

Programowanie komputerów.

Ćwiczenia 3.

ZADANIA

Zadanie 1.

Napisz program, który tworzy klasę **Sprzedaz**, zawierającą trzy prywatne pola składowe: `cena`, `szt` i `upust` (wartość logiczna). Klasa ma posiadać dwie właściwości:

- `Wartosc` – własność „do odczytu” zwracająca wartość sprzedaży;
- `Upust` – własność „do odczytu” zwracająca informację „udzielono upustu”, gdy upust został przyznany, „nie udzielono upustu”, gdy upust nie został przyznany.

Klasa ma posiadać także konstruktor oraz nadpisanie metody `ToString` pozwalającej wyświetlić informację o wartości sprzedaży wraz z informacją o przydzieleniu lub nie upustu.

- a) W metodzie **Main** zdefiniuj 2 obiekty tej klasy i wyświetl informację o nich używając metody `ToString`.

Zadanie 2.

Zmodyfikuj zadanie 1:

tak aby możliwe było zliczanie ile wykonano transakcji sprzedaży.

- a) W metodzie **Main** zdefiniuj 4 obiekty tej klasy i wyświetl informację o nich używając metody `ToString` wraz z informacją którą z kolei transakcję wykonano.

Zadanie 3.

Zmodyfikuj zadanie 2:

tak aby możliwe było utworzenie nowego obiektu klasy na podstawie już istniejącego obiektu tej samej klasy.

- a) W metodzie **Main** zdefiniuj 4 obiekty tej klasy (w tym dwie kopie obiektów już istniejących) i wyświetl informację o nich używając metody `ToString` wraz z informacją którą z kolei transakcję wykonano.

Zadanie 4.

Utwórz klasę **Agregacja** składającą się z trzech transakcji sprzedaży (trzy obiekty klasy **Sprzedaz**). W klasie **Agregacja** zdefiniuj metody:

- `Suma` – sumująca wartości sprzedaży;
- `Srednia` – wyznaczająca wartość średnią z wartości sprzedaży.

WSKAZÓWKI

- a) W tym programie wystąpią dwie klasy (**Sprzedaz** i **Agregacja**).
- b) Przykładowa deklaracja pól w klasie **Agregacja**:
`Sprzedaz s1, s2, s3;`
Natomiast w metodzie **Main** przykładowa definicja agregacji może wyglądać:
`Agregacja agregaty123 = new Agregacja (new Sprzedaz(2.5, 10),
new Sprzedaz(4,6,)
new Sprzedaz(11,5));`
- c) Nie wykorzystuj żadnych dodatkowych bibliotek poza biblioteką `System`.

Zadanie 5.

Utwórz klasę statyczną **Dodatki**, w której znajdą się dwie metody statyczne pozwalające na przeliczenie ceny netto na brutto i odwrotnie. Stawkę podatku VAT (0%, 5%, 8%, 23%) oraz cenę podaje użytkownik. Należy przyjąć zaokrąglenie do 1 gr.

Zadanie domowe 1.

Napisz program tworzący klasę **Punkt** do obsługi punktów na płaszczyźnie. Współrzędne punktu mają być prywatnymi polami tej klasy. Klasa ta ma zawierać: konstruktor, którego argumentami będą współrzędne punktu, metodę składową **Przesun**, realizującą przesunięcie o zadane wartości oraz metodę **ToString** zwracającą tekst opisujący aktualne współrzędne punktu. Zdefiniuj także metodę **Wyświetl**, która wyświetla aktualne informacje o punkcie na konsoli.

Przykładowa zawartość metody **Main**:

```
static void Main()
{
    Punkt a = new Punkt(1,4);    // Wpisanie współrzędnych poprzez konstruktor
    a.Wyświetl();                // Wyświetlone zostaną wartości 1,4
    a.Przesun (3,5);              // Przesunięcie punktu o zadane wartości
    a.Wyświetl ();                // Wyświetlone zostaną wartości 4,9
    Console.ReadKey();
}
```

Zadanie domowe 2.

Napisz program (używając klasy **Punkt**), który przechowuje współrzędne trzech punktów w tablicy i sprawdza przy pomocy statycznej metody, czy leżą one na jednej prostej.

Przykładowa zawartość metody **Main**:

```
Punkt[] tab = new Punkt[3];
tab[0] = new Punkt(1, 1);
tab[1] = new Punkt(2, 2);
tab[2] = new Punkt(3, 3);
foreach(Punkt a in tab)
    a.Wyświetl();                // Wyświetla położenie punktów
Punkt.NaProstej(tab);           // Wyświetla komunikat czy są współliniowe
```

Zadanie domowe 3.

Zdefiniuj klasę **Odcinek** składającą się z dwóch punktów (klasy **Punkt**). W klasie **Odcinek** zdefiniuj metodę, która obliczy długość odcinka.

WSKAZÓWKI

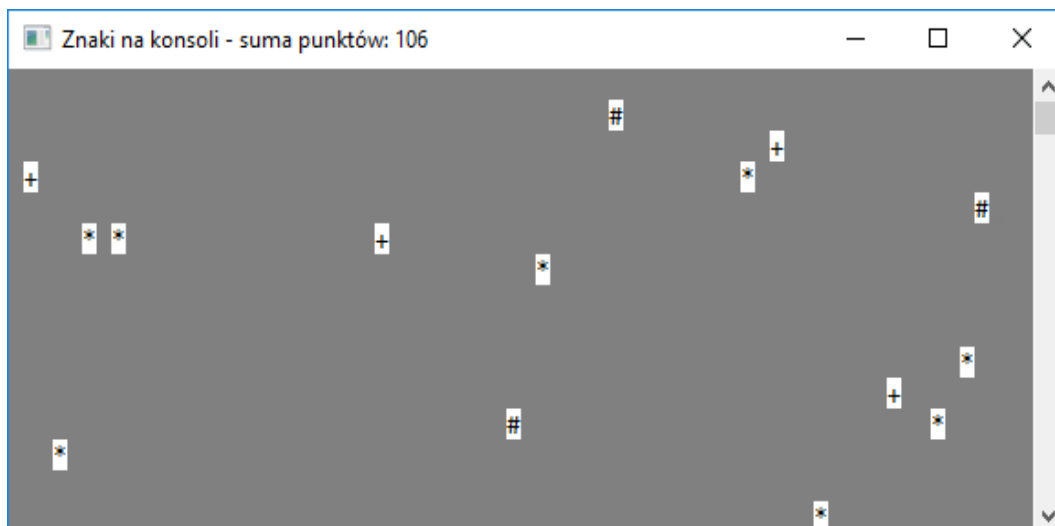
- W tym programie wystąpią dwie klasy (**Punkt** i **Odcinek**). Należy przyjąć, że odcinek składa się z punktów i jego położenie na płaszczyźnie będzie opisane przy pomocy punktów wyznaczających oba końce odcinka.
- Przykładowa deklaracja pól w klasie **Odcinek**:
`Punkt punktA, punktB;`
Natomiast w metodzie **Main** przykładowa definicja odcinka może wyglądać:
`Odcinek odcinekAB = new Odcinek(new Punkt (4,0), new Punkt(4,6));`
- W zadaniu możesz skorzystać z dodatkowych bibliotek.

Zadanie domowe 4.

Zapoznaj się z treścią podręcznika „*Wstęp do programowania w C#*” dot. klasy **System.Object**, str. 178-180.

Zadanie domowe 5.

Napisz program, który wyświetla na konsoli znaki w określonych miejscach. Załóżmy, że program stanowi część jakiejś gry, np. znaki na konsoli mogą pełnić rolę „nagród” dla gracza poruszającego postacią po planszy jak w grze *snake*. Do wyboru mamy 3 rodzaje nagród: znak * (10 punktów), znak + (5 punktów) i znak # (2 punkty). Klasa realizująca to zadanie powinna mieć dwa konstruktory. Jeden bezargumentowy, który przydziela losowo nagrodę i jej miejsce (współrzędne kolumny i wiersza). Drugi ma mieć argumenty dla rodzaju nagrody i jej miejsca. W programie należy utworzyć kilkanaście obiektów (przy pomocy obu konstruktorów) i wyświetlić nagrody na konsoli. W przypadku gdy w danym miejscu jest przydzielonych więcej nagród, należy na ekranie napisać znak X, oznaczający „worek nagród” stanowiący niespodziankę dla gracza. W tytule okna należy wyświetlić łączną sumę punktów dla wszystkich umieszczonych na konsoli nagród.



LITERATURA

„*Wstęp do programowania w C#*”, podrozdział:

- 6.3 Użycie zdefiniowanej klasy