Cwiczenia z C# do wykladu  
  
**Dodawanie**  
**Zadanie:** Napisz metodę, która przyjmuje dwie tablice liczb całkowitych o różnej długości i zwraca nową tablicę, w której każdy element jest sumą odpowiadających sobie elementów obu tablic. Jeśli jedna tablica jest krótsza, uzupełnij brakujące elementy zerami.  
**Przykład:** [1, 2, 3] + [4, 5] = [5, 7, 3]

1. **Odejmowanie**  
   **Zadanie:** Stwórz metodę, która odejmuje dwie liczby zmiennoprzecinkowe podane jako ciągi znaków (string). Metoda powinna obsługiwać wyjątki (np. błędny format) i zaokrąglać wynik do 2 miejsc po przecinku.  
   **Przykład:** "5.75" - "2.1" = 3.65
2. **Mnozenie**  
   **Zadanie:** Zaimplementuj mnożenie macierzy 2D (reprezentowanych jako int[,]). Metoda powinna sprawdzać, czy mnożenie jest możliwe (zgodność wymiarów), i zwracać wynik jako nową macierz.  
   **Przykład:** [[1, 2], [3, 4]] \* [[5, 6], [7, 8]] = [[19, 22], [43, 50]]
3. **Dzielenie**  
   **Zadanie:** Napisz metodę, która dzieli dwie liczby całkowite i zwraca wynik jako krotkę (wynik, reszta). Obsłuż przypadek dzielenia przez zero, rzucając odpowiedni wyjątek z komunikatem.  
   **Przykład:** 10 / 3 = (3, 1)
4. **Inkrementuj**  
   **Zadanie:** Zmodyfikuj metodę, aby inkrementowała wszystkie elementy w liście liczb całkowitych o wartości podanej przez użytkownika. Wynik wyświetl jako posortowaną listę.  
   **Przykład:** Lista [3, 1, 4], inkrementacja o 2 → [3, 5, 6]
5. **Dekrementuj**  
   **Zadanie:** Stwórz metodę, która dekrementuje wartości w słowniku (Dictionary<int, int>), gdzie klucze to indeksy, a wartości to liczby. Jeśli wartość osiągnie 0, usuń klucz z słownika.  
   **Przykład:** {1: 3, 2: 1} → {1: 2}
6. **Modulo1**  
   **Zadanie:** Napisz metodę, która dla podanej liczby n i zakresu m zwraca listę wszystkich liczb od 1 do n, które przy dzieleniu przez m dają resztę większą niż 2.  
   **Przykład:** n = 10, m = 3 → [2, 5, 8]
7. **LogicznyAnd**  
   **Zadanie:** Zaimplementuj metodę, która sprawdza, czy w podanej liście wartości logicznych wszystkie są prawdziwe tylko wtedy, gdy ich liczba jest parzysta.  
   **Przykład:** [true, true] → true, [true, true, true] → false
8. **LogicznyOr**  
   **Zadanie:** Napisz metodę, która dla listy wartości logicznych zwraca true, jeśli co najmniej połowa elementów jest prawdziwa.  
   **Przykład:** [true, false, true] → true, [true, false, false] → false
9. **LogicznyXor**  
   **Zadanie:** Stwórz metodę, która dla dwóch ciągów znaków zwraca true, jeśli dokładnie jedna z nich zawiera podciąg podany przez użytkownika.  
   **Przykład:** "abcde", "xyz", podciąg "cd" → true
10. **LogicznyNot**  
    **Zadanie:** Zaimplementuj metodę, która neguje warunki w liście wyrażeń logicznych zapisanych jako stringi (np. "x > 5" → "x <= 5") i zwraca nową listę.  
    **Przykład:** ["a > 3", "b == 0"] → ["a <= 3", "b != 0"]
11. **BitowyAnd**  
    **Zadanie:** Napisz metodę, która dla dwóch liczb całkowitych wykonuje bitowe AND na ich reprezentacjach binarnych i zwraca wynik jako ciąg znaków w formacie binarnym.  
    **Przykład:** 5 & 3 = "001" (binarnie: 101 & 011 = 001)
12. **ZrobCos2**  
    **Zadanie:** Zmodyfikuj metodę, aby wykonywała operacje logiczne na liście funkcji zwracających bool, zatrzymując się przy pierwszym false dla && i kontynuując dla &. Wyświetl wyniki dla obu operatorów.  
    **Przykład:** [() => true, () => false, () => true] → &&: false, &: false

