Konwersje i typy liczbowe

Czas: 2 × 45 min | praca indywidualna | Wykład 1.03

Cel: Napiszesz przeliczniki jednostek i sprawdzisz w praktyce różnice między typami `int`, `double` i `decimal` oraz sposobami konwersji.

Po zajęciach potrafisz:

- dobiera typ liczbowy do rodzaju danych (pieniądze — `decimal`)
- porównuje dokładność obliczeń na `double` i `decimal`
- rozróżnia rzutowanie `(int)` od `Convert.ToInt32`
- wyjaśnia przepełnienie typu `int`

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 Przelicznik temperatur

Program wczytuje temperaturę w °C i wypisuje ją w °F i K.

1. Utwórz projekt `Przeliczniki`. Wzory: F = C × 9/5 + 32, K = C + 273,15.
2. Wczytaj liczbę przez `double.TryParse`; przy błędzie wypisz komunikat.
3. Sprawdź wyniki dla 0, 100, −40 i 36,6 °C i wpisz je do tabeli.
4. Sprawdź, co się stanie, gdy zapiszesz wzór jako `C * (9 / 5) + 32` — wyjaśnij wynik.

2 Przelicznik walut

Kurs podaje użytkownik — program nie pobiera go z internetu (zrobimy to w roku 2).

1. Wczytaj kwotę w złotych i kurs euro jako `decimal`.
2. Odrzuć kurs mniejszy lub równy zero.
3. Wypisz wynik z dwoma miejscami po przecinku i symbolem EUR.

3 Eksperyment: double kontra decimal

Uruchom kod i zapisz wyniki.

```
double sd = 0; decimal sm = 0;
for (int i = 0; i < 10; i++) { sd += 0.1; sm += 0.1m; }
Console.WriteLine($"double:  {sd:R}, równe 1? {sd == 1.0}");
Console.WriteLine($"decimal: {sm}, równe 1? {sm == 1.0m}");
```

1. Dodaj dziesięć razy 0,1 do zmiennej `double` i do zmiennej `decimal`.
2. Wypisz obie sumy (dla `double` użyj formatu `:R`) i wynik porównania z 1.
3. Zapisz wniosek: którego typu użyć do kwot pieniężnych i dlaczego.

4 Eksperyment: konwersje i przepełnienie

Sprawdź, jak zachowują się różne konwersje.

1. Porównaj `(int)2.7` i `Convert.ToInt32(2.7)`.
2. Porównaj `Convert.ToInt32(2.5)` i `Convert.ToInt32(3.5)` — sprawdź w dokumentacji, co to jest „zaokrąglanie bankierskie".
3. Do zmiennej `int duza = int.MaxValue;` dodaj 1 w bloku `unchecked` i wypisz wynik.
4. Powtórz z blokiem `checked` i zapisz, jaki wyjątek wystąpił.
5. Zatwierdź pracę w Git.

W sprawozdaniu umieść

- ☐ kod programu
- ☐ tabela wyników przeliczeń
- ☐ wyniki eksperymentów z wnioskami

Pytania kontrolne

1. Dlaczego `9 / 5` w C# daje 1, a `9.0 / 5` daje 1,8?
2. Dlaczego suma dziesięciu 0,1 w typie `double` nie jest równa 1?
3. Czym różni się `(int)x` od `Convert.ToInt32(x)`?
4. Co to jest przepełnienie i jak się przed nim zabezpieczyć?

Kryteria oceny

Kryterium	Punkty
Przelicznik temperatur i tabela	5
Przelicznik walut na decimal z walidacją	4
Eksperyment double/decimal z wnioskiem	4
Eksperyment konwersji i przepełnienia	4
Pytania	3
Razem	**20**

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