasa Walidator z czytelnymi komunikatami	5
Oznaczanie pól i komunikaty	5
Czyszczenie z potwierdzeniem	2
Testy jednostkowe (min. 10 przypadków)	5
Pytania	3
Razem	**20**

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