Programowanie komputerów.
Ćwiczenia 4.

METODY

```
[Modyfikatory] Typ Nazwa ([Lista argumentów])
{
    // Ciało metody
}
```

Elementy definicji metody:

- **Deklaracja metody**:
  - **Modyfikatory** - określają zachowanie i dostępność metody,
  - **Typ** – typ danej zwracanej przez metodę,
  - **Nazwa** – nazwa metody, różna od nazwy klasy, w której została zdefiniowana,
  - **Lista argumentów** – lista argumentów przekazywanych do metody (może być pusta).
- **Ciało metody** – inaczej treść metody, kod (zbiór instrukcji) realizujący działanie metody. Jeśli metoda ma coś zwrócić, w jej ciele musi wystąpić polecenie *return*.

Przekazywanie tablic do metody i zwracanie tablic

- Tablice jako obiekty referencyjne przekazywane są przez referencje (tak jest domyślnie). Patrz przykład 5.7 z podręcznika „Wstęp do programowania w C#”.
- Tablica może być zwracana przez metodę. Patrz przykład 5.8 z podręcznika „Wstęp do programowania w C#”.

Przekazywanie argumentów przez referencję

Przekazywanie argumentów przez referencję. Patrz rozdział 5.4 z podręcznika „Wstęp do programowania w C#”.

Obiekty klasy są przekazywane przez referencję.

Natomiast zmienne typów prostych domyślnie są przekazywane przez wartość, ale można przekazać także przez referencję (ref lub out).

Każda metoda może zwrócić maksymalnie jedną wartość. Jeśli zależy nam na większej liczbie dostępnych po wykonaniu metody wartości – można w tym celu wykorzystać argumenty (ref lub out) lub typ referencyjny (np. tablica, obiekt klasy).

Weryfikacja typu

Struktury definiujące typy wartościowe udostępniają metody `TryParse` służące do konwersji łańcucha znaków do danej danego typu. Metoda `TryParse` zwraca `true`, jeśli konwersja powiodła się.

Metody przeładowane (przeciążone)

- Metody, które mają tę samą nazwę i różnią się listą argumentów nazywane są metodami przeładowanymi lub przeciążonymi (ang. overloaded).
- Odpowiednia wersja przeładowanej metody jest dobierana na podstawie listy argumentów umieszczonej w wywołaniu (patrz przykład 5.11 z podręcznika „Wstęp do programowania w C#").
- Należy także uwzględnić argumenty domyślne oraz niejawne konwersje (patrz przykłady 5.12-5.14 z podręcznika „Wstęp do programowania w C#").

ZADANIA

Zadanie 1.

Korzystając z klas utworzonych na ćwiczeniach 3. **`Sprzedaz`**, **`Agregacja`**, **`Dodatki`**. Zmodyfikuj klasę w taki sposób aby klasa przyjmowała argument typu referencyjnego tablicę. Dostosuj metody klasy do jej pól.

a) Nie wykorzystuj żadnych dodatkowych bibliotek poza biblioteką System.

Zadanie 2.

Korzystając z klas utworzonych w zadaniu 2 **`Sprzedaz`**, **`Dodatki`**. Zmodyfikuj metody Brutto, Netto w taki sposób aby możliwe było wyświetlenie kwoty podatku VAT poza metodą.

a) Nie wykorzystuj żadnych dodatkowych bibliotek poza biblioteką System.

Zadanie 3.

Korzystając z klas utworzonych w zadaniu 3 **`Sprzedaz`**, **`Dodatki`**. Zmodyfikuj metody `Brutto`, `Netto` w taki sposób aby możliwe było wyświetlenie kwoty podatku VAT poza metodą oraz sumowanie kwot dla wszystkich kolejno tworzonych obiektów sprzedaży.

a) Nie wykorzystuj żadnych dodatkowych bibliotek poza biblioteką System.

Zadanie 4.

Napisz program, który będzie prosił użytkownika o podanie poprawnej wartości boku kwadratu aż do uzyskania poprawnej wartości. Program powinien sprawdzić, czy użytkownik podaje liczbę dodatnią.

Następnie należy wyznaczyć pole powierzchni kwadratu.

Zadanie domowe 1.

Napisz program z użyciem klasy, której polem składowym jest łańcuch znakowy. Klasa ma posiadać metodę, która pozwoli użytkownikowi wpisać tekst do pola. Ponadto ma mieć metodę, która z łańcucha znaków (będącego polem tej klasy) pobiera same cyfry i zwraca liczbę (`int`) konwertowaną z uzyskanego łańcucha cyfr (Przy pomocy pola publicznego `Int32.MaxValue` upewnij się, czy uzyskany ciąg cyfr może być przekonwertowany na liczbę typu `int`).

Przykładowe działanie obu metod: prezentuje poniższy zrzut ekranu:

```
Wpisz tekst: dr450-?ks>123h3
4501233
```

Zadanie domowe 2.

Do klasy z poprzedniego zadania dodaj metodę zamieniającą w łańcuchu znaków (będącego polem klasy) wszystkie małe litery na duże, a duże na małe i zwracającą nowy łańcuch znaków. Znaki, które nie są literami, mają pozostać bez zmian.

Przykładowe działanie tej metody prezentuje poniższy zrzut ekranu:

```
Wpisz tekst: ŚcieMNiało SIĘ już!
Nowy tekst:  śCIEmnIAŁO się JUŻ!
```

Wskazówki: wykorzystaj metody statyczne ze struktury *Char*: Char.IsLower, Char.IsUpper oraz Char.ToUpper i Char.ToLower.

https://msdn.microsoft.com/en-us/library/system.char(v=vs.110).aspx

Zadanie domowe 3.

Napisz program, który zawiera klasę Wektor z tablicą jednowymiarową typu int jako polem składowym. Klasa ma mieć konstruktor pozwalający inicjalizować obiekt, konstruktor kopiujący, metodę wyświetlającą w jednej linii elementy wektora oraz metodę obliczającą iloczyn skalarny dwóch wektorów, która ma zwrócić obliczoną wielkość.

Przykładowe działanie prezentuje poniższy zrzut ekranu:

```
  1  2  0
  3  1 -1
Iloczyn skalarny: 5
```