**Region Sterowanie**

1. **Wybierz**  
   **Zadanie:** Napisz metodę, która dla podanej daty (typu DateTime) sprawdza, czy jest to dzień roboczy, weekend, czy święto (załóż listę świąt jako stałą tablicę).  
   **Przykład:** 2025-12-25 → "Święto"
2. **Wybierz2**  
   **Zadanie:** Stwórz metodę switch, która dla podanego kodu produktu (string) zwraca kategorię, cenę i dostępność na podstawie słownika produktów.  
   **Przykład:** "ABC123" → "Elektronika, 599.99 PLN, Dostępny"
3. **Wyswietl**  
   **Zadanie:** Zmodyfikuj metodę, aby wyświetlała tekst w formie tabeli ASCII z ramką, gdzie szerokość dostosowuje się do długości tekstu.  
   **Przykład:** "Test" →

text

CollapseWrapCopy

+------+

| Test |

+------+

1. **pobierzINT**  
   **Zadanie:** Rozszerz metodę, aby akceptowała zakres dopuszczalnych wartości (min, max) i ponawiała pytanie, dopóki użytkownik nie poda poprawnej liczby.  
   **Przykład:** min = 1, max = 10, input 15 → "Podaj liczbę od 1 do 10"`
2. **Menu**  
   **Zadanie:** Zaimplementuj dynamiczne menu, które wczytuje opcje z pliku tekstowego i pozwala użytkownikowi dodawać nowe pozycje w trakcie działania programu.  
   **Przykład:** Plik: 1. Start\n2. Stop → Menu z opcją dodania 3. Reset
3. **WyborDzialania**  
   **Zadanie:** Rozszerz metodę o obsługę podmenu – dla każdej opcji użytkownik może wybrać dodatkowy parametr (np. dla Dodawanie – czy dodać trzecią liczbę).  
   **Przykład:** "1" → "Czy dodać trzecią liczbę? (tak/nie)"
4. **wyrazeniaSwitch**  
   **Zadanie:** Napisz metodę, która dla podanego miesiąca (string) zwraca liczbę dni w tym miesiącu, uwzględniając rok przestępny (dla lutego).  
   **Przykład:** "luty", 2024 → 29
5. **wyrazeniaSwitch2**  
   **Zadanie:** Stwórz metodę, która dla obiektu dowolnego typu zwraca jego właściwości jako JSON-like string, używając pattern matching.  
   **Przykład:** obj = "Cześć" → { "Type": "string", "Value": "Cześć" }
6. **wyrazeniaSwitch3**  
   **Zadanie:** Zaimplementuj metodę, która dla podanej temperatury (int) klasyfikuje ją jako "zimno", "ciepło", "gorąco" z uwzględnieniem wilgotności (drugi parametr).  
   **Przykład:** 25, wilgotność 80% → "gorąco"

**Pozostałe metody**

1. **InstrWhile**  
   **Zadanie:** Napisz metodę, która generuje ciąg Fibonacciego w pętli while, zapisując wyniki do pliku, dopóki suma nie przekroczy podanej wartości.  
   **Przykład:** 100 → Plik: 0, 1, 1, 2, 3, 5, 8, 13, 21, 34
2. **InstrDoWhile**  
   **Zadanie:** Stwórz metodę, która symuluje rzut kostką (losowa liczba 1-6) w pętli do-while, sumując wyniki, dopóki suma nie przekroczy 20. Wyświetl liczbę rzutów.  
   **Przykład:** Wynik: 4 rzuty, suma = 22
3. **zalozFoldery**  
   **Zadanie:** Zmodyfikuj metodę, aby tworzyła strukturę folderów na podstawie daty (rok/miesiąc/dzień) dla całego miesiąca, z opcją podania zakresu dat przez użytkownika.  
   **Przykład:** 2025-03-01 do 2025-03-31 → 2025/03/01, 2025/03/02, itd.
4. **instrukcjaFor**  
   **Zadanie:** Napisz metodę, która dla podanego zakresu generuje ciąg geometryczny (np. a, a\*r, a\*r^2) i zapisuje go do tablicy, używając pętli for.  
   **Przykład:** a = 2, r = 3, n = 4 → [2, 6, 18, 54]
5. **instrukcjaForeach**  
   **Zadanie:** Stwórz metodę, która dla tablicy stringów generuje raport: dla każdego elementu podaje liczbę samogłosek i spółgłosek.  
   **Przykład:** ["kot", "pies"] → kot: 1 samogłoska, 2 spółgłoski; pies: 2 samogłoski, 2 spółgłoski