ŚCIĄGAWKA

Zmienna:

Składnia: Typ nazwaZmiennej; lub Typ nazwaZmiennej = wartość;
Przykład: int car
Ważne informacje:

- Koniecznie musimy podać typ zmiennej
- Z dobrych praktyk nazwy zmiennych piszemy z małej litery, a jeśli nazwa składa się z więcej niż jednego wyrazu to stosujemy tzw camelCase

Przykłady podstawowych typów zmiennych:

- Zmienne numeryczne:
  - int
  - double
  - decimal
  - float
- Zmienne tekstowe:
  - string
  - char

Pole:

Składnia: modyfikatorDostępu Typ _nazwaPola;
Lub
modyfikatorDostępu Typ _nazwaPola = wartość
Przykład: private int number;
Ważne informacje:

- Umieszczamy wewnątrz klasy
- Możemy traktować pole jak zmienną globalną dostępną z poziomu klasy
- Z dobrych praktyk pole nazywamy zaczynając od znaku „_” a następnie z małej litery podajemy nazwę pola

Właściwość

Składnia: modyfikatorDostępu typ NazwaWłaściwości { get; set; }
Przykład: public string Name { get; set; }
Ważne informacje:

- Umieszczamy wewnątrz klasy
- Właściwość to element klasy który przechowuje informację o stanie naszego obiektu, np. klasa Samochód może mieć właściwość Marka która przechowuje wartość „Mercedes”
- Dostęp do właściwości uzyskujemy poprzez bezpośrednie wywołanie właściwości na obiekcie np. car.Brand (nazwaZmiennejKlasy.Właściwość). To podejście działa jedynie wtedy kiedy Właściwość ma ustawiony modyfikator dostępu na public.W przeciwnym wypadku musimy użyć tzw. setterów i getterów czyli funkcji które ustawiają lub pobierają wartość Właściwości. Możemy również ustawić prywatne właściwości za pomocą konstruktora (więcej o tym przy klasach)
- Właściwości nazywamy z wielkiej litery

Funkcja

Składnia: modyfikatorDostępu typ NazwaFunkcji(typ parametr){ }

Przykłady:

```
public void SetProperty(string brand)
{
    Brand = brand;
```

}; - ta funkcja jest typu void tzn że nie zwraca ona żadnej wartości po wykonaniu. Funkcja przyjmuje jeden parametr typu string o nazwie brand. Działanie funkcji polega na przypisaniu wartości tej zmiennej do właściwości (seter)

```
public string  GetProperty()
{
    return Brand;
```

}; - ta funkcja jest typu string tzn że zwraca ona zmienną typu string. Funkcja nie przyjmuje żadnych parametrów. Działanie funkcji polega na pobraniu wartości z Właściwości Barnd i przekazaniu jej do zmiennej w programie.

Przykłady wywołania:

- SetProperty(„Mercedes”);
- string brand = GetProperty();

Ważne informacje:

- Funkcja to element programu który jest wywoływany po nazwie i tymczasowo przejmuje kontrolę nad działaniem programu w celu wykonania jakiejś operacji. Po zakończeniu operacji zwraca kontrolę nad programem do głównej funkcji main.
- Funkcja może przyjmować tzw parametry, czyli zmienne które są przekazywane do funkcji z zewnątrz (np. w głównej funkcji main wywołujemy funckję SetProperty i przekazujemy jej zmienną typu string której w funkcji main przypisaliśmy wcześniej wartość „Mercedes”. Dzięki temu po przekazaniu parametru string brand do funkcji, funkcja SetProperty ma do niej dostęp.
- Funkcja może ale nie musi zwracać wartości. To znaczy, że funkcja może przejąć działanie programu i wykonać jakąś operację, a następnie może zwrócić wynik tej operacji lub nie. Np. ustawienie właściwości nie zwraca żadnej wartości, pobranie właściwości zwraca wynik operacji w postaci zmiennej typu string.

Klasa

Składnia: modyfikatorDostępu słowoKluczoweClass NazwaKlasy
{
}
Przykłady:

```
public class Car
{
        W tym miejscu możemy umieścić konstruktor
        Tu umieszczamy: właściwości, pola oraz funkcje (metody)
}
```

Ważne informacje:

- Klasa to obiekt który możemy zdefiniować za pomocą właściwości np. klasa Car opisywana przez właściwości: string Brand, string Model, string EngineType...
- Utworzenie obiektu w programie polega na tzw utworzeniu instancji klasy oraz inicjalizacji tej klasy:
    - Car car = new Car(), po lewej stronie nic się nie zmienia, podajemy typ zmiennej oraz nazwę zmiennej. Natomiast po prawej stronie korzystamy z konstruktora (w tym przypadku domyślnego konstruktora który jest tworzony razem z klasą, nie musi on być deklarowany) do utworzenia nowej instancji klasy Car przypisanej do zmiennej car.
- Nazwy klas piszemy z dużej litery
- Możemy utworzyć własny konstruktor który będzie przyjmował jakieś parametry i już w tym miejscu wymagać od użytkownika aby przypisał wartości do właściwości (konstruktor może również ustawić właściwości które mają modyfikator dostępu ustawiony na private)
    - Przykład

```
public class Car
{
        public Car(string brand) // to jest nasz konstruktor
        {
                Brand = brand
        }
        private string Brand { get; set; }
}
```