# **C# Classes and Objects**

# **(Класи та об'єкти)**

У C# все пов'язано з класами та об'єктами, а також їхніми атрибутами та методами.

**Клас** - це як конструктор об'єктів, або "план" для створення об'єктів.  
**Клас** - це ніби план конкретного об'єкта, який має певні атрибути та особливості. Наприклад, автомобіль повинен мати певні атрибути, такі як чотири колеса, двоє чи більше дверей, кермо, лобове скло тощо. Він також повинен мати певні функціональні можливості, такі як запуск, зупинка, рух, переміщення тощо. Будь-який об'єкт, який має ці атрибути та функції, є автомобілем. Тут автомобіль - це клас, який визначає деякі специфічні атрибути та функції. Кожен окремий автомобіль є об'єктом класу car. Ви можете сказати, що автомобіль, який ви маєте, є об'єктом класу car.

Так само в об'єктно-орієнтованому програмуванні клас визначає деякі властивості, поля, події, методи тощо. Клас визначає типи даних і функціональність, яку матимуть його об'єкти.

**Створення класу**

Щоб створити клас, використовуйте ключове слово **class**:

class Car

{

string color = "red";

}

**Створення об'єкта**

Об'єкт створюється з класу. Ми вже створили клас з іменем **Car**, тож тепер можемо використовувати його для створення об'єктів.

Щоб створити об'єкт класу **Car**, вкажіть ім'я класу, а потім ім'я об'єкта і використовуйте ключове слово **new**:

class Car

{

string *color* = "red";

static void Main(string[] args)

{

Car **myObj** = new Car();

Console.WriteLine(**myObj.***color*);

}

}

Зверніть увагу, що ми використовуємо крапковий синтаксис (.) для доступу до змінних/полів всередині класу (myObj.color).

# **C# Class Members**

Поля та методи всередині класів часто називають "членами класу" (Class Members):

// The class

class MyClass

{

// Class members

string color = "red"; // field

int maxSpeed = 200; // field

public void fullThrottle() // method

{

Console.WriteLine("The car is going as fast as it can!");

}

}

**Поля(Fields)**

У попередній главі ви дізналися, що змінні всередині класу називаються полями, і що доступ до них можна отримати, створивши об'єкт класу і використовуючи синтаксис крапки ***(.)***.

У наступному прикладі створюється об'єкт класу Car з іменем **myObj**. Потім ми виводимо значення полів **color** і **maxSpeed**:

class Car

{

string color = "red";

int maxSpeed = 200;

static void Main(string[] args)

{

Car **myObj** = new Car();

Console.WriteLine(myObj.color);

Console.WriteLine(myObj.maxSpeed);

}

}

Ви також можете залишити поля порожніми і змінити їх під час створення об'єкта:

class Car

{

string color;

int maxSpeed;

static void Main(string[] args)

{

Car myObj = new Car();

myObj.color = "red";

myObj.maxSpeed = 200;

Console.WriteLine(myObj.color);

Console.WriteLine(myObj.maxSpeed);

}

}

Це особливо корисно при створенні декількох об'єктів одного класу:

class Car

{

string model;

string color;

int year;

static void Main(string[] args)

{

Car Ford = new Car();

Ford.model = "Mustang";

Ford.color = "red";

Ford.year = 1969;

Car Opel = new Car();

Opel.model = "Astra";

Opel.color = "white";

Opel.year = 2005;

Console.WriteLine(Ford.model);

Console.WriteLine(Opel.model);

}

}

**Методи об'єктів**

З розділу Методи C# ви дізналися, що методи використовуються для виконання певних дій.

Методи зазвичай належать до класу і визначають поведінку об'єкта класу.

Як і у випадку з полями, ви можете отримати доступ до методів за допомогою крапкового синтаксису. Однак зверніть увагу, що метод повинен бути загальнодоступним. І пам'ятайте, що для виклику (виконання) методу ми використовуємо ім'я методу, за яким слідують дві парентези ***()*** і крапка з комою ***;*** :

class Car

{

string color; // field

int maxSpeed; // field

public void fullThrottle() // method

{

Console.WriteLine("The car is going as fast as it can!");

}

static void Main(string[] args)

{

Car **myObj** = new Car();

myObj.fullThrottle(); // Call the method

}

}

До **статичного (static)** методу можна отримати доступ без створення об'єкта класу, в той час як до **загальнодоступних (public) методів** можуть звертатися лише об'єкти.

**Використання декількох класів**

Пам'ятаєте з попереднього розділу, що ми можемо використовувати декілька класів для кращої організації (один для полів і методів, а інший для виконання).

#### **prog2.cs**

class Car

{

public string model;

public string color;

public int year;

public void fullThrottle()

{

Console.WriteLine("The car is going as fast as it can!");

}

}

#### **prog.cs**

class Program

{

static void Main(string[] args)

{

Car Ford = new Car();

Ford.model = "Mustang";

Ford.color = "red";

Ford.year = 1969;

Car Opel = new Car();

Opel.model = "Astra";

Opel.color = "white";

Opel.year = 2005;

Console.WriteLine(Ford.model);

Console.WriteLine(Opel.model);

}

}

Ключове слово **public** називається модифікатором доступу, який вказує, що поля класу **Car** є доступними і для інших класів, таких як **Program**.