Przetwarzanie tekstu

Czas: 2 × 45 min | praca indywidualna | Wykład 1.10

Cel: Napiszesz klasę `Tekst` z metodami do sprawdzania palindromów, liczenia samogłosek, szyfrowania szyfrem Cezara i weryfikacji numeru PESEL.

Po zajęciach potrafisz:

- przetwarza napisy znak po znaku i metodami klasy `string`
- korzysta z `StringBuilder` przy budowaniu napisu
- implementuje algorytm sumy kontrolnej
- testuje metody na danych poprawnych i błędnych

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.
- Numer PESEL to dane osobowe — w testach używaj wyłącznie numerów przykładowych.

Przebieg zadania

1 Palindrom i samogłoski

Utwórz projekt `Teksty` i statyczną klasę `Tekst`.

1. Metoda `CzyPalindrom(string s)` ignoruje wielkość liter, spacje i znaki interpunkcyjne (zostaw tylko `char.IsLetterOrDigit`).
2. Sprawdź: „Kajak", „Kobyła ma mały bok", „Ala", „Programista".
3. Metoda `Samogloski(string s)` liczy samogłoski z polskimi znakami: a ą e ę i o ó u y.
4. Sprawdź na „Zażółć gęślą jaźń" (wynik: 5).

2 Szyfr Cezara

Każdą literę alfabetu łacińskiego przesuń o zadaną liczbę pozycji; inne znaki pozostaw.

```
if (z is >= 'a' and <= 'z')
    sb.Append((char)('a' + ((z - 'a' + przes) % 26 + 26) % 26));
```

1. Napisz `Cezar(string s, int przes)` z użyciem `StringBuilder`.
2. Obsłuż małe i wielkie litery oraz zawijanie (z → c dla przesunięcia 3).
3. Deszyfrowanie to szyfrowanie z przesunięciem ujemnym — upewnij się, że działa (reszta z dzielenia liczb ujemnych!).
4. Zaszyfruj i odszyfruj zdanie z wielkimi literami, cyframi i znakami interpunkcyjnymi.

3 PESEL

Metoda `PeselPoprawny(string p)` sprawdza długość, cyfry i cyfrę kontrolną.

1. Sprawdź, czy napis ma 11 znaków i same cyfry (`char.IsAsciiDigit`).
2. Oblicz sumę iloczynów pierwszych 10 cyfr przez wagi 1, 3, 7, 9, 1, 3, 7, 9, 1, 3.
3. Cyfra kontrolna = (10 − suma mod 10) mod 10; porównaj z ostatnią cyfrą.
4. Sprawdź przykładowy numer z dokumentacji MSWiA `44051401359` (poprawny) i ten sam ze zmienioną ostatnią cyfrą.
5. Nie używaj prawdziwych numerów PESEL swoich ani kolegów.

4 Menu programu

Połącz metody w program z menu.

1. Użytkownik wybiera funkcję i wpisuje tekst; program wypisuje wynik.
2. Zatwierdź pracę w Git.

W sprawozdaniu umieść

- ☐ kod klasy Tekst
- ☐ tabela testów wszystkich metod
- ☐ wyjaśnienie wzoru `((z - 'a' + przes) % 26 + 26) % 26`

Pytania kontrolne

1. Dlaczego `string` w C# jest niezmienny i jakie ma to znaczenie przy sklejaniu w pętli?
2. Co zwraca `-1 % 26` w C# i dlaczego dodajemy 26?
3. Jakie informacje zawiera numer PESEL poza cyfrą kontrolną?
4. Dlaczego szyfr Cezara nie jest bezpieczny?

Kryteria oceny

Kryterium	Punkty
Palindrom i samogłoski	4
Szyfr Cezara z zawijaniem i deszyfrowaniem	5
Weryfikacja PESEL	5
Menu i testy	3
Pytania	3
Razem	**20**

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