Pętle

Czas: 2 × 45 min | praca indywidualna | Wykład 1.06

Cel: Rozwiążesz cztery krótkie problemy, dobierając odpowiedni rodzaj pętli: `for`, `while` lub `do…while`.

Po zajęciach potrafisz:

- dobiera rodzaj pętli do problemu
- stosuje pętle zagnieżdżone
- przerywa pętlę instrukcją `break` i pomija krok `continue`
- wyszukuje błędy typu „o jeden za dużo” (off-by-one)

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.

Przebieg zadania

1 Tabliczka mnożenia

Wypisz tabliczkę mnożenia 10 × 10 w równych kolumnach.

1. Utwórz projekt `Petle`. Użyj dwóch zagnieżdżonych pętli `for`.
2. Każdą liczbę wypisz w polu o szerokości 4 (`{i * j,4}`).
3. Zmień program tak, aby rozmiar tabliczki podawał użytkownik (2–20).

2 Suma cyfr

Wczytaj liczbę naturalną i oblicz sumę jej cyfr.

1. Użyj pętli `do…while`: dodawaj `n % 10`, potem dziel `n /= 10`.
2. Dlaczego `do…while`, a nie `while`? Sprawdź dla liczby 0.
3. Użyj typu `long`, aby program działał dla dużych liczb.

3 Liczby pierwsze

Wypisz wszystkie liczby pierwsze od 2 do podanej granicy i ich liczbę.

1. Dla każdej liczby `k` sprawdzaj dzielniki `d` od 2, dopóki `d * d <= k`.
2. Po znalezieniu dzielnika przerwij pętlę `break`.
3. Sprawdź wynik dla granicy 50 (powinno być 15 liczb).
4. Zmierz „na oko", ile trwa obliczenie do 1 000 000 z warunkiem `d * d <= k` i z `d < k`.

4 Gra „Zgadnij liczbę"

Program losuje liczbę 1–100, a użytkownik zgaduje z podpowiedziami „za mało / za dużo".

```
var los = new Random();
int tajna = los.Next(1, 101);   // 1..100 — górna granica nie jest losowana!
```

1. Losuj liczbę: `var los = new Random(); int tajna = los.Next(1, 101);`
2. Użyj pętli `do…while` kończącej się po trafieniu; licz próby.
3. Gdy użytkownik wpisze tekst niebędący liczbą, wypisz komunikat i użyj `continue` (nie licz próby).
4. Po trafieniu wypisz liczbę prób. Ile prób wystarczy przy najlepszej strategii? Uzasadnij.
5. Zatwierdź pracę w Git.

W sprawozdaniu umieść

- ☐ kod programu
- ☐ zrzuty ekranu każdej części
- ☐ porównanie czasu sprawdzania liczb pierwszych
- ☐ odpowiedź: liczba prób przy najlepszej strategii

Pytania kontrolne

1. Czym różni się pętla `while` od `do…while`?
2. Dlaczego przy sprawdzaniu liczby pierwszej wystarczy warunek `d * d <= k`?
3. Co robi `break`, a co `continue`?
4. Dlaczego `los.Next(1, 100)` nigdy nie wylosuje liczby 100?

Kryteria oceny

Kryterium	Punkty
Tabliczka mnożenia z rozmiarem od użytkownika	4
Suma cyfr (do...while)	3
Liczby pierwsze z break	5
Gra z licznikiem prób i obsługą błędów	5
Sprawozdanie i pytania	3
Razem	**20**

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