Tablice jednowymiarowe — statystyki ocen

Czas: 2 × 45 min | praca indywidualna | Wykład 1.07

Cel: Obliczysz statystyki ocen klasy zapisanych w tablicy, napiszesz własne sortowanie bąbelkowe i porównasz je z metodą `Array.Sort`.

Po zajęciach potrafisz:

- przechodzi po tablicy pętlami `for` i `foreach`
- wyznacza minimum, maksimum, średnią i medianę
- implementuje sortowanie bąbelkowe i porównuje je z `Array.Sort`
- zlicza wystąpienia wartości w tablicy pomocniczej

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 Dane i podstawowe statystyki

Utwórz projekt `Statystyki` z tablicą co najmniej 12 ocen (1–6).

1. Wypisz oceny w jednym wierszu: `string.Join(", ", oceny)`.
2. W jednej pętli `foreach` wyznacz minimum, maksimum i sumę — bez metod `Min()`, `Max()`, `Sum()`.
3. Oblicz średnią jako `double` (uwaga na dzielenie całkowite) i wypisz z dwoma miejscami po przecinku.

2 Sortowanie bąbelkowe

Napisz własne sortowanie na kopii tablicy.

```
for (int i = 0; i < kopia.Length - 1; i++)
    for (int j = 0; j < kopia.Length - 1 - i; j++)
        if (kopia[j] > kopia[j + 1])
            (kopia[j], kopia[j + 1]) = (kopia[j + 1], kopia[j]);
```

1. Skopiuj tablicę: `int[] kopia = (int[])oceny.Clone();` — dlaczego nie `int[] kopia = oceny;`?
2. Zaimplementuj sortowanie bąbelkowe według szablonu; zamianę zapisz krotką.
3. Dodaj licznik zamian i wypisz go po sortowaniu.
4. Porównaj wynik z `Array.Sort` na drugiej kopii (`SequenceEqual`).

3 Mediana i rozkład ocen

Wykorzystaj posortowaną tablicę.

1. Oblicz medianę: dla nieparzystej liczby elementów — środkowy, dla parzystej — średnia dwóch środkowych.
2. Sprawdź oba przypadki (dodaj lub usuń jedną ocenę).
3. Utwórz tablicę `int[] ile = new int[7]` i zlicz, ile było jedynek, dwójek itd.
4. Wypisz prosty wykres z gwiazdek lub znaków `#`: `new string('#', ile[o])`.

4 Wyszukiwanie

Odszukaj ocenę w tablicy.

1. Napisz pętlę, która znajduje pierwszą pozycję podanej oceny lub wypisuje „brak”.
2. Porównaj wynik z `Array.IndexOf`.
3. Zatwierdź pracę w Git.

W sprawozdaniu umieść

- ☐ kod programu
- ☐ zrzut ekranu wyników
- ☐ liczba zamian w sortowaniu bąbelkowym
- ☐ obliczenia mediany dla liczby parzystej i nieparzystej

Pytania kontrolne

1. Dlaczego `int[] kopia = oceny;` nie tworzy kopii tablicy?
2. Ile porównań wykonuje sortowanie bąbelkowe dla 12 elementów?
3. Dlaczego do obliczenia mediany tablica musi być posortowana?
4. Jaki wyjątek wystąpi przy odwołaniu `oceny[oceny.Length]`?

Kryteria oceny

Kryterium	Punkty
Min, max, średnia w jednej pętli	4
Sortowanie bąbelkowe z licznikiem zamian	5
Mediana (oba przypadki)	3
Rozkład ocen z wykresem	4
Wyszukiwanie i pytania	4
Razem	**20**

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