

Definiowanie klas w C#, składowe klasy, modyfikatory dostępu do klas i składowych klas

Deklaracja i definicja metod, zwracanie wartości z metod, przekazywania parametrów do metod

Zadanie 1

Napisz program wykorzystujący klasę **Punkt** reprezentującą punkty na płaszczyźnie. Klasa ta ma zawierać:

- konstruktor, którego argumentami będą współrzędne punktu,
- właściwości umożliwiające odczytanie i zmodyfikowanie współrzędnych punktu,
- metodę do przesuwania punktu o zadane wartości,
- metodę wypisującą współrzędne danego punktu.

Współrzędne punktu mają być prywatnymi składowymi klasy, natomiast metody mają być publiczne.

Przygotuj przykład wykorzystania klasy.

Zadanie 2

Przygotuj klasę, która pozwoli na rejestrację zużycia energii elektrycznej. Klasa powinna pozwalać na:

- rejestrację początkowego i bieżącego stanu licznika energii,
- uzyskanie danych o początkowym oraz bieżącym stanie licznika,
- obliczanie zużytej energii.

Zadanie 3

Napisz program tworzący klasę **Punkt** do obsługi punktów na płaszczyźnie. Klasa ta ma zawierać: konstruktor, którego argumentami będą współrzędne punktu, metodę składową *Przesun()*, realizującą przesunięcie o zadane wielkości oraz metodę składową *Wyswietl()* wypisującą aktualne współrzędne punktu. Współrzędne punktu mają być zdefiniowane poprzez właściwości.

Zadanie 4

Napisz program (używając klasy **Punkt** zdefiniowanej w poprzednim zadaniu), który przechowuje dane o trzech punktach w tablicy obiektów i sprawdza przy pomocy statycznej metody, czy leżą one na jednej prostej.

Zadanie 5

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

Zadanie 6

Zdefiniuj klasę **Prostopadloscian**, która pozwoli na reprezentację danych opisujących długość, szerokość i wysokość prostopadłościanu. W klasie zaimplementuj metody pozwalające na obliczenie objętości prostopadłościanu, oraz porównanie objętości dwóch prostopadłościanów.

Zadanie 7

Przygotuj klasę **Prostopadloscian**, która pozwoli na reprezentację danych opisujących długość, szerokość i wysokość prostopadłościanu w centymetrach. W klasie zaimplementuj metody pozwalające na obliczenie objętości prostopadłościanu, oraz porównanie objętości dwóch prostopadłościanów. W konstruktorze użyj wartości 1 centymetr jako domyślnej dla wszystkich rozmiarów prostopadłościanu

Zadanie 8

Napisz program, który tworzy klasę **Prostokat**, zawierającą dwie prywatne dane składowe: *dlugosc*, *szerokosc*, dwie prywatne metody: *powierzchnia()*, *obwod()* oraz metodę publiczną – *Prezentuj()* (która wyświetla powierzchnię i obwód prostokąta) i konstruktor inicjalizujący. W metodzie *Main()* zdefiniuj obiekt i uruchom dla niego metodę *Prezentuj()*.

Zadanie 9

Uzupełnij program z poprzedniego zadania o definicję tablicy obiektów dla prostokątów. W metodzie *Main()* wyświetl powierzchnie oraz obwód wszystkich prostokątów w tablicy używając (wewnątrz pętli *foreach*) metody publicznej *Prezentuj()*.

Zadanie 10

Uzupełnij program z poprzedniego zadania o definicję metody statycznej, która podaje powierzchnię największego prostokąta.

Zadanie 11

Utwórz klasę **Student** (imie, nazwisko, nr albumu, średnia). Utwórz metody, które pobiorą imię, nazwisko, nr albumu i średnią oraz metodę, która sprawdzi czy i w jakiej wysokości dostanie student stypendium (wykorzystaj tabelę poniżej). Pamiętaj, że średnia powinna być z przedziału 2,0-5,0. Wykorzystując utworzoną klasę napisz program w Main().

Średnia ocen		Kwota stypendium
od	do	
2,00	4,00	0,00 zł
4,00	4,80	300, 00 zł
4,80	5,00	600,00 zł

Zadanie 12

Przygotuj klasę **Pracownik**, która pozwoli na reprezentację danych o pracowniku: *imię, nazwisko, stawka godzinowa, ilość godzin pracy*. Klasa ma posiadać konstruktor wprowadzający dane, właściwości umożliwiające na modyfikowanie wymienionych danych oraz metodę *drukuj()*, która ma wyświetlić w jednej linii: *imię, nazwisko, stawka, godzina, wynagrodzenie zasadnicze, np.:*

Jan	Kowalski	20.00	160.00	3200.00
-----	----------	-------	--------	---------

Następnie utwórz 4-elementową tablicę obiektów klasy Pracownik. Załaduj dane do tablicy poprzez konstruktory. Oczekiwany rezultat powinien wyglądać następująco:

Jan	Kowalski	20.00	160.00	3200.00
Anna	Nowak	25.00	140.00	3500.00
Jerzy	Adamski	15.00	170.00	2550.00
Ewa	Czekaj		17.00	150.00
				2550.00

Zadanie 13

Napisz klasę **Trojmian**, która umożliwi obliczenie pierwiastków równania kwadratowego. Klasa powinna zawierać trzy metody:

- `czytaj_dane()` - odpowiedzialna za wczytanie danych do programu oraz obsłużenie sytuacji, kiedy $a=0$,
- `przetworz_dane()` - odpowiedzialna za wykonanie niezbędnych obliczeń,
- `wyswietl_wynik()` - odpowiedzialna za wyświetlanie wyników na ekranie monitora.

Zadanie 14

Napisz statyczną klasę **Temperatura**, która będzie zawierała dwie metody:

- **DoCelsjusza** – która przeliczy temperaturę w stopniach Fahrenheita na temperaturę w stopniach Celsjusza,
- **DoFahrenheita** – która przeliczy temperaturę w stopniach Celsjusza na temperaturę w stopniach Fahrenheita.

Sprawdź działanie klasy w praktyce, przeliczając podane przez użytkownika temperatury zarówno w stopniach Celsjusza jak i Fahrenheita.

Zadanie 15

Przygotuj klasę **Osoba** pozwalającą na reprezentację danych o osobach. Przyjmij, że osoba opisana jest przez: *imię, nazwisko, datę i miejsce urodzenia, adres zamieszkania*. *Adres zamieszkania* powinien być obiektem klasy **Adres**. W klasie zaimplementuj:

- konstruktor,
- właściwości pozwalające na odczytanie i zmianę wartości każdego z pól,
- metodę pozwalającą na obliczenie wieku osoby,
- metodę sprawdzającą, czy dana osoba jest pełnoletnia.

Zadanie 16

a). Stwórz klasę **Licz** z:

- publicznym polem *wartosc* przechowującym wartość liczbową
- metodą *Dodaj* przyjmującą jeden parametr i dodającą przekazaną wartość do wartości trzymanej w polu *wartosc*
- analogiczną operację *odejmij*

W *Main()* utwórz kilka obiektów klasy *Licz* i wykonaj różne operacje.

b). Do klasy *Licz* dodaj konstruktor z jednym parametrem - który inicjuje pole *wartosc* na liczbę przekazaną w parametrze.

c). Zmień widoczność pola na `private` i dodaj funkcję wypisującą stan obiektu (pole *wartosc*